# SPiiPlus MMI Application Studio

## User Guide

**September 2020**

**Document Revision: 3.02**

ACS MOTION CONTROL

SPiiPlus MMI Application Studio

Release Date: September 2020



www.acsmotioncontrol.com

support@acsmotioncontrol.com

sales@acsmotioncontrol.com

# Revision History

| Date | Revision | Description |
|---|---|---|
| September 2020 | 3.02 | Corrections to SinCos Encoder Compensation section |
| July 2020 | 3.01 | Bug fixes, ADK Update |
| June 2020 | 3.00 | Changed to WPF |
| April 2019 | 2.70 | Layout corrections |
| July 2018 | 2.60 | Added Current Loop Auto Tuning<br>Added Open Loop Verification<br>Added XRMSD, XRMSM, XRMSTD, XRMSTM, CERRK, KDEC |
| December 2017 | 2.50 | Updated Adjuster Wizard and Scope. |
| July 2017 | 2.40 | Added PdoIndex to the #ETHERCAT command<br>Removed the obsolete description of basic and advanced SPiiPlusMMI Application Studio Packages<br>Added a description of the Backup Machine and Duplicate Machine to Upgrade and Recovery Wizard feature<br>Updated Windows Platform Support<br>Removed UDPPORT |
| August 2016 | 2.30 | Added optional groups to System Setup<br>Added to Safety and Faults Configurator and Safety and Faults Monitor<br>> STO<br>> Hardware Emergency Stop<br>Removed "Component Failure" inversion checkbox from Safety and Faults Configurator<br>Added Component Failure to Safety and Faults Monitor<br>Renamed "Drive Fault" to "Drive Fault/ Laser Fault"<br>Clarified the default response for a hardware emergency stop<br>Added note stating that to save any work in a buffer window that it must be compiled first.<br>Added new scope features |

| Date | Revision | Description |
|---|---|---|
| | | > Save MMI data as Excel CSV<br>> Quadrature grid for XY-CH# display<br>> Setting Y-axis limits per channel |
| March 2015 | 2.29 | Upgrade and Recovery Wizard Updated |

# Conventions Used in this Guide

### Text Formats

| Format | Description |
|---|---|
| **Bold** | Names of GUI objects or commands |
| **BOLD + UPPERCASE** | ACSPL+ variables and commands |
| `Monospace + grey background` | Code example |
| *Italic* | Names of other documents |
| Blue | Hyperlink |
| [ ] | In commands indicates optional item(s) |
| \| | In commands indicates either/or items |

### Flagged Text

| | |
|---|---|
|  | **Note** - includes additional information or programming tips. |
|  | **Caution** - describes a condition that may result in damage to equipment. |
|  | **Warning** - describes a condition that may result in serious bodily injury or death. |
|  | **Model** - highlights a specification, procedure, condition, or statement that depends on the product model |
|  | **Advanced** - indicates a topic for advanced users. |

# Related Documents

Documents listed in the following table provide additional information related to this document.

The most updated version of the documents can be downloaded by authorized users from www.acsmotioncontrol.com/downloads.

Online versions for all ACS software manuals are available to authorized users at ACS Motion Control Knowledge Center.

| Document | Description |
|---|---|
| *SPiiPlus Setup Guide* | A complete guide for setting up the SPiiPlus system. |
| *SPiiPlus Utilities User Guide* | A guide for using the SPiiPlus User Mode Driver (UMD) for setting up communication with the SPiiPlus motion controller. |
| *SPiiPlus Command & Variable Reference Guide* | Complete description of all variables and commands in the ACSPL+ programming language. |
| *SPiiPlus C Library Reference* | C/C++ library for host PC applications. This guide is applicable for all the SPiiPlus motion control products |
| *SPiiPlus .NET Library Reference* | .NET Methods, Properties, and Events for communication with the controller. |
| *SPiiPlus Programmer's Guide* | Guide for using the ACSPL+ high level programming language. |
| *SPiiPlus NTM Network Manager Hardware Guide* | Technical description of the SPiiPlus NTM Network Manager that provides complete communication control between the ACS Motion Control products on an EtherCAT network |

# Table of Contents

# 1. SPiiPlus MMI Application Studio Overview

SPiiPlus MMI Application Studio is a comprehensive environment for setting up and managing ACS Motion Control SPiiPlus motion controllers as well as control modules such as the MC4U product line.

## 1.1 System Requirements

The SPiiPlus MMI Application Studio can be run in any of the following Microsoft Windows® operating systems:

- Windows® 7 (32-bit and 64-bit)
- Windows® Server 2003 (32-bit and 64-bit)
- Windows® Server 2008 (32-bit and 64-bit)
- Windows® Server 2008 R2 (64-bit)
- Windows® 8.1 (32 bit and 64 bit)
- Windows Server 2012® (32 bit and 64 bit)
- Windows 10® (32 bit and 64 bit)

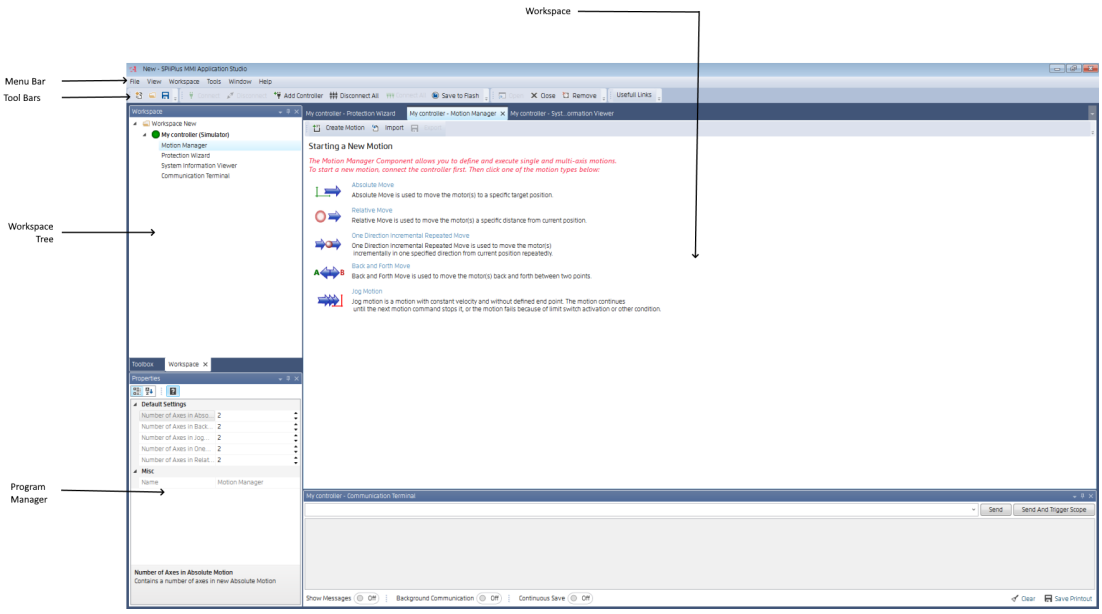
### 1.1.1 SPiiPlus MMI Application Studio

The SPiiPlus MMI Application Studio consists of the following:

- Adjuster Wizard
- Safety and Faults Configurator
- Application Wizard
- Program Manager
- Protection Wizard
- Motion Manager
- FRF Analyzer
- HSSI Manager
- Inputs/Outputs Manager
- Safety and Faults Monitor
- Scope
- Variables Manager and Watch
- Communication Terminal
- System Information Viewer
- Upgrade and Recovery Wizard

## 1.2 SPiiPlus MMI Application Startup Window

The SPiiPlus MMI Application Studio startup window:




### Workspace

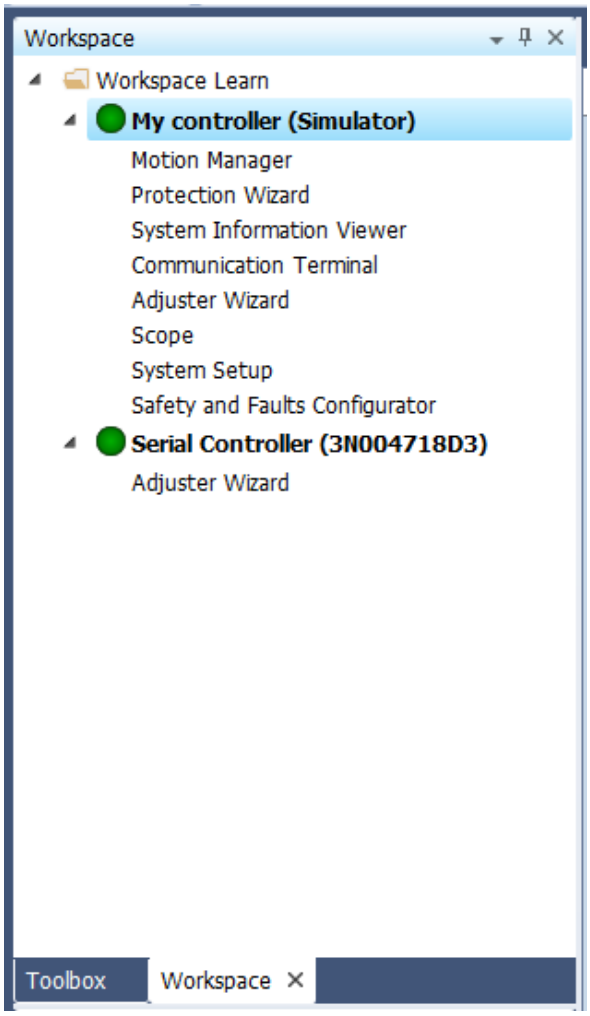
The Workspace is where all data is entered. The contents of the Workspace depend on the MMI option that is currently being used.

### Workspace Tree

The Workspace Tree displays all MMI components associated with the Workspace that you have activated.

The pod in which the Workspace Tree is displayed has two tabs:

- Workspace Tree Tab - clicking this tab displays the Workspace Tree.
- Toolbox Tab - clicking this tab displays the SPiiPlus MMI Application Studio tools.




### Status Bar

This displays messages of the MMI status and the status of any component action you have executed.

## 1.3 Menu Bar

The SPiiPlus MMI Application Studio Menu Bar has the following options:

| | | |
|---|---|---|
| **File** | New | Create a new Workspace file. |
| | Open | Open an existing Workspace file. |
| | Close Workspace | Close the currently displayed Workspace. |
| | Save Workspace | Save the Workspace data. |
| | Save Workspace As | Save the Workspace data to a different filename. |
| | Print | Access Print Manager for printing Workspace data. |
| | Recent | Display list of Workspace data files to enable selecting a specific Workspace. |
| | Exit | Close SPiiPlus MMI Application Studio. |
| **View** | Toolbox Window | Toggle Toolbox display on and off. |
| | Properties Window | Toggle Properties display on and off. |
| | Workspace Window | Toggle Workspace Tree display on and off. |
| **Workspace** | Add Controller | Add a new controller to the Workspace. |
| | Add File | Add an external file to the Workspace Tree - when selected, this file will open in the appropriate application. |
| | Set as Default | Sets the current Workspace as the default - it is displayed on the next time SPiiPlus MMI Application Studio is initiated. |

| | | |
|---|---|---|
| **Tools** | Language | Select language used by MMI.<br>Default: US English |
| | Style | Change the look of the MMI to suit your tastes. |
| | Settings | Select log settings, your options are:<br>> Off - Turns logging off.<br>> Errors - only errors are logged.<br>> Warnings - Errors & warnings are included in the log.<br>> Info - Errors, warnings and messages are included in the log. Verbose - Everything is included in the log.<br>In addition you can set the display time for ToolTips in seconds. |
| **Window** | Close All | Closes the display of all component windows in the Workspace. |
| | Tile Horizontal | If the **Floating** option is selected, each component window is displayed horizontally in the Workspace. |
| | Tile Vertical | If the **Floating** option is selected, each component window is displayed vertically in the Workspace. |
| | Cascade | If the **Floating** option is selected, each component window is displayed cascaded (one behind the other) in the Workspace. |
| **Help** | Contents | Displays the Help topics for the function in the active Workspace. |
| | Index | Displays a list of keywords from which you can perform a search for a particular help. |
| | Knowledge Center | Open the Knowledge Center in the web browser. |
| | About | Displays SPiiPlus MMI Application Studio version information. |

When you press **Alt + the Underlined Letter**, the selections for the Menu option are displayed.

## 1.4 Toolbars

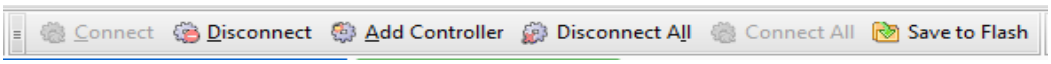
The SPiiPlus MMI Application Studio contains four built-in toolbars:

- **Workspace**

This toolbar contains Workspace handling function buttons. They are:

- Open a new Workspace
- Load a Workspace from a file on the computer
- Save the Workspace data to computer

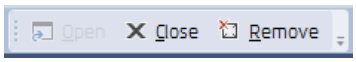
- **Controller**




This toolbar contains controller handling function buttons. They are:

- Connect - connects the selected controller
- Disconnect - disconnects the selected controller
- Add Controller - adds a new controller to Workspace
- Disconnect All - disconnects all controllers in Workspace
- Connect All - connects all controllers in Workspace

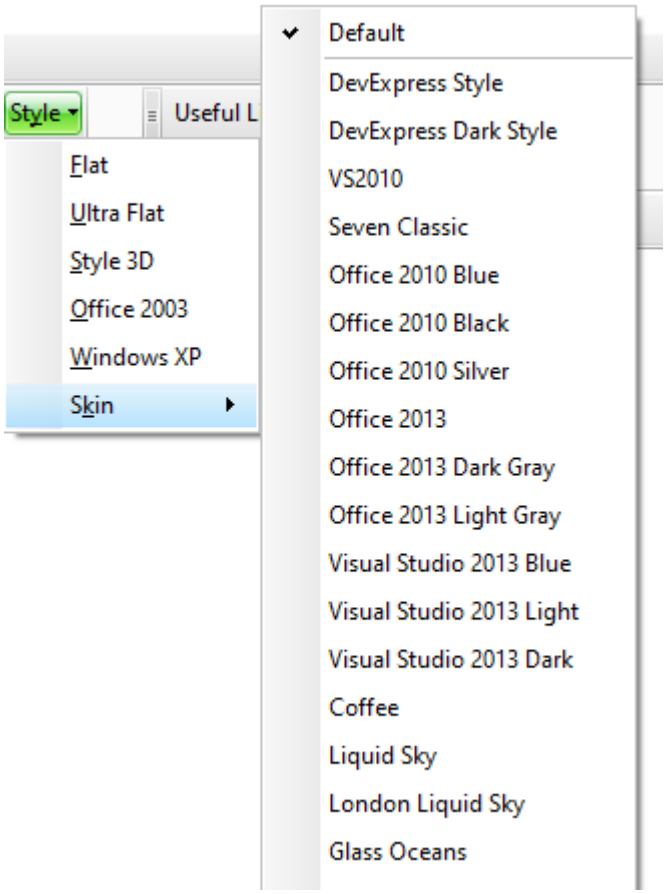
- **Component**




This toolbar contains function buttons for components in the Workspace tree. They are:

- Open - Displays the selected component in the Workspace
- Close - Closes the selected component in the Workspace
- Remove - Removes the selected component from the Workspace

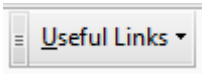
- **Style**




This button opens a menu offering style options:

- Flat
- Ultra Flat
- Style 3D
- Office 2003
- Windows XP
- Skin (Opens sub-menu with various options)

- **Useful Links**



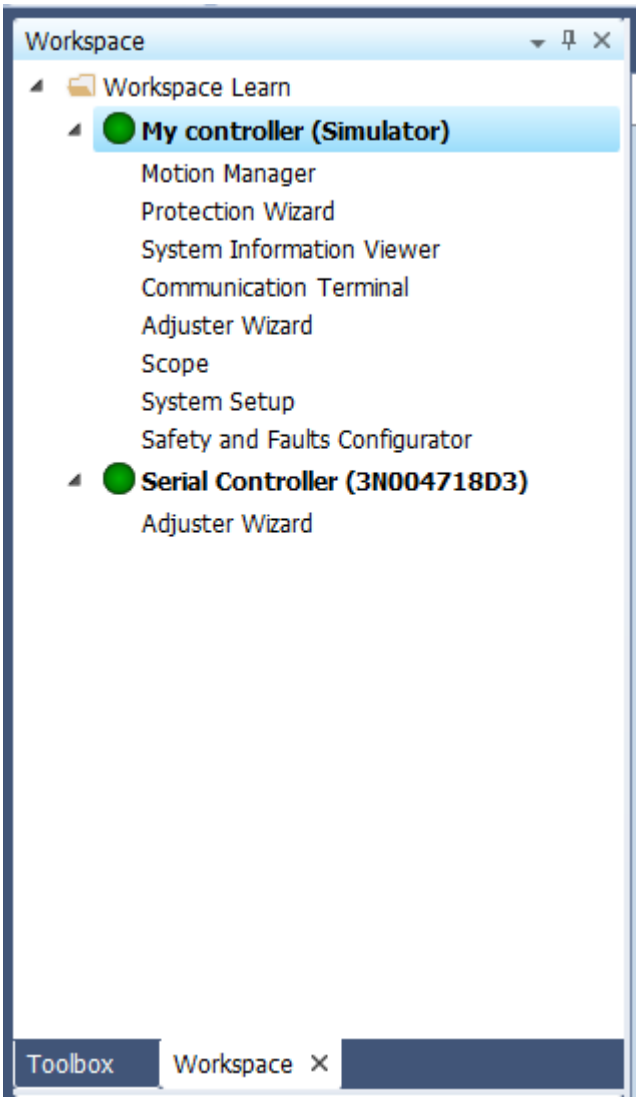

This toolbar opens a menu offering shortcuts to the following functions.

- Setup New System
- Tune an Axis
- Create Watch
- Save Application to PC

> Load Application to Controller

## 1.5 Workspace Tree Tab

The Workspace Tree displays all MMI functions that you have used for a particular session.




When you initiate SPiiPlus MMI Application Studio, the following functions are loaded by default:

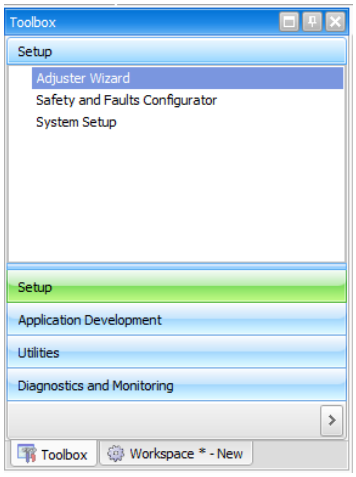
> Program Manager
> Motion Manager
> Communication Terminal

As you work within the MMI, each function that you employ is added to the list.

For details on the Workspace Tree operations, see Working with Workspace Tree.

## 1.6 Toolbox Tab

SPiiPlus MMI Application Studio Toolbox provides the following tools:




> **Setup**

  The Setup set of tools enables you to configure your SPiiPlus Motion Controller

> **Application Development**

  The Application Development set of tools enables you to program your SPiiPlus Motion Controller.

> **Utilities**

  The Utilities enable you to upgrade the SPiiPlus Motion Control firmware and software.

  Utilities also is your access point to Communication Terminal which enables you to send commands directly to the SPiiPlus Motion Controller.

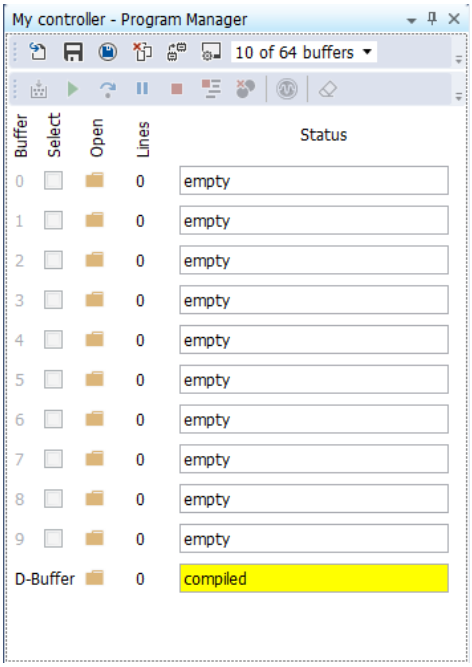
> **Diagnostics and Monitoring**

  The Diagnostics and Monitoring set of tools enables you to monitor, graphically through the Scope and the FRF Analyzer, and analyze the motion as well as keep abreast the values of critical variables.

## *1.7 Program Manager*

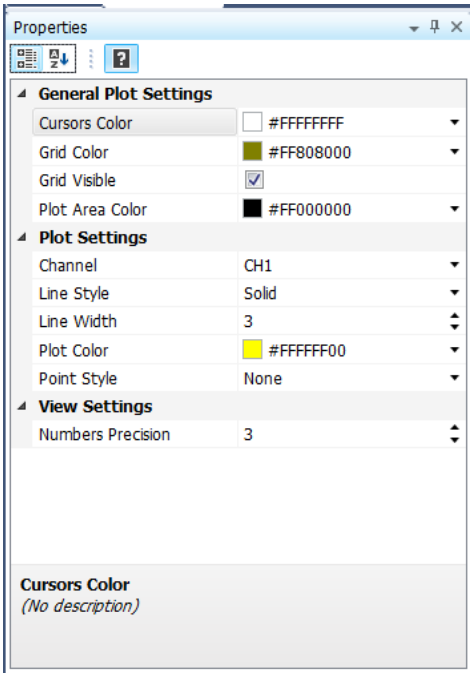
**Program Manager** is the principal tool for programming your SPiiPlus Motion Controller.

The Program Manager window displays information for the controller selected in the Workspace Tree.




## 1.8 Properties Window

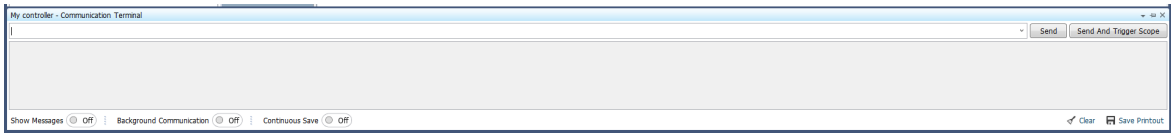
The Properties window enables you to set certain attributes for the object, such as color, numeric value, identifying name, and the like.




For details of the Properties window see Working with Workspace Tree.

## 1.9 Communication Terminal

The utility enables you to enter commands directly to the controller.




There are two types of commands that you can enter:

1. Terminal commands - these are commands specific to the Terminal and are not part of the ACSPL+, SPiiPlus C Library, or SPiiPlus COM Library command sets.
2. Certain ACSPL+ commands.

## 1.10 Emergency Stop Button

When you enter SPiiPlus MMI Application Studio, it connects to the SPiiPlus motion controller via the **SPiiPlus User Mode Driver** (UMD). The UMD displays the Emergency Stop button in the lower right side of the screen.


Clicking this button immediately stops all Program Buffers along with halting all motors and sets all axis offsets to 0.

By default, the button is always overlaid on top of the display. To change this, right-click the button and uncheck **Always On Top**.

## 1.11 Wizards

Wizards are special tools where your operations have been bundled into a single tool. For each option of a wizard there is a logical sequence of actions that need to be taken, and the wizard leads through them step-by-step. The SPiiPlus MMI Application Studio has the following wizards:

> Adjuster Wizard

The **Adjuster Wizard**, one of the Setup tools, enables you to improve the system and modify its behavior. The **Adjuster Wizard** provides tuning in terms of high-level parameters, namely parameters that have a clear meaning for you. Once you have assigned values to the high-level parameters, the **Adjuster Wizard** automatically transforms them into values for the low-level parameters, that is, controller parameters.

> Application Wizard

The **Application Wizard**, one of the Application Development tools, enables you to manage the controller program files, such as loading applications, saving applications, copying files to the controller, as well as loading/saving system and axis parameters.

> Upgrade and Recovery Wizard

The **Upgrade and Recovery Wizard**, one of the Utilities tools, enables you to upgrade your controller's firmware as well as to recover the firmware data should you need to reinstall the controller.

> Protection Wizard

The **Protection Wizard**, one of the Application Development tools, enables you to define the controller program Protection mode parameters, modify the Protection mode parameters, and view the Protection status. By applying Protection, you assure that your ACSPL+ programs are protected from unauthorized changes to system and axis parameters.

## 1.12 Graphic Monitors

SPiiPlus MMI Application Studio provides you with three graphic monitors to enable you to view and analyze the motion. These monitors are:

> FRF Analyzer

**Frequency Response Function** (FRF) is the steady-state transfer function of a dynamic system describing the relation between an input and an output as a function of frequency in terms of gain and phase. You have the option of displaying the data as either a Nyquist Diagram or a Bode Diagram.

> Scope

**Scope** presents a real time display of the values of selected motion variables. Signal values are tracked through various program variables, such as ACSPL+ variables, SP variables, user-defined variables, and the like. You can set the Scope to display any motion parameters you like.

> Sin Cos Encoder Compensation and Monitoring

The **Sin Cos Encoder Compensation** displays a Sin-Cos Lissajous curve graph of the wave form which enables you to read a controller's gain, phase and offsets compensation parameters (**SCGAIN**, **SCPHASE**, **SCSOFFS**, and **SCCOFFS**) and make adjustments where necessary to improve the encoder's response.

## 1.13 Variables Monitoring Tools

In addition to the graphic monitors, SPiiPlus MMI Application Studio provides you with a number of tools for viewing the values and status of various variables to aid you in diagnosing the motion. These tools are:

> Inputs/Outputs Manager

The Inputs/Outputs Manager displays the status of the input and output variables of the SPiiPlus controller as well as those of the HSSI if defined.

> Safety and Faults Monitor

The Safety and Faults Monitor displays the status of the SPiiPlus controller faults as defined through the Safety and Faults Configurator.

> Variables Manager and Watch

The Variables Manager displays the values currently stored in a program that is running. This includes ACSPL+, global and local variables.

- HSSI Manager

  The HSSI Manager is used for monitoring the status of HSSI channels. The manager displays the number and states of the HSSI channels.

## 1.14 Motion Manager

The Motion Manager is used control the axes' motion. It also displays when the motion is in progress, accelerating, slowing, and so forth. It enables you to operate each axis independently.

There are five types of motion available:

- **Absolute Move** - Moves the motor(s) to a specific target.
- **Relative Move** - Moves the motor(s) a specific distance from current position.
- **One Direction Incremental Repeated Move** - Repeatedly moves the motor(s) incrementally in one specified direction from current position.
- **Back and Forth Move** - Moves the motor(s) back and forth between two specified points.
- **Jog Motion** - Moves the motor(s) with constant velocity and without a defined end point. The motion continues until the next motion command stops it, or the limit switch has been activated, or some other condition.

# 2. MMI Basics

This chapter presents the procedures used in performing the basic functions in the SPiiPlus MMI Application Studio Workspace.

The basic functions covered by this chapter include:

- Workspace Operations
- Handling Controllers
- Handling Workspace Components
- Working with Workspace Tree
- Working with Wizards

## 2.1 Workspace Operations

Workspace operations include:

- Creating a New Workspace
- Saving a New Workspace
- Loading a New Workspace
- Closing the Workspace
- Handling Controllers
- Handling Workspace Components
- Working with Workspace Tree

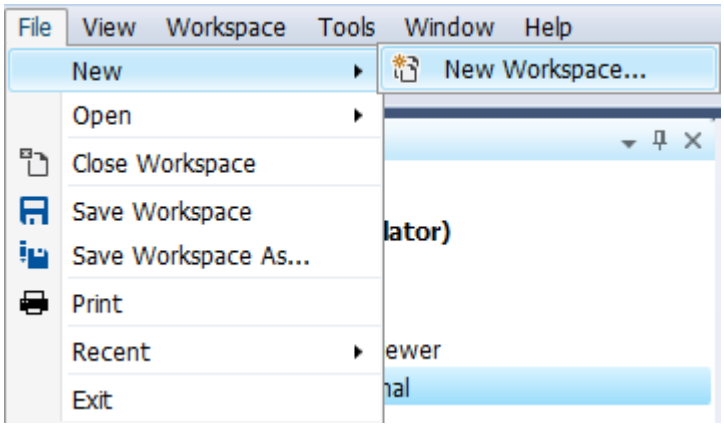
### 2.1.1 Creating a New Workspace

You may create as many Workspaces as you want. Some reasons for creating Workspaces are:

- Retaining certain setups
- Maintaining separate Workspaces for each controller

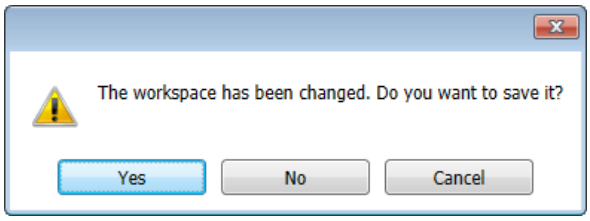
To create a new Workspace:

1. Click **File** in the Main Menu.
2. Select **New**.
3. Click **New Workspace**.

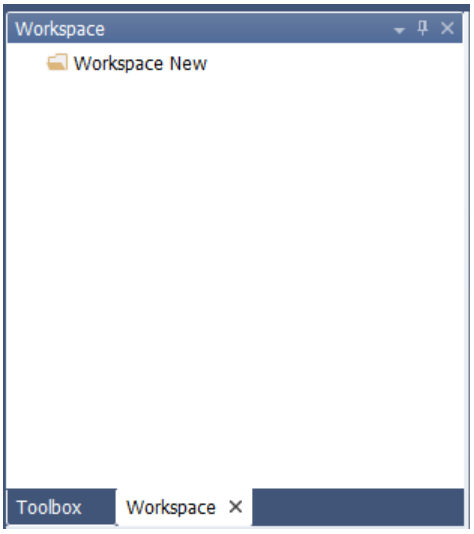



You can also create a new Workspace by pressing **Ctrl+N**, or by clicking in the top bar.

4. You are prompted with:




5. If you want to save the current Workspace, click **Yes**; otherwise click **No**. The new Workspace is loaded.

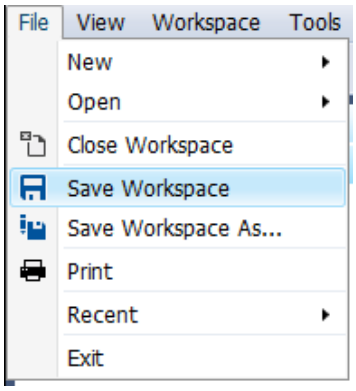



6. To rename the **Workspace** click the Workspace folder and type in the new name.

### *2.1.2 Saving a New Workspace*

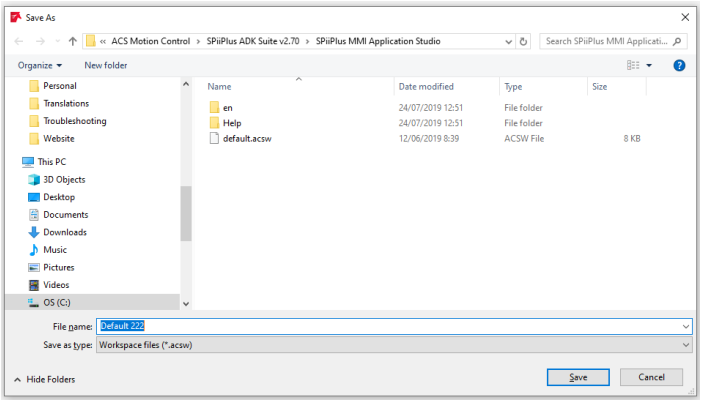
Once created, you save the new Workspace by:

1. Click **File** in the Main Menu.
2. Select **Save Workspace**.




You can also save the new Workspace by pressing **CTRL+S**, or by clicking in the toolbar.

3. You are prompted with:



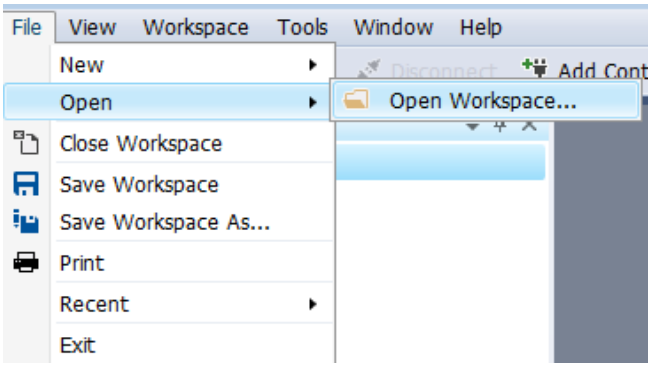

4. Type in the filename and click **Save**.

The Workspace configuration is stored in files having **.acsw** file extension.

### *2.1.3 Loading a New Workspace*

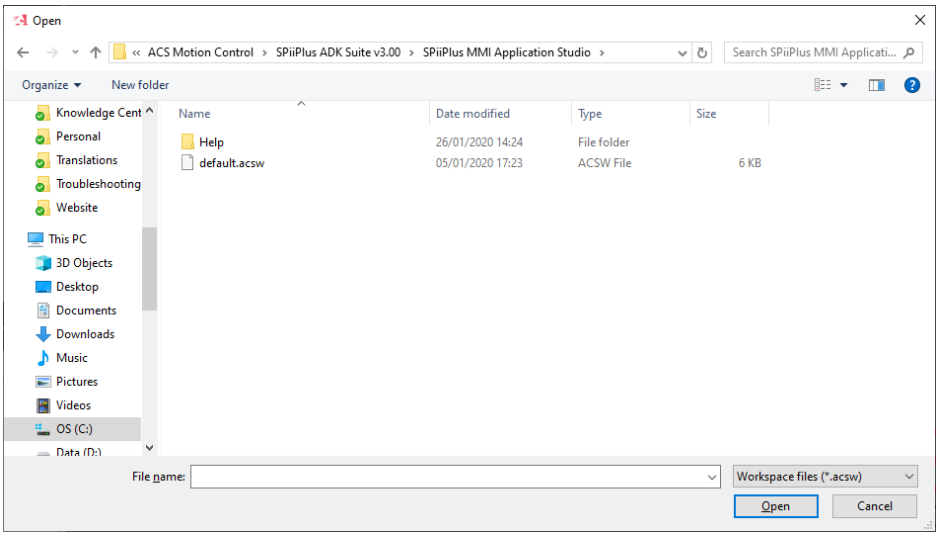
To load a new Workspace:

1. Click **File** in the Main Menu.
2. Select **Open**.




You can also open the new Workspace by pressing **Ctrl+O**, or by clicking toolbar.

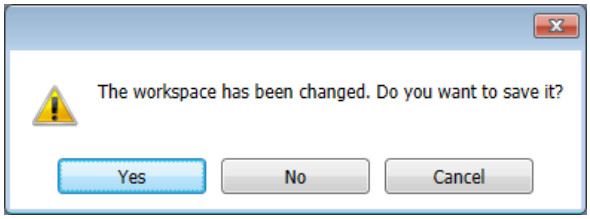
3. You are prompted with:




4. Use the Open window to browse for the location of the file, select the file and click **Open**.

The Workspace configuration is stored in files having **.acsw** file extensions.

5. You are prompted with:

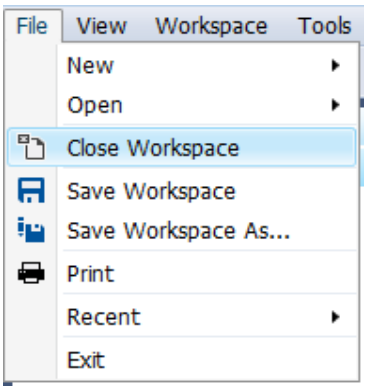



6. If you want to save the current Workspace, click **Yes**; otherwise click **No**. The new Workspace is loaded.

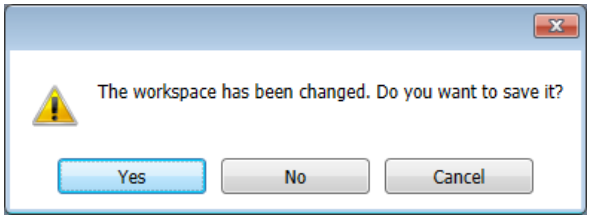
### 2.1.4 Closing the Workspace

To close the Workspace:

1. Click **File** in the Main Menu.
2. Select **Close Workspace**.




3. You are prompted with:




4. If you want to save the current Workspace, click **Yes**; otherwise click **No**.

## 2.2 Handling Controllers

The Workspace controller handling functions are:

- Adding a Controller
- Disconnecting a Controller
- Connecting a Controller
- Setting Controller Properties
- Saving to Flash
- Controller Reboot

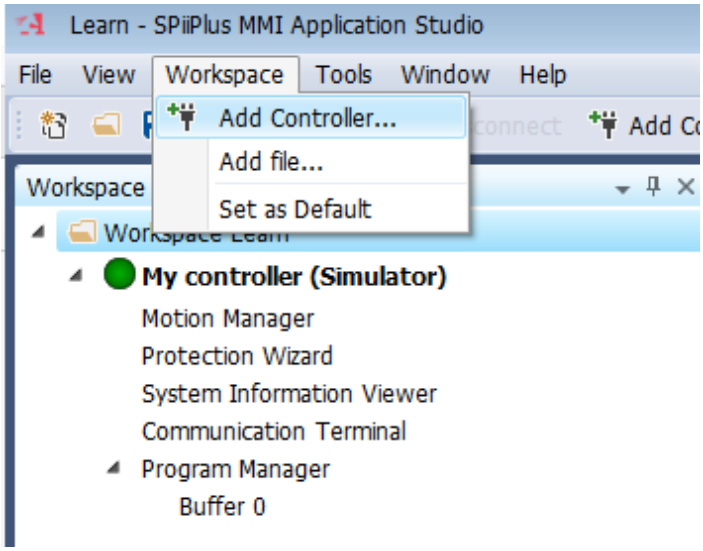
> Setting Factory Defaults

> Removing a Controller

### *2.2.1 Adding a Controller*

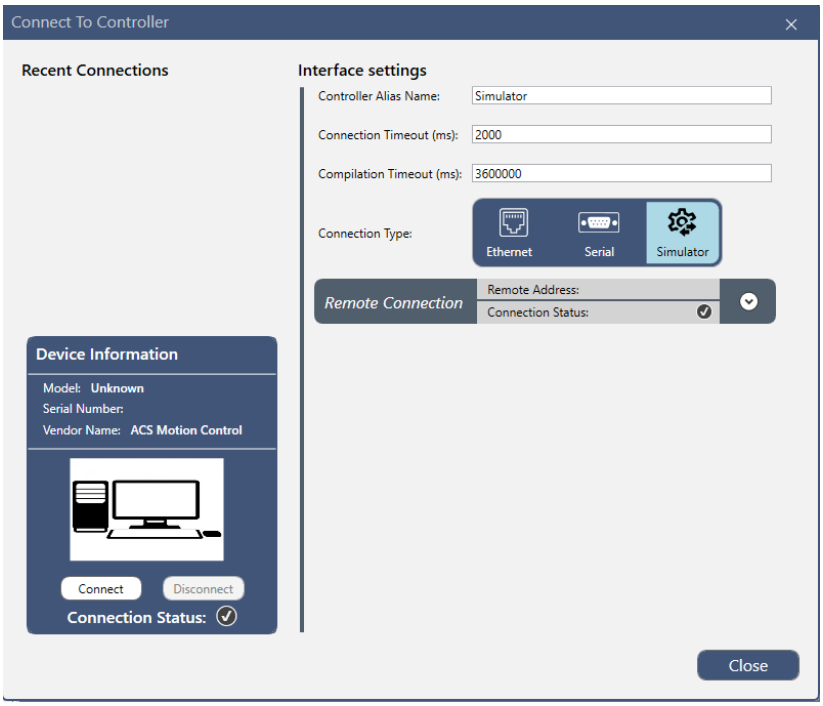
To add a controller:

1. Click **Workspace** on Menu bar, and select **Add Controller**.

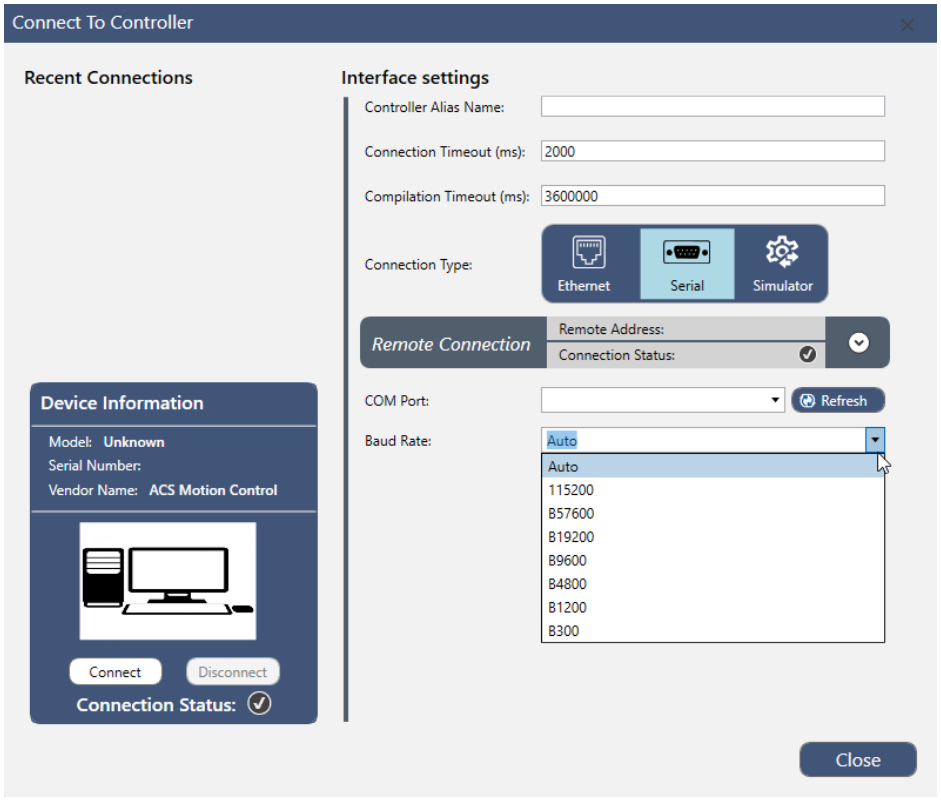



You can also click **ADD CONTROLLER** on the toolbar or press **ALT+A**.

The Connect to Controller dialog is displayed:

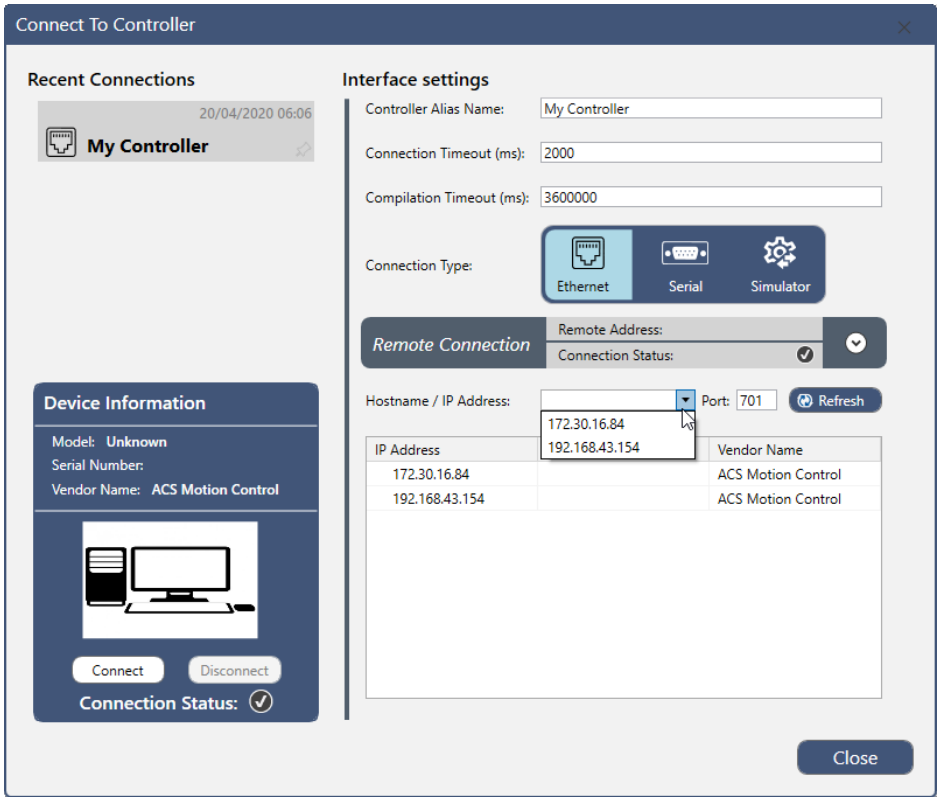



2. Enter an identifying name in the **Controller Alias Name** field.
3. Enter the connection timeout value, in milliseconds, in the **Connection Timeout** field.
4. If your computer is connected directly to the controller, select **Serial**, and
   > Select the communication port from the **Port** dropdown list. If not all the ports are displayed, press the Refresh button.
   > Select the baud rate from the **Rate** dropdown list:




> Click **Connect**.

5. If your computer is connected to the controller through a LAN, select **Ethernet**.

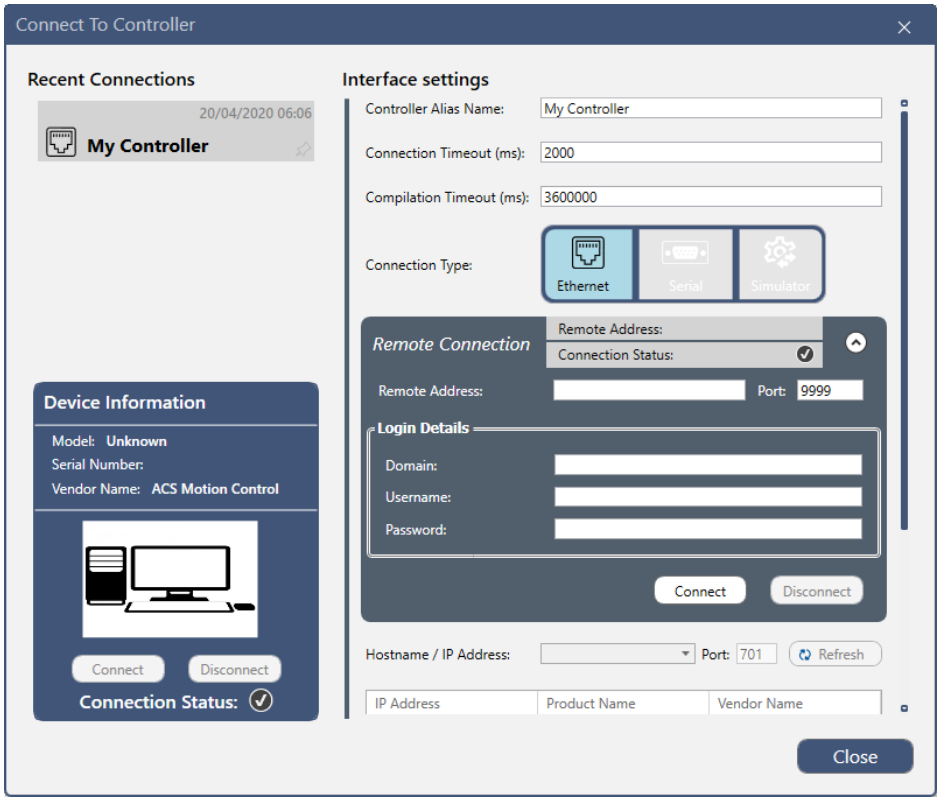



> Select the IP Address from the **Controller IP Address** dropdown list.

The factory default TCP/IP address is 10.0.0.100.

> Click **Connect** .

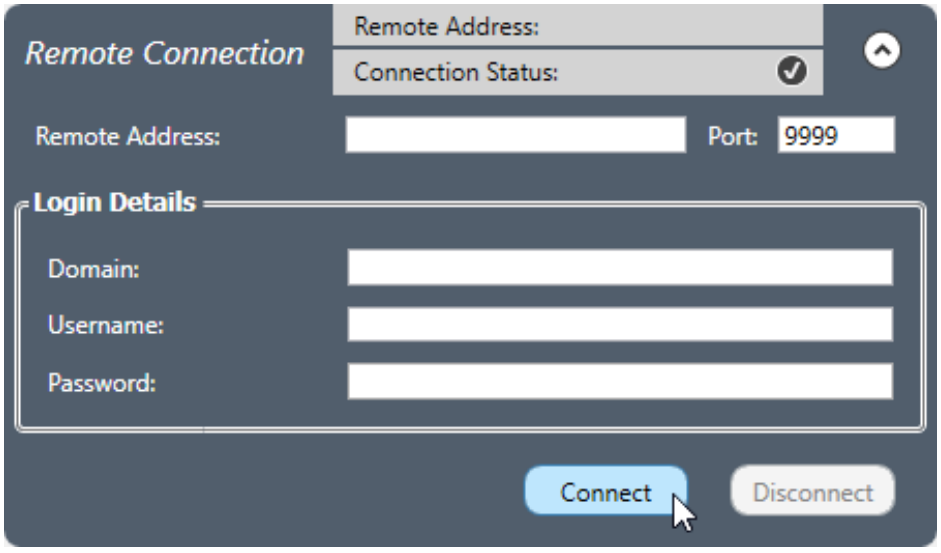
6. If the controller is connected to a different computer, you have the option of establishing a remote connection by selecting the **Remote Connection** checkbox.




> Enter the IP of the remote computer in the **Remote Address** field.

> Enter the port number of the remote computer in the **Remote Port** field.

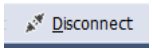
> If the remote computer has login requirements, enter the Domain name, Username and Password in the **Login Details**.




> Click **Connect**. The controller is automatically stored in the Workspace folder and appears in the Workspace Tree.

### 2.2.2 Disconnecting a Controller

To disconnect a controller:

1. In the Workspace Tree select the controller you want to disconnect.
2. Click **Disconnect** ( 

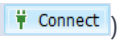
) or press **Alt+D**.

To disconnect all the controllers click **Disconnect All** or press **Alt+L**.

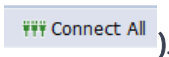
The "LED" ( ) indicating that the controller is connected goes out.

### 2.2.3 Connecting a Controller

To connect a controller that has been disconnected:

1. In the Workspace Tree select the controller you want to connect.
2. Click **Connect** ( 

) or press **Alt+C**.

To connect all the controllers that have been disconnected click **Connect All** ( 

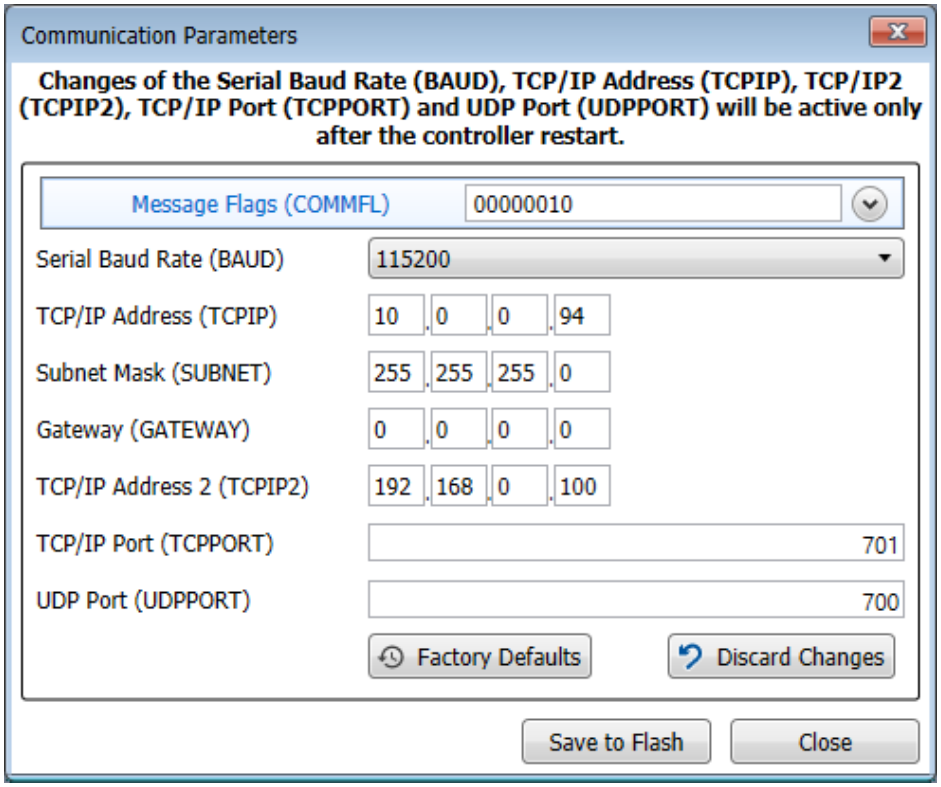
).

The "LED" ( ) indicating that the controller is connected is on.

### 2.2.4 Setting Controller Properties

This option enables you to set various communication properties for your controller.

To set controller properties: click the controller, the SPiiPlus Communication Parameters window is displayed:

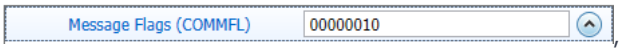



This window allows you to set the following communication parameters:

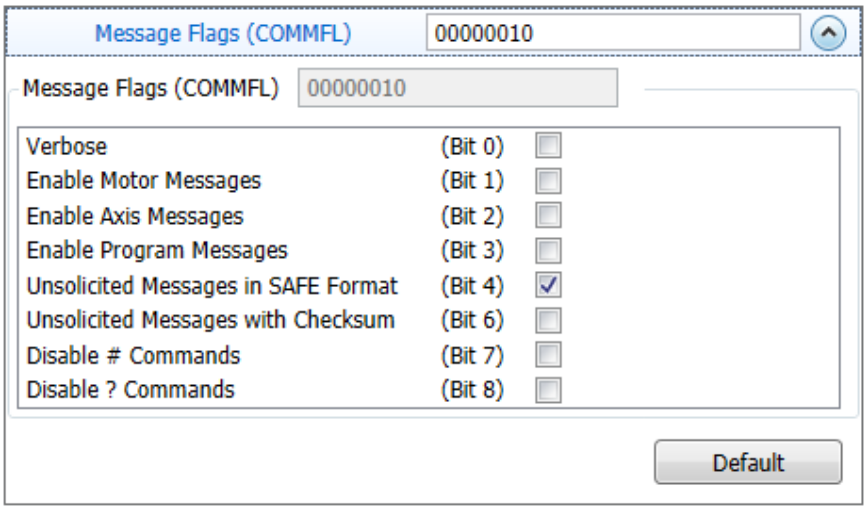
- COMMFL
- DISPCH
- BAUD
- TCPIP
- SUBNET
- GATEWAY
- TCPIP2
- TCPPORT
- UDPPORT

### *2.2.4.1 COMMFL*

**COMMFL** is a scalar variable containing a set of 9 bits that affect controller communication. To set the bits:

1. Click 

, the Message Flags Form is displayed:




2. You can either type the number in the **Message Flags (COMMFL)** field, or select the bits you want to enable by clicking the appropriate checkbox. The meanings of the bits are:

| Bit No. | Description |
|---|---|
| 0 | 1 = Controller includes an extended explanation with error code messages. |
| 1 | 1 = Enables motor messages. |
| 2 | 1 = Enables Axis messages. |
| 3 | 1 = Enables Program messages. |
| 4 | 1 = Controller sends unsolicited messages in Safe communication format. |
| 6 | 1 = A checksum is included in unsolicited messages. Normally you do not need to enable this bit. |
| 7 | 1 = The controller ignores any command received via communication channels except the queries that start with the '?' character. The bit is not effective if the controller is in the Configuration mode. The default value is 0. |
| 8 | 1 = The controller ignores any query received via communication channels. The bit is not effective if the controller is in Configuration mode. The default value is 0. |

3. Click **Close** to save the settings. Or you can click **Default** to enter the default settings.

### *2.2.4.2 DISPCH*

**DISPCH** is a scalar integer variable that defines the communication channel between the controller and a host application, MMI or any device connected to the controller's communication ports.

**DISPCH** is relevant only to messages sent with **DISP** and **SEND** (referred to as "Unsolicited Messages"). In order to view unsolicited messages in the window, select the check box in the lower right corner of the Terminal window to enable **Show Unsolicited Messages**.

If **DISPCH** specifies a valid communication channel, all unsolicited messages (messages that are sent with **DISP** and **SEND** from the program buffers) are sent to this channel irrespective of the channel used for immediate commands.

To set **DISPCH**: Type the required number in the **Default Channel (DISPCH)** field. Your choices are:

| Channel Number | Description |
|---|---|
| -1 | No default channel is specified, the command uses the last channel activated by the host. |
| -2 | All channels (default) |
| 1 | Serial port 1 |
| 2 | Serial port 2 |
| 6 | Ethernet network (TCP) |
| 7 | Ethernet network (TCP) |
| 8 | Ethernet network (TCP) |
| 9 | Ethernet network (TCP) |
| 10 | Ethernet Point-to-Point (UDP) |
| 12 | PCI bus |

### *2.2.4.3 BAUD*

**BAUD** is an integer variable that defines the serial communication rate, given in bits per second.

Enter the required Baud rate from the dropdown menu by clicking the down arrow of the **Serial Baud Rate (BAUD)** field. The options are:

- 300
- 1200
- 4800
- 9600

- 19200
- 57600
- 115200 (default)

#### *2.2.4.4 TCPIP*

**TCPIP** sets IP Address for the 1st Ethernet Port. If **TCPIP** has a non-zero value, the controller uses the value as its TCP/IP address.

The default address is 10.0.0.100.

To set the IP Address enter the address in the **TCP/IP Address (TCPIP)** field.

If you set the **TCPIP** to: **0.0.0.0**, the controller uses the DHCP protocol to retrieve the network configuration from the DHCP server. The network configuration received from the DHCP server includes the following parameters:

- Controller's TCP/IP address
- Subnet mask
- Gateway address

#### *2.2.4.5 SUBNET*

**SUBNET** is used to determine to what subnet an IP address belongs.

The **SUBNET** value consists of four individual bytes, each being a decimal number between 0 and 255, separated by dot. The right most value is the least significant byte.

The default is 255.255.255.0.

To set the Subnet Mask enter the values in the **Subnet Mask (SUBNET)** fields.

If controller is configured to get network settings from DHCP server, that is, **TCP/IP** is set to **0.0.0.0**, **SUBNET** receives its value from the DHCP server subnet mask.

#### *2.2.4.6 GATEWAY*

**GATEWAY** is the address of a network router that serves accessing another network segments.

The **GATEWAY** value consists of four individual bytes, each being a decimal number between 0 and 255, separated by dot. The right most value is the least significant byte.

The default is 0.0.0.0.

To set the Gateway address enter the values in the **Gateway (GATEWAY)** fields.

If controller is configured to get network settings from DHCP server, that is, **TCP/IP** is set to **0.0.0.0**, **GATEWAY** receives its value from the DHCP server gateway address.

#### *2.2.4.7 TCPIP2*

**TCPIP2** sets IP Address for the 2nd Ethernet Port. If **TCPIP2** is zero, the address will be automatically obtained at the controller start-up through the DHCP protocol.

The default address for second Ethernet port is 192.168.0.100.

To change the IP Address enter the address in the **TCP/IP Address2 (TCPIP2)** field.

#### 2.2.4.8 TCPPORT

**TCPPORT** defines Ethernet ports in the controller for TCP. The value that can be set in **TCPPORT** ranges between 0 and 65536. By default, this variable is set to 701.

In order to establish communication with the controller through a port different from default port numbers, do the following:

1. Set **TCPPORT** to a value other than 701 (you can use the up and down arrows of the **TCP/IP Port (TCPPORT)** field to increment/decrement the value).

Some of the ports are used by the controller OS and cannot be used. It's recommended to use ports starting from 1024.

2. Save system parameters to the flash (see Saving to Flash).
3. Restart the controller.
4. Try to establish communication using new ports by providing them in the client user application. If communication is not established, try to set other values.

#### 2.2.4.9 Controller Parameters Function Buttons

The Controller Parameters tab has the following function buttons:

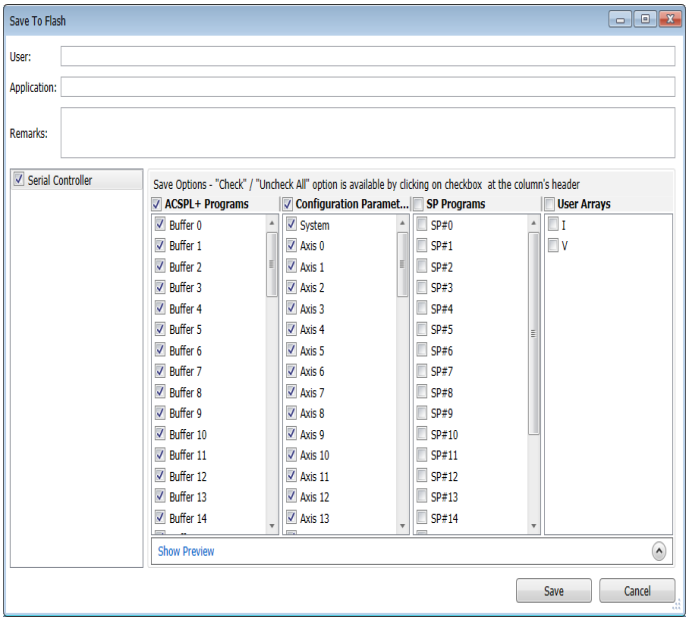
| | |
|---|---|
| Factory Defaults | Loads factory default values into the fields. |
| Discard Changes | Removes any values you have entered and restores the previous values. |

### 2.2.5 Saving to Flash

You use this function to save your program and configuration data to the selected controller's flash memory.

To save to flash:

1. In the Workspace Tree select the controller to whose flash memory you want to save the data.
2. Click **Save to Flash** ( Save to Flash ). The Save to Flash window is displayed:



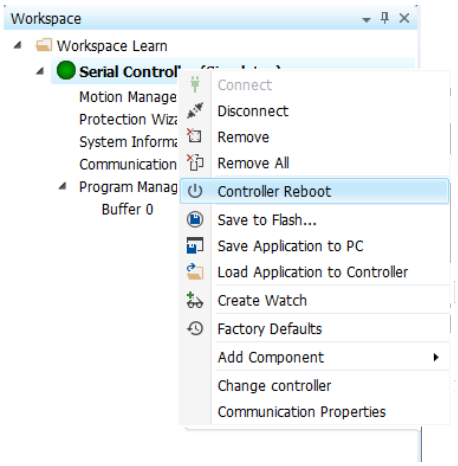

3. Enter your user name in the **User** field.
4. Enter the application name in the **Application** field.
5. You can, if you desire, enter free text remarks in the **Remarks** field.
6. By default all data is selected. Select the data you want to load into the controller by clicking (thereby deselecting) the checkbox of the data you do not want to be saved.
7. Click **Save**.

### *2.2.6 Controller Reboot*

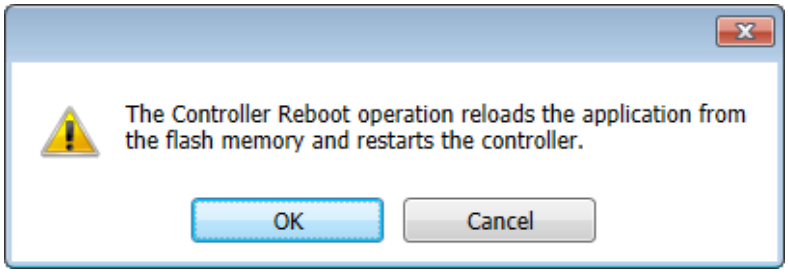
You use this function whenever you've made changes to the memory of the SPiiPlus Motion Controller. You have to reboot the controller in order to activate these changes.

To reboot the controller:

1. Right-click the controller in the Workspace Tree and select **Controller Reboot** from the menu.




2. The following prompt appears:




3. Click **OK**. The rebooting operation begins.


4. Wait until you are notified of a successful reboot:


5. Click **OK**.

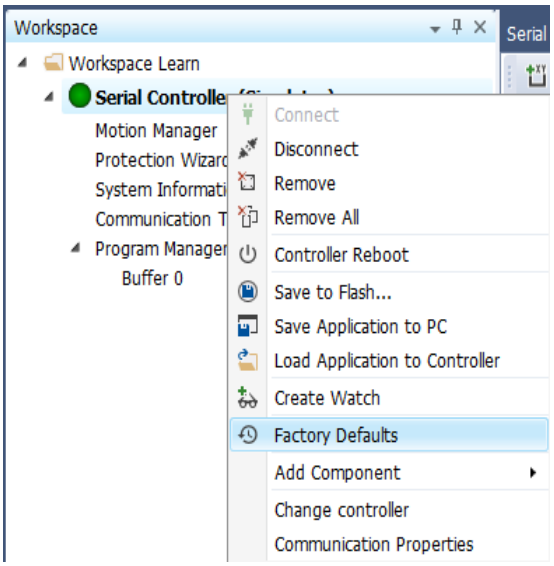
### *2.2.7 Setting Factory Defaults*

You use this function to set factory default values into the controller.

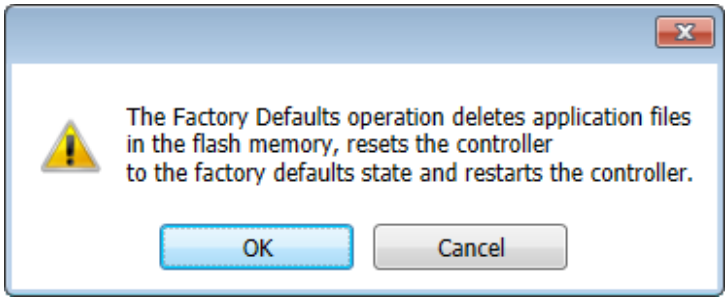
While this option exists, it should be used with some care in that the factory defaults may be inconsistent with your system, and may cause you to lose communication with the controller.

To set the factory defaults into the controller:

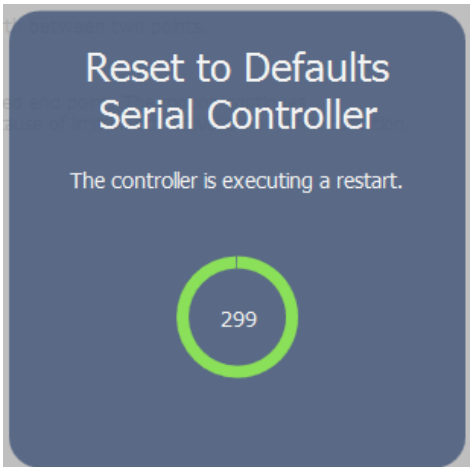
1. Right-click the controller in the Workspace Tree and select **Factory Defaults** from the menu.



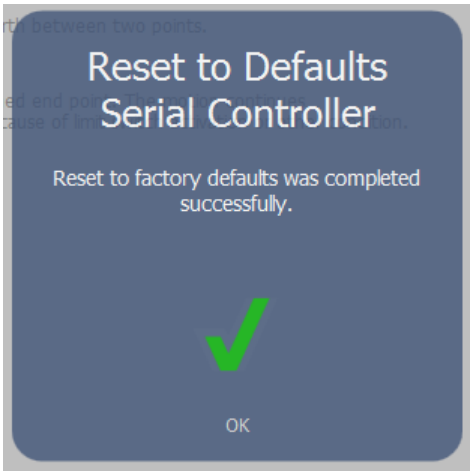

2. The following prompt appears:




3. Click **OK**. The rebooting operation begins.




4. Wait until you are notified of a successful reboot:

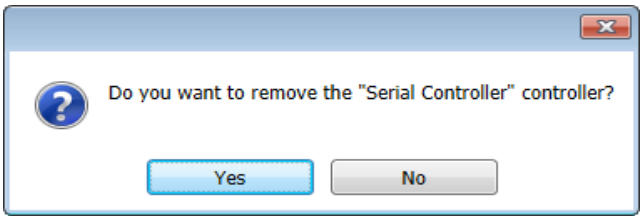



5. Click **OK**.

### 2.2.8 Removing a Controller

To remove a controller:

1. In the Workspace Tree select the controller you want to remove.
2. Click **Remove** ( Remove ) on the toolbar or press **Alt+R**. You are asked:




3. Click **Yes**.

## 2.3 Handling Workspace Components

Workspace components consist of all the various SPiiPlus MMI Application Studio functions and utilities that can be applied to the controller, such as the Program Manager, Safety and Faults Configurator, Motion Manager, Scope, and the like.

The Workspace component handling functions are:

> Opening a Component
> Closing a Component
> Removing a Component

These functions are activated by clicking the appropriate button on the Toolbars.

### 2.3.1 Opening a Component

To activate a component in the Workspace Tree:

1. Select the component.
2. Click **Open** on the toolbar or press **Alt+O**. The component is displayed in the Workspace.

### 2.3.2 Closing a Component

To close a component:

1. Select the component.
2. Click **Close** ( Close ) on the toolbar or press **Alt+C**. The component is no longer displayed in the Workspace.

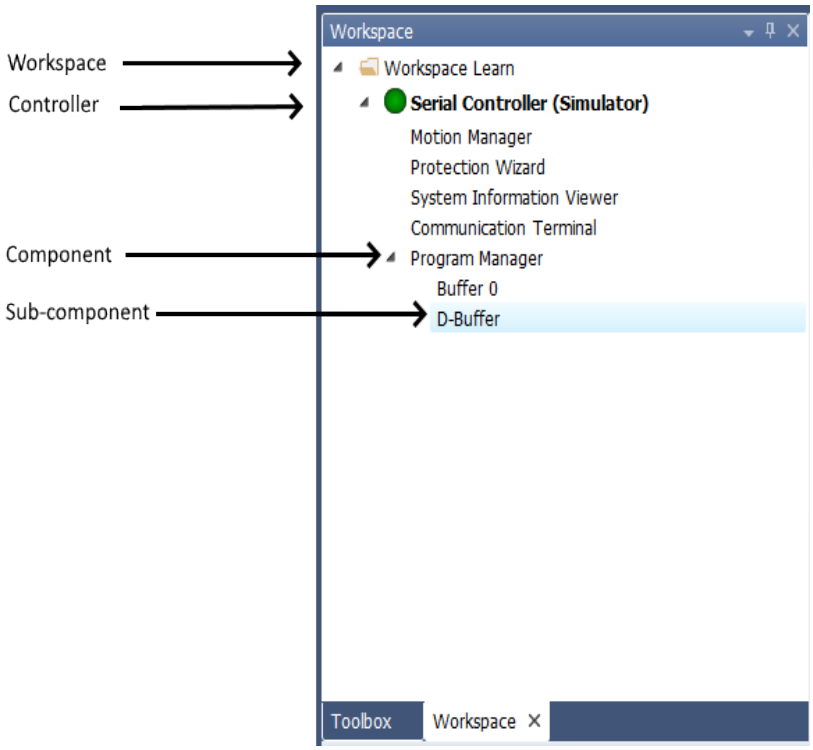
### 2.3.3 Removing a Component

To remove a component:

1. Select the component.
2. Click **Remove** ( Remove ) on the toolbar or press **Alt+R**. The component is no longer displayed in the Workspace and it has been removed from the Workspace Tree.

## 2.4 Working with Workspace Tree

The Workspace Tree is constructed as follows:



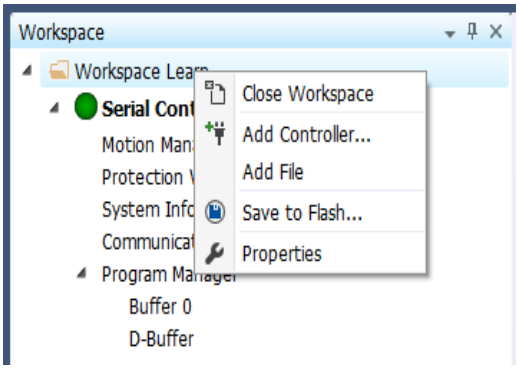

- **Workspace** - Top level
- **Controller** - Level Two
- **Components** - Level Three
- **Sub Component** - Level Four

Each level has functions associated only with that level that are activated by right-clicking the element in the level.

Only Program Manager and Variables Manager and Watch have sub-components, but their right-click options are the same as level three components.

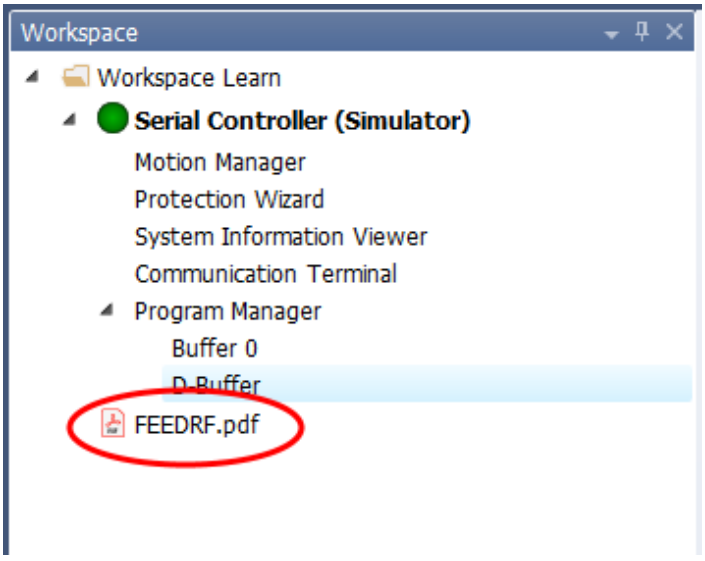
### 2.4.1 Workspace Right-Click Options

Right-clicking the Workspace top level displays the following list of options:




The options that are available are:

- **Close Workspace** - see Closing the Workspace
- **Add Controller** - see Adding a Controller
- **Save to Flash** - see Save to Flash
- **Add File** - enables you to add a shortcut to any file in your system, for example:




When you click the file, it is opened in the appropriate application. When you save the Workspace, the shortcut is saved as part of the Workspace.

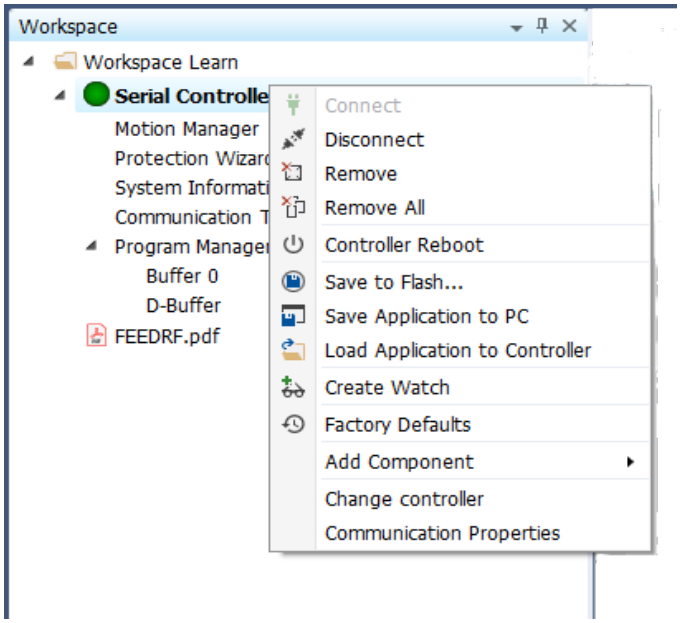
This function can also be selected from list in the **Workspace** option of the Main Menu.

A good use for this capability is to link ACS Motion Control documentation, such as the *SPiiPlus Command & Variable Reference Guide* and the *SPiiPlus ACSPL+ Programmer's Guide* for looking up information while working in the SPiiPlus MMI Application Studio.

> **Properties** - this activates the Workspace **Properties** window.

### *2.4.2 Controller Right-Click Options*

Right-clicking the controller displays the following list of options:




The options that are available are:

> **Connect** - this is the same as Connecting a Controller.

> **Disconnect** - this is the same as Disconnecting a Controller.

> **Remove** - this is the same as Removing a Controller

> **Remove All** - this removes all components under the selected controller.

> Controller Reboot - this reboots the selected controller, generally used when you have saved new data to the controller's flash and you want it to take effect.

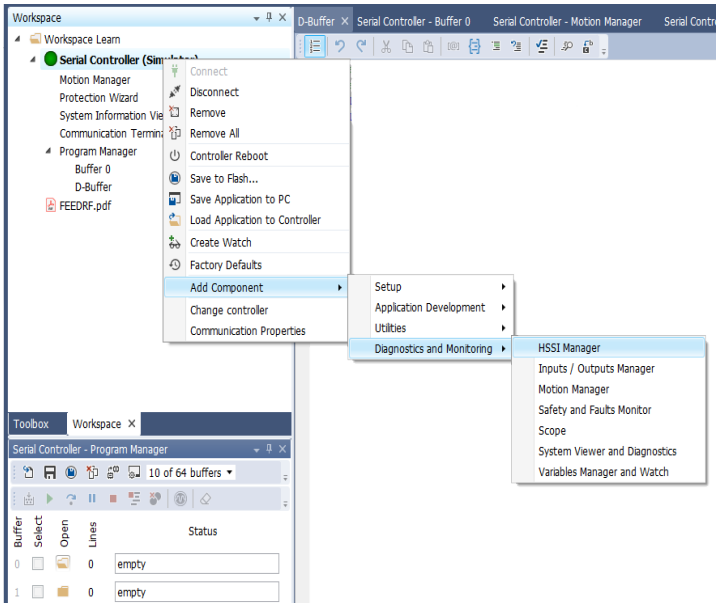
> **Save to Flash** - this is the same as Save to Flash.

> **Factory Defaults** - this restores the factory-set default values of the selected controller, see Setting Factory Defaults.

> **Add Component** - this enables you add a component to the Controller branch.

The **Add Component** is a very handy option in that it saves you time navigating through the SPiiPlus MMI Application Studio to activate a particular component.

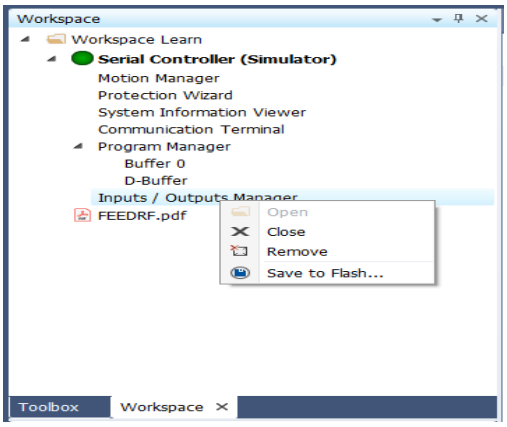
For example, say you want to activate the **HSSI Manager**, rather than display **Toolbox** and selecting **Utilities**, then selecting **HSSI Manager**, all you need to do is drill down through this option:




> **Properties** - this activates the controller's **Properties** window, see Working with Properties Window.

### *2.4.3 Component Right-Click Options*

Right-clicking the component displays the following list of options:




The options that are available are:

> **Open** - this is the same as Opening a Component.

- **Close** - this is the same as Closing a Component.
- **Remove** - this is the same as Removing a Component.
- **Save to Flash** - this is the same as Save to Flash.
- **Properties** - this activates the component's **Properties** window, see Working with Properties Window.

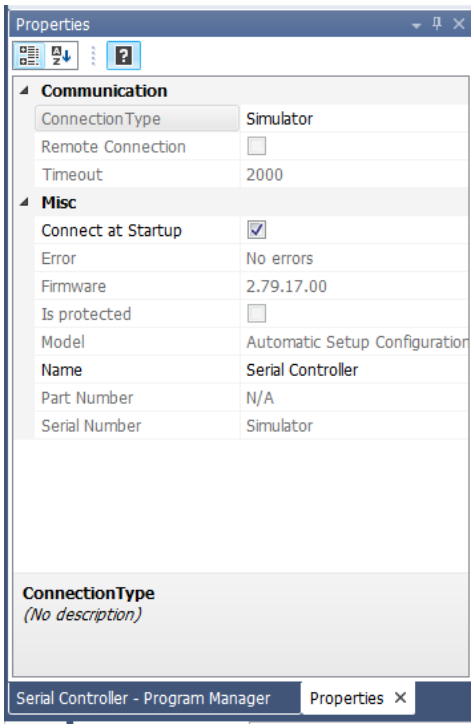
## 2.5 Working with Properties Window

All SPiiPlus MMI Application Studio components have an associated **Properties** window. The properties often determine how a component is displayed or how it is communicated with, and just as often it merely determines the name of the component.

To display a component's Properties window:

1. Select the component in the Workspace Tree.
2. Press **Ctrl+Alt+P** (or select **Properties Window** from **View** in the Main Menu, or use the Workspace right-click option - see Working with Workspace Tree).

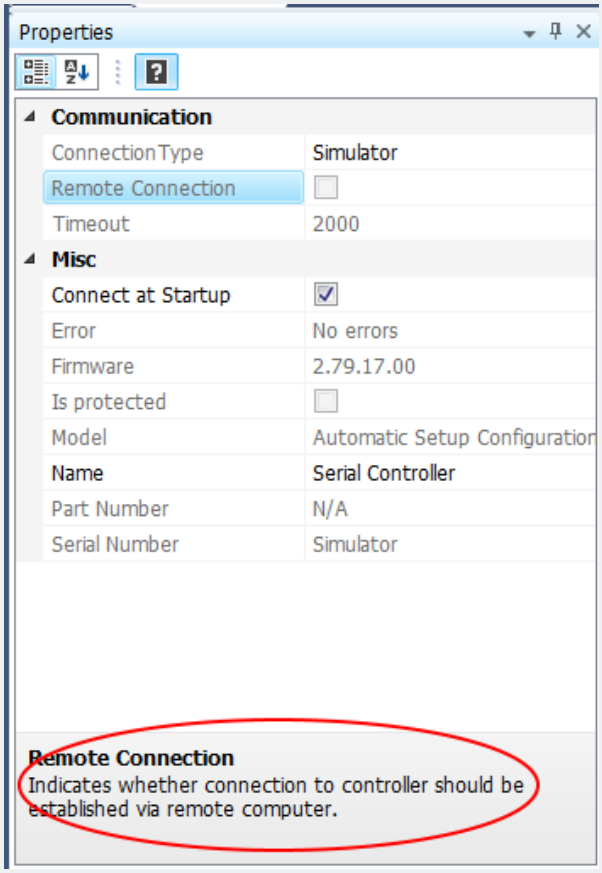
   The Properties pane is displayed, for example:




Properties that are highlighted in **bold** can be changed in the pod by clicking the field to input-focus it. Those properties that are grayed-out are for information only and cannot be changed through the Properties pod.

There are three action buttons associated with the Properties window:

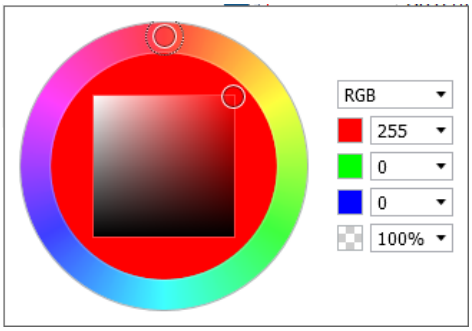
| | |
|---|---|
|  | **Categorize** - displays the Properties according to various categories. |

| | |
|---|---|
|  | **Alphabetic** - displays the Properties in alphabetic order. |
|  | **Show Description** - displays a short description of the selected property at the bottom of the pod, for example  |



### Setting Color Properties

The Color property is very common to most of the Property windows. You set the Color property by clicking  in the color field which displays the color options from which you can choose.

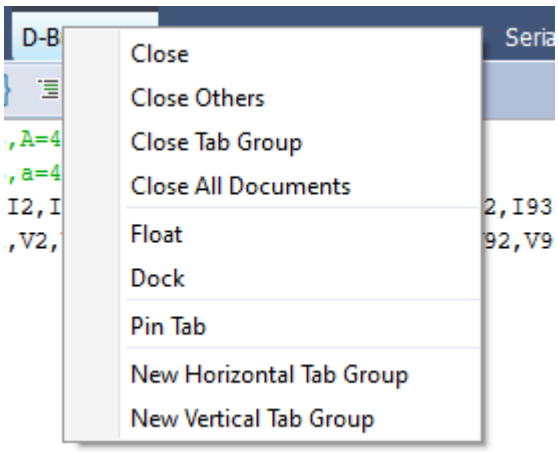



The Custom tab presents gradations of the most common colors. You select the color by clicking the required box.

## 2.6 Workspace Tab Options

When you add a component, SPiiPlus MMI Application Studio opens its data window in a separate tab in the Workspace. You navigate from window to window by clicking its tab.

Right-clicking the tab displays the following menu:




Your options are:

- **Close** - closes the window (you can reopen it by clicking the component in the Workspace Tree).
- **Close Others** - closes the other windows in the tab group (you can reopen them by clicking the component in the Workspace Tree).
- **Close Tab Group** - closes all windows and removed the pod containing the tab group (you can restore windows by clicking the component in the Workspace Tree).
- **Close All Documents** - closes the window (you can reopen it by clicking the component in the Workspace Tree).
- **Float**- the window is displayed separately apart from the rest of the windows and you can resize it or move it to any place on the screen.
- **Dock** - the window is arranged in a line together with the rest of the windows (default).
- **Pin Tab** - the window is pinned to the left side of the tab group

- **New Horizontal Tab Group** - the area of the tab group is split in half horizontally and the tabbed window is moved to the lower area
- **New Vertical Tab Group** - the area of the tab group is split in half vertically and the tabbed window is moved to the area on the right

## 2.7 Panel Controls

To expand the view of the Workspace, you have the following available controls:

Clicking  places the panel on the side of the window:


The panel display can be restored by placing the mouse cursor over it. Then when you move the mouse cursor outside the pod, it slides back to the side.

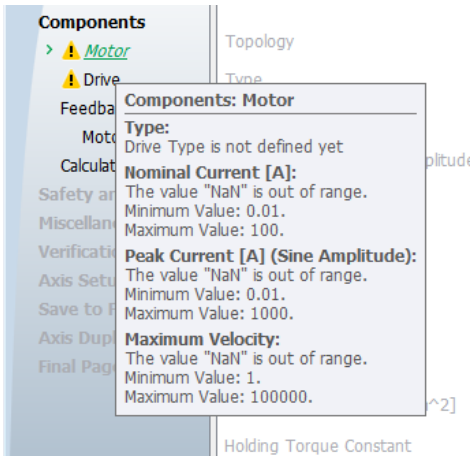
If you want to restore its display permanently, click  .

You can cancel its display by clicking  .

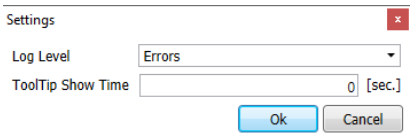
## 2.8 Tool Tips

In addition to the instructions that appear at the bottom of the Workspace window, if you allow the mouse cursor to linger on any field in the window, a popup Tool Tip is displayed to aid you in

connection with the data in the field. For example:




You can set the length of time, in seconds, that the Tool Tip remains displayed by selecting **Tools➔Settings** from the Main Menu. This displays a dialog window:

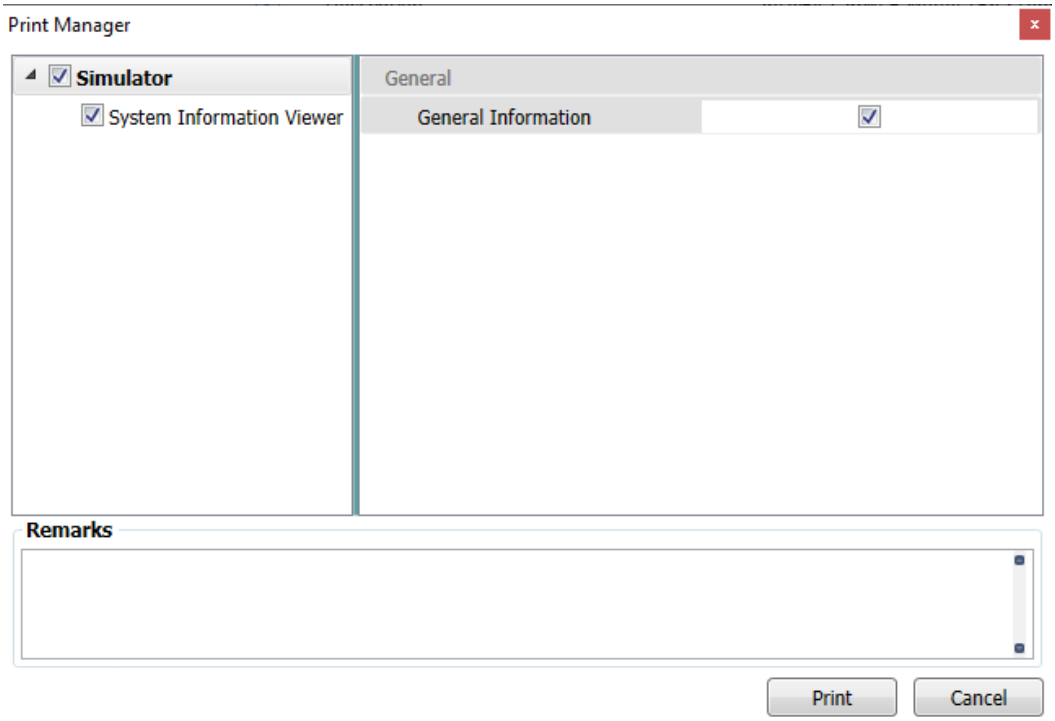



Enter the display time (the default is 30 seconds) and click **OK**.

## 2.9 Printing Component Data

SPiiPlus MMI Application Studio provides you with a wide variety of Workspace printing options. To print component data in the Workspace:

1. Select **Print** from the Main Menu **File** option. This displays the Print Manager:

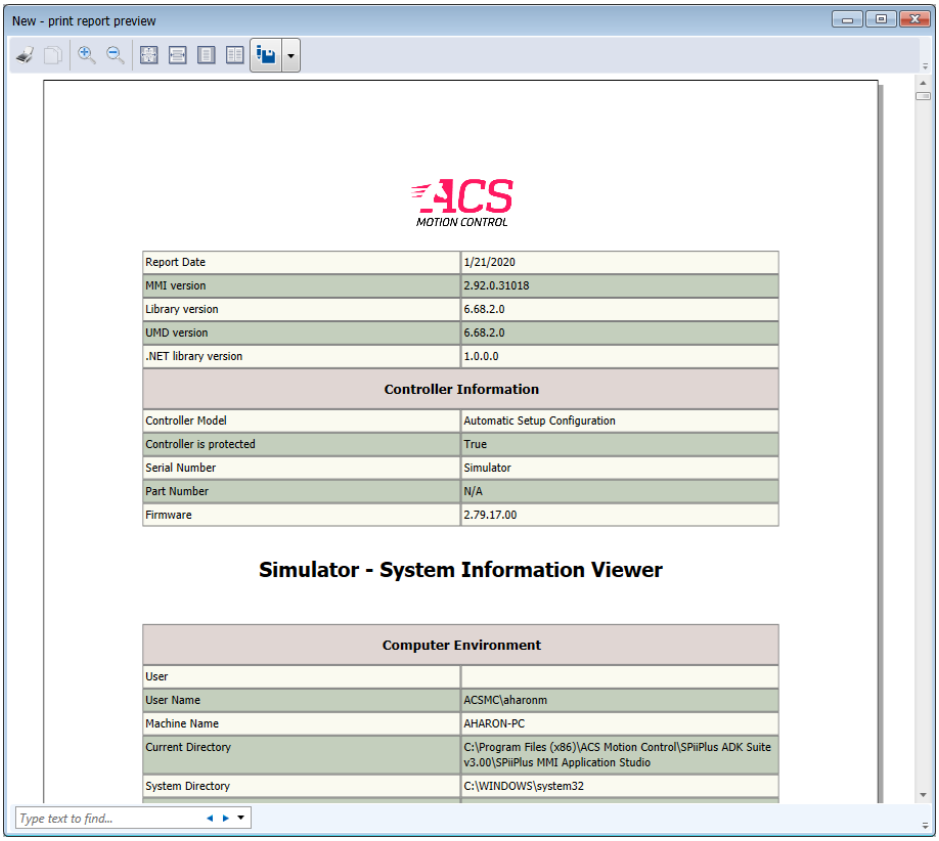



The Print Manager lists all components currently in the Workspace.

2. Select the data you want to print. By default Print Manger selects all checkboxes. If there is data you do not want to print, unselect the appropriate checkbox.
3. Click **Print**. The Print Report Preview window is displayed.

### *2.9.1 Print Report Preview Window*

In addition to displaying the data that will be printed, the Print Report Preview window provides you with complete functionality for printing.




## Toolbar

**The Print Report Preview Toolbar contains the following buttons:**

| | |
|---|---|
|  | **Print...** - displays a Print window so set printer and printing preferences. |
|  | **Copy** - copies selected text |
|  | **Expand** - expand report in preview window |
|  | **Contract** - contract report in preview window |
|  | **100%** - resize report to fit page in preview window |
|  | **Resize Page Width** - resize report to fit page width to window |

| | |
|---|---|
|  | **Resize Whole Page** - resize report to fit page width to window |
|  | **Resize Two Pages** - resize report to fit two pages in display window |
|  | **Export Document** - export report to one of the following formats:<br>> PDF<br>> RTF<br>> XLS<br>> XLSX<br>> CSV<br>> Text file<br>> Image file |

## 2.10 Working with Wizards

Wizards are groups of procedures. The procedures are broken down by tasks, and once you have selected a particular task, the wizard leads you through each task step-by-step.

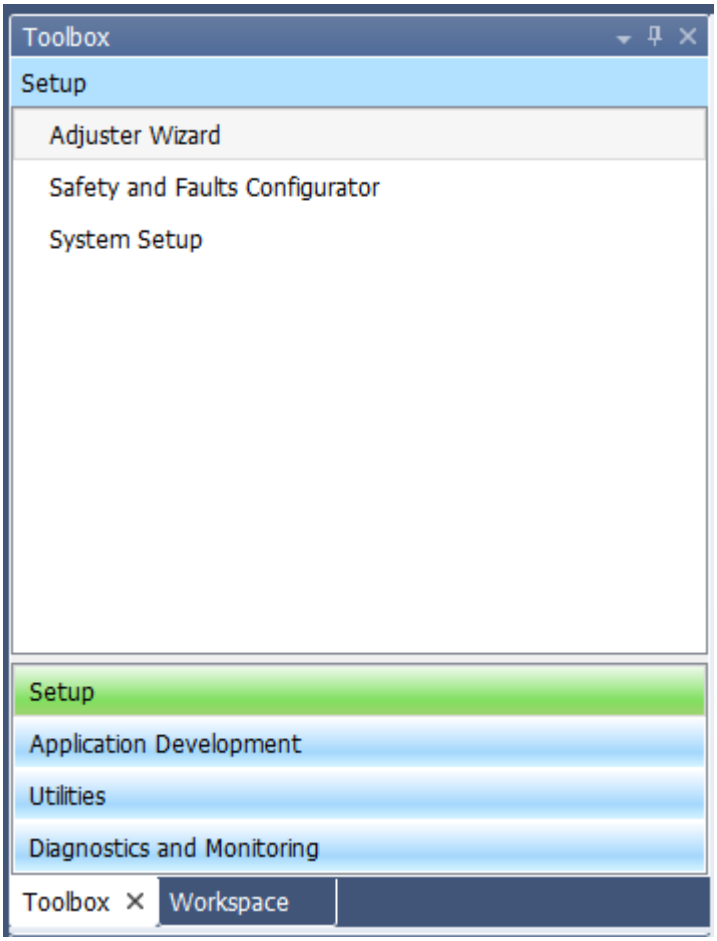
The following wizards are available:

- Adjuster Wizard
- Protection Wizard
- Upgrade and Recovery Wizard
- Application Wizard

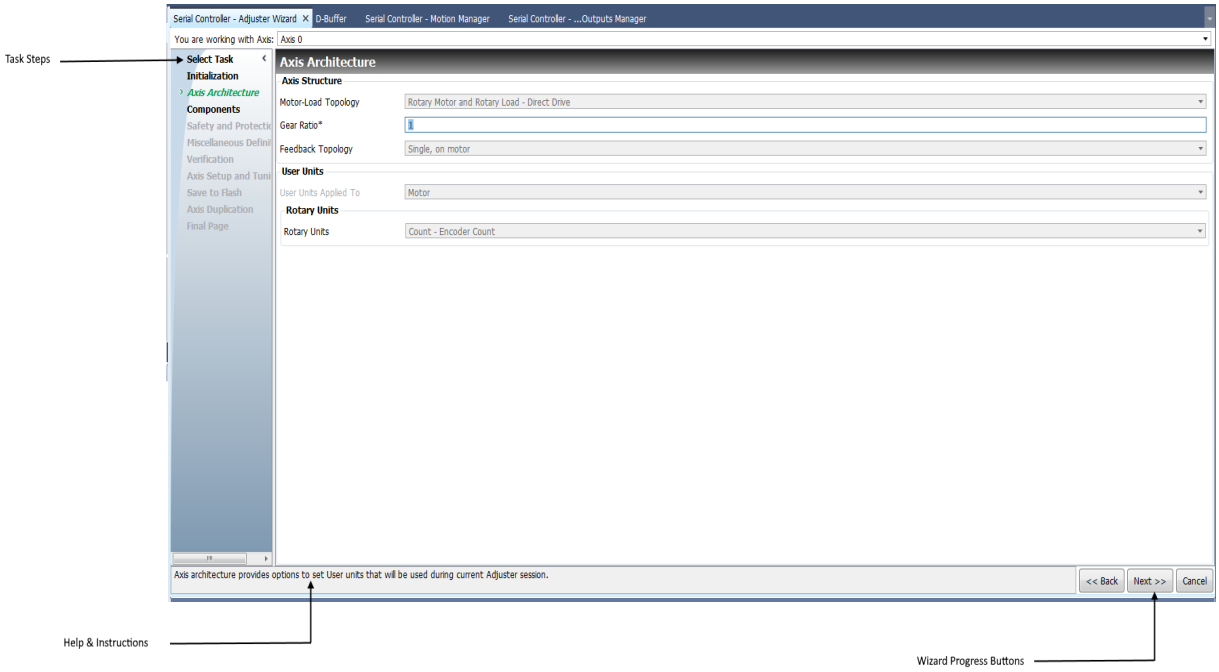
### 2.10.1 Starting a Wizard

To start a wizard:

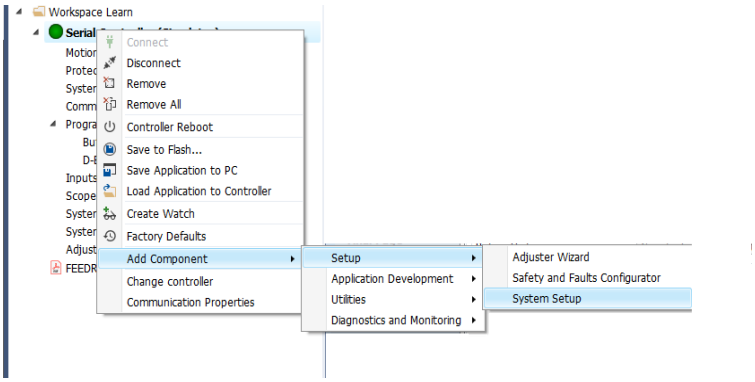
1. From the Toolbox select the tool of which the wizard is a component, for example:




2. Click the required wizard in the list. The **Select a Task** window for the selected wizard is displayed in the workspace, for example:



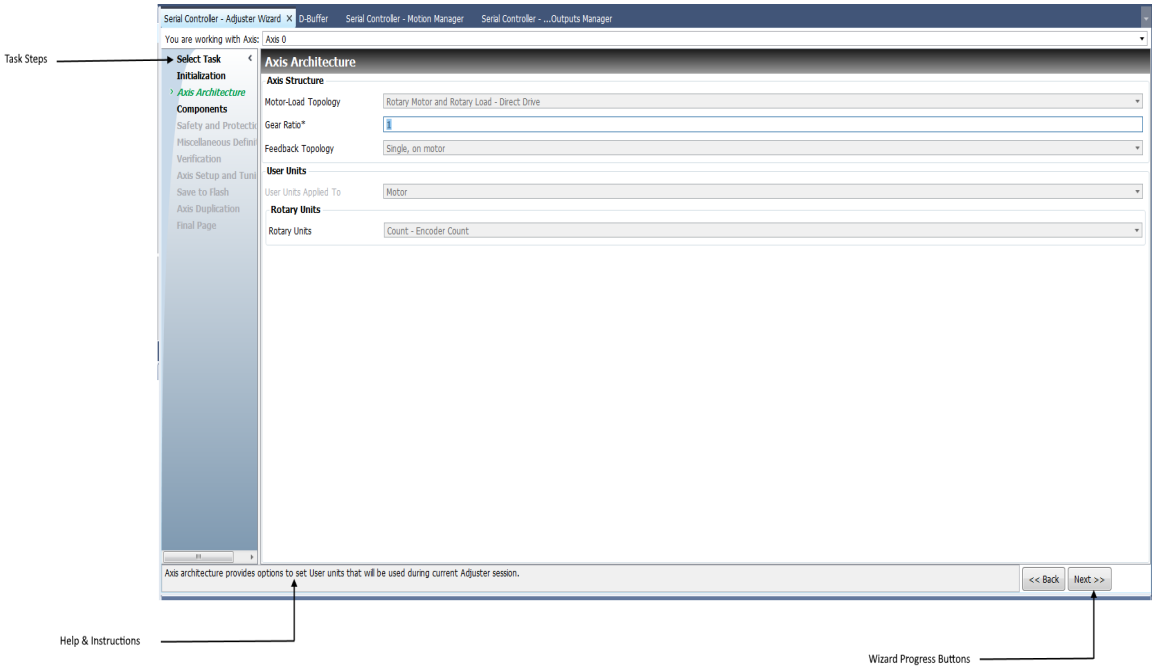

The wizard can also be selected directly by right-clicking the controller and selecting **Add component➔<Tool>➔<Wizard>**, from the dropdown menu, for example:




3. Select the Task by clicking the associated button and then click **Next**.

## *2.10.2 Wizard Window Elements*

The Wizard Window elements are:



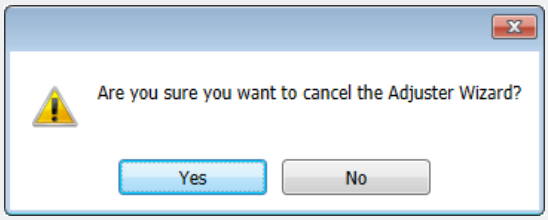

> Task Steps
This lists the steps that are performed during the course of working through the wizard. When the step is highlighted, indicates on what step you are on.

> Task Workspace
This is the area in which you enter the data required by the task step.

> Help & Instructions
This area provides quick instructions and tips of what you are to do in the current step.

> Wizard Progress Buttons
You use these buttons to navigate from one task to another. The buttons are:

| | |
|---|---|
| Next >> | Displays the next task in the Task Workspace. It should be noted that this button is not available until you have entered all of the data required by the task step. |
| << Back | Displays the previous task in the Task Workspace. |

| | |
|---|---|
| Cancel | Aborts the wizard. When you click this button, you are prompted with, for example:<br>  <br>Click **Yes** to cancel. |
| Finish | Appears in the final task of the wizard - exits the wizard. |

There are other action buttons that are specific to particular tasks, such as **Restart** ( 

) which, when clicked, reboots the SPiiPlus Motion Controller.

# 3. Setup

**Setup** contains the following tools for setting up your controller and system:

- Adjuster Wizard for setting up your motion parameters and adjust them to maximize your system's motion performance.
- Safety and Faults Configurator - for selecting specific faults that you want monitored during a program run.
- Automatic System Setup - for performing a manual or automatic system configuration.

## 3.1 Accessing Setup Tools

To access the Setup tools:

1. Open the **TOOLBOX**.
2. Click **SYSTEM SETUP**.
3. Click the required tool.

## 3.2 Adjuster Wizard

Only personnel having a good background in servo parameters, stability, frequency response, bandwidth, etc., and are capable of analysis and diagnostics in time domain and of setting parameters of pre-defined control filters/algorithms should use the **Adjuster Wizard**. For details on the algorithms and control filters, see the *SPiiPlus Setup Guide*.

The **Adjuster Wizard** is used to setup the system and tune the control loops. You use the **Adjuster Wizard** to set the parameters for the:

- Axis architecture
- Drive
- Motor
- Feedback

The **Adjuster Wizard** provides tuning in terms of high-level parameters, namely parameters that have a clear meaning for you. Once you have assigned high-level parameters, the **Adjuster Wizard** automatically translates them into low-level parameters, that is, controller parameters.

Since the **Adjuster Wizard** works together with the **Scope**, the Adjuster provides interactive tuning. The **Adjuster Wizard** sends a trial signal to the motor and automatically selects suitable **Scope** settings and sends the low-level parameter values. The **Scope** displays the motor response and you can immediately view the results.

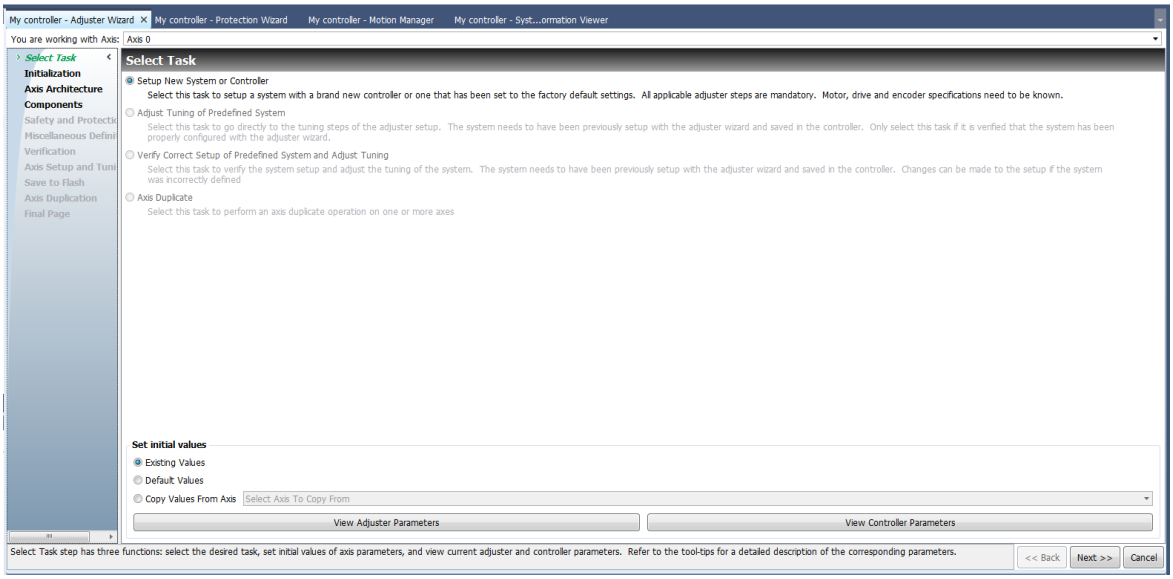
Some operations during the adjustment process may cause the motor to begin moving unexpectedly. Some of the limits and safety features may be temporarily disabled. To avoid personal injury or damage to the equipment, check the following before starting the adjustment process:

> Verify that NOTHING (people, electrical cables, or other obstacles) is in the path of the motor or objects connected to the motor.

> Verify that the motor is securely anchored and that proper safety barriers, stops, and/or limits are installed.

### *3.2.1 Starting Adjuster Wizard*

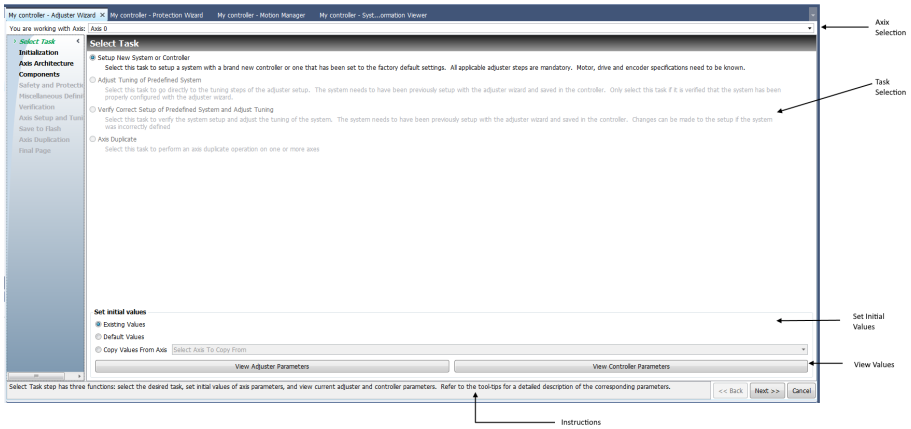
1. In the Toolbox click **Setup** to display the Setup list of tools.
2. Click **Adjuster Wizard** in the list of tools.
   The Adjuster Wizard **Select Task** window is displayed in the workspace.




You can also activate the **Adjuster Wizard** using the right-click **Add Component** option of the Workspace Tree - see Controller Right-Click Options.

### 3.2.2 Adjuster Wizard Select Task Window

The **AdjusterWizardSelect Task** window consists of the following elements:

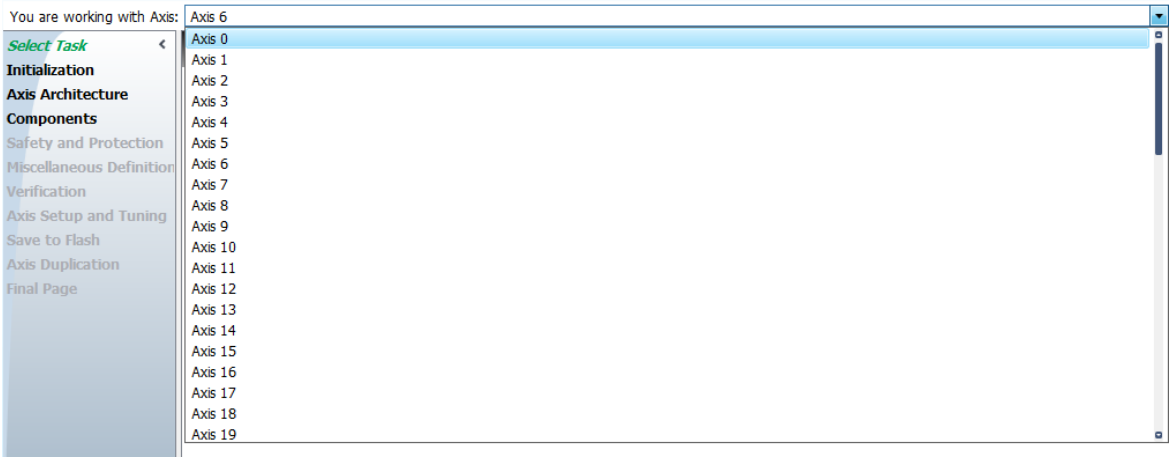



Selecting a task is done in the following sequence:

1. Select Axis

   The Adjuster Wizard is applied on an axis-by-axis basis. That is, you have to perform it for each axis individually. When you activate the wizard, the first thing you have to do before selecting a Task is select the axis to which the data is to be applied.

   You select the axis by clicking the Axis Selector field and then select the axis from the dropdown list:



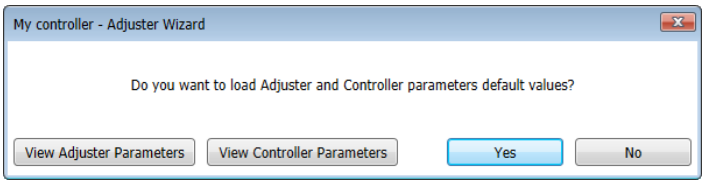

2. Set Initial Values

   Your choices of initial values to be used by the Adjuster are:

   > **Existing Values** - using the values that have been built into the Adjuster database.

   > **Default Values** - using factory default values that have been set in the controller.

When you select this option, a prompt window is displayed:




To view the Adjuster database default values click **View Adjuster Parameters**, the View Adjuster Parameters window is displayed with the values for the selected axis.

To view the controller parameter values click **View Controller Parameters**, the View Controller Parameters window is displayed with the values for the selected axis.

View Controller Parameters [Axis 0]

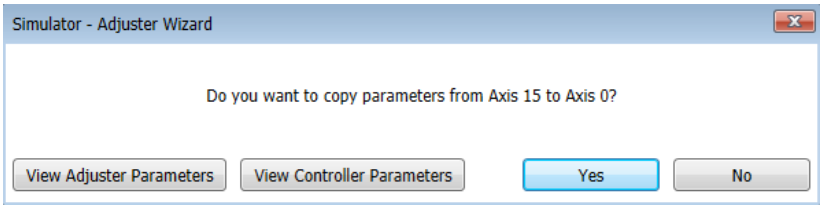
| Parameter | Description | Current Value | New Value | Modified |
|---|---|---|---|---|
| ACC | Acceleration | 100000 | 100000 | No |
| BOFFTIME | Brake Deactivation Time | 50 | 50 | No |
| BONTIME | Brake Activation Time | 50 | 50 | No |
| CERRA | Critical Error (Acceleration) | 1000 | 1000 | No |
| CERRI | Critical Error (Idle) | 1000 | 1000 | No |
| CERRK | StandardVariable_CERRK | 1000 | 1000 | No |
| CERRV | Critical Error (Velocity) | 1000 | 1000 | No |
| CTIME | Control Cycle Time | 1 | 1 | No |
| DCOM | Drive Command | 0 | 0 | No |
| DEC | Deceleration | 100000 | 100000 | No |
| DELI | Delay on Transition to Idle State | 50 | 50 | No |
| DELK | Delay on Transition to Idle State (on Kill) | 50 | 50 | No |
| DELV | Delay on Transition to Velocity State | 50 | 50 | No |
| E_FREQ | Encoder Frequency | 20 | 20 | No |
| E_SCMUL | Encoder Sin-Cos Multiplier | 10 | 10 | No |
| E_TYPE | Encoder Type | 3 | 3 | No |
| | Secondary Encoder Sin-Cos | | | |

Modify parameters as needed for the controller and application as appropriate and then click **Yes**.

> **Copy Values From Axis** - using values that have previously been set for another axis.

When you select this option, you choose the axis whose values are to be used from the **Select Axis To Copy From** (which becomes active when you select this option) dropdown list:


A prompt window is displayed:




You can view the Adjuster and Controller parameters values that will be copied.
If these are the values you want to copy, click **Yes**.

3. Select Task

There are four tasks that can be performed using the Adjuster **Wizard**:

> Setup New System or Controller
You perform this task when setting up a new controller, or if the controller has been set to the factory default values.
When performing the task you not only set the control loop values, but you also set system hardware values, that is, values for the Drive and Motor.

> Adjust Tuning of Predefined System
You perform this task to fine-tune the control loop values.
When you select this task, you are taken directly to the Axis Setup and Tuning step.

> Verify Correct Setup of Predefined System and Adjuster Tuning
You perform this task to verify the setup that was saved to the controller from the **Setup New System of Controller** task.
When you select this task, you are taken directly to the Verification step and then to the Axis Setup and Tuning step. Both of these steps are available only if you have first setup the system.

> Axis Duplication
You perform this task to copy an axis set of parameters to any other axis in the controller.
When you select this task, a window appears in which you can select the source axis and the destination axis.

### *3.2.3 Adjuster Wizard Task Steps*

In all, the **AdjusterWizard** consists of nine task steps:

> **Initialization**

In this step you provide information that serves to identify the database containing the high-level parameter values.

- **Axis Architecture**

  In this step you define the electro-mechanical structure of the controller's working application as well as the units for measuring feedback.

- **Components**

  In this step you define the parameters governing the:

  - Drive
  - Motor
  - Feedback

  Once you have defined these, you trigger the Adjuster to calculate the optimal values for the high-level parameters and you load them into the system.

- **Safety and Protection**

  In this step you define the parameters for:

  - Motion limits
  - Current limits
  - Position errors
  - Position limits

- **Miscellaneous Definitions**

  In this step you define the parameters for such things as:

  - Motion completion
  - Enable/Disable/Brake
  - Dynamic brake
  - Home switch

- **Verification**

  In this step the **AdjusterWizard** runs various verification tests for such things as:

  - Feedback
  - Motor
  - Switches
  - Stop, Alarm and Brake

  This enables you to see where you need to adjust the values of the parameters.

- **Axis Setup and Tuning**

  In this step you are given the opportunity to refine your values and fine tune your system.

- **Save to Flash**

  Once your high-level values are correct for your system, in this step you save the low-level values to the controller flash.

> **Final Page**

This step is your exit point from the **AdjusterWizard**.

Many of task options are dependent on the nature of the three basic components you have defined. That is, the type of motor, drive and required feedback signal defined in the **Axis Architecture** and **Components** steps determine what you can define in the subsequent steps.

### *3.2.4 Component Database*

The adjustment procedure affects both the volatile and non-volatile memory of the controller. The Adjuster keeps the controller memory in-sync with the application database.

The **AdjusterWizard** maintains the high-level adjusted values for each component separately on your computer. Upon selecting **Apply changes**, the **AdjusterWizard** saves the values into low-level parameters, that is the parameters that the controller recognizes. Only after you have selected **Save to Flash**, does the wizard save the low-level parameter values into the flash, i.e., non-volatile, memory of the controller.

If the controller is shut off during an adjustment session and the data has not been saved to the controller's flash memory, synchronization is lost and the adjustment session must be repeated.

**Incompatibility Symbol**

The Incompatibility Symbol indicates that the value in the component field is incompatible with what you have defined in previous windows.

You should return to a previous window and change the value with which it is in conflict, or, if possible, change the value in the field in the current window. In any case you should resolve all conflicts before continuing.

This symbol can be ignored when an element is not yet defined. For example, a Drive has been defined but not the Motor.

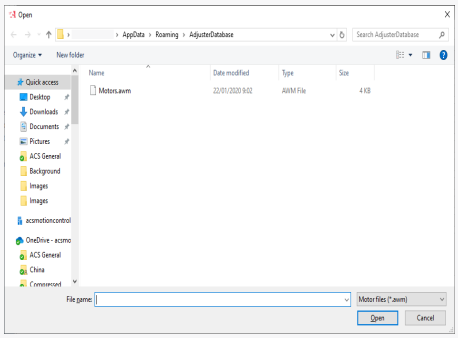
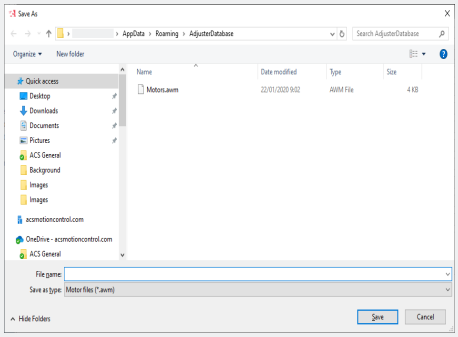
### *3.2.5 Working in Adjuster Wizard Task Windows*

There are several elements that are common to most of the **Adjuster Wizard** task windows. These are:

**Data Action Buttons**

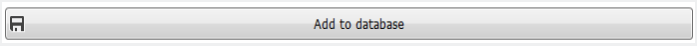
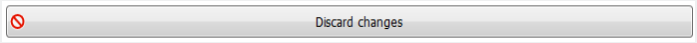
There are five data action buttons:

| | |
|---|---|
|  | Add an item to the field. |

| | |
|---|---|
|  | Edit the data fields. |
|  | Deletes the item from the database. |
|  | Import data from a previously saved file on the computer.<br>When you click this button, a browser window is displayed, for example:<br>  <br>Drill down to the subdirectory containing database files and select the file, all Adjuster database files have an **awd** file extension. Then click **Open**. |
|  | Export data to a file on the computer.<br>When you click this button, a browser window is displayed, for example:<br>  <br>Drill down to the subdirectory containing database files and enter the filename, all Adjuster database files have an **awd** file extension. Then click **Save**. |

### Adding Data to Component Database

When you define the values for the Motor, Drive and Feedback components, the following action buttons are available:

| | |
|---|---|
|   | Active if you have entered a component name that is not in the database. Clicking this button adds the data to the Component database under the component name. |
|   | Active if you have made changes to the data that exists in the database. Clicking this button puts changes into the Component database. |
|   | Deletes all changes you have made to the data fields. |

### *3.2.6 Setup New System or Controller*

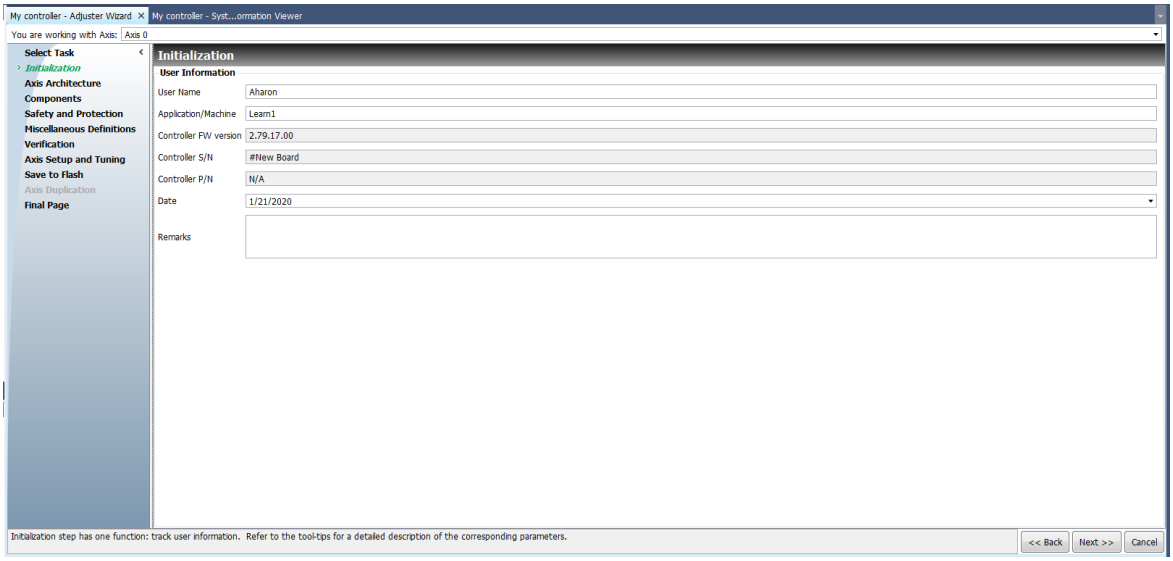
You must perform this task for any new system or a controller whose data has been reset to factory defaults.

The steps involved in setting up a new controller are:

- Axis Architecture
- Components
- Safety and Protection
- Miscellaneous Definitions
- Axis Setup and Tuning

#### *3.2.6.1 Initialization*

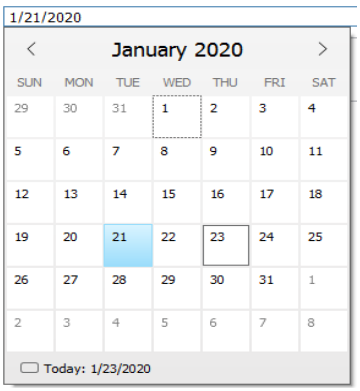
Initialization serves only for tracking purposes.




1. Enter your name in the **User Name** field (optional - this field may be skipped).
2. Enter the name of your application or the ID of the machine for which the controller is being employed in the **Application/Machine** field (optional - this field may be skipped).

You will note that Adjuster has extracted the controller's firmware version, S/N and P/N from the controller and filled in these fields.

3. Enter the date in the **Date** field - its format is: mm/dd/yyyy. Or you can click the down arrow and select the date from the calendar.




4. Enter any remarks you want in the **Remarks** field (optional - this field may be skipped).
5. Click **Next**.

### *3.2.6.2 Axis Architecture*

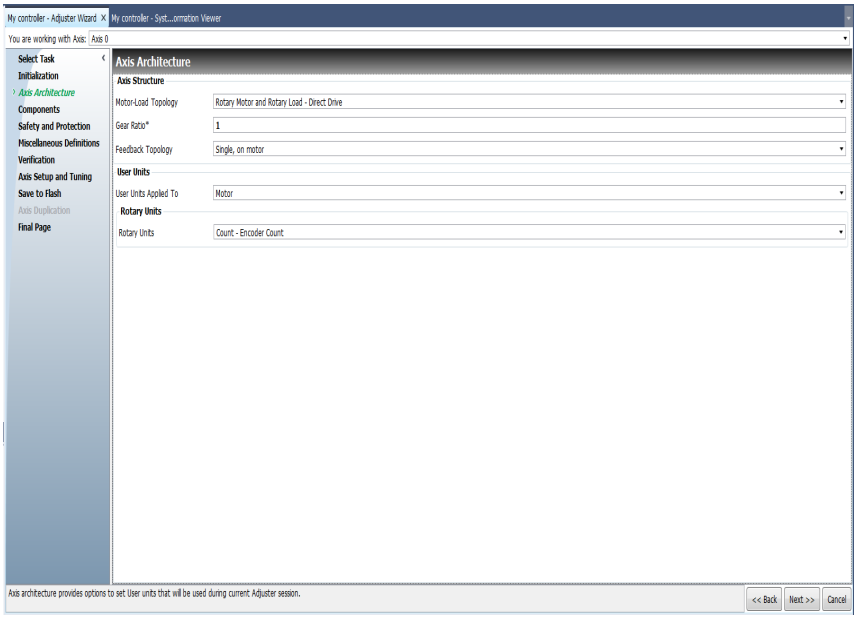
In this step you are to define the general parameters of the feedback mechanism to include the electro-mechanical characteristics of the motion and how feedback is to be measured and from what.

Data must be entered in all the fields. The data is entered by selecting the appropriate value from the dropdown list of each field.

### Feedback Topology

The feedback topology choices are:

| Controller Drive Interface | Options |
|---|---|
| ACS PWM drive | All current options |
| PDICL | > Verification, on motor<br>> Verification, on load<br>> None |
| PDMNT | None |




Once you have entered all the values, click **Next** to go on to the next step.

### User Units

Of particular import in this step is the setting of User Units to be used by Adjuster in measuring feedback. The User Unit can be millimeters, microns, nanometers, degrees or any other unit that defines a distance for a linear axis or an angle for a rotary axis.

The User Unit is defined per axis. By default, the User Unit is the Encoder Count. For example, for a quadrature encoder with resolution of 500 lines per millimeter, each default User Unit equals 2mm/4 = 0.5mm.

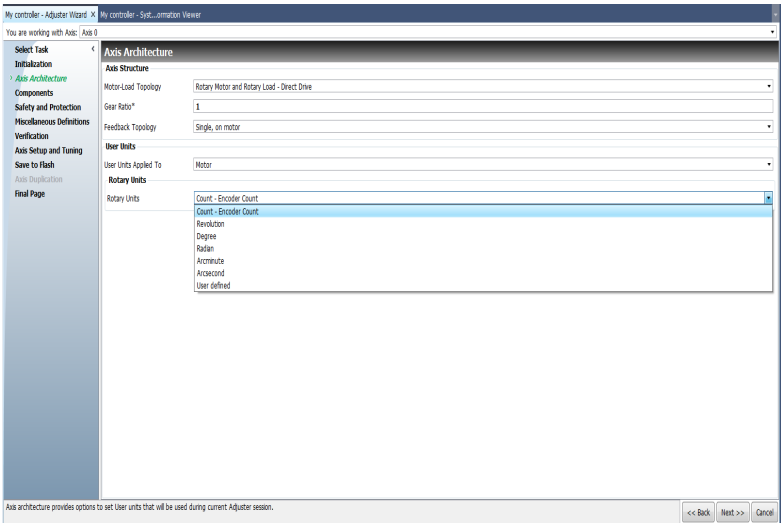
The User Unit is applied to the **EFAC** variable. **EFAC** is a 64 bits member (one for each possible axis) real array used for defining a factor between the raw feedback in encoder counts and the **FPOS** value calculated by the controller. See *SPiiPlus Command & Variable Reference Guide* for complete details.

### Advanced Parameters

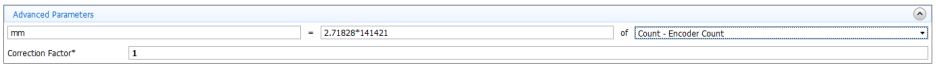
**Advanced Parameters** enables you define your own feedback units.

> The **Advanced Parameters** option should be used with caution. Badly defined units may cause serious problems with measuring the actual feedback.

To access **Advanced Parameters** click in Units field and select **User defined**, for example:




The **Advanced Parameters** panel is displayed:




1. Enter a unique parameter name in **User Defined Unit Name**.
2. Enter a mathematical expression in **Function**, you can use the standard arithmetical operations:
   - \+ (Add)
   - \- (Subtract)
   - * (Multiply)
   - / (Divide)

   The function is used as a conversion factor applied to the selected standard unit. For example: 2.71828*1.41421 (***e*** times the square root of 2).
3. Select a standard unit from the dropdown list in the **of** field.

4. Enter a correction factor (if different than **1**) in **Correction Factor**. The correction factor is used to correct the difference between actual movement (as measured by a laser interferometer) and the software commanded move (indicated in the user units).

### *3.2.6.3 Components*

The Components step consists of defining the values for:

> Motor

> Drive

> Feedback

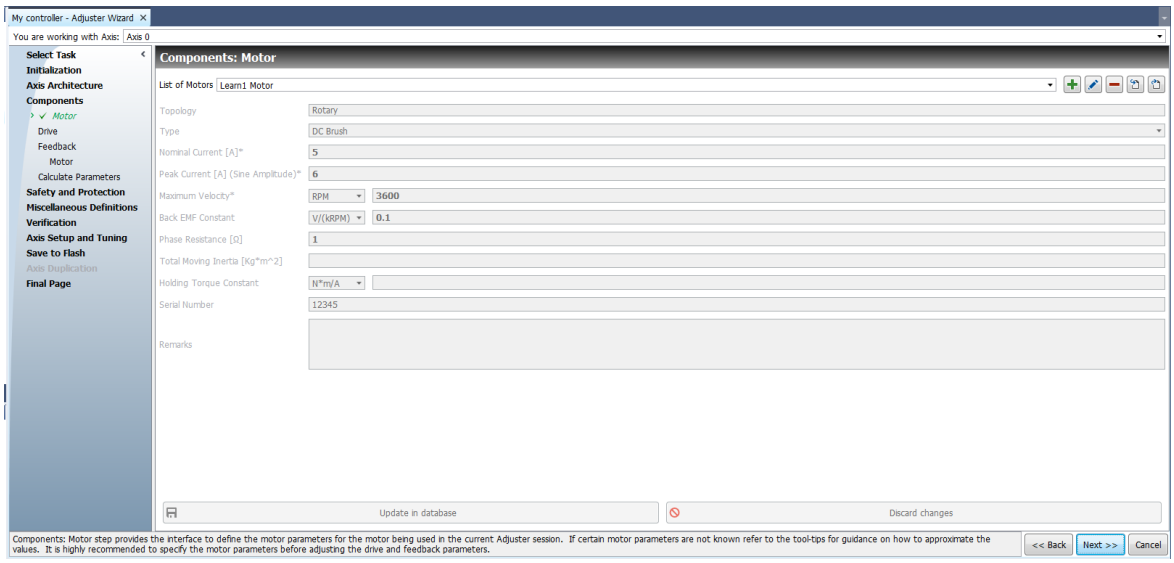
At the end of the step the Adjuster calculates the parameters that are to be used and inserts them into the appropriate program variables.

For this step you will need the technical specifications for the motor and drive from which to extract the correct values.

The screen shots presented in this step are for example only. The actual fields that appear depend on the values set in the **Axis Architecture** step.

#### *3.2.6.3.1 Motor*

The first component for which you need to enter values is the Motor.




Data must be entered in all the fields. The data is entered by selecting the appropriate value from the dropdown list of each field or entering the values taken from the motor's technical data sheet.

In entering the values the incompatibility icon ( ) may appear. Check the tool-tip. If it states that there is incompatibility between components that have not been set yet, ignore it. Continue to set component values.

However, do not forget to go back after all the components have been set up and ensure that it no longer appears. If it still appears, then resolve it by changing the component's values or changing the **Axis Setup** values.

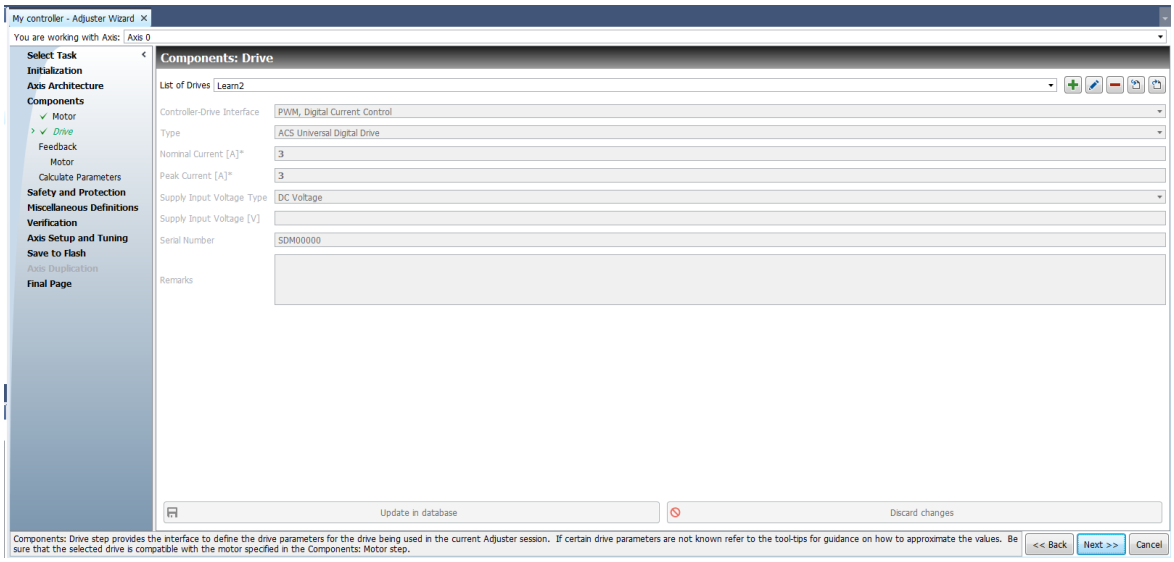
Once you have entered all the values, click **Next** to define the Drive component.

*3.2.6.3.2 Drive*

The second component for which you need to enter values is the Drive.

The relevant drive parameters are:

> Step Motor




Data must be entered in all the fields. The data is entered by selecting the appropriate value from the drop-down list of each field or entering the values taken from the drive's technical data sheet.

In entering the values the incompatibility icon ( ) may appear. Check the tool-tip. If it states that there is incompatibility between components that have not been set yet, ignore it. Continue to set component values.

However, do not forget to go back after all the components have been set up and ensure that it no longer appears. If it still appears, then resolve it by changing the component's values or changing the **Axis Setup** values.

**Step Motor**

| Controller Drive Interface | Feedback topology | Window Behaviour |
|---|---|---|
| ACS PWM | Verification, on motor<br>Verification, on load<br>None | Available - as Stepper Motor |
| ACS PWM | Any option not listed above | Unavailable |
| PDICL<br>PDMNT | Any | Availble - as Pulse Direction Setup |

Once you have entered all the values, click **Next** to define the Feedback component.

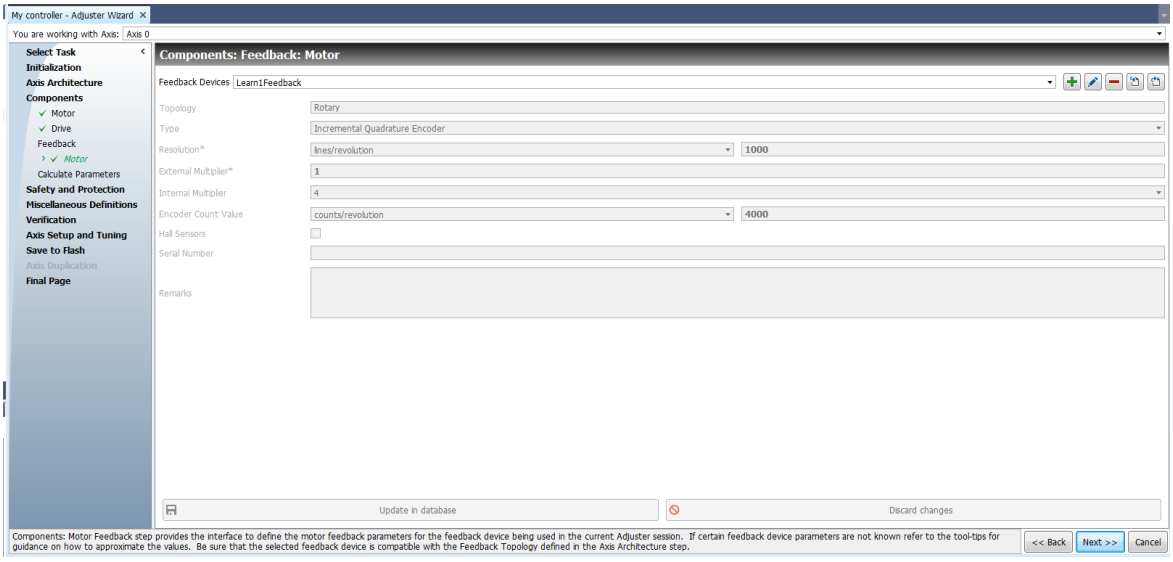
*3.2.6.3.3 Feedback*

The **Feedback** component can be either the Motor, the Load, or both depending on what you defined in **Feedback Topology** in the Axis Architecture step.

Data must be entered in all the fields that are active (many, you will note, have already been selected by Adjuster from data that you have previously entered for the Drive and Motor components). The data is entered by selecting the appropriate value from the dropdown list of each field or entering the values directly in the fields.

In entering the values the incompatibility icon ( ⚠ ) may appear. Check the tool-tip to see what is causing the incompatibility. If it is a value of a previously set up component, then return to the component and either change the component's value, or make changes in the **Axis Setup**.

Make sure that all component incompatibilities are resolved before continuing.




The feedback choices are:

| Controller Drive Interface | Feedback topology | Window behavior |
|---|---|---|
| ACS PWM | Single, on motor<br>Single, on load<br>Verification, on motor<br>Verification, on load | Available |
| ACS PWM | None | Unavailable – the entire step is skipped |
| PDICL | Verification, on motor<br>Verification, on load | Available |
| PDICL | None | Unavailable – the entire step is skipped |
| PDMNT | None | Unavailable – the entire step is skipped |

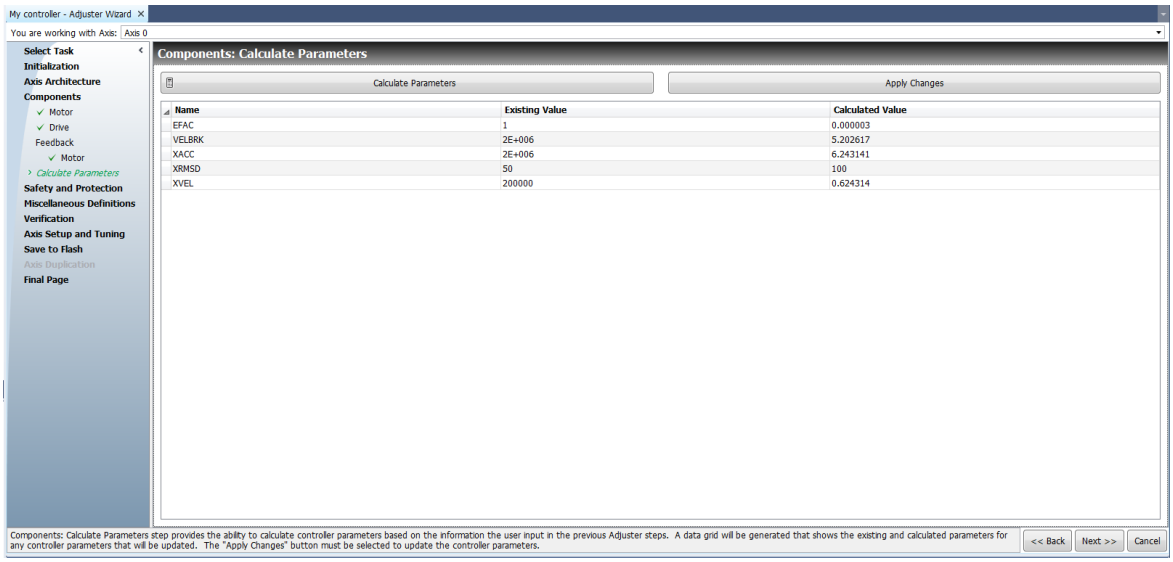
Click **Next** to advance to calculate and apply the feedback parameters.

*3.2.6.3.4 Calculate Parameters*

At this stage, the Adjuster verifies that the Axis Architecture, Motor, Drive and Feedback definitions are all compatible with each other.

If an inconsistency is detected, a prompt is displayed about it with recommended solutions.

If there are no inconsistencies, click: Calculate Parameters. The Adjuster calculates the values for the controller parameters and displays a table with each parameter name, its old and its new value, for example:




Click: Apply Changes and click **Next** to advance to the next step.

At this stage values are stored in the low-level parameters within the **Adjuster**. They do not take effect until you have saved them into the controller's flash memory.

### *3.2.6.4 Safety and Protection*

The Safety and Protection step enables you to set certain safety and protection variables. There are a total of four possible types of Safety and Protection parameters that you can set.

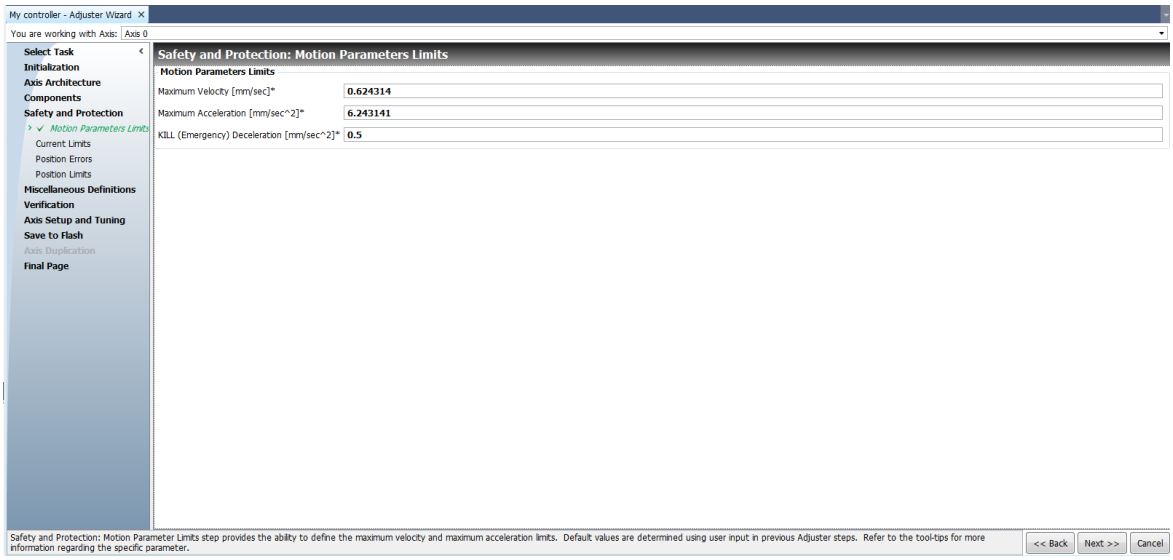
The actual number of parameter types and parameters available to you depends on what you defined in the Axis Architecture and Components steps.

The Adjuster displays values it has calculated in the window of each type. You have the option of changing these within the limits of valid values.

A complete explanation of each parameter is given in the Tool Tips associated with the parameter.

For the meanings of the variables see the *SPiiPlus Command & Variable Reference Guide*.

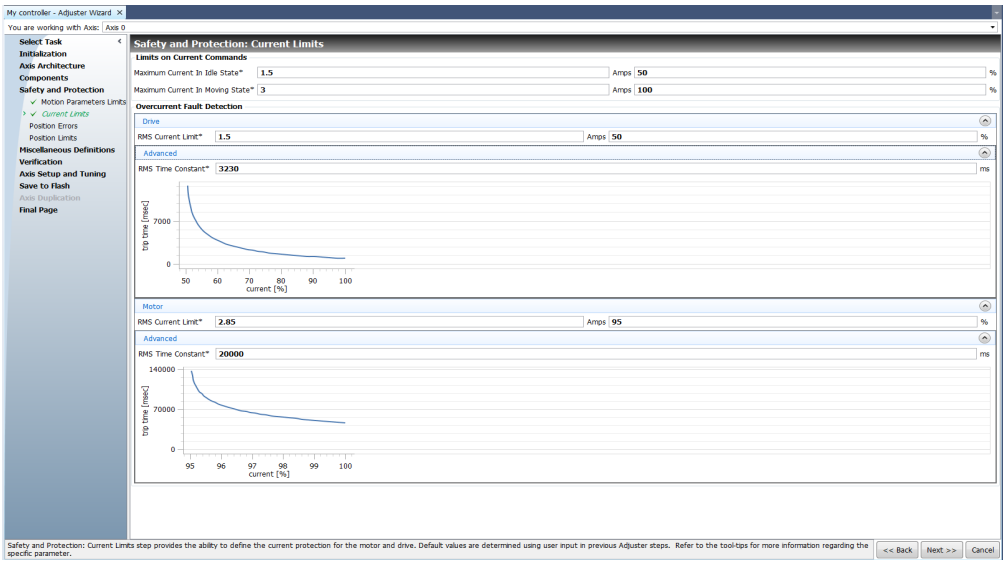
### *3.2.6.4.1 Motion Parameters Limits*




This window is used for entering the Maximum Velocity and Maximum Acceleration Limits values.

- **MAXIMUM VELOCITY (XVEL) [UNITS/SEC]** - The maximum allowed value is automatically calculated based on motor parameters that were defined in Components step. Letting the mouse cursor linger on the field displays a tool-tip guiding you in permissible values.
- **MAXIMUM ACCELERATION (XACC) [UNITS/SEC$^2$]** - The Adjuster puts in a value based on motor parameters that were defined in Components step. Letting the mouse cursor linger on the field displays a tool-tip guiding you in permissible values.
- **KILL (Emergency Deceleration - KDEC)[deg/sec$^2$]**

### *3.2.6.4.2 Current Limits*



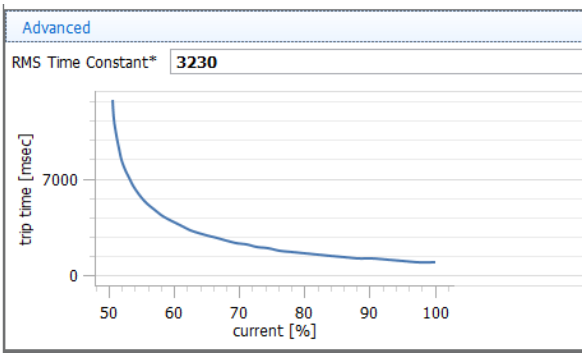

This window is used for defining motor and drive protection parameters:

- **Maximum Current In Idle State (XCURI)** - The Adjuster automatically calculates the value from the maximum torque/force required during standstill to overcome gravitation, friction, etc.
  If the value cannot be automatically calculated, that is, you have not specified gravitation, friction, etc., the parameter XCURI is set to default.
  If you want to set values other than the value displayed, letting the mouse cursor linger on the field displays a tool-tip guiding you in permissible values.
- **Maximum Current In Moving State (XCURV)** - The value is automatically calculated according to minimum of either maximal acceleration (XACC) or motor peak current that you have specified.
  If you want to set values other than the calculated one, letting the mouse cursor linger on the field displays a tool-tip providing guidance in changing maximal acceleration or motor peak values.
- **Overcurrent Fault detection** - The following numeric controls defines Overcurrent Fault detection criterion:
  - **RMS CURRENT LIMIT (XRMSD, XRMSM)** - The value is automatically calculated based on motor/drive parameters that were defined in the Components step.
    If you want to set values other than the calculated ones, letting the mouse cursor linger on the field displays a tool-tip guiding you in permissible values for the motor and/or drive current limits.
  - **RMS CURRENT TIME CONSTANT (XRMSTD, XRMSTM)** - This is the time interval for checking whether an overcurrent fault has occurred (meaning that XRMSM or XRMSD has been exceeded). The overcurrent fault activation time is according to the following formula:

> Motor: fault activation = -ln(1-(XRMSM/DCOM)$^2$)*XRMSTM
For example: if XRMSM=50, DCOM = 100 and XRMSTM = 3300 msec, then fault activation = -ln (0.75)*3300 = 950msec.

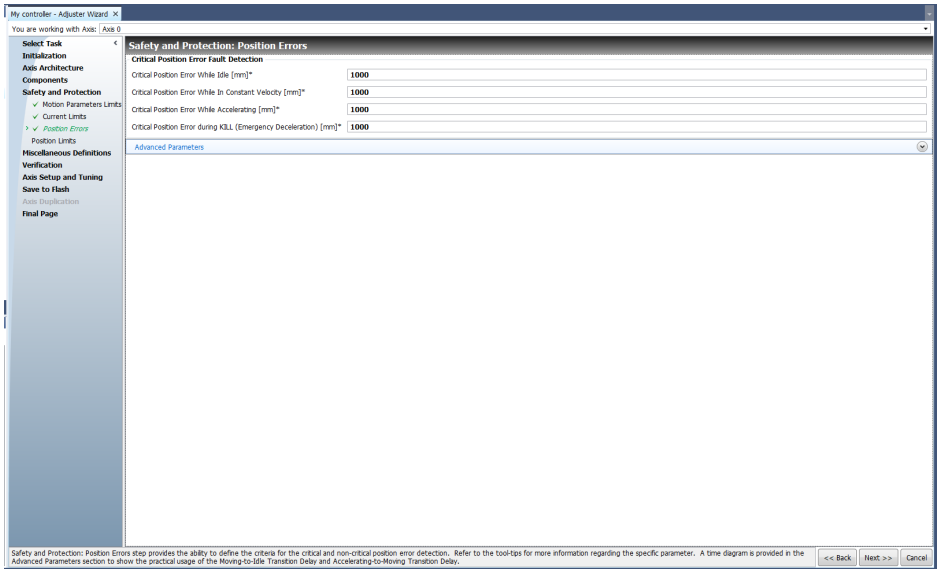
> Drive: fault activation = -ln(1-(XRMSD/DCOM)$^2$)*XRMSTD

If you click the **Advanced** down arrow, the Adjuster displays a graph of the Overcurrent Fault Detection:




When you change parameter values, you can see the effect on the graph.

### *3.2.6.4.3 Position Errors*




This window is used for defining Critical Position Errors criteria:

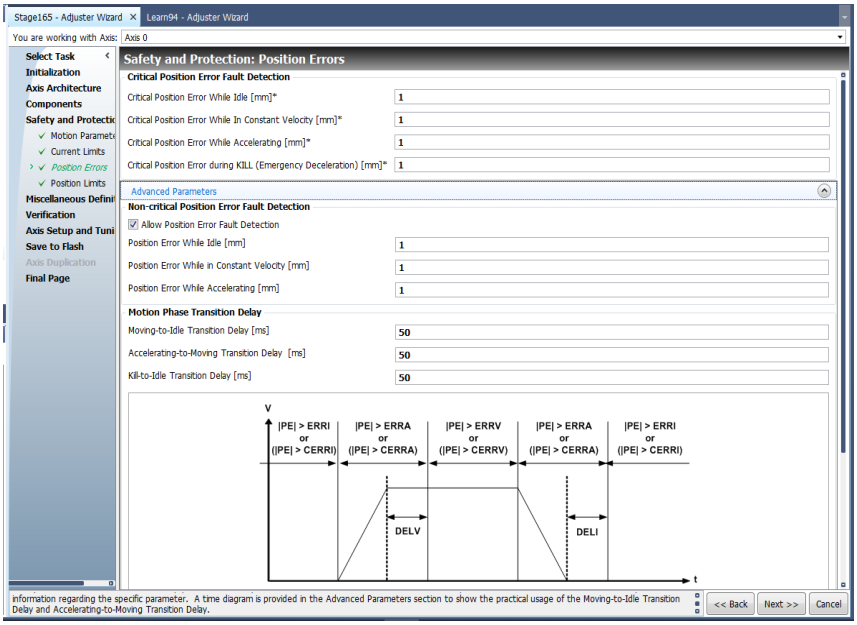
> Critical Position Error While Idle (CERRI)

> Critical Position Error While In Constant Velocity (CERRV)

> Critical Position Error While Accelerating (CERRA)

> Maximum Critical Position Error during kill (CERRK)

If you click the **Advanced** down arrow, the Adjuster displays the following:

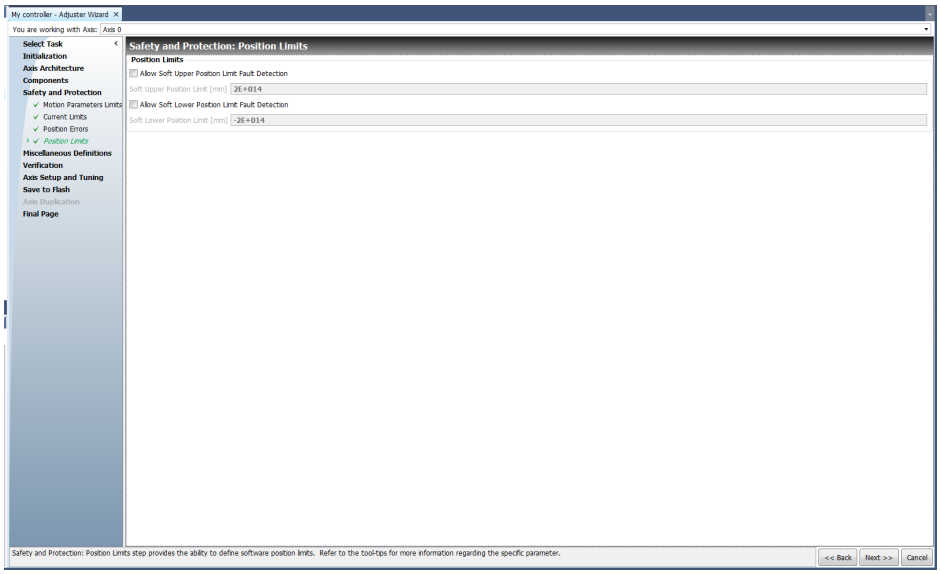



This enables you to set values for Non-critical Position Errors whereby excessive errors that might affect the application are flagged:

- Non-critical Position Error Fault Detection
  - Position Error In Idle (ERRI)
  - Position Error While In Constant Velocity (ERRV)
  - Position Error While Accelerating (ERRA)
- Motion Phase Transition Delay
  - Moving-to-Idle Transition Delay (DELI)
  - Accelerating-to-Moving Transition Delay (DELV)
  - Kill-to-Idle Transition Delay (DELK)

A diagram illustrating the meaning of the delay is included.

#### *3.2.6.4.4 Position Limits*




This window defines Software Position Limits and enables you to select:

- **ALLOW SOFT UPPER POSITION DETECTION (SRLIMIT)** - If Software Right Limit Mask is selected, then the controller examines the fault and kills the motion that is beyond the defined limit.
- **ALLOW SOFT LOWER POSITION DETECTION (SLLIMIT)** - If Software Left Limit Mask is selected, then the controller examines the fault and kills the motion that is beyond the defined limit.

### *3.2.6.5 Miscellaneous Definitions*

The Miscellaneous Definitions step enables you to set values of sets of variables of up to four possible types, depending on the system configuration.

The actual number of parameter types and parameters available to you depends on what you defined in the Axis Architecture and Components steps.

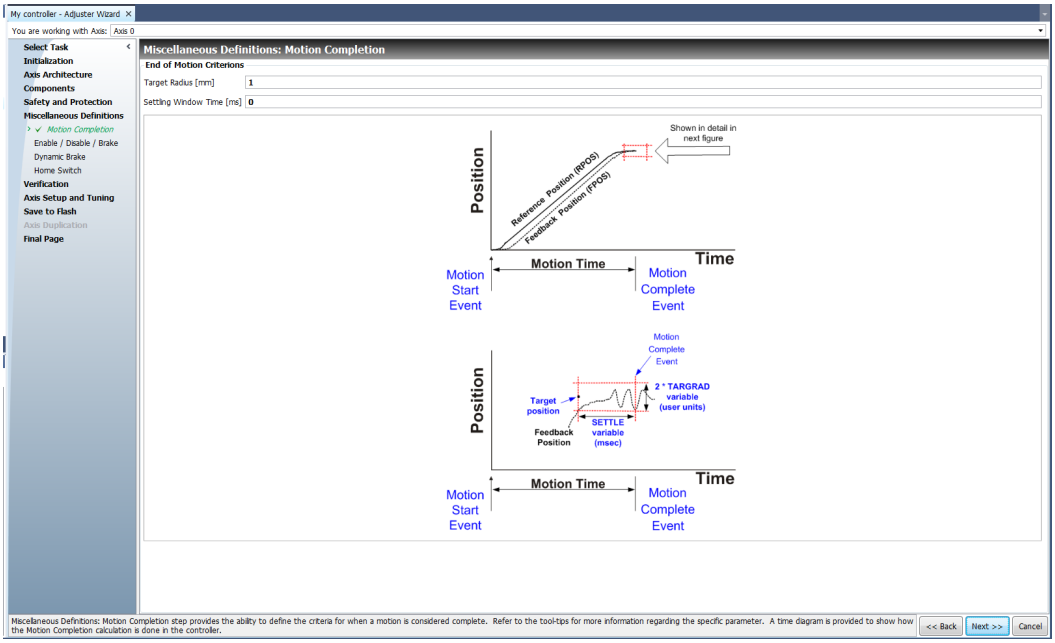
The Adjuster displays values it has calculated in the window of each type. You have the option of changing these within the limits of valid values.

A complete explanation of each parameter is given in the Tool Tips associated with the parameter.

For the meanings of the variables see the *SPiiPlus Command & Variable Reference Guide*.

### *3.2.6.5.1 Motion Completion*

Motion is considered completed once the position is within a window of TARGRAD user units and has not exited this window for a period of SETTLE ms.



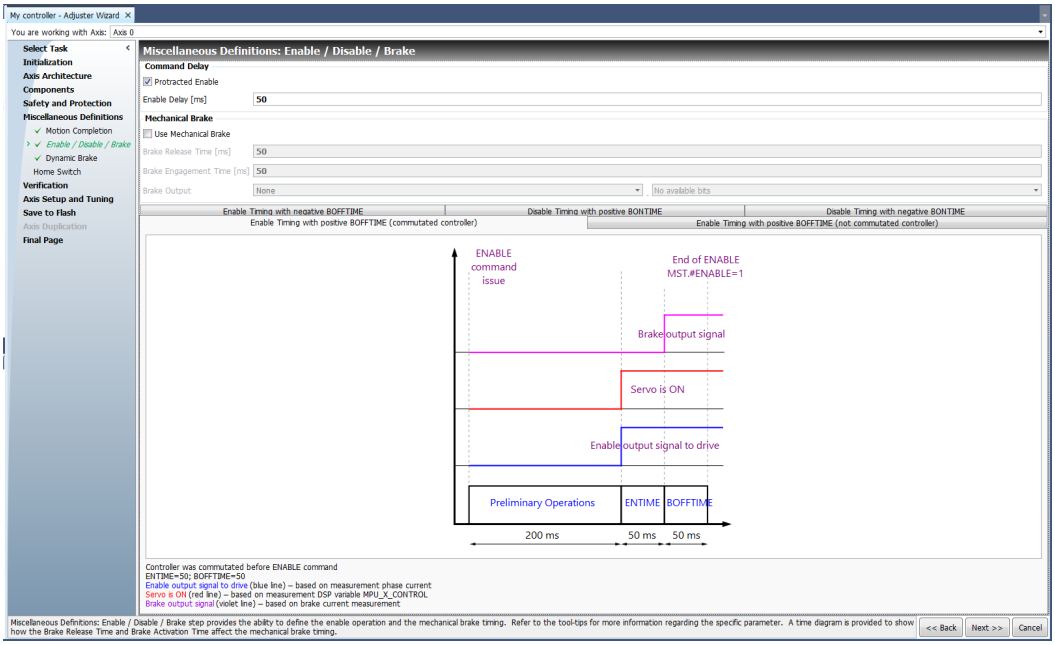

There are two variables, both optional, for which you can enter values:

- Target Radius (TARGRAD)
- Settling Time (SETTLE)

To aid your understanding a diagram is included showing the meaning of the two variables.

### *3.2.6.5.2 Enable/Disable/Brake*

This window is used for defining the Enable operation and Mechanical Brake timing.




The variables you can define are:

- **PROTRACTED ENABLE** - This is optional, but if you select it, you must put a value in **Enable Delay** (ENTIME). The controller delays the Enable Command by **ENTIME** mSec.
- **USE MECHANICAL BRAKE** - You need to select this only if the application has a mechanical brake; otherwise leave it unselected. If you select it, you have to enter values for:
  - **BRAKE RELEASE TIME (BOFFTIME)** - This value can be positive or negative and sets when the brake is released.
  - **BRAKE ACTIVATION TIME (BONTIME)** - This value can be positive or negative and sets when the brake is applied.

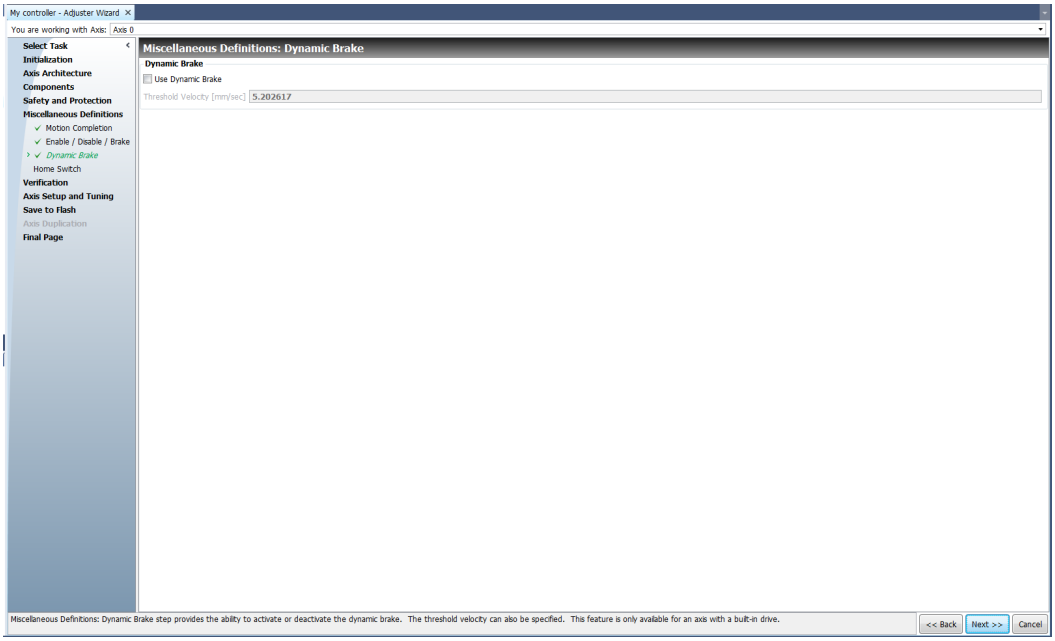
You are also informed of the Digital Output variable (**OUTx.x**) that will contain the state of brake. Make sure that you include it in your program to test the brake status.

To aid you in determining the values of **BOFFTIME** and **BONTIME** you can see diagrams of the effect these values have by clicking the appropriate tab.

*3.2.6.5.3 Dynamic Brake*

This window enables you to activate (or deactivate) the dynamic brake.

This option is active only for those axes having a built-in drive.




You activate a dynamic brake by:

1. Selecting **Use Dynamic Brake** (this sets the **MFLAGS#DBRAKE** bit for the axis), and then
2. Entering a **Threshold Velocity** (**VFLBRK**) value.

Braking will occur when both of the following two conditions are met:

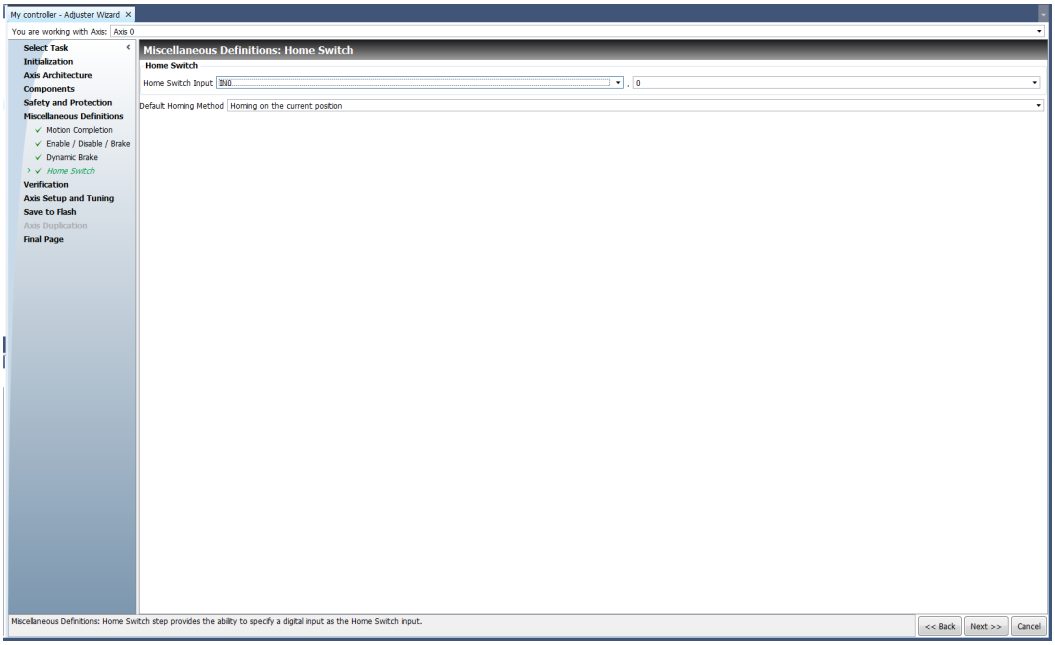
> The motor is disabled.

> The feedback velocity (**FVEL**) is less than **VFLBRK**.

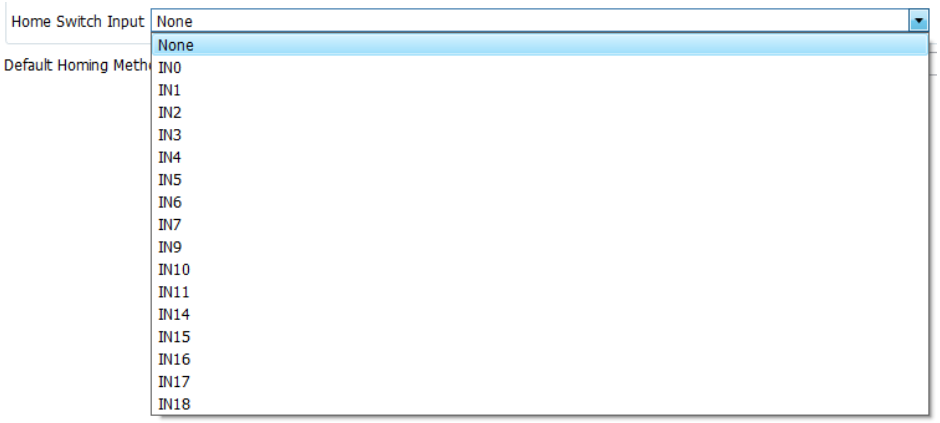
#### *3.2.6.5.4 Home Switch*

This window enables you to specify a digital input (**IN**) to signal when the motor is in its home position.

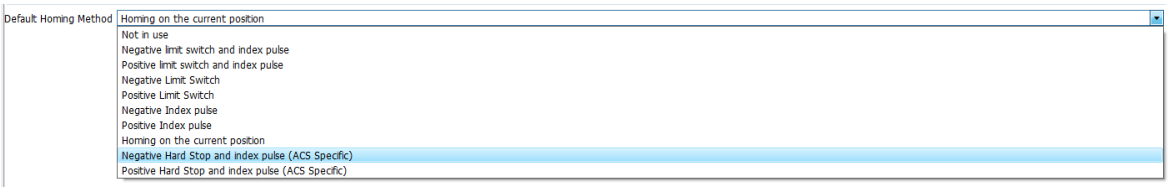
In a SPiiPlus motion controller there is no dedicated Home Switch. You use this **Adjuster** option to check a value that you want to be the Home Switch. Once you have completed the **Adjuster**, and tested the value (adjusting it as needed), you then have to incorporate the value into your motion program.




The Home Switch can be **None** or you can select the digital input variable from the **Home Switch** dropdown list:




In addition, the user can chose the default homing method from the list shown below.




### *3.2.6.6 Verification*

The Verification step examines the values you have entered and verifies that they are valid.

Verification is performed on the following:

- Feedback devices
  - Limit switches

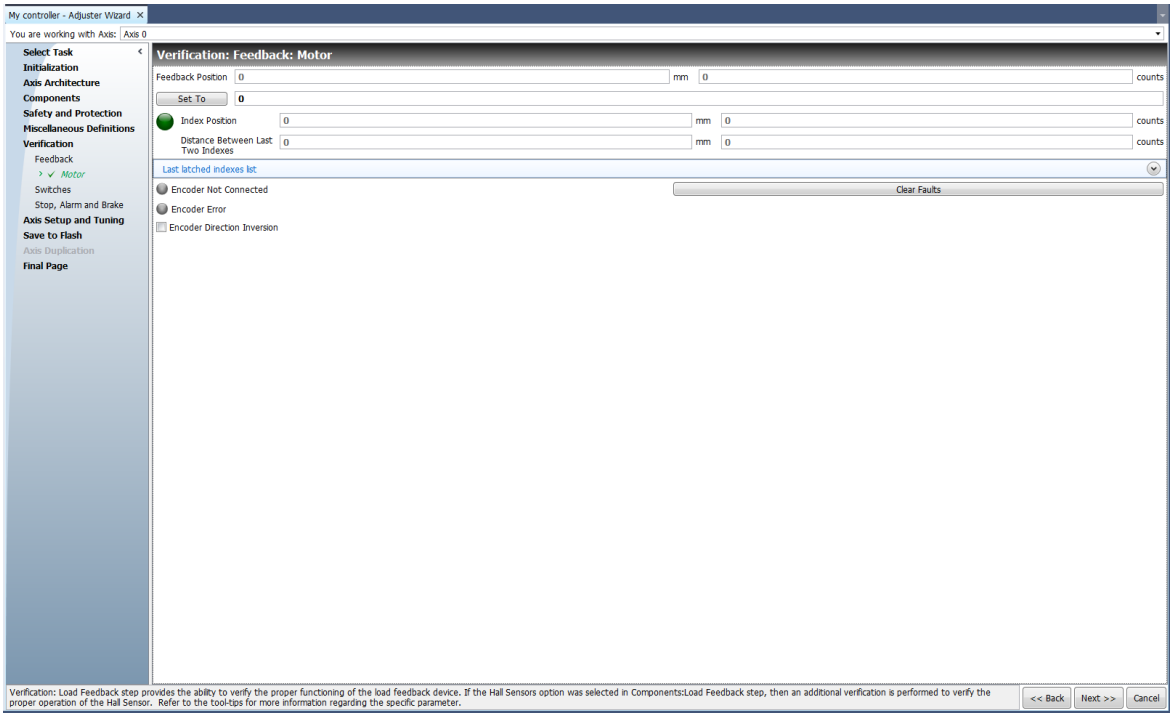
- Hardware Emergency Stop
- Mechanical Brake

There are three "LED" indicators used throughout the Verification windows:

| | |
|---|---|
|  | Indicates that the item passed verification |
|  | Indicates that the item failed verification |
|  | Indicates that the item is not active |

*3.2.6.6.1 Feedback*

This window verifies that the feedback values are valid.




Depending on what you defined in Components, the window displays the values for:

- Load
- Motor

If you have defined both Load and Motor as feedback components, there is one page for Load and one page for Motor.

On this page:

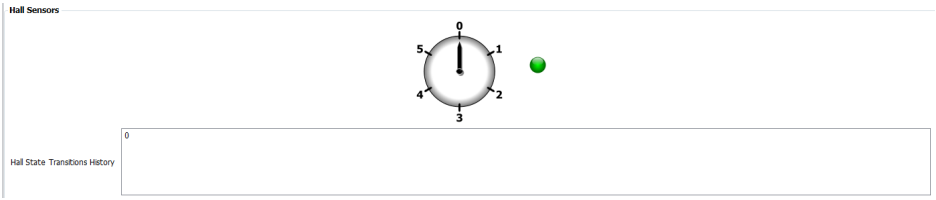
- Check that there are no encoder faults.
- Check that encoder counts in the right direction. If it does not, click the **Encoder Direction Inversion** check box to reverse the direction
- Check that the index is working properly by manually moving the motor. The motor should be moved slowly to ensure that each index is latched and displayed.
- Check that calculated distance between indexes is correct.

You can view a list of the last latched indexes by clicking the down arrow of the **Last latched indexes list**. To clear this list click **Clear List**.

There are two LEDs in this window relevant to the encoder:

- **Encoder Not Connected Fault** (FAULT.#ENCNC) - green indicates that this fault was not detected.
- **Encoder Error Fault** (FAULT.#ENC) - green indicates that no encoder fault was detected.

If you selected **Hall Sensors** in **Components:** Feedback, the following is displayed:




The connection sequence of the three Hall Sensors is not important. You only have to verify that the three Hall sensors are connected and the Hall counter counts 0,1,2,3,4,5 or vice versa. It does not matter if the Hall counters count opposite to the encoder. This will be identified and taken care of during Axis Setup and Tuning.

The diagram shows:

- Feedback Position (FPOS) - the position in user units.
- Hall State - the Hall State transitions: $0^{a}1^{a}2^{a}3^{a}4^{a}5$ or $5^{a}4^{a}3^{a}2^{a}1^{a}0$.

The history of the Hall State transitions is displayed under the diagram.

To test the Hall Sensors:

1. Rotate the motor by hand in one direction. The diagram should show the expected Hall state transitions.
2. Rotate the motor by hand in opposite direction. The diagram should show the expected Hall state transitions.

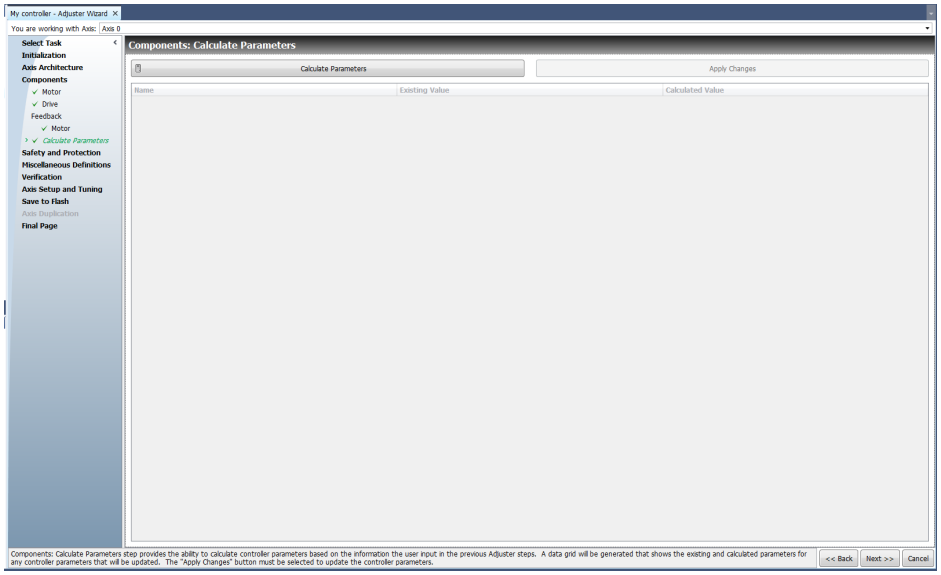
### Correcting Feedback Values that Fail Verification

Where feedback values fail verification:

1. Return to the **Components:**Feedback step by clicking **Components** in the Task List, then click **Feedback**. The **Components:Feedback** window is displayed.
2. Click **Edit** ( ) and enter a new value or values.

Those values that can be changed are displayed in bold face. Values that cannot be changed are grayed out.

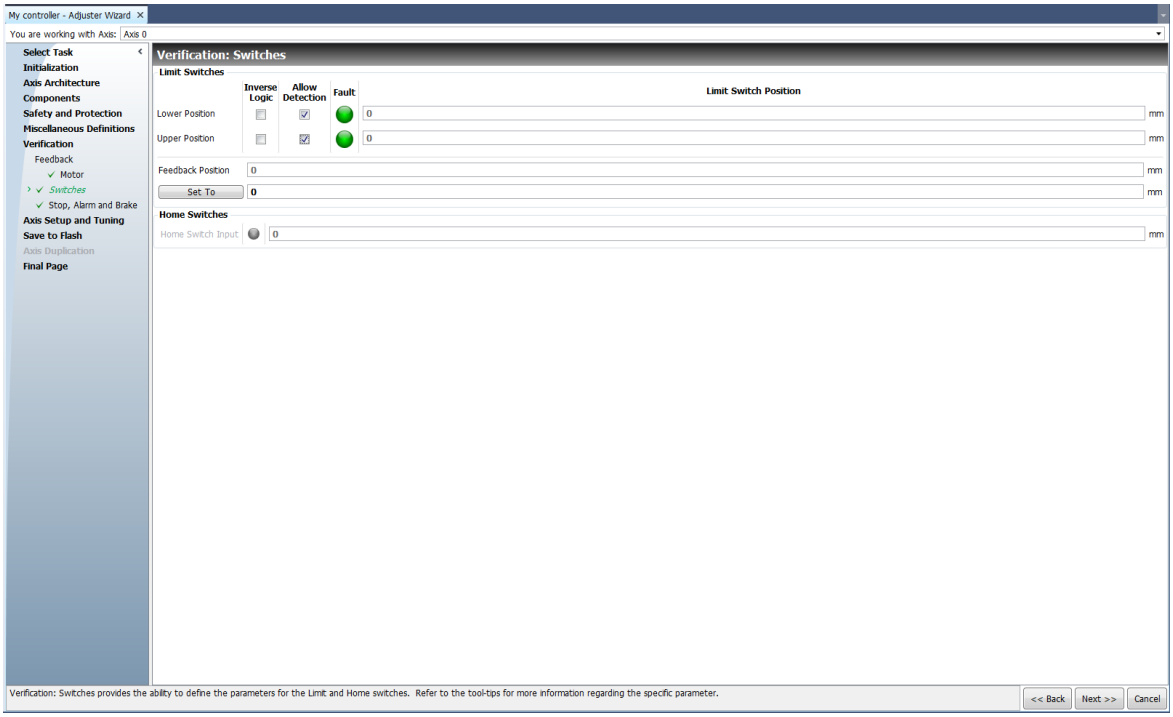
3. Click **Update in Database** and click **Next**. The **Components:Calculate Parameters** window is displayed.




4. Click **Calculate Parameters** then click **Apply changes**.
   **Adjuster** recalculates the values of the low-level controller parameters and applies them to the appropriate variables.
5. Click **Verification** in the Task List to return to Verification and check if the new value(s) pass verification.

*3.2.6.6.2 Switches*

This window is used for verifying the that actual limit switches are properly connected.




### Verification of Switch Operability

The LEDs indicate if the switches are properly connected or not. The switches are:

- Upper Position Limit (SAFINI.#RL)
- Lower Position Limit (SAFIN.#LL)

that you set in **Safety and Protection:**Position Limits.

If you have selected a digital input variable in **MISCELLANEOUS DEFINITIONS** Home Switch, its value is also displayed.

To verify that the switches are properly connected manually move the motor into the switch and check if the Fault LED turns red.

If it does not turn red, this means that there is something wrong with the switch. Check its connection and if it is operational.

### Fault Detection Settings

You have the options of:

- Enable/disable fault detection (FMASK.#RL for upper limit, FMASK.#LL for lower limit) by selecting the appropriate check box.

If the bit is empty, the controller does not detect the fault.

> Inverting the detection logic (SAFINI.#RL for upper limit and SAFINI.#LL for lower limit) by selecting the appropriate check box.

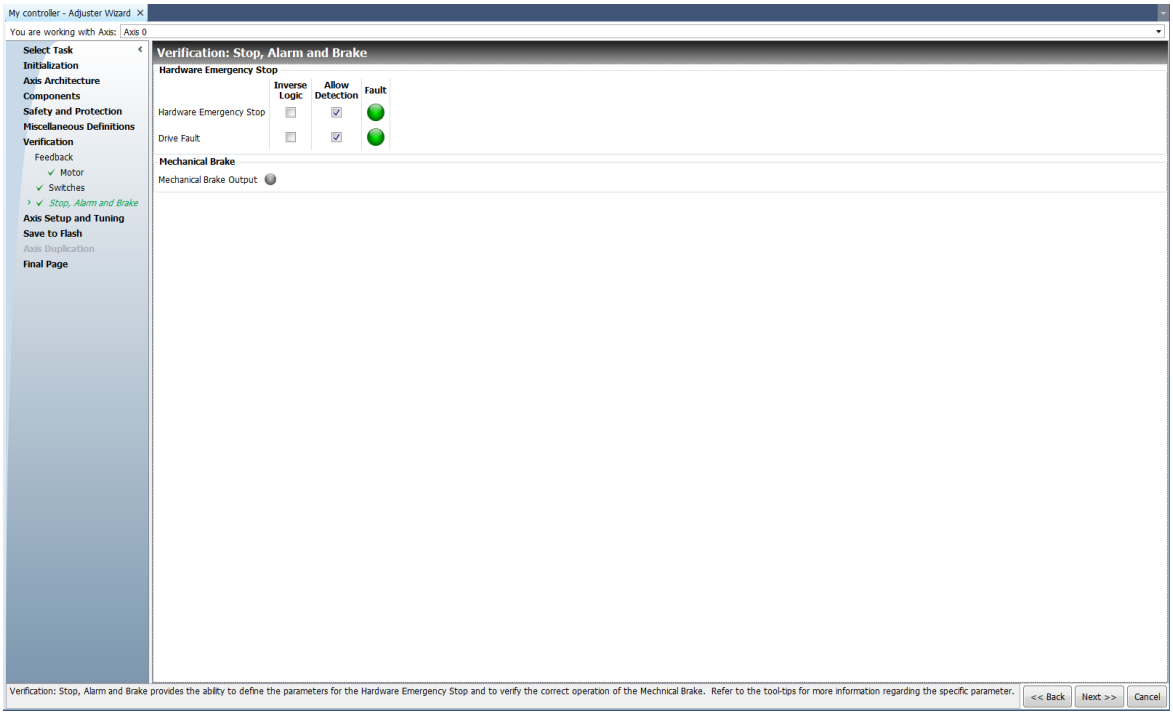
**Feedback Setting**

1. To set the feedback (FPOS) to a particular value:
2. Enter the value in the **Set To** field.

Click Set To - this resets the position and the Adjuster starts counting from this position.

*3.2.6.6.3 Stop, Alarm and Brake*

This window verifies the status of Hardware Emergency Stop and Mechanical Brake.




The window shows the status of the:

> Hardware Emergency Stop (S_SAFIN.#ES)
> Drive Fault / Laser Fault (SAFIN.#DRIVE)

You have the options of:

> Enable/disable fault detection (S_FMASK.#ES for Hardware Emergency Stop and S_FMASK.#DRIVE for Drive Fault / Laser Fault) by selecting the appropriate checkbox.

If the bit is empty, the controller does not detect the fault.

> Inverting the detection logic (S_SAFINI.#ES for Hardware Emergency Stop and SAFINI.#DRIVE for Drive Fault / Laser Fault) by selecting the appropriate checkbox.

### *3.2.6.7 Axis Setup and Tuning*

The Axis Setup and Tuning step enables you to fine tune your system. There are a total of six possible options:

1. Current Loop
2. Current Phase Offset
3. Commutation
4. Open Loop Verification
5. Position and Velocity Loops
6. Position Verification

The options that are available to you depend on the definitions you entered in Components.

#### *3.2.6.7.1 Current Loop*

Current Loop enables you to adjust the current loop.

Current Loop adjustment applies only to:

- PWM, Digital Current Control, controller-drive interface with:
  - DC brushless/AC servo motor (two or three phase)
  - Step motor (with feedback, close loop)
  - DC brush or single phase motor
  - AC induction motor
- Analog (±10V), Digital Current Control, controller-drive interface with
  - Three phase DC brushless/AC servo motor
  - DC brush or single phase

Two current loop tuning methods are available:

> Auto tuning - The auto tuning process calculates the current loop parameters from values measured by the frequency response.

> Manual tuning - The manual tuning process calculates the current loop parameters from a step response with parameters you set. Manual tuning is optional.

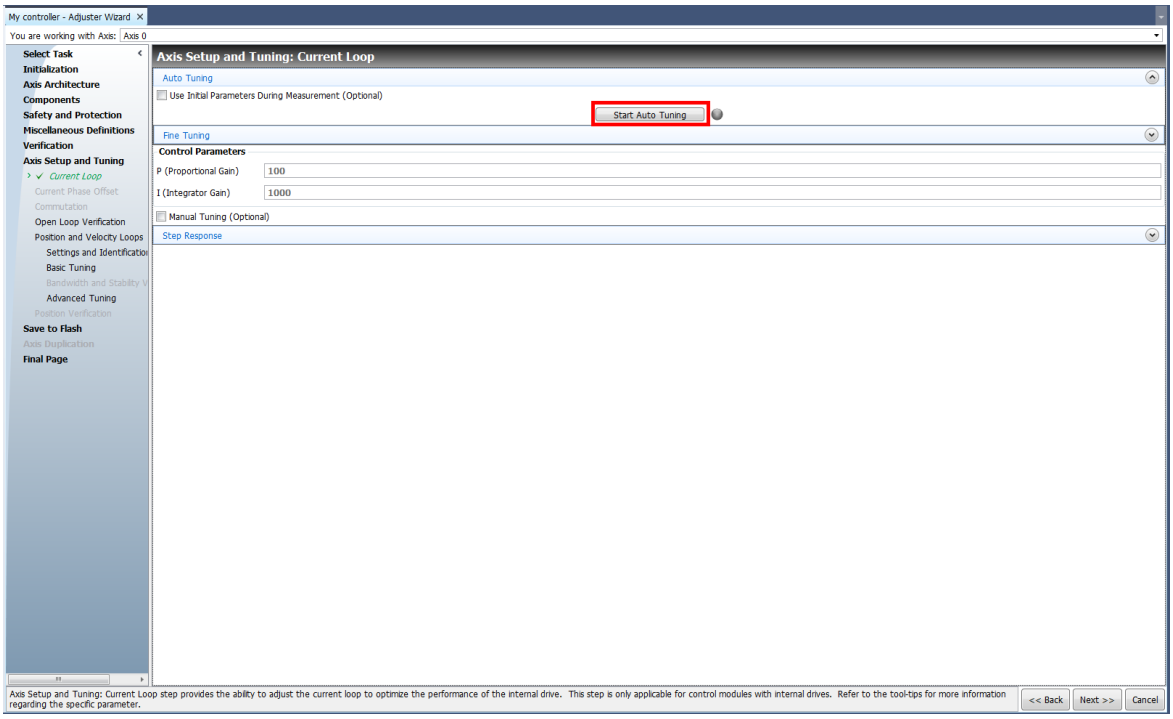
### Auto tuning

The calculated current loop parameters are:

> The Proportional Gain (SLIKP)

> The Integrator Gain (SLIKI)

1. To start the auto tuning process, click **Start Auto Tuning**.




> When the auto tuning process starts, the **Start Auto Tuning** button changes to **Stop Auto Tuning** and the LED next to it flashes "green".

> When the auto tuning process is complete, the LED returns to "gray" and **Done** shows to the right of it.

2. When the auto tuning is complete, click **Next** to go to the next adjustment.

During current loop tuning, the motor is enabled and no position or velocity control is executed. To remove the risk of motor motion due to gravity, do the following:

For motors with mechanical brake, it is necessary to:

1. Set MFLAGS.23 = 0
2. Tune the current loop.
3. When the current loop tuning is complete,set MFLAGS.23=1 to activate the mechanical brake.

For products that do not have a dedicated mechanical brake output, it is necessary to:

1. Send setconf(29, 0, AANN00) with an appropriate AANN00 value
2. Tune the current loop.
3. Send setconf(29, 0, AANN00) with an appropriate AANN00 value.

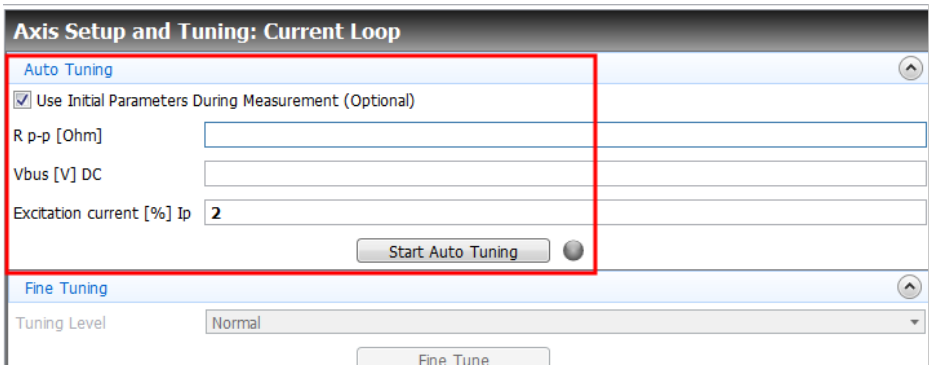
### Auto tuning with Use Initial Parameters During Measurement (Optional)

If necessary, the current loop auto tuning can be done from initial parameters that you set. The parameters are:

- Motor resistance phase-to-phase
- The rectified bus voltage
- Initial excitation current

To start the Auto Tuning process with the optional **Use Initial Parameters During Measurement**:

1. Click **Use Initial Parameters During Measurement (Optional)** check box.
2. Enter the values in the applicable field.
3. Select **Start Auto Tuning**.




4. When the auto tuning is complete, click **Next** to go to the next adjustment.

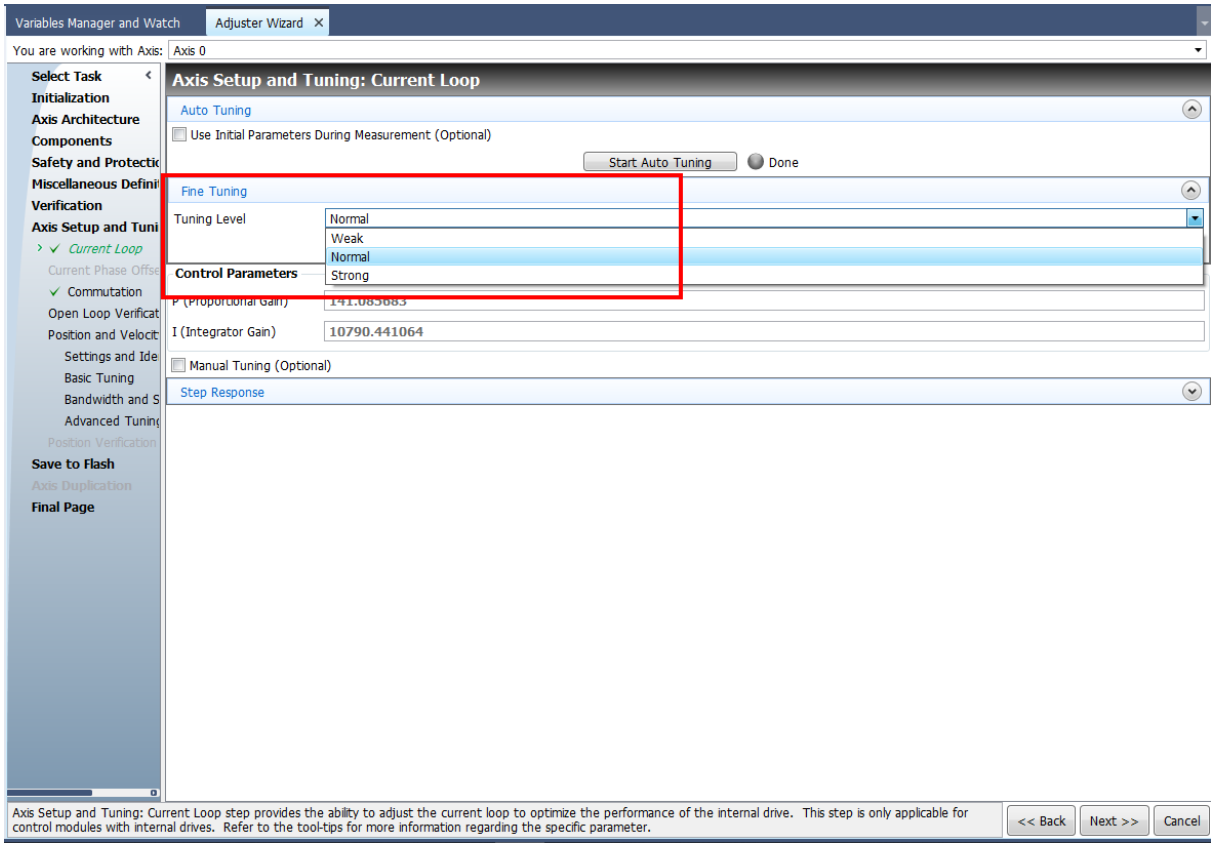
### Fine Tuning

When Auto tuning is complete, a **Fine Tuning** process can be done. Auto fine tuning is only for PWM drives with good linear properties. You can do fine tuning for three different types of drives :

- Weak (PWM drives with poor linear properties)
- Normal (PWM drives with good linear properties)

> Strong (PWM drives with good linear properties)

1. To start fine tuning, click **Fine Tuning** drop down icon to show the different tuning levels.
2. Select the applicable drive.
3. Click **Fine Tune**.

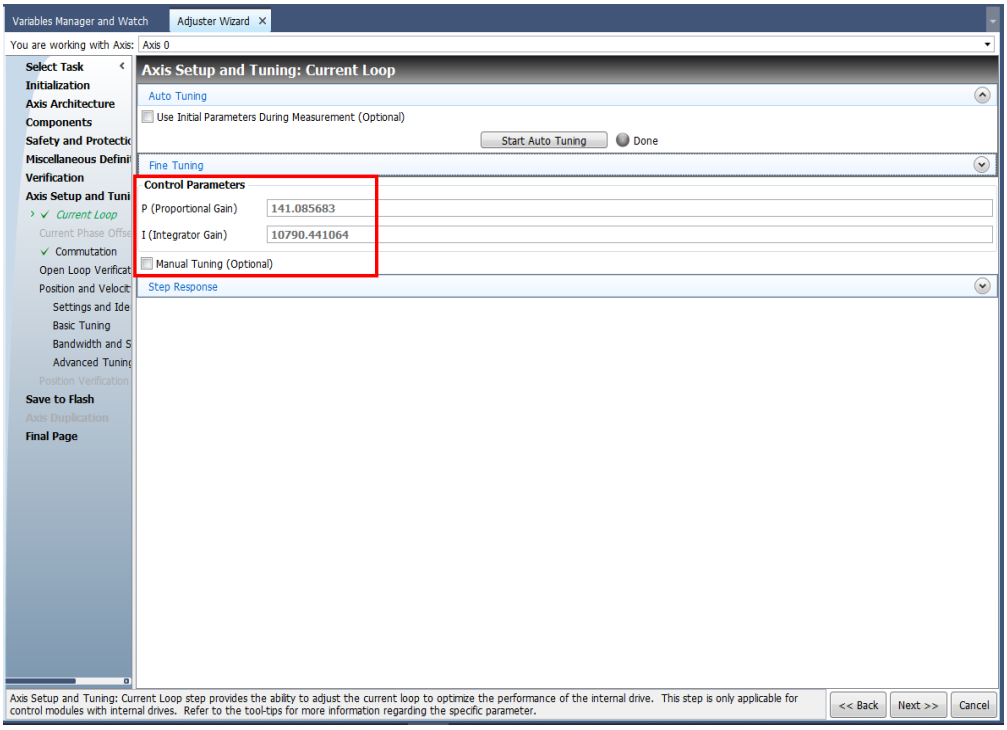



4. When the fine tuning is complete, click **Next** to go to the next adjustment.

### Manual Tuning (Optional)

If you select manual tuning, you can set the proportional gain (SLIKP) and the integrator gain (SLIKI).

1. Click the **Manual Tuning** check box.
2. Enter the values in the applicable field.




3. Click **Next** to go to the next adjustment.

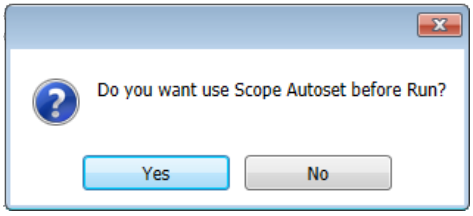
### Step Response

Use step response to verify the proportional gain and integrator gain. The required parameters to do a step response are:

> Current level

> Pulse width

1. To start the step response, click the **Step Response** drop down box to show the parameter fields available.
2. Enter the parameter values.
3. Click **Check Step Response**. A dialog box opens and shows the message.

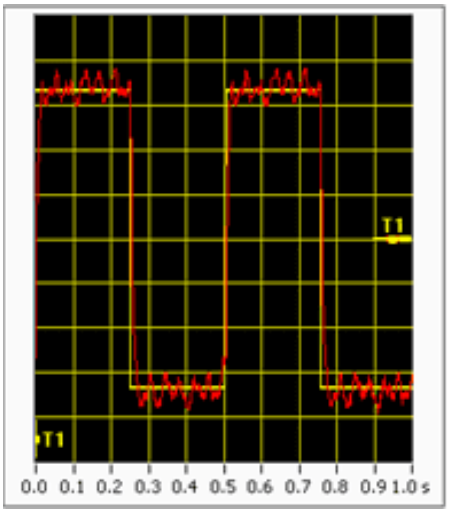



4. Select the applicable condition.

When the step response starts, the **Check Step Respnse** button changes to **Stop** and the LED next to it flashes "green".

5. To make adjustments on the signal:

> Increase the Proportional Gain (SLIKP) until the response waveform approximates a square and a small overshoot appears, for example:




> Increase the Integrator Gain (SLIKI) by hundreds, until you get a narrow overshoot of 10% to 20%.

6. When you achieve the optimal values, click **Stop**. Close the Scope.
7. Click **Next** to go to the next adjustment.

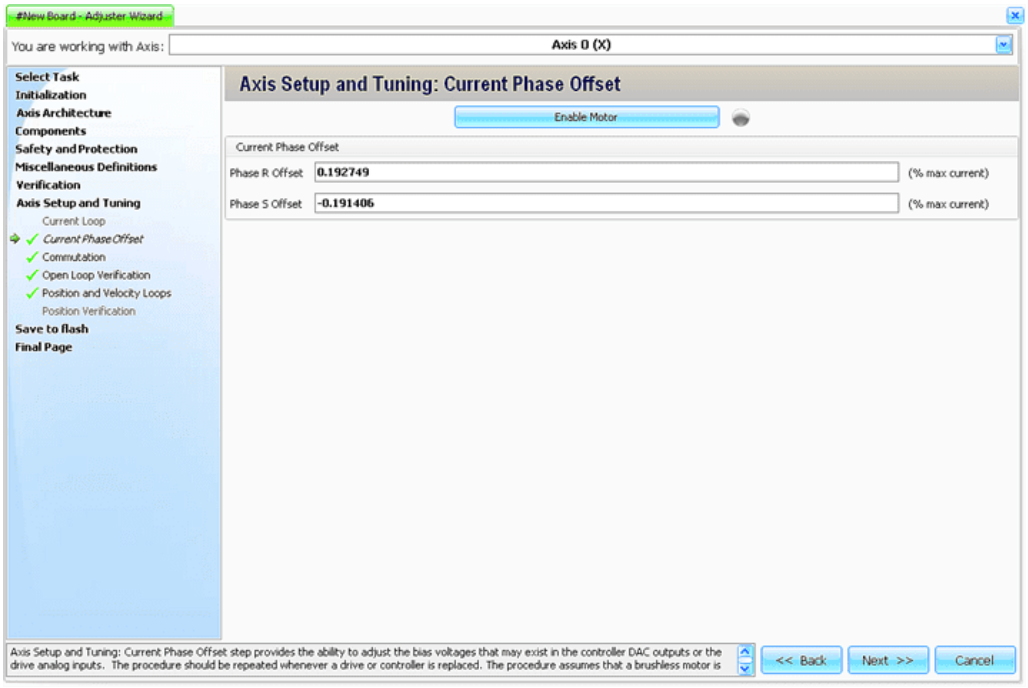
*3.2.6.7.2 Current Phase Offset*

The Current Phase Offset window enables you to verify, and, if needed, adjust the values of the current phase offset.

Current Phase Offset applies to Analog (±10V) controller-drive interfaces with two or three phase DC brushless (AC servo) motor connected to an external drive with two drive command inputs (also known as a UV drive).

Theoretically, the drive should produce zero voltage in the three phases when the controller drive commands are zero. If, however, the drive outputs have a bias voltage when the controller commands are zero, this can be corrected with the SLBIASA and SLBIASB variables.

If the external drive is a current drive, it must be connected to the motor while doing the bias adjustment. If it is a voltage drive, it doesn't have to be connected.




There are two current phase offset variables:

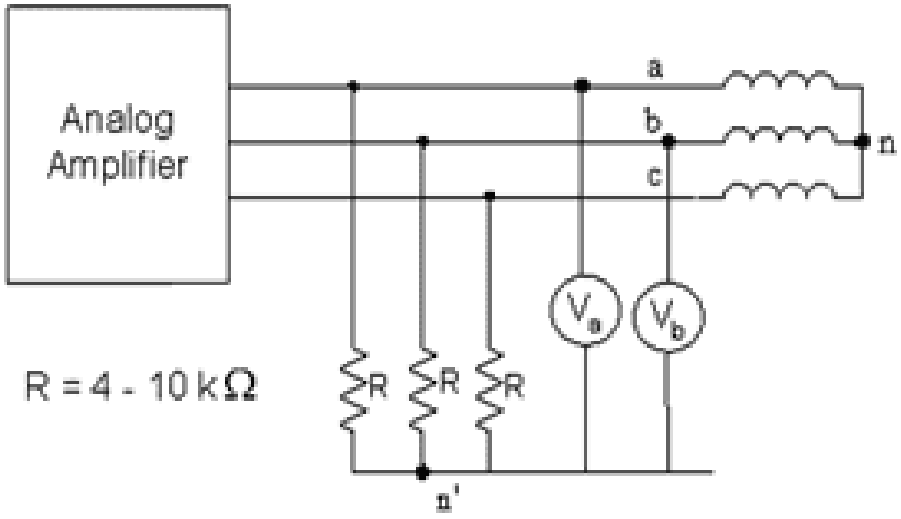
- **Phase R Offset** (SLBIASA) which contains the maximum controller voltage output (expressed as a percentage).
  - For integrated models: SLBIASA specifies the measured value of the current input bias of Phase R.
  - For nonintegrated models: SLBIASA specifies the bias of the drive output. The controller uses the value only for brushless motors commutated by the controller.
- **Phase S Offset** (SLBIASB) which is used for defining the maximum controller current (expressed as a percentage).
  - For integrated models: SLBIASB specifies the measured value of the current input bias of Phase S.
  - For nonintegrated models: SLBIASB specifies the bias of the drive output. The controller uses the value only for brushless motors commutated by the controller.

Click **Enable Motor** and check that the LED turns green.

If the LED turns red, this means that one or both values need to be adjusted, in which case, the variables **SLBIASA**, **SLBIASB** have to be adjusted to bring both readings as close to zero as possible. This can be done by a single voltmeter (or multimeter) as follows:

1. Make sure that the value of the **DOUT** variable is zero (you can query it from the Communication Terminal, for example: **?X_DOUT**).
2. Connect the voltmeter between phase "R" and the motor neutral point, and adjust **SLBIASA** to make the reading as close to zero as possible. A typical range is ±10%. The value can be either positive or negative.
3. Connect the voltmeter between phase "S" and the motor neutral point, and adjust **SLBIASB** to make the reading as close to zero as possible.

If the neutral point is not accessible, or the motor is connected in delta, you may create an "artificial" neutral point by connecting three-phase resistors in parallel to the motor. The 3 resistors should be identical, in the range of 4-10 kΩ. Then you should measure relative to the artificial neutral as shown in the following drawing:




When you are done, click **Next** to go to the next window.

*3.2.6.7.3 Commutation*

The Commutation window enables you to adjust motor commutation and generate a commutation startup program.

Commutation applies only for DC brushless (AC Servo) motors. It will appear in the Adjuster tasks only if motors of this type have been defined as part of the system.

**Adjuster and Commutation**

In a DC brush motor, a fixed magnetic field created by permanent magnets in the stator interacts with the armature current flowing in the rotor winding. The interaction of the current-carrying conductors with the magnetic field produces a torque (or force in the case of a linear motor). This torque/force is at its maximum value when the magnetic field vector is perpendicular to the resultant current vector.

The mechanical commutator of a DC brush motor distributes the motor current among the windings such that the resultant current vector remains perpendicular to the magnetic field vector at any position and speed. This process is referred to as commutation.

In a DC brushless/AC servo motor, on the other hand, an electronic drive takes the place of the mechanical commutator, keeping the resultant current vector perpendicular to the magnetic field vector by controlling the phase currents.

The **Adjuster** Commutation process creates a startup program that is loaded into the controller which enables the controller to commutate the drive correctly based on the motor and encoder electrical properties.

For this reason the **Adjuster** Commutation has to be performed for every new system. Once it has been performed, whenever a new motor or encoder is introduced into your system so long as you do not change the type of motor or encoder, you do not have to perform it again.

SPiiPlus motion controllers support three types of drives for 3-phase DC brushless/AC servo motors:

- **Universal Internal Drive** (such as SPiiPlus CM and MC4U drives). Commutation is performed by the controller as part of its Field-Oriented Control algorithm.
- **Two-Input Drive** (such as SPiiPlus SA controller or SPiiPlus PCI card with a Sine Wave Brushless Amplifier). The drive receives two current commands from the controller, reflecting required current for two of the motor phases. In this case **the controller is responsible for commutation**.
- **One-Input External Drive** (such as a SPiiPlus SA controller with an Analog Input Brushless Amplifier). This drive receives a single current command from the controller (±10V), reflecting the required current amplitude, and takes care of the commutation by itself.

Commutation needs to be done only for types 1 and 2.

Motor commutation requires an initial setup process in which an initial current vector is generated and the magnetic field of the motor aligns with it (at what is called a ***detent point***). Once the magnetic field is aligned with the current vector, it is shifted $90^{o}$ thereby achieving maximum torque/current ratio.

Since the commutation angle depends on the motor position, the commutation process has to be done only once for an absolute encoder but after every power-up for incremental quadrature and Sin/Cos encoders – since the motor position is not known.

Therefore, when working with an incremental encoder and a DC brushless motor, commutation setup must be executed after every powerup. For this reason you can use **Adjuster** to generate and store a **Commutation Startup Program**.

The **Adjuster** executes a program that performs Commutation Adjustment setup. The program supports several commutation schemes (procedures). You select the most appropriate scheme for your application. In most cases, the commutation scheme will include the following steps:

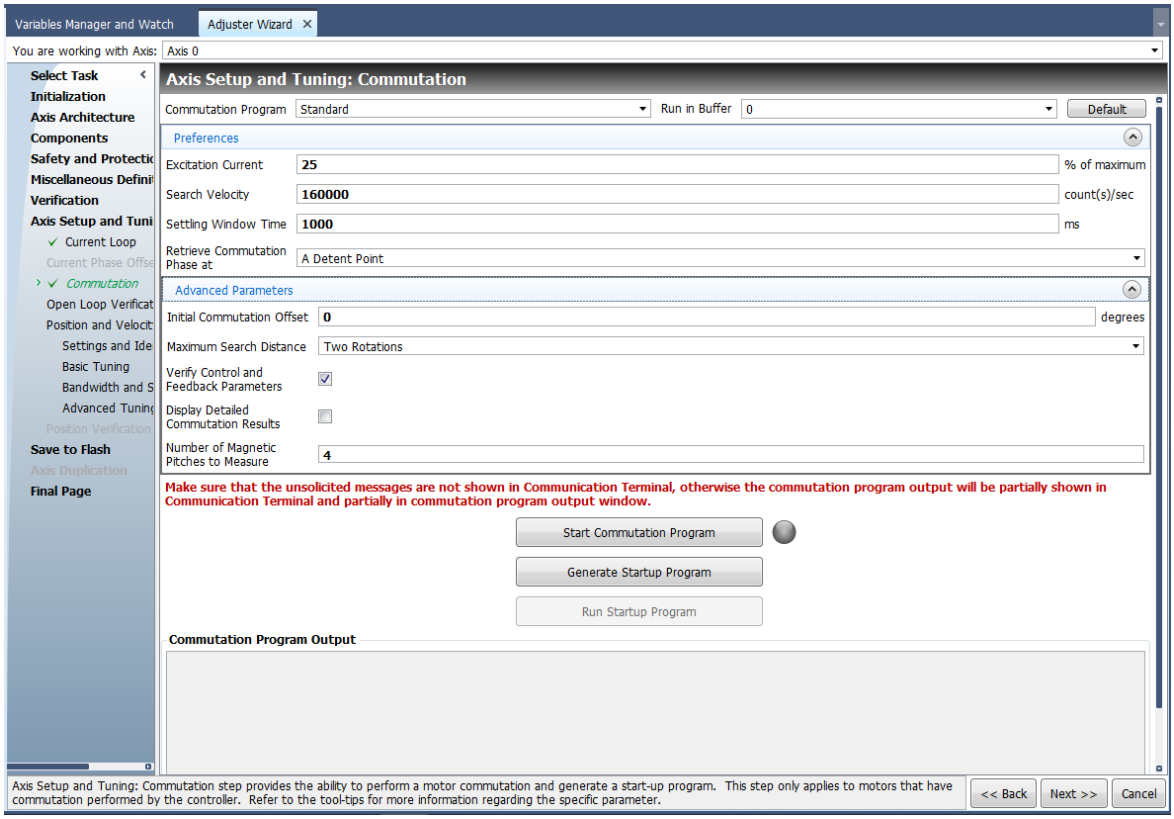
1. Find a detent point.
2. Identify the phase sequence both of the motor and the encoder.

3. (Optional): Verify variables (number of poles or magnetic pitch, encoder resolution) by moving the motor several magnetic pitches and comparing the encoder feedback with the expected results.
4. (Optional): Move to the index and save the commutation phase at that point to the controller flash memory.

The **Commutation Startup Program** is an ACSPL+ program that is used to retrieve commutation. It can be executed upon every powerup (once the system has been setup). The startup program can be based either on bringing the motor to a detent point (usually involves movement) or on **automatic commutation (COMMUT** command)**,** which is fast commutation retrieval using a closed-loop algorithm (and involving almost no motor movement).

When the current vector is aligned with the magnetic field during commutation, the motor can jump. The maximum jump is one magnetic pitch (180 electrical degrees) of the motor in either direction. If the motor bumps into an obstacle, the commutation setup algorithm will attempt recovery. The recovery may involve additional abrupt moves.

To prevent possible damage or injury, it is recommended that the motor be initially positioned at least one magnetic pitch away from any obstacles.




When the page is first displayed, Preferences and Advanced Parameters are not displayed. It is strongly recommended that you expand both and view all the options available to you.

Your options are:

### Preferences

When initially setting **Preferences**, you should first use the default values (by clicking Default). This will set the commutation parameters in accordance with the settings you have entered in previous steps of the Adjuster.

| | |
|---|---|
| Excitation Current | Determines the amplitude of the current vector that is used for commutation setup. The value should not exceed the nominal current ratings of the drive and the motor. The default value is: 0.95 × **XRMS**.<br>Things you should consider:<br>> Due to friction and load, a low excitation current may result in poor alignment between the current vector and the magnetic field, resulting in unsatisfactory commutation.<br>> On the other hand, too high an excitation current with a low friction system (inadequate damping) may result in excessive oscillation and even in mechanical damage.<br>It is recommended starting the commutation process with an excitation current of 0.4 x **XRMS**. If the results are not successful, repeat this task with higher excitation current values. |
| Search Velocity | Determines the velocity of the current vector during the commutation process. During this process the motor position follows the current vector in order to align the magnetic axes. It is recommended to set a velocity lower than one magnetic pitch per second and higher than one fifth of a magnetic pitch per second. |
| Settling Window Time | Determines the time assigned to the motor for settling at detent points. |
| Retrieve Commutation Phase at | In all the schemes except the powerup position scheme, the following steps are performed:<br>> Settle at a first detent point.<br>> Move the current vector to identify the phase sequence. Automatically correct for wrong polarity.<br>> If the feedback indicates that the motor has moved less than 50% of the expected distance, it is assumed to have run into an obstacle and the program tries to recover in the opposite direction.<br>> Retrieve the commutation phase according to the selected scheme.<br>Considerations when selecting a Commutation Scheme:<br>1. Based on detent point: If the feedback device has no index, this is the only scheme that can be used. If the feedback device does have an index it is recommended to use one of the "index" schemes so that the index can be used as an absolute reference point for commutation phase retrieval at subsequent powerups.<br>2. Based on index: Saves the value of the commutation phase at the index position. This value can be used at subsequent system |

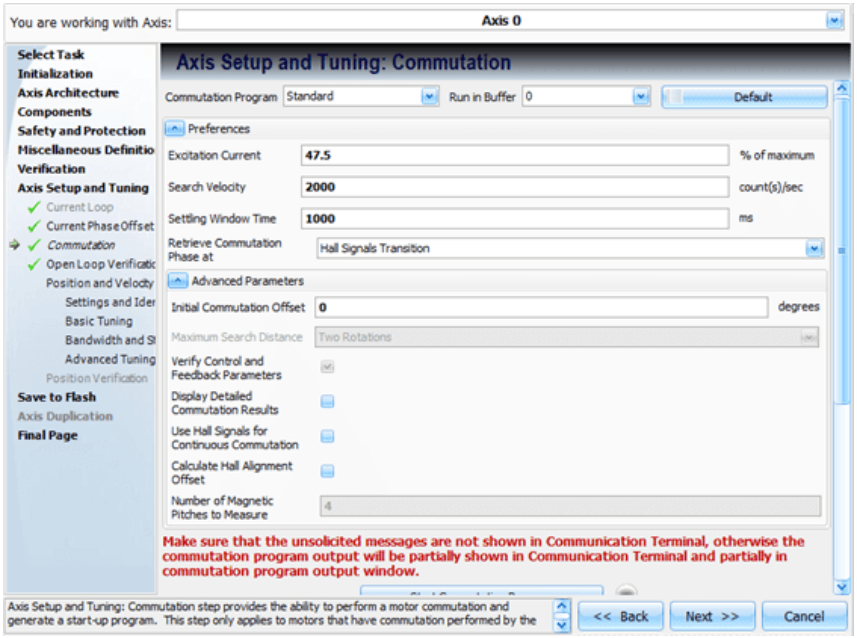
| | |
|---|---|
| | powerups to achieve the same commutation result. Six variations of this scheme are available:<br>> First Index in Positive Direction<br>> First Index Next to Right Hard Stop<br>> First Index Next to Right Limit Switch<br>> First Index in Negative Direction<br>> First Index Next to Left Hard Stop<br>> First Index Next to Left Limit Switch<br>3. Other<br>> Powerup Position<br>> Hall Signals Transition<br>> Current Absolute Position |

**Advanced Parameters**

| | |
|---|---|
| Initial Commutation Offset | Determines the initial value of the commutation phase, and, thereby, the initial orientation of the current vector and the initial detent point.<br>By properly setting this offset, you can avoid an initial jump in a system with a predefined startup position. For other systems it can be used to determine a specific target position for the initial jump.<br>If no information is known about the relationship between the current vector and the motor position, it should be set to **0** (the default). |
| Maximum Search Distance | Sets the maximum allowed distance for searching for targets (index or limits). It is recommended to set the variable according to the operational distance. |
| Verify Control and Feedback Parameters | Causes the system verify the encoder and motor variables during the initial commutation process. If the variables seem to be configured wrong, a general message will be displayed stating that improperly configured variables are a possible reason for failure. Once corrected, you will be able to continue. In a case of more than 20% deviation between the defined and measured pitch, the commutation process reports failure. |
| Display Detailed Commutation Results | Displays commutation phase data (in electrical degrees). |

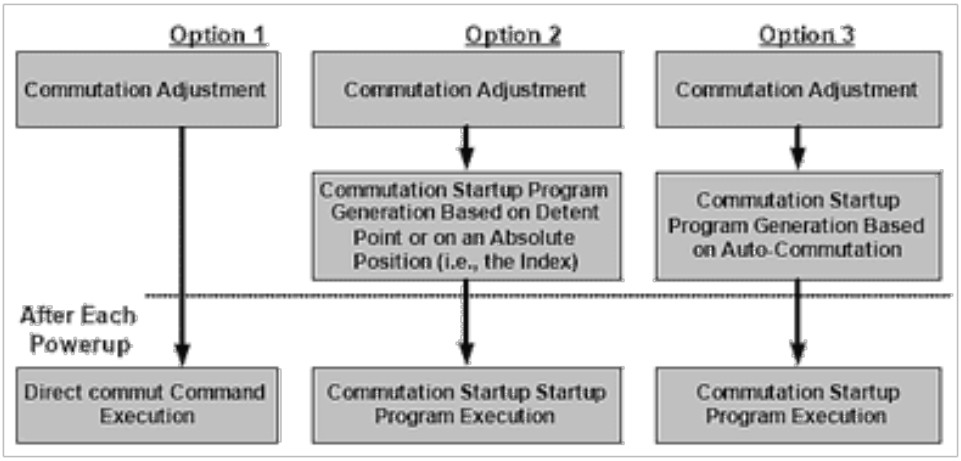
| | |
|---|---|
| Use Hall SIgnals for Continuous Commutation* | This option resolves a commutation loss due to improper incremental encoder-to-motor rotation counts. When this option is selected, the commutation angle is adjusted at each hall transition. In between hall transitions, a sinusoidal commutation is calculated based on incremental feedback. |
| Calculated Hall Alignment Offset* | Compensates for Improper hall alignment and optimizes torque/force production. |
| Number of Magnetic Pitches to Measure | A word of caution: Deviation between defined and measured pitch does not necessarily indicate poor commutation quality. For example, incorrect definitions of feedback variables may be compensated for by high friction, thus the deviation may be low but the commutation will be poor. |

*Only available when Hall Signals Transition is the selected option for "Retrieve Commutation Phase at". This is the default condition if Hall sensors were selected.



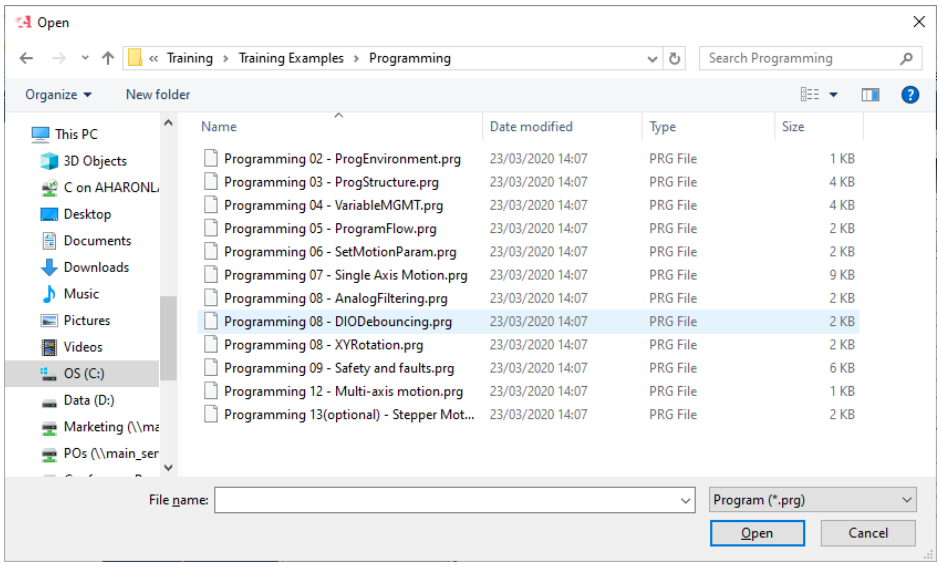

### Commutation Startup Program

The options that you have when setting the Commutation Startup program are illustrated in the following diagram:




To set the Commutation Startup program:

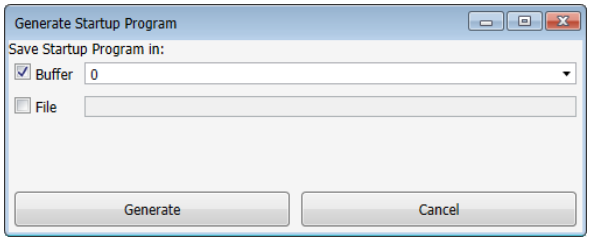
1. Select one of the two Commutation Program options from the **Commutation Program** dropdown list:
   - > **Standard**: Uses a built-in default commutation scheme that is incorporated in **Adjuster**.
   - > **User Defined**: For selecting a commutation program that was previously saved in a file (in this case you need to have prepared the Commutation Startup program through Program Manager). A browser is displayed enabling you to search for and load your program:




2. Select the required buffer for storing the Commutation Startup program from the **Run in Buffer** dropdown list (the controller supports up to 9 simultaneously executing program buffers). Usually, you will want to select an empty buffer to avoid overwriting an existing program.
3. Set your preferences (see Preferences and Advanced Parameters), or click **Default** which will enter default values in the preferences.
4. At this point you have two choices:

a. Start Commutation Program - causes the **Adjuster** to automatically generate the Commutation program taking the **Retrieve Commutation Phase at** parameter as its starting point. It compiles the program and runs it.

b. Generate Startup Program - causes the **Adjuster** to generate the Commutation Startup program. You are prompted with:




The **Buffer** checkbox should be selected and the buffer number that you entered in Step 2 should appear. If the program is to be generated from a user file, select the **File** checkbox and click ... to browse for the file. If you want to use automatic commutation, select the **Use Auto-commutation** checkbox.

Automatic commutation is a way to retrieve the commutation offset automatically, using closed-loop control.
Commutation setup (without using automatic commutation) is required once: during Commutation Adjustment. Once this has been done, automatic commutation is recommended at every controller powerup.

5. Click **Generate**. In the **Program Manager** pod, note that program appears in the specified buffer, and a notification that the program was successfully generated is displayed.
6. Click **Start Startup Program** to run the Commutation Startup program.
7. Observe the motion of the motor and the messages that appear in the **Commutation Program Output** panel. If errors appear or the motor does not respond as it should, make adjustments to the parameters and run again - refer to Troubleshooting Commutation.

### Commutation Program Output Panel

There are two buttons associated with the **Commutation Program Output** panel:

> Clear Output - clears the messages in panel.

> Save Printout - enables you to save all the messages to a text file.

### Troubleshooting Commutation

If the motor does not move during **Start Commutation Program** or **Start Startup Program** execution, consider the following:

- Check the connections between the controller and the drive and motor.
- If a high friction or active load is applied, increase the **Excitation Current** and try again.

If **Start Commutation Program** or **Start Startup Program** execution displays a message about wrong motor variables, return to Components and verify the following:

- **Number of Poles** or **Magnetic Pitch** are specified correctly.
- Encoder variables are specified correctly.

If the motion failed during Commutation Adjustment or Commutation Startup program execution, probable causes are:

- The required **Excitation Current** is greater than the minimum of **XRMSM** value and **XRMSD** value. If the motor and drive rating allow it, try to increase this value.
- There is an obstacle in the path of the motion.

Another problem that can arise is the case where **Verify Control & Feedback Parameters** is not selected in the **Advance Preferences**. Inaccurate variables can cause the commutation process to report successful completion even though the commutation is actually wrong. Therefore it is recommended to select this field.

Possible reasons for failure of the commutation process:

- Hard stop: motor has bumped into a hard stop or an obstacle and is unable to move.
- Excitation current too low: results in poor field alignment.
- Inaccurate motor and feedback variables.
- Hardware problems, such as encoder or drive fault, wiring error, or bad grounding.

8. When you are done, click **Next** to go to the next window.

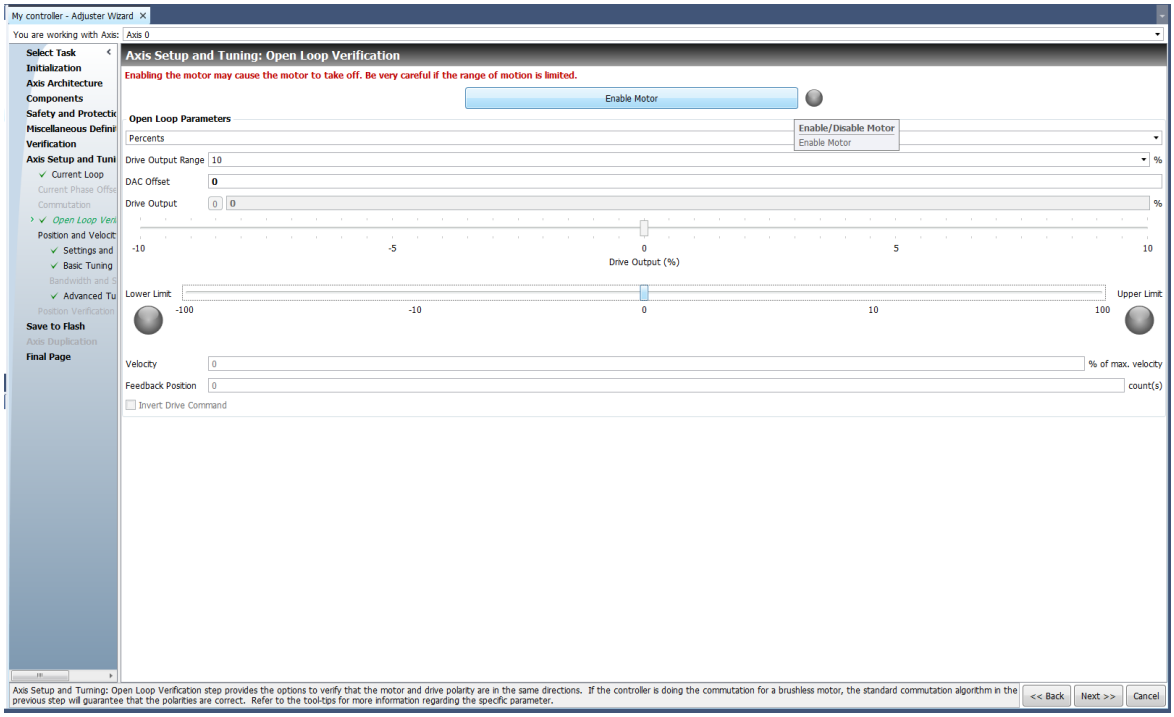
*3.2.6.7.4 Open Loop Verification*

The Open Loop Verification window enables you to check if the motor responds correctly to the drive command, that is, the motor velocity should be in the same direction as drive command.

Enabling the motor may cause it to jump. To avoid personal injury or damage to the equipment, check the following before enabling the motor:

- Ensure that nothing (people, cables, or other obstacles) is in the path of the motor or objects connected to the motor.
- Ensure that the motor is securely anchored and that proper safety barriers, stops, and limits are installed.

Be ready to engage the Emergency Stop switch.




Values can be expressed either as:

- % - Percentages
- Amps - Amperes

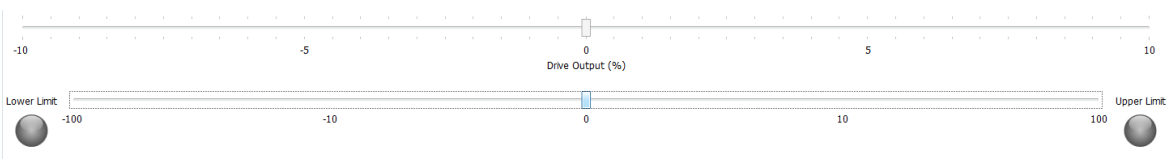
which you select from the dropdown list.

The parameters on this window are:

- **Drive Output Range** - Percentage (or amperage) relative to the maximum voltage as defined in **Components** - see Drive.
- **DAC Offset** - Offset to be added to the results of the current loop control. This value is stored in the **SLIOFFS** variable.

  The offset can be used to compensate for an active component of the motor load. For example, in a vertical axis the weight of the carriage can be compensated.

1. Click **Enable** to enable the drive.
2. If an encoder is active in the system, click **Zero** to reset the encoder.
3. Drag the Drive Output slider slowly to the right until the motor moves. If the motor does not move (due to friction), increase the **Drive Output Range** (with the slider or by entering a value) until the motor moves.




4. Verify that the velocity gauge indicator moves to the right (same direction you moved the slider). If the velocity arrow moves in the opposite direction, select **Invert Drive Command**.

If the direction of the velocity gauge is opposite to that of the Drive Output slider, it is an indication of positive feedback. This is likely to cause a motor run away. If the direction of the velocity gauge is opposite to that of the Drive Output slider, it is an indication of positive feedback. This is likely to cause a motor run away.

5. For DC brush motor only: If there is bias (an offset) in the drive (motor moves while the controller command is zero), then with **Driver Output Range** set to zero, use the **DAC Offset** (SLIOFFFS) field to compensate.
6. When you are done, click **Next** to go to the next window.

*3.2.6.7.5 Position and Velocity Loops*

The Position and Velocity Loops window enables you to tune the position and velocity loops in order to optimize the servo system performance.

To tune the Position and Velocity Loops we recommend using One Parameter Based Tuning as described below. One Parameter Based Tuning has been tested on several demanding applications and achieved excellent results.

Experienced control engineers can make use of Manual Tuning or Advanced Tuning.

**One Parameter Based Tuning**

This feature enables you to tune the system based on a single parameter - the required bandwidth of the Position loop.

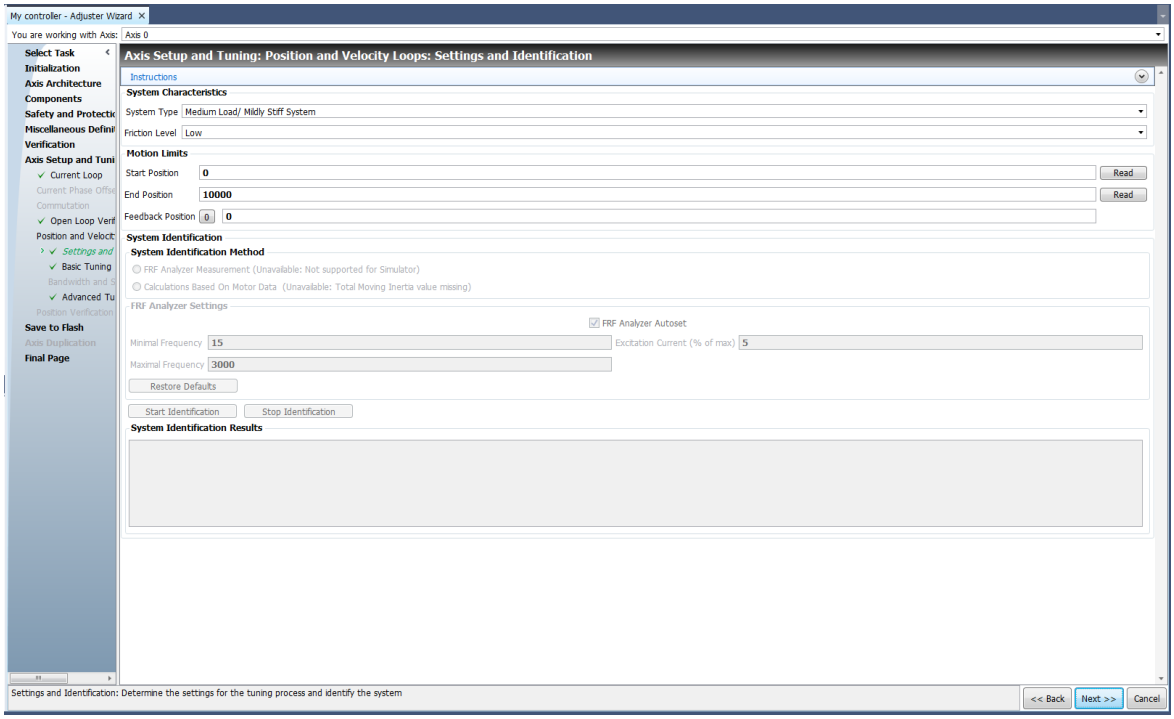
The Adjuster Wizard measures and identifies the characteristics of the electro-mechanical system. Once identified, you specify the required bandwidth and the Adjuster determines the relevant servo parameters and gains required to achieve the specified bandwidth.

You can now verify that the system performs as required and is stable by activating a motion and monitoring the response with the scope. You can modify the bandwidth parameter accordingly and repeat the process until an optimal time response is achieved.

Afterwards, you can use the FRF to verify that the required bandwidth is achieved and that the stability margins are adequate.

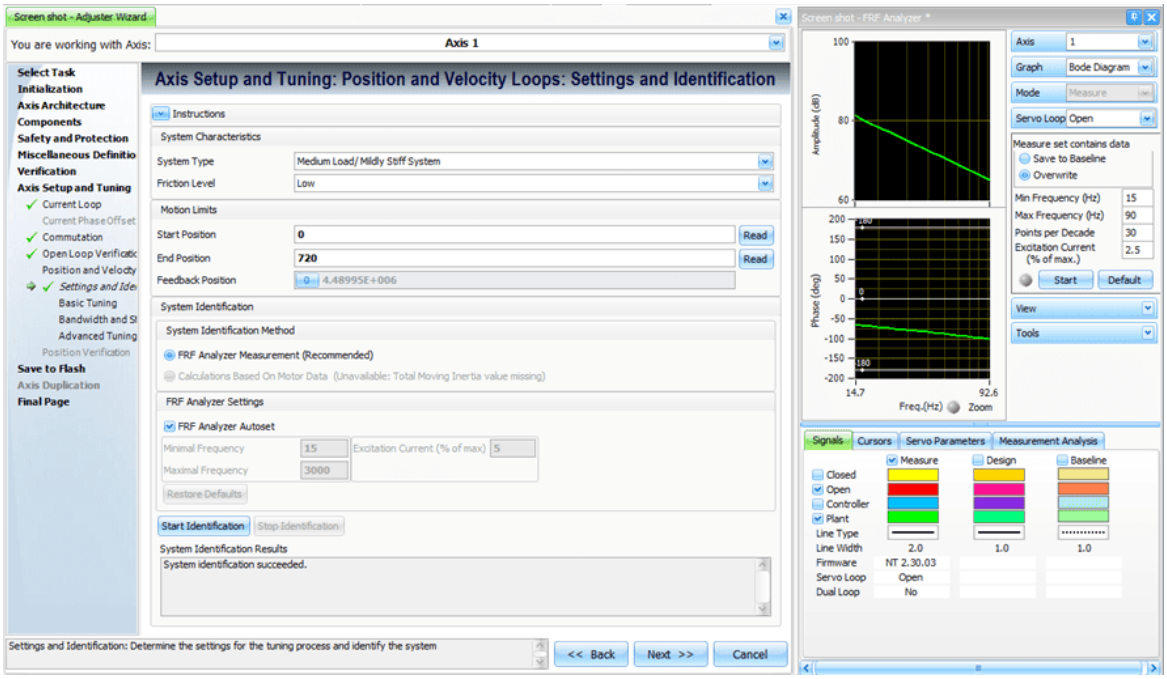
**To perform one parameter tuning:**

1. In the **Settings and Identification** window, select the characteristics of the system in the **System Characteristics** section. If unknown, use the defaults.
2. Define start and end motion limits position in the Motion Limits section either by entering values or by clicking the **Read** buttons. Ensure that the values are adequately distanced from the actual travel limits.

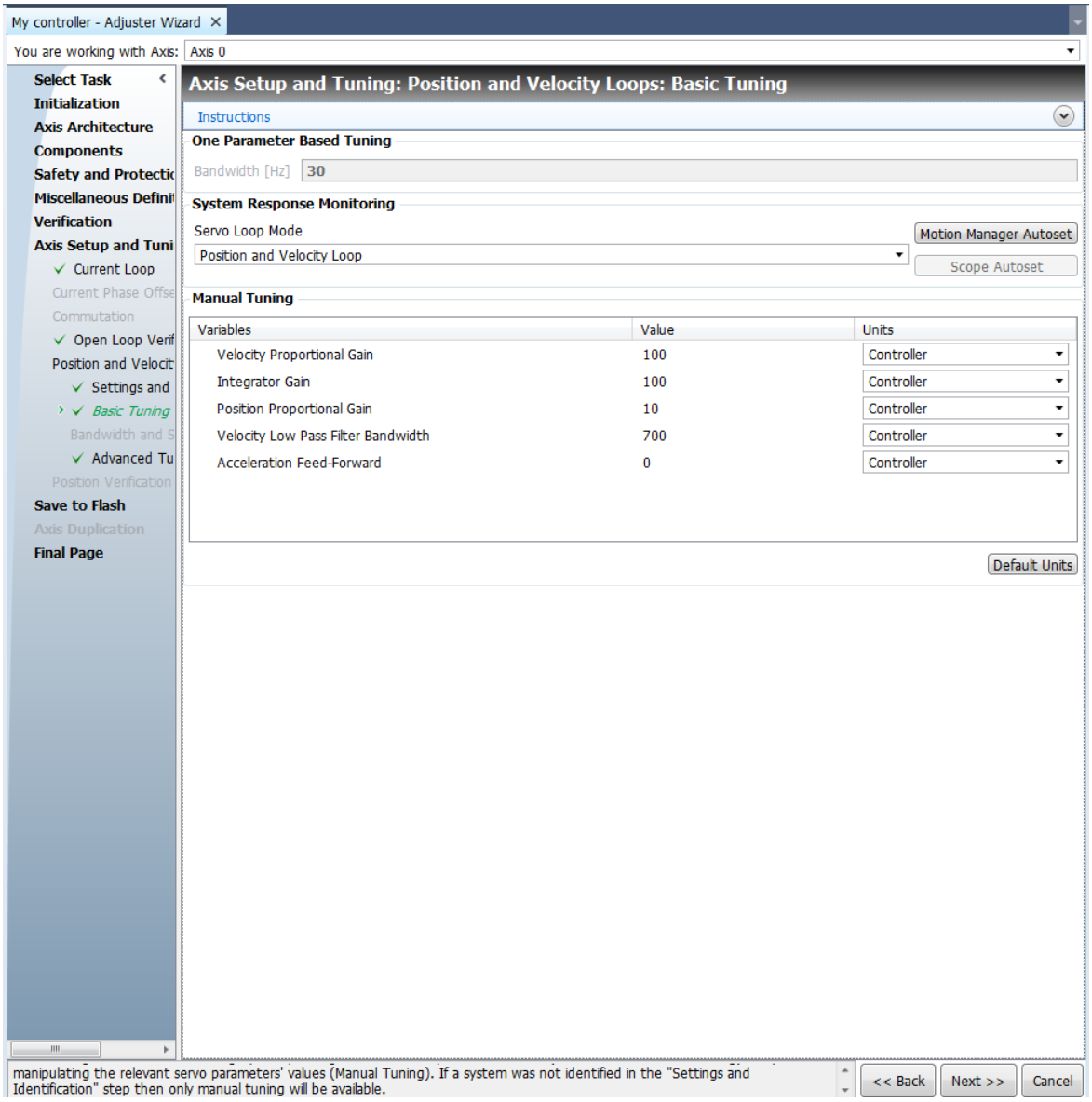



In case of an encoder fault, the motion limits will be zeroed by default. After the fault is cleared, entering new values may be required.

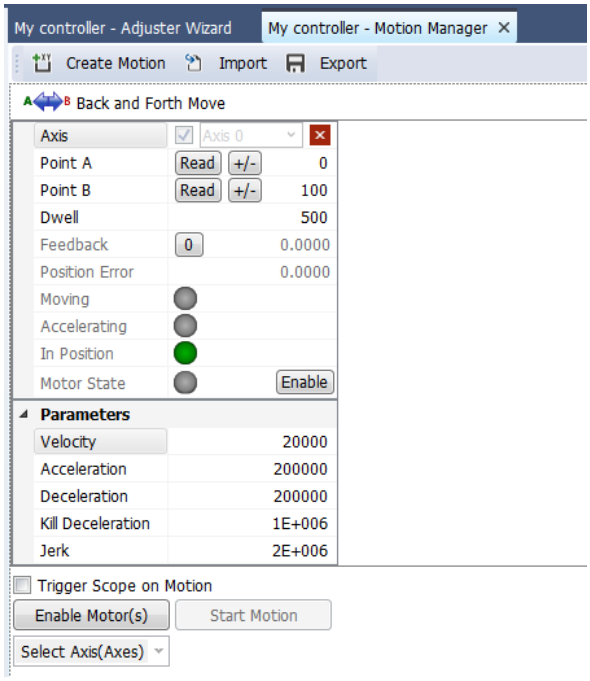
3. Under System Identification, select **FRF Analyzer Measurement**.
   > The values in the **FRF Analyzer Settings** are initially calculated according to the system characteristics.
   > If you select the **FRF Analyzer Autoset** option, the optimal excitation current is measured at the beginning of the system identification measurement (in this case the frequency range is disabled for changes).
4. Click **Start Identification**.
   The FRF Analyzer is displayed showing graphs of the movement.



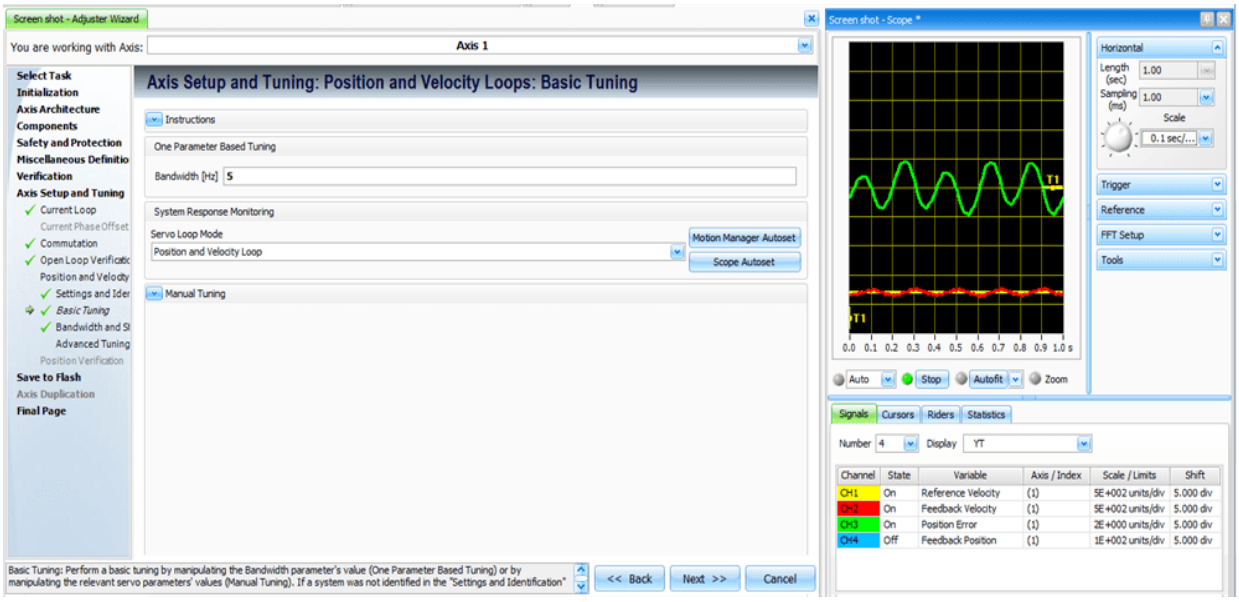

5. When the system identification is successful, click **Next** to continue to the next step. If the system identification is not successful, follow the recommendations under **System Identification Results**. If you were not able to identify the system, only Manual Tuning is available in the **Basic Tuning** window.



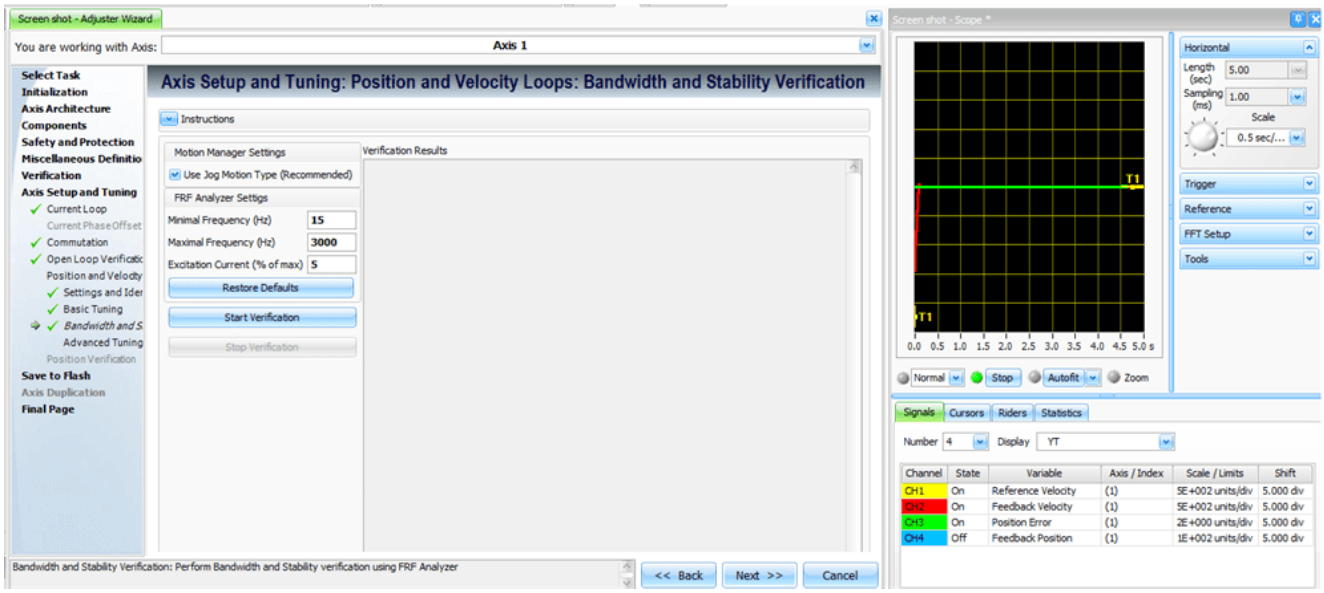

6. In the **Basic Tuning** window, click **Motion Manager Autoset**.
   The Motion Manager is displayed.



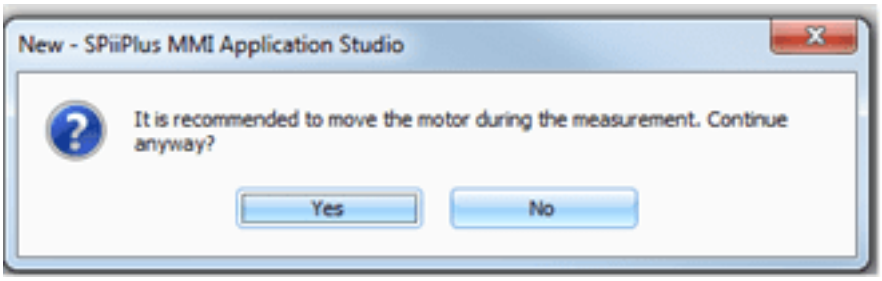

7. Click **Enable Motor(s)** and then **Start Motion**.
8. Click **Scope Autoset** and monitor the scope:
   Increase the **Bandwidth** parameter as long as position error improves and stability is maintained.
   Decrease the bandwidth to improve stability.
   The ideal Scope display is a square wave indicating that the Position and Velocity Loops are well adjusted.

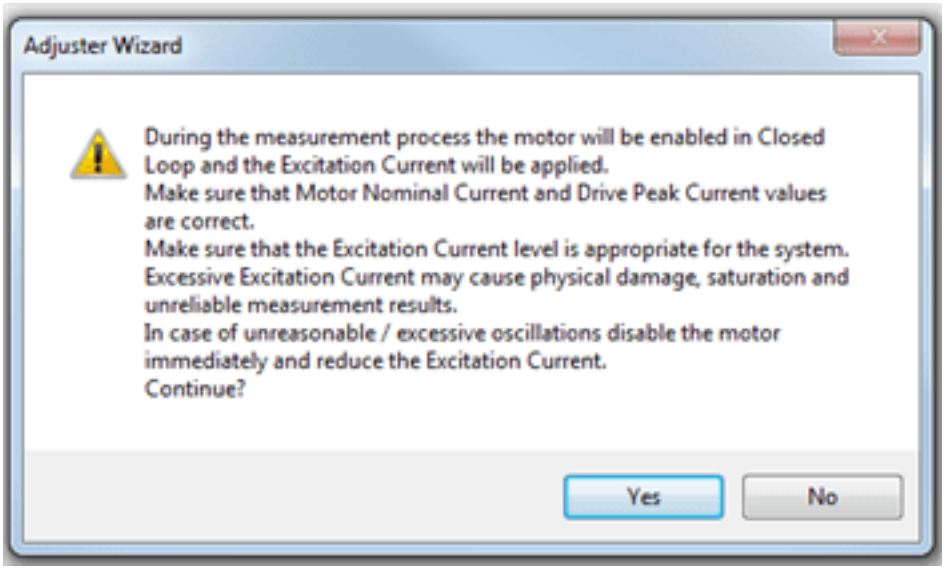



9. When you achieve a satisfactory bandwidth, click **Next** to continue to the next step.

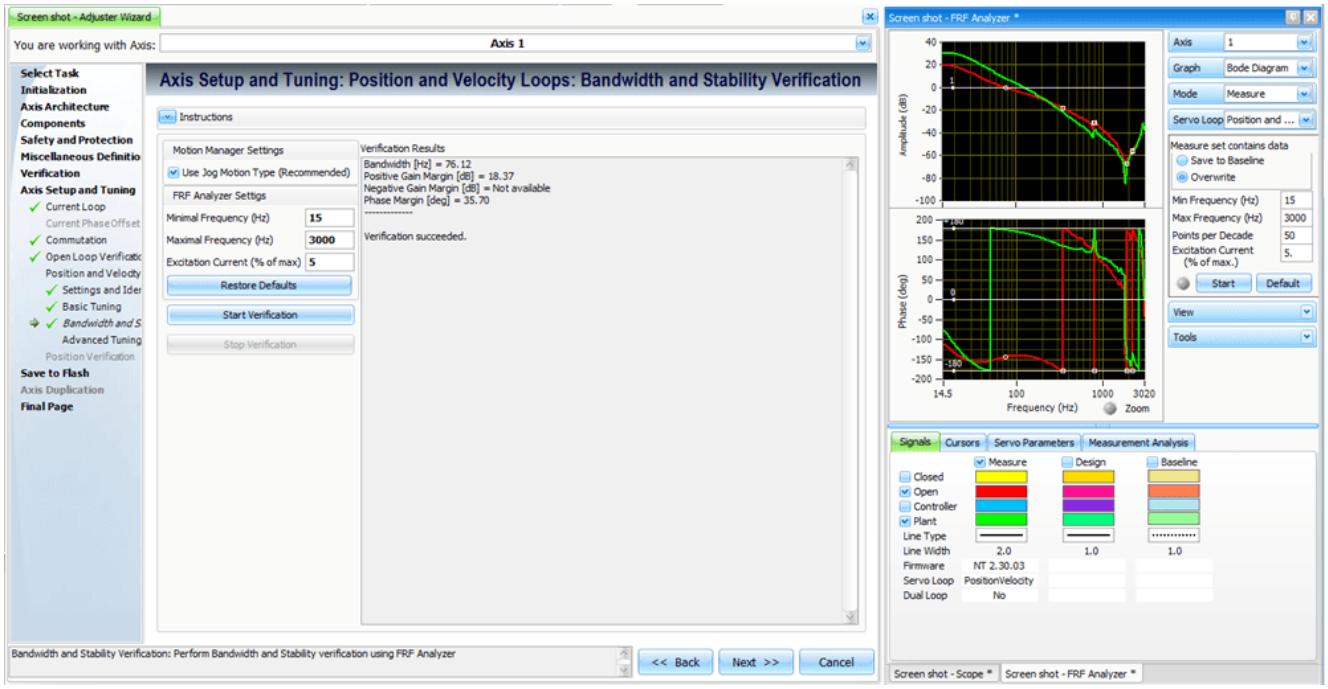



10. In the **Bandwidth and Stability Verification** window, click **Start Verification**.
11. Click **Yes** to the following warning messages.




The Bandwidth and stability verification is based on FRF closed loop measurement and comparing the measured stability gains to the predefined thresholds. The measured bandwidth is also compared to the required one.




> If the verification is unsuccessful, follow the recommendations under the **Verification Results** section.
> You may need to return to the **Basic Tuning** window and repeat the process described in **Step 6**.

> If the verification is successful, you can click **Next** to Advanced Tuning.
> Advanced tuning is optional and requires experience in Control field.

### Manual Tuning

Manual tuning involves tuning controller parameters in the **Basic Tuning** window. You can modify each parameter during a motion in both Velocity Loop mode and Position and Velocity Loop mode. Available parameters are:

> **Velocity loop Proportional gain (SLVKP)** - in both servo loop modes

> **Velocity loop Integrator gain (SLVKI)** - in both servo loop modes

> **Position loop Proportional gain (SLPKP)** - only in Position and Velocity Loop

> **Low pass filter bandwidth (SLVSOF)** - in both servo loop modes

> **Acceleration Feed-Forward (SLAFF)** - only in Position and Velocity Loop

**To perform manual tuning:**

1. In the **Basic Tuning** page, click **Motion Manager Autoset**.
   The Motion Manager is displayed.
2. Click **Enable Motor(s)** and then **Start Motion**.
3. Click **Scope Autoset** and monitor the scope:
   Change the relevant controller parameters values as long as position error improves and stability is maintained.

The ideal Scope display is a square wave indicating that the Position and Velocity Loop is well adjusted.

4. When you achieve a satisfactory bandwidth, click **Next** to continue to the next step.
5. In the **Bandwidth and Stability Verification** page, click **Start Verification**.
   The Bandwidth and stability verification is based on FRF closed loop measurement and comparing the measured stability gains to the predefined thresholds.
   > If the verification is unsuccessful, follow the recommendations under the **Verification Results** section.
   > You may need to return to the **Basic Tuning** page and repeat the process described in **Step 3**.
   > If the verification is successful, you can click **Next** to Advanced Tuning.
   > Advanced tuning is optional and requires experience in Control field.

**Advanced Tuning**

Use the Advanced controller parameters tuning for further improvements.

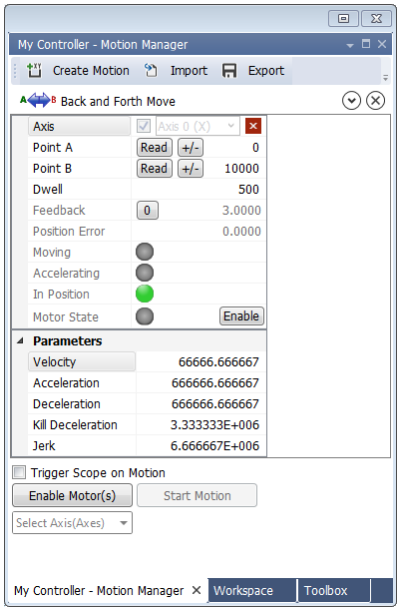
1. To tune the filters (Notch, BiQuad or Low-Pass) select **Filters**.
   If the verification measurement was successfully completed then the FRF component will be opened with already existing measurement.
   Otherwise, measure the FRF of the Position and Velocity loop; We recommend to move the motor slowly in constant velocity during the measurement (1 revolution/sec for rotary motor, or a few mm/sec for linear motor).
   By default, **Open Motion Manager** is selected for that purpose.
2. Click **Design** to open the design mode and tune the filters.
   When the Design FRF is satisfactory, click **Finish** and choose **yes** in the message asking to save the design parameters to the controller.
   Then, measure the FRF again to verify it is close to the design.
3. To tune other controller parameters, select **Algorithms**, then open the required servo variables group and click **Motion Manager & Scope** to open the Motion Manager and Scope.
4. In the Motion Manager, apply a required motion, monitor the Scope and modify the controller parameters as needed.
5. When you achieve satisfactory results, the tuning is finished. Click **Next** to continue to the next step to finish the adjustment.

*3.2.6.7.6 Position Verification*

The Position Verification window is relevant only for open loop modes for step motors and enables you to verify that the actual move corresponds to a commanded one, and make changes where necessary.


To tune position verification:

1. Click **Motion Manager Autoset**. The Motion Manager panel is displayed.




The **Motion Type** dropdown list gives you the option of defining the type of motion, the options are:

> One Direction Repeated Move

> Back and Forth Move (Default)

If you select One Direction Repeated Move, you specify how much movement by entering a value in **Move By**.

If you select Back and Forth Move, the position is measured between two points:

> Point A

> Point B

**Point A** is the start position and **Point B** is the end position. The motion is then measured from Point A to Point B.

You can set these points by manually moving the motor to Point A and clicking Read, and repeating this for Point B.

You can change the sign of the value by clicking +/-

The **Dwell** value is Dwell time between moves.

The Motion Parameters that you can manipulate are:

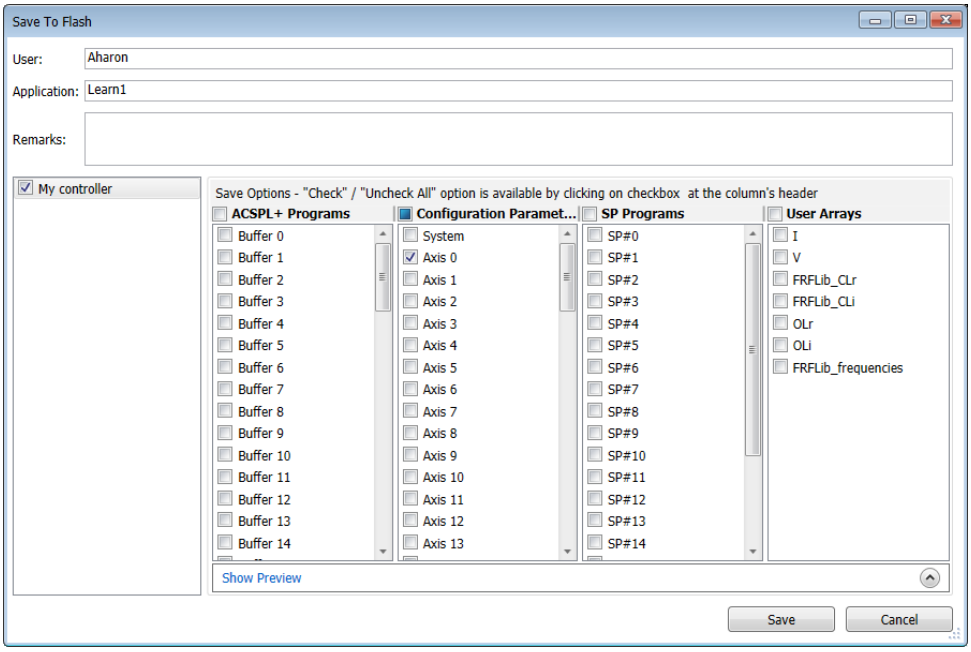
- **Velocity** (VEL)
- **Acceleration** (ACC)
- **Deceleration** (DEC)
- **Kill Deceleration** (KDEC)
- **Jerk** (JERK)

To start the measurement:

1. Click Enable Motor(s), this enables the motor of the axis you are working on.
2. Click Start Motion, this starts the motion.
   Visually check the motion. If there is a positional error, exit **Adjuster** and correct the value in your program, then run **Adjuster** again.
3. When you are done, click **Next** to go to the next window.

### *3.2.6.8 Save to Flash*

The Save to Flash step enables you to save all the **Adjuster** data to the controller's flash. When you enter this step, the Save to Flash prompt is displayed.




For saving to the controller's flash see Saving to Flash.

If you do not save the data to the controller's flash, all of the data will be lost.

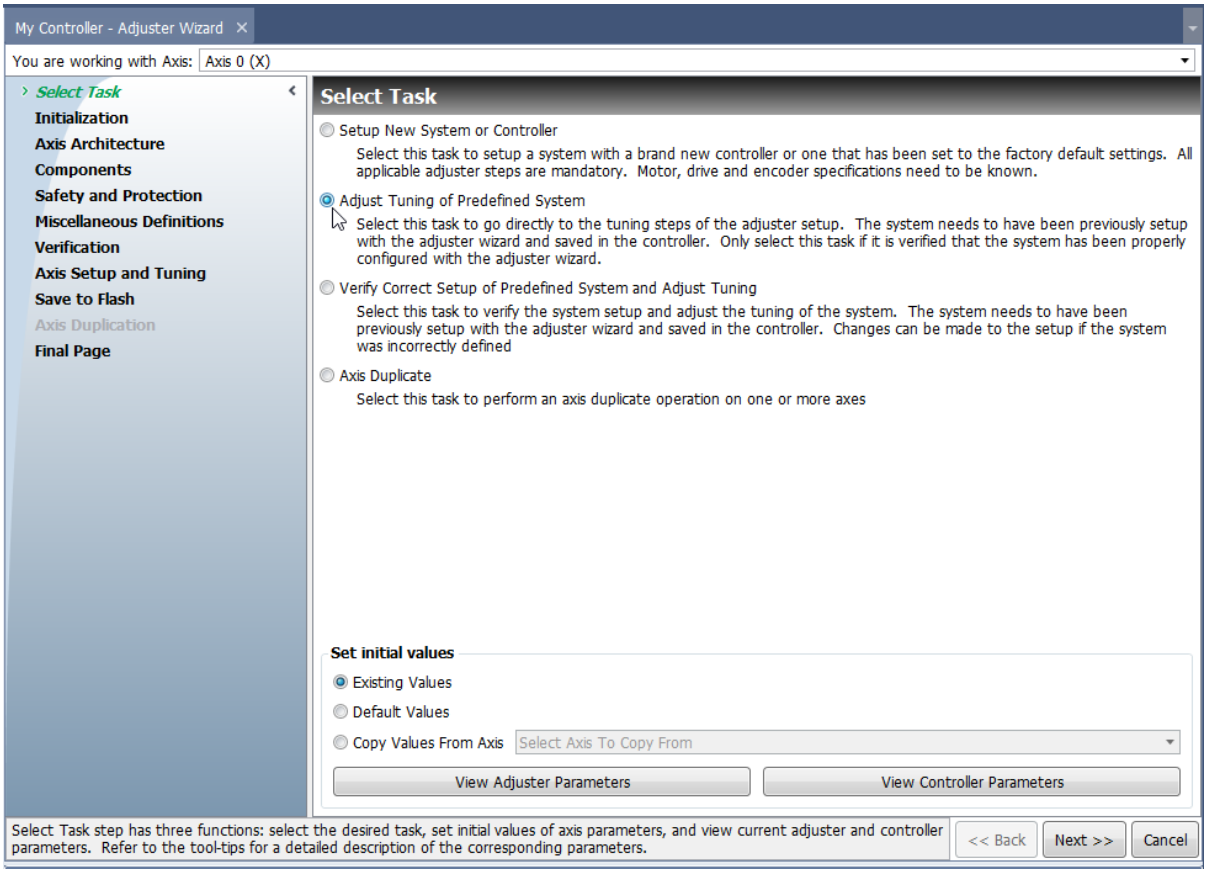
Once you have saved the data, the Final Page is displayed. This marks the completion of the **Adjuster**. Click **Finish** to exit.

### *3.2.7 Adjust Tuning of Predefined System*

The **Adjust Tuning of Predefined System** task of the Adjuster enables you to fine tune your system in order to optimize performance.

Although you can perform this task before performing Setup New System or Controller, it is not recommended.

1. In the Adjuster **Select Task** window, select **Adjust Tuning of Predefined System**.

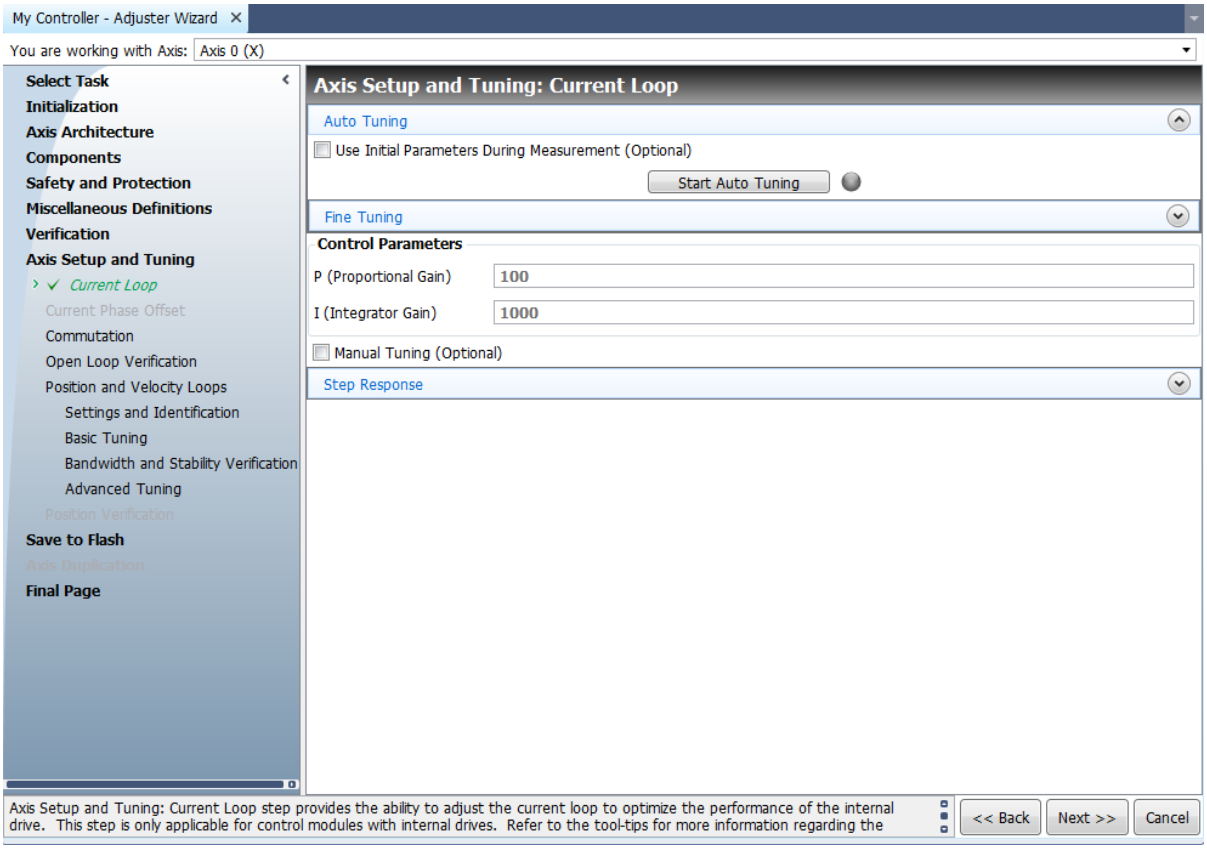



You have a choice of selecting:

> **Existing Values** (default), or

> Copy Values From Axis

2. Click **Next**, you jump to the **Axis Setup and Tuning** task page:

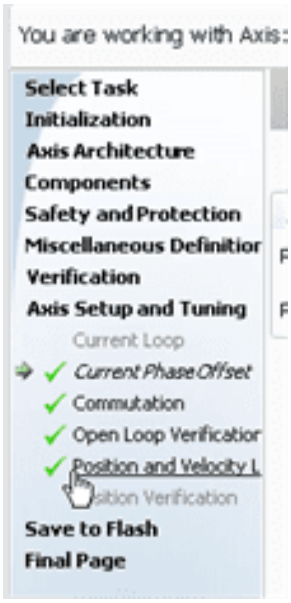



The Axis Setup and Tuning step enables you to fine tune your system. There are a total of six possible options:

1. Current Loop
2. Current Phase Offset
3. Commutation
4. Open Loop Verification
5. Position and Velocity Loops
6. Position Verification

The options that are available to you depend on the definitions that were entered in Components during the performance of Setup New System or Controller

You can skip steps either by clicking **Next**, or by clicking the step in the Task Steps list that you want to activate:



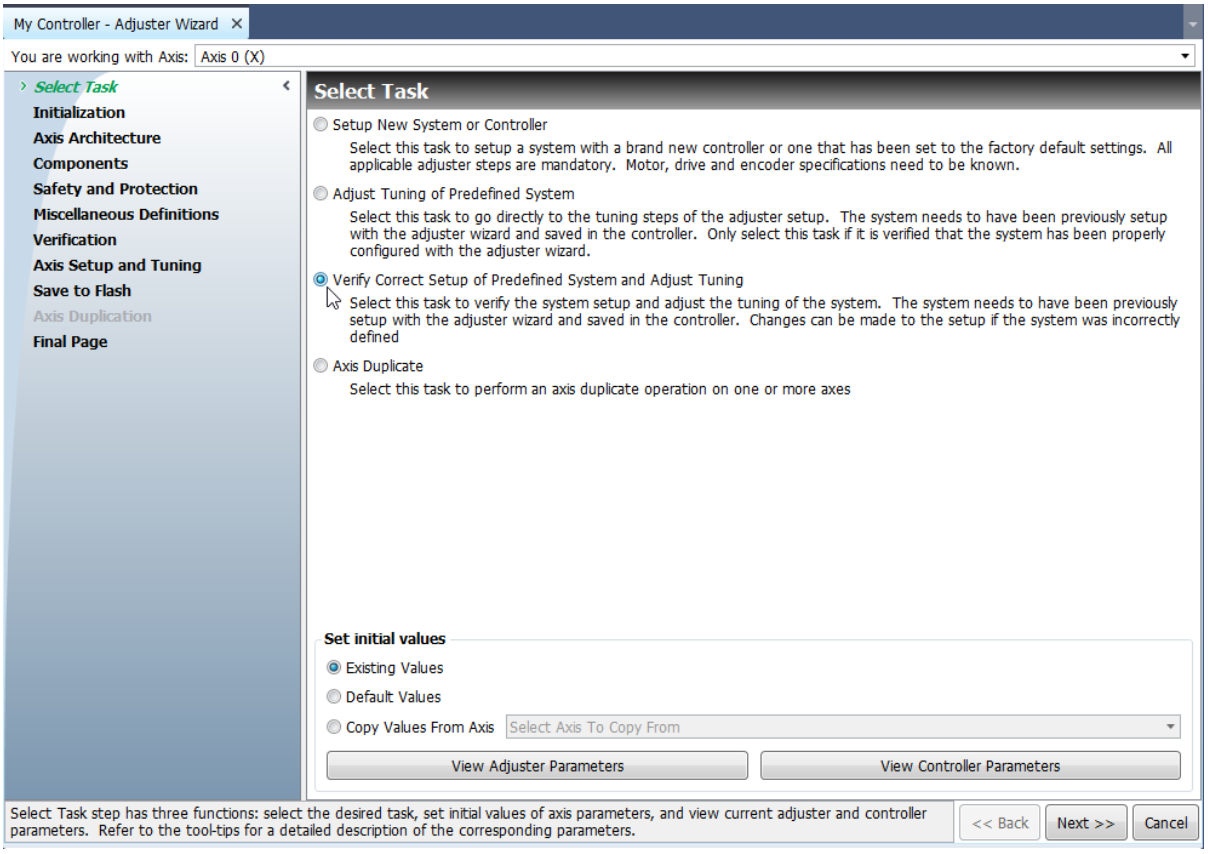

In the same manner you can return to previous task steps by clicking **Back** or clicking the required step in the Task Steps list.

7. Once you have tuned your values to your satisfaction, you move to Save to Flash.
8. After saving the values to the controller's flash, click **Finish** to exit the **Adjuster Wizard**.

### *3.2.8 Verify Correct Setup of Predefined System and Adjuster Tuning*

You use the **Verify Correct Setup of Predefined System and Adjuster Tuning** task of the Adjuster to verify the setup recorded in the performance of the Setup New System or Controller task, and, where needed, fine tune your system in order to optimize performance.

1. From the Adjuster **Select Task** window select **Verify Correct Setup of Predefined System and Adjuster Tuning**.

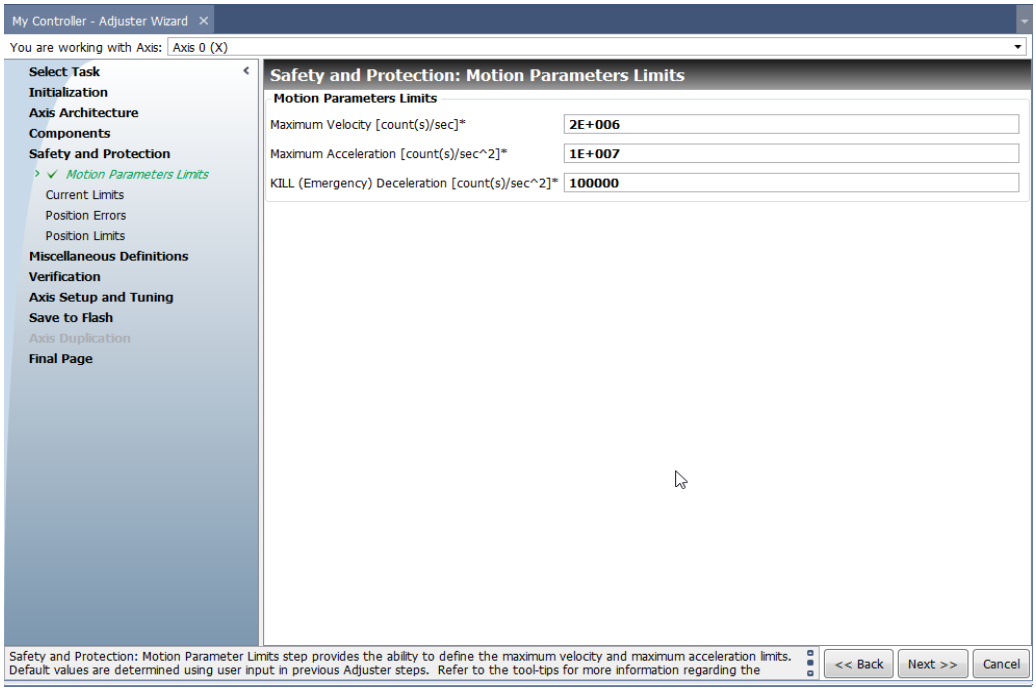



You have a choice of selecting:

> **Existing Values** (default), or

> Copy Values From Axis

2. Click **Next**, you jump to the **Safety and Protection** task window:

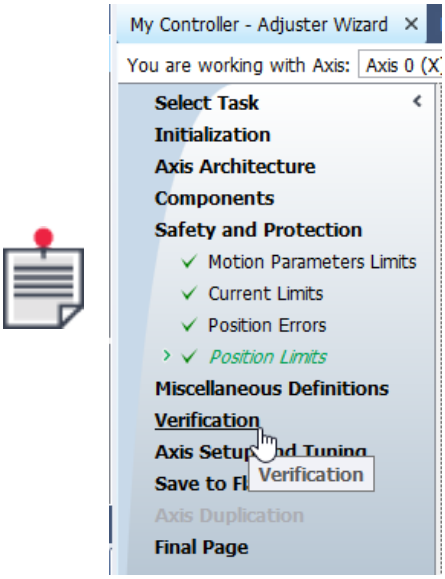



In this task you verify that the settings are correct for:

- Motion Parameters Limits
- Current Limits
- Position Errors
- Position Limits

In the event that you find an incompatibility, this task provides you with the ability to make changes.

You can skip steps either by clicking **Next**, or by clicking the step in the Task Steps list that you want to activate:




In the same manner you can return to previous task steps by clicking **Back**, or clicking the required step in the Task Steps list.

3. The next task is Miscellaneous Definitions. In this task you check the values for:
   - Motion Completion
   - Enable/Disable/Brake
   - Dynamic Brake
   - Home Switch
4. The next task is Verification. In this task you check the values for:
   - Feedback
   - Switches
   - Stop, Alarm and Brake
5. The next task is Axis Setup and Tuning. In this task you check the values for:
   - Current Loop
   - Current Phase Offset
   - Commutation
   - Open Loop Verification
   - Position and Velocity Loops
   - Position Verification

The options that are available to you depend on the definitions that were entered in Components when running Setup New System or Controller.

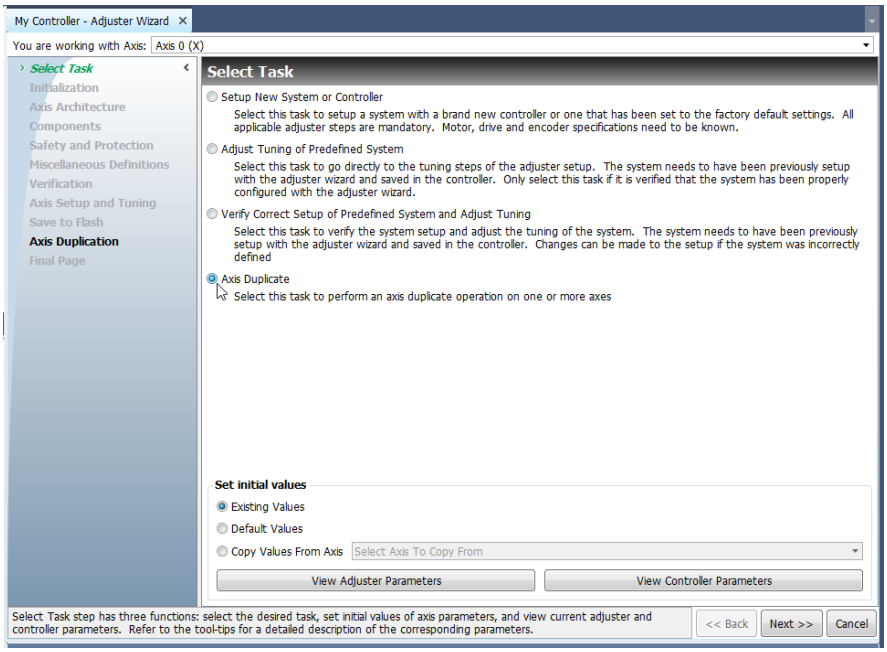
6. Once you have checked the values, and tuned them, if needed, you move to Save to Flash.
7. After saving the values to the controller's flash, click **Finish** to exit the **Adjuster Wizard**.

### *3.2.9 Axis Duplication*

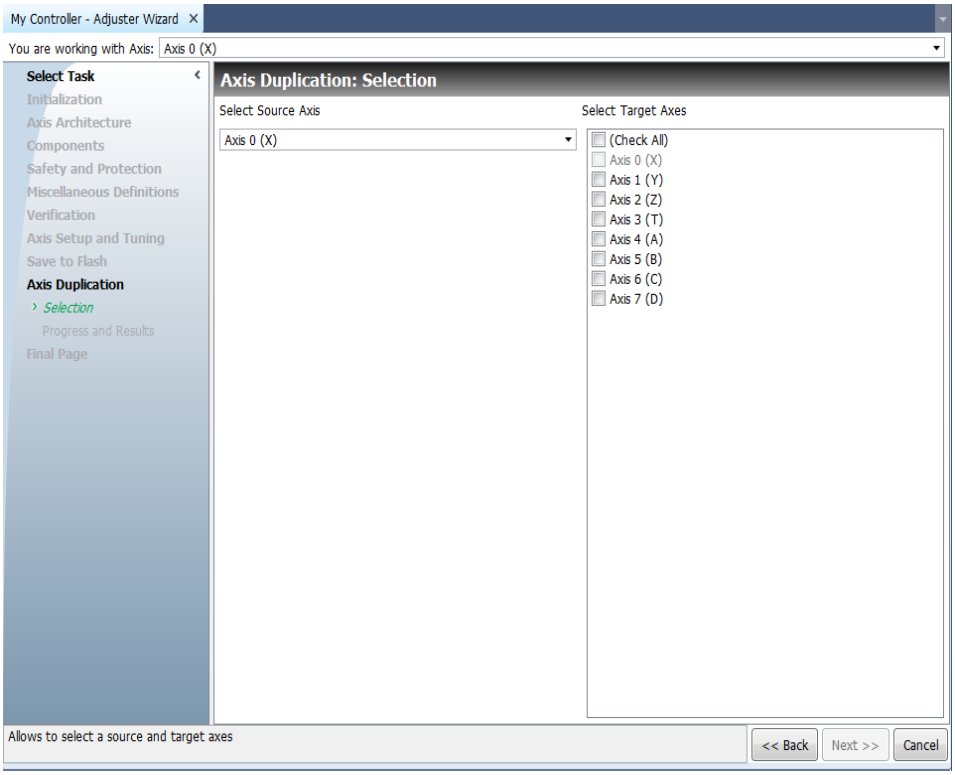
This option allows you to copy adjuster and controller parameters from one axis to another axis, typically the one you currently adjusting.

**To duplicate an axis:**

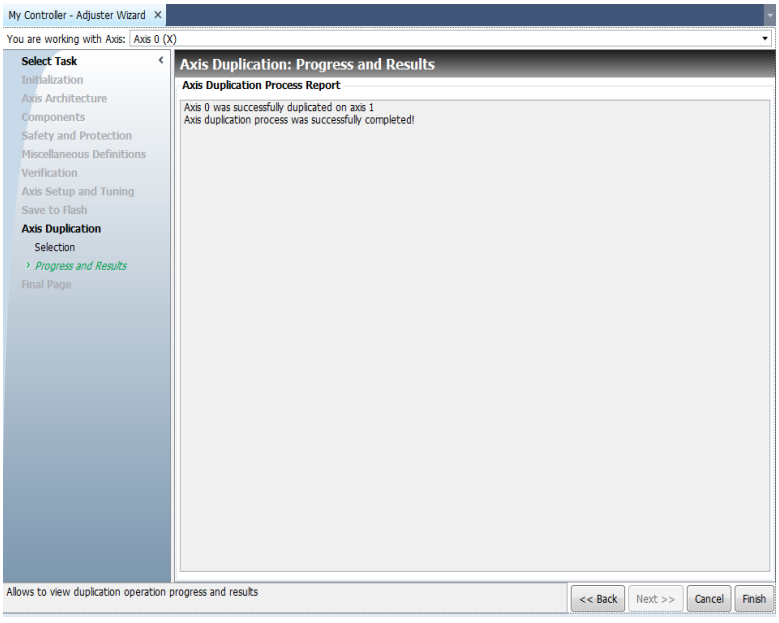
1. Select **Axis Duplication** in the Select Task step and click **Next**.




The **Axis Duplication: Selection** window substep appears.



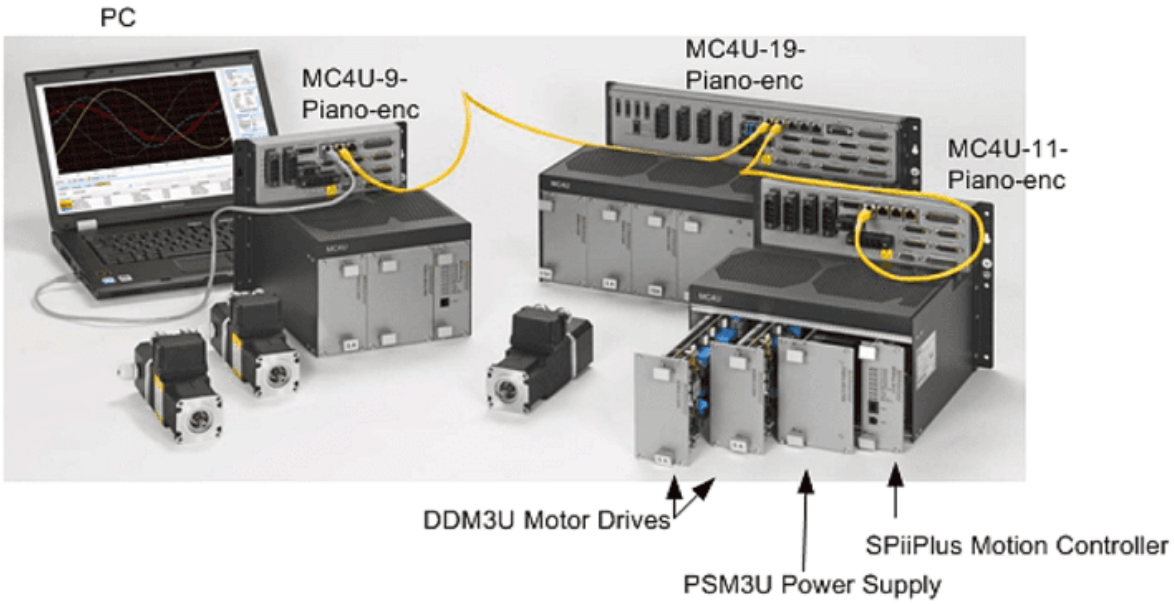

2. Select the Source Axis. By default the source axis is the working axis.
3. Select the Target Axis and click **Next**.
   The Wizard prompts you to Save to Flash the source axis (the Wizard copies only data that is saved to the controller flash). When the duplication is completed, the Wizard presents the results in the Progress and Results substep.




4. Click **Next** to complete the Adjuster Wizard and then click **Finish** to close the wizard.

## 3.3 System Setup

ACS Motion Control's NT technology allows several units to be chained together using EtherCAT. **System Setup** enables you to configure a single or multiple Unit Control Module along with stand-alone NT modules and non-ACS units as a network, as shown in the following figure:




Use **System Setup** to:

- Define the system configuration
- Update the system configuration

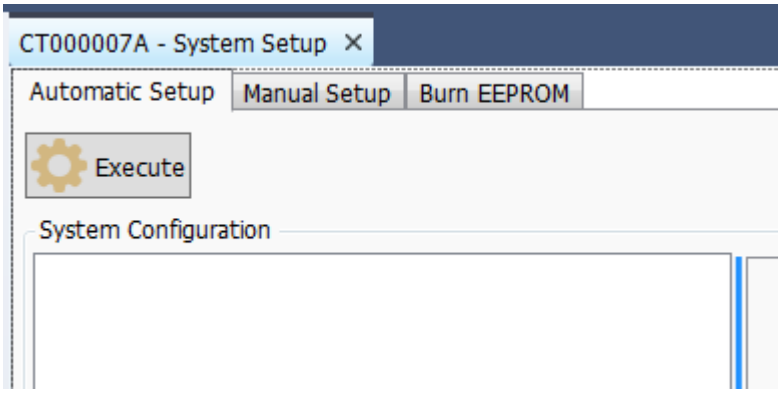
You can setup the system either automatically or manually.

You should run the automatic system setup on initialization and whenever the system is changed.

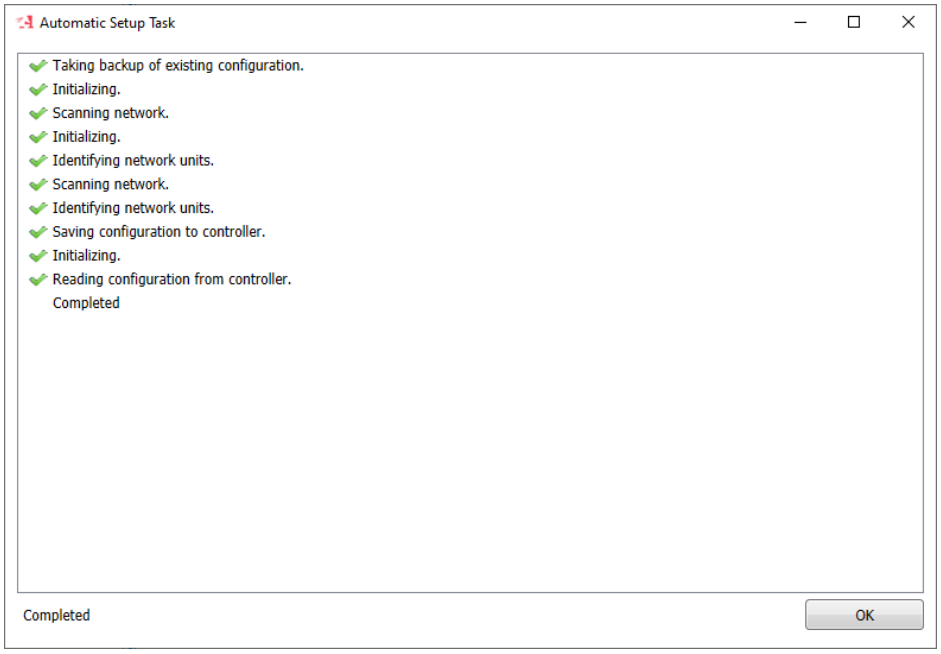
### 3.3.1 Automatic System Setup

To activate **System Setup**:

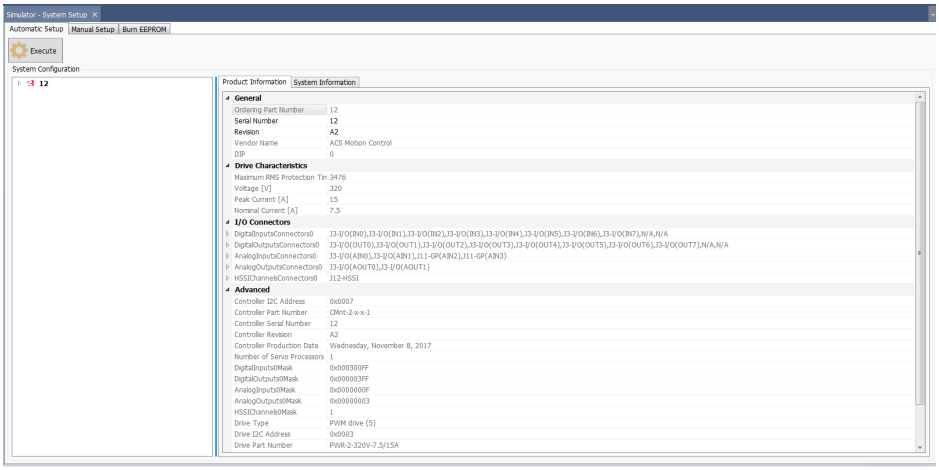
1. In the Workspace right-click the controller.
2. Select **Add component** > **Setup**> **System Setup**.
   The System Configuration window opens.
3. In the Automatic Setup tab, press **Execute**.




4. Approve the message about the controller being rebooted.
   The system is setup automatically. This may take several minutes to complete.
5. When the setup is complete, click **OK**.




The screen shown shows representative output of the automatic system configuration process:



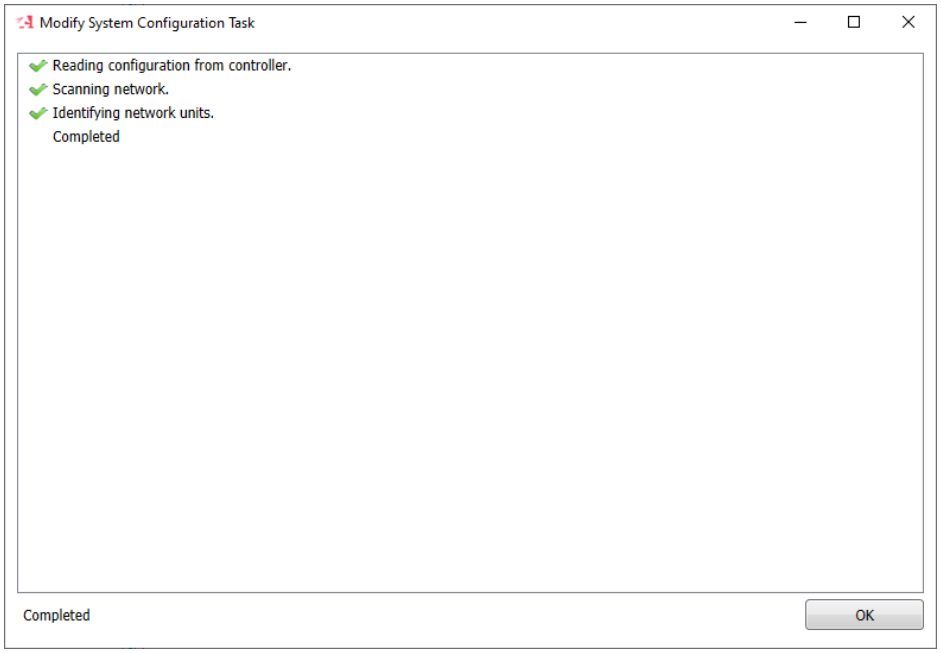

## 3.3.2 Manual System Setup

The following tasks can be done using manual setup:

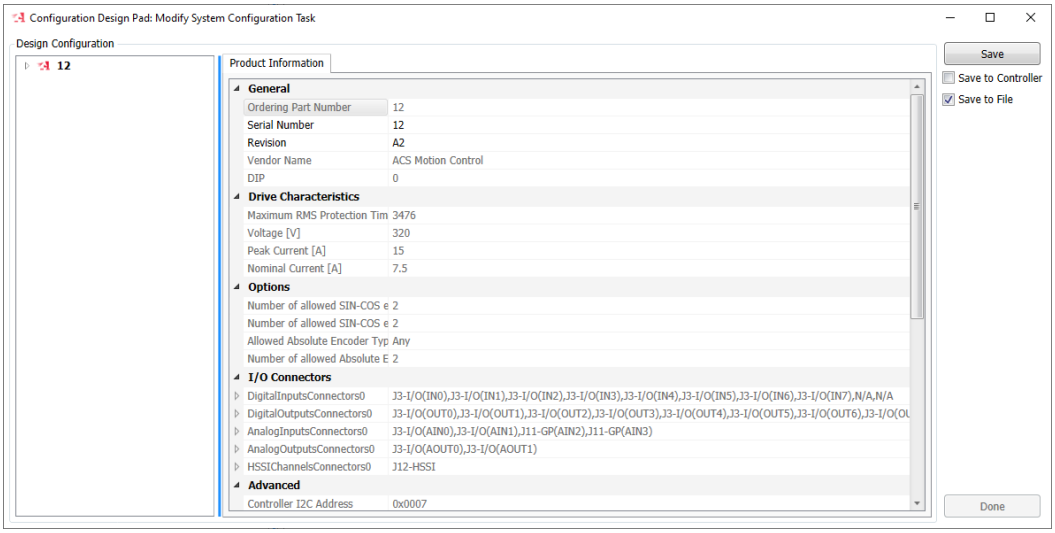
- Modifying a system configuration
- Modifying a configuration from a database
- Loading a configuration from a database

### 3.3.2.1 Modifying a system configuration

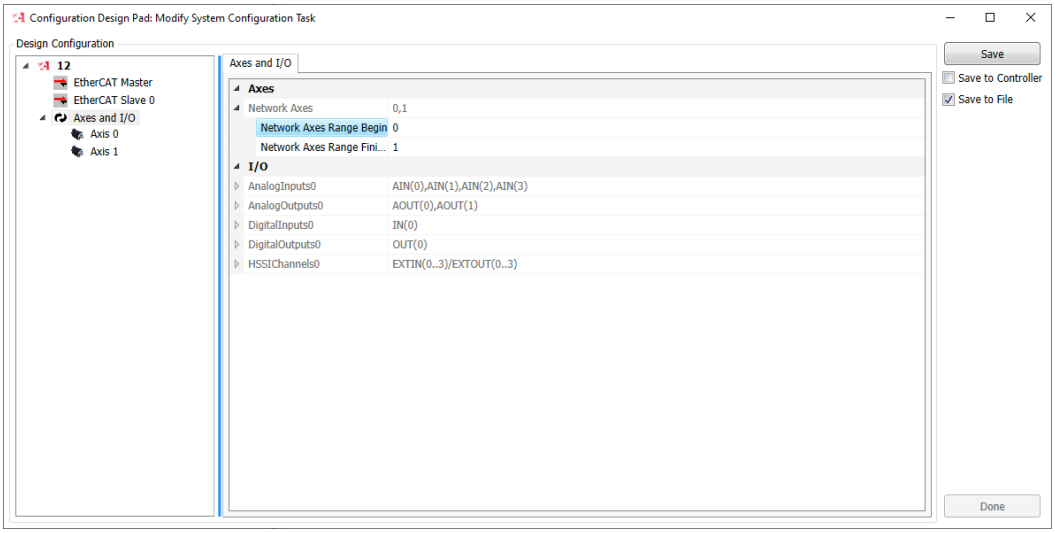
1. In the Workspace, right-click the controller.
2. Select **Add component** > **Setup**> **System Setup**.
3. In the System Configuration window, click **Manual Setup**.
4. Click **Modify System Configuration**. The system will load the configuration and may take several minutes.



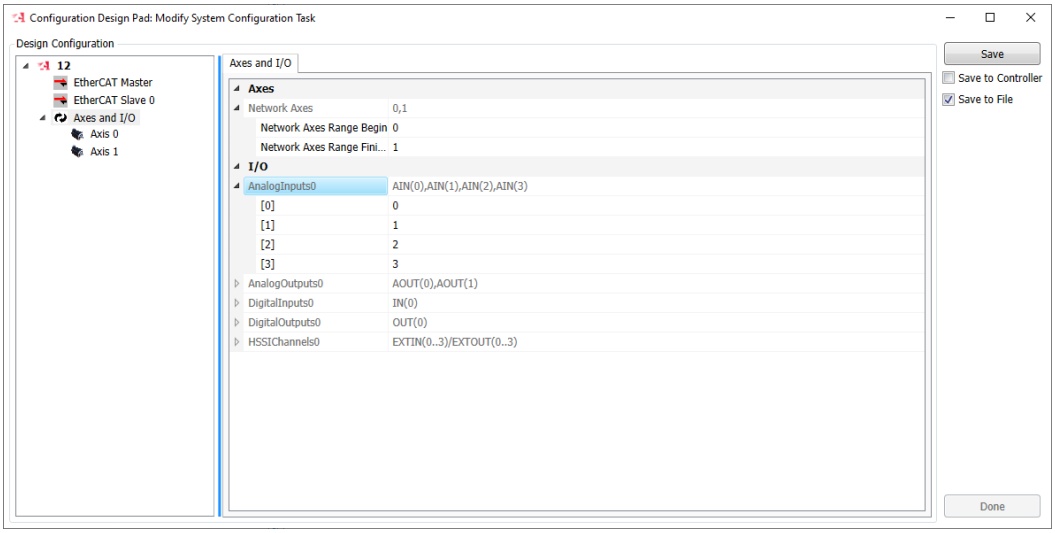

5. When the **Configuration Design Pad** window appears, you can modify settings such as the axes and I/O allocations:



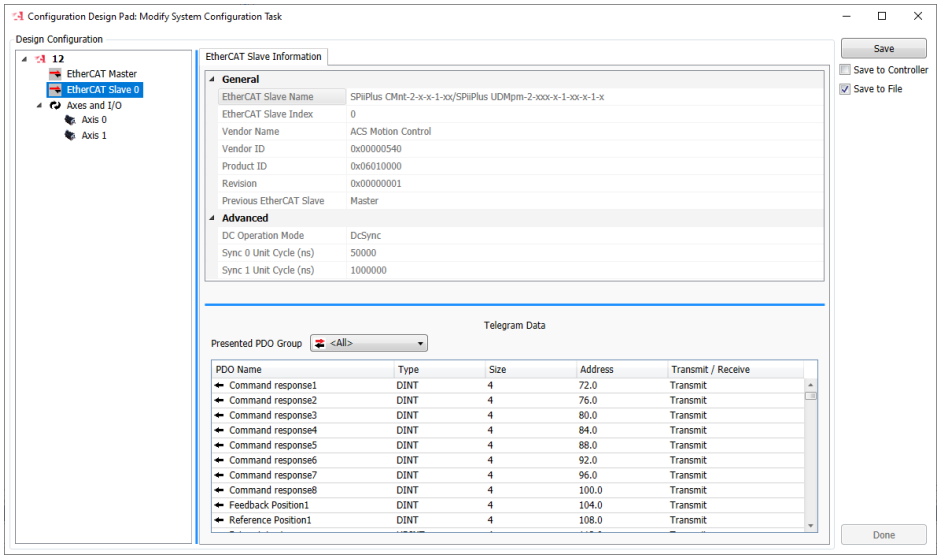

6. Modify axes allocation in the following window:



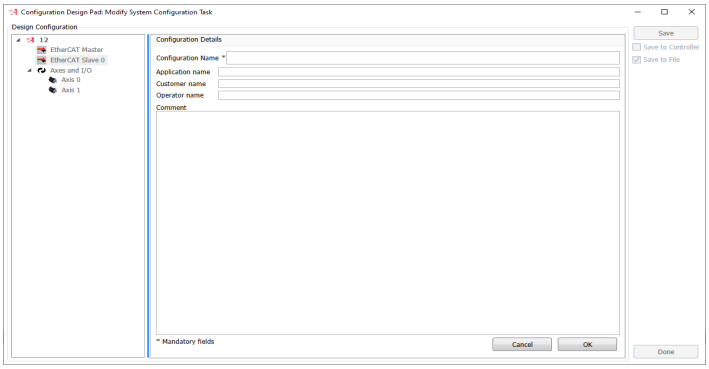

7. Modify I/O allocation in the following window:




8. Modify PDO configuration in the following window:




9. To save the configuration to the controller or to the database, select the required option under **Save** and click **Save**.
10. In the **Configuration Details** window, enter a name for the configuration and optionally enter additional information.

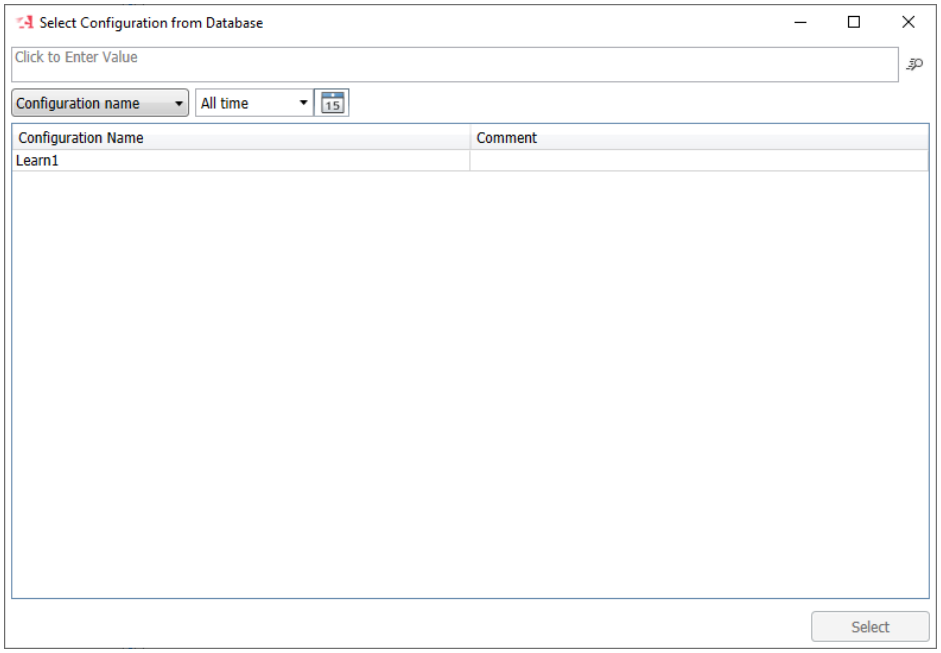



11. Click **OK**.

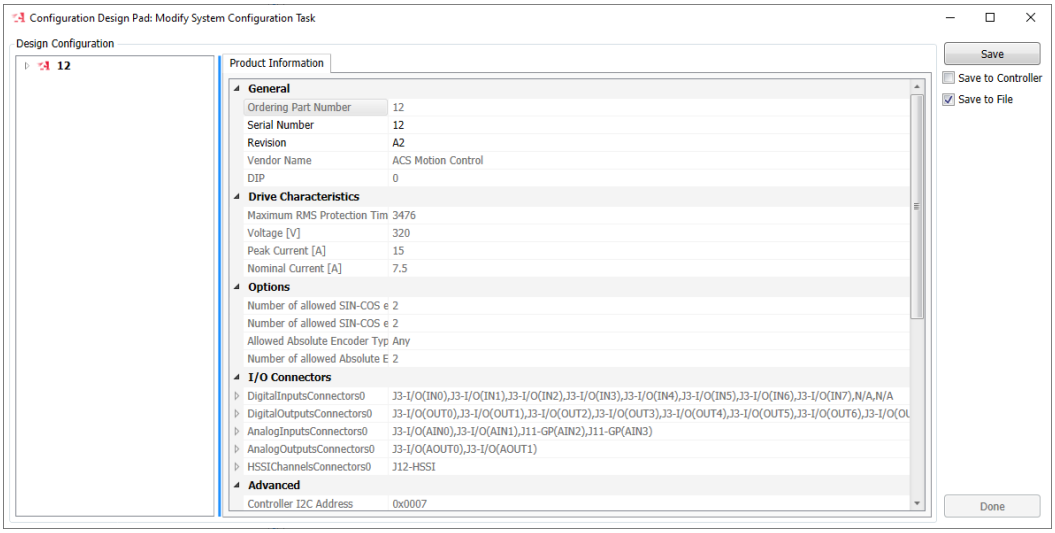
### *3.3.2.2 Modifying a Configuration from the Database*

A configuration that has been saved to a database can be edited.

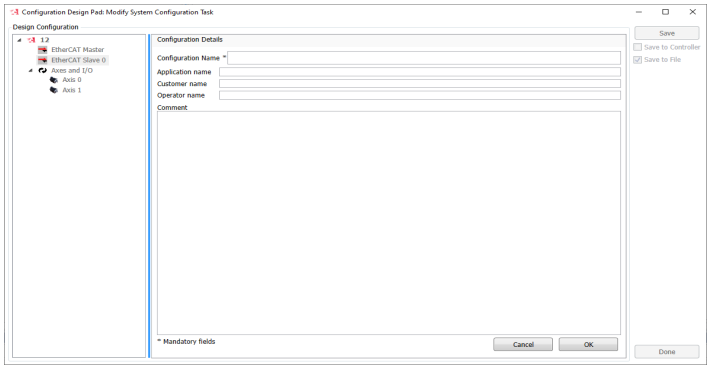
1. In the Workspace right-click the controller.
2. Select **Add component** > **Setup**> **System Setup**. The System Configuration window opens.
3. Press the **Modify Configuration from Database** button. A window displaying the configurations saved in the database appears.




4. Select the saved configuration to use.
5. Edit the configuration as required.
6. To save the configuration to the controller or to the database, select the required option under **SAVE** and click **Save**.




7. In the **CONFIGURATION DETAILS** window, enter a name for the configuration and optionally enter additional information.

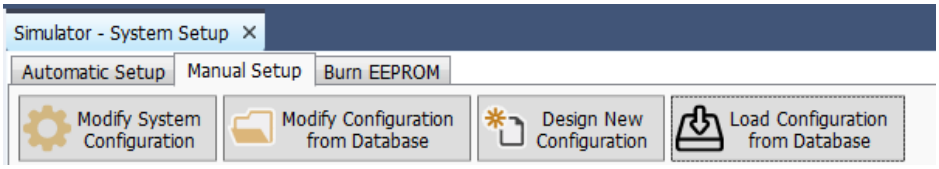


8. Click **OK**.

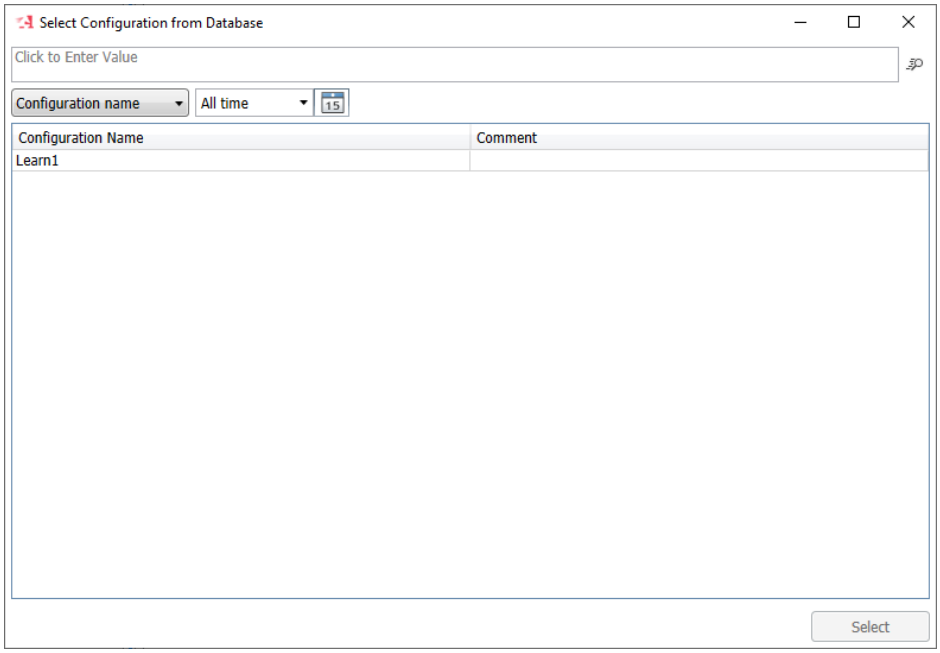
### *3.3.2.3 Loading a Configuration from the Data Base*

A configuration that has been saved to a database can be loaded and edited.

1. In the Workspace right-click the controller.
2. Select **Add component** > **Setup**> **System Setup**. The System Configuration window opens.

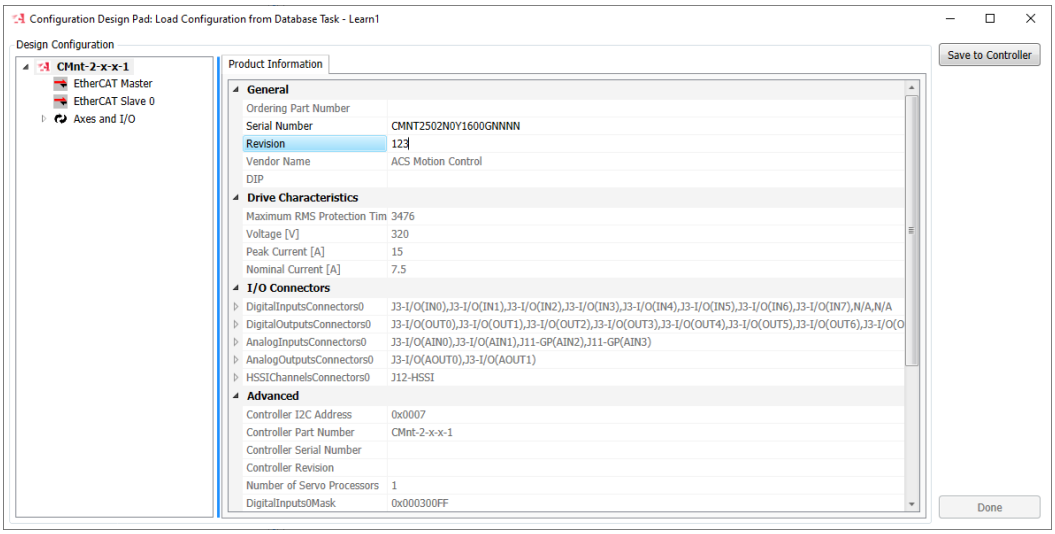



3. Press the **Load Configuration from Database** button.
4. A message stating that the controller will be rebooted will appear, click **Yes**.
5. A window displaying the configurations saved in the database appears.

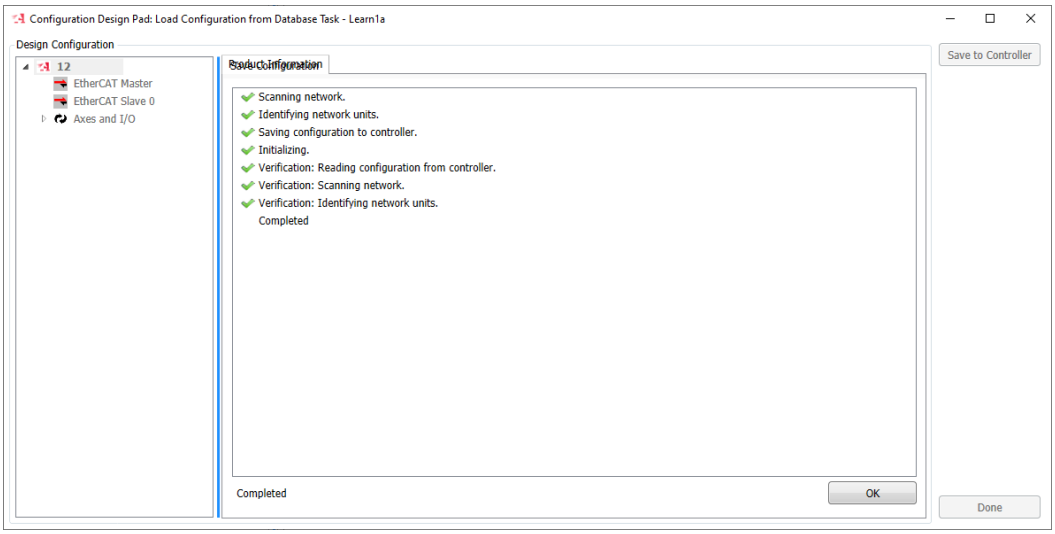



6. Press **Select** to load the desired configuration.
7. Enter the **Serial number** and **Revision** for the relevant products.

8. Click **Save to Controller**.




9. When the configuration has been saved to the controller, click **OK** to close the progress report window.




10. Click on **Done** to close the configuration design pad.

### *3.3.2.4 MC4U Configuration Codes*

The following are the standard codes used for indicating the configuration of the MC4U unit. The general format is:

**MC4U [MMM] AA-B#-CC-DD-[E]-[F]-[G]-[H]**

Where:

**MMM** – Motherboard type

Depending on the **MC4U** model, motherboard codes can be:

> Z - MB5U-Z Motherboard

> ZZ - MB5U-ZZ Motherboard

- 2 - MB5U-2 Motherboard
- 90A - MB5U-2-90A Motherboard
- 4 - MB5U-4 Motherboard
- YYYY – MB5U-YYYY Motherboard
- ZZW – MB5U-ZZW Motherboard
- ZZZ – MB5U-ZZZ Motherboard
- PS - MB5U-CON-PS Motherboard
- PS2 - MB5U-CON-PS2 Motherboard

**AA** – MC4U Configuration Type, which can be:

- CS – Customization Solution
- CM – Control Module

**B#** – SPiiPlus Controller Type and number of axes, the controller can be:

- P – SPiiPlus 3U-HP
- T – SPiiPlus 3U-LT
- NTP - SPiiPlus NT-HP
- DCP - SPiiPlus DC-HP
- NTT - SPiiPlus NT-LT
- DCT - SPiiPlus DC-LT

  The number (#) of axes can be:

  - 4 – Four axes
  - 6 – Six axes
  - 8 – Eight axes
  - 4E – Four axes and eight feedback interfaces

**CC** – Power Supply Type (there can be more than one), which can be:

- 00 – None
- A - PSM3U-28V-0.5kW
- H – PSM3U-320V-8kW
- L – PSM3U-48V-0.7kW
- V – PSM3U-560V-8kW
- HH – PSMU3U-320V-11Kw
- LL – PSMU3U-48V-1.4Kw
- HL – PSMU3U-320/48V-0.7/8Kw
- HM – PSMU3U-320/48V-0.7/8kW
- VL – PSMU3U-560/48V-0.7/8kW

**DD** – Drive Type (there can be more than one), which can be:

- A – DDM3U-2-60V-2A
- B – DDM3U-1-320V-5A
- C – DDM3U-1-320V-10A
- D – DDM3U-1-320V-20A
- E – DDM3U-1-560V-2A
- F – DDM3U-1-560V-5A
- G – DDM3U-1-560V-10A
- H – DDM3U-1-560V-15A
- K - DDM3U-1-320V-45
- M - DDM3U-1-320V-30
- S – DDM3U-2-680V-2A
- T – DDM3U-2-680V-5A
- U – DDM3U-2-680V-10A
- V – DDM3U-2-680V-15A
- W – DDM3U-4-60V-2A
- Wa - DDM3U-4-60V-4A
- W1 - DDM3U-4-320-1A
- W2 - DDM3U-4-320-2A
- W3 - DDM3U-4-320-3A
- X – DDM3U-2-320V-5A
- Y – DDM3U-2-320V-10A
- Z – DDM3U-2-320V-20A
- Pa – LDM3U-55V-8A (Analog)
- Pd – LDM3U-55V-8A (Digital)
- Ps – LDM3U-55V-8A (Space Vector Modulation)
- Ra – LDM3U-55V-16A (Analog)
- Rd – LDM3U-55V-16A (Digital)
- Rs – LDM3U-55V-16A (Space Vector Modulation)

**E** – Total number of Sin-Cos multipliers (optional), which can be:

- 1
- 2
- 3
- 4
- 5
- 6

- 7
- 8

**F** – Number of 2.5 MHz Sin-Cos multipliers (optional), which can be:

- 1
- 2
- 3
- 4

When included, **E** must also be included as **F** represents the number of 2.5 MHz Sin-Cos multipliers out of the total number of Sin-Cos multipliers, for example, 63 which means there is a total of 6 Sin-Cos multipliers, of which 3 are 2.5 MHz Sin-Cos multipliers.

**F** cannot be greater than **E**.

**G** - Input Shaping (optional), its code is:

- I - Convolve Input Shaping® algorithm enabled

**H** - PLC and axis and I/O extension network (optional), its code is:

- C - PLC and axis extension enabled

The code string consists of a maximum of 15 characters, beginning with: MC4U. In the event that the string exceeds 15 characters, the leading MC4U can be shortened or omitted completely.

### *3.3.3 Optional Groups Setup*

The System Setup component now includes the capability to define optional groups to support flexible configuration. Flexible configuration provides the ability to work with a subset of an initial configured system without requiring reconfiguration of the system. Using flexible configuration enables an optional group of EtherCAT slaves to be disconnected from the system without a need to reconfiguration, for details see *AN Flexible Configuration Application Note*.

The following operations on optional groups are now supported in System Setup:

- Define optional group
- Discard optional group
- Discard all optional groups
- View optional groups

### *3.3.3.1 Head of optional group*

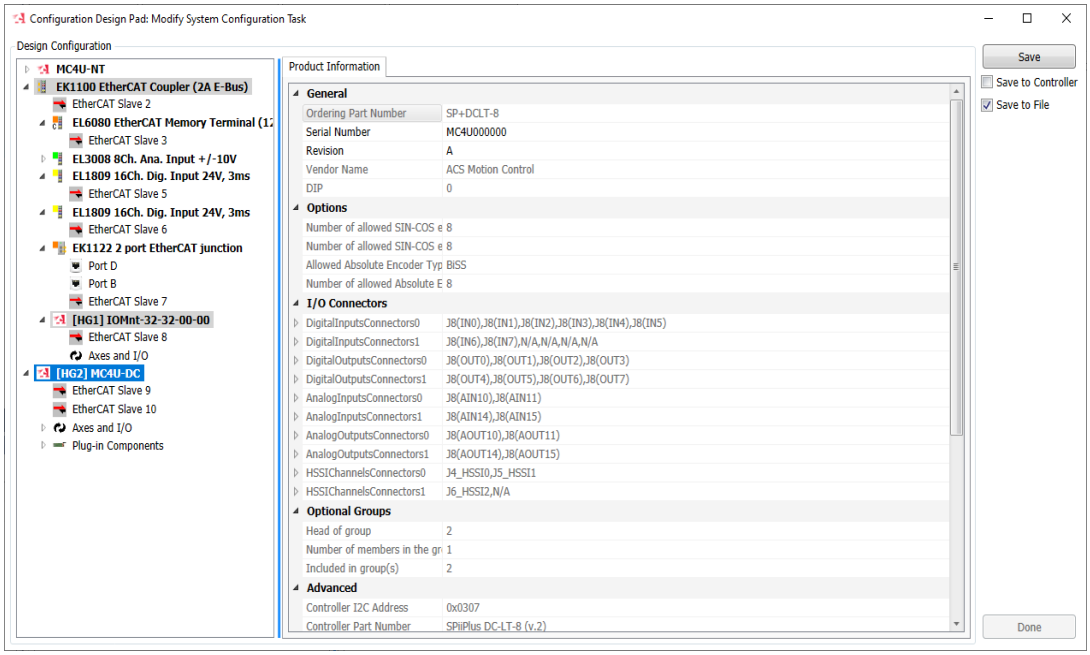
#### Product Name Prefix

The presentation name of the product which is defined as head of optional group in the tree will start with prefix "[HG<*group ID*>]" ("HG" = Head of Group).

#### Properties

On the product property panel of each head of optional group, the following information is presented:

| Optional group section | Values |
|---|---|
| Head of group [R/O] | *<Group ID>* |
| Number of members in the group [R/O] | *<[N]>* |
| Included in group(s) [R/O] | *<i, j, k, … > where i, j, k are including groups' IDs, i ϵ j ϵ k* |




### *3.3.3.2 Member of optional group*

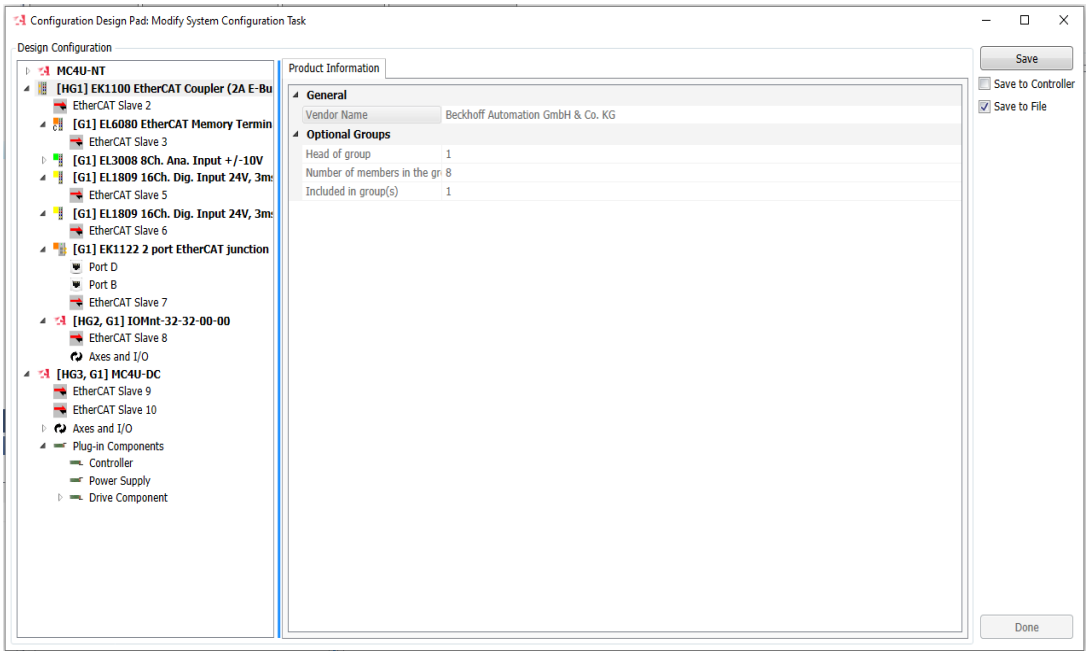
#### Product Name Prefix

The presentation name of the product which is defined as a member of optional group in the tree will start with [G<*group ID*>] prefix ("G" = Group). If this product is also a head of nested optional group then the member of group prefix will be preceded by head of group prefix: [HG<*group i ID*> G<*group j ID*>] .

**Properties**

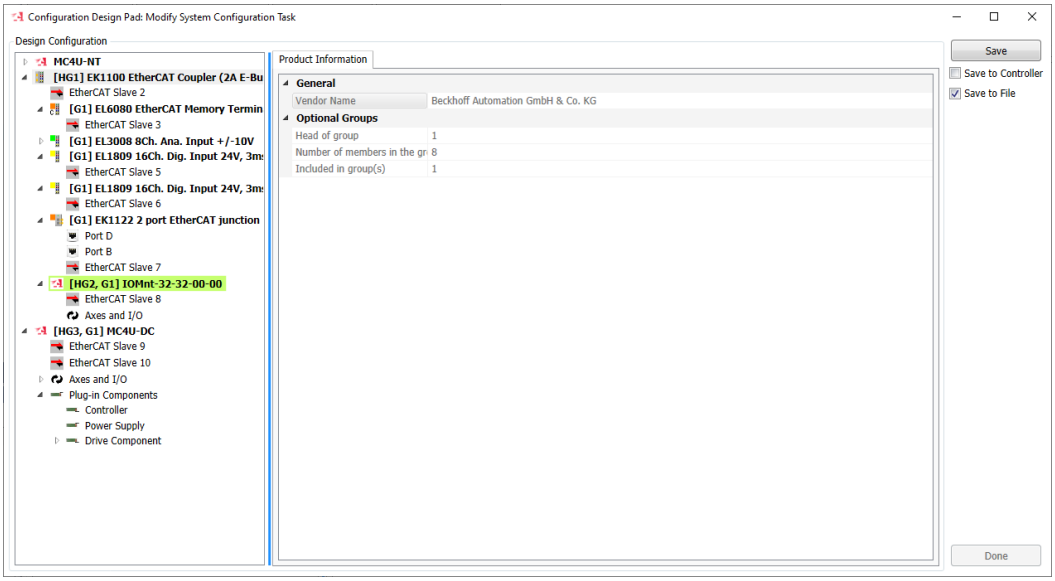
On the product property panel of each member of optional group the following information is presented:

| Optional group section | Values |
|---|---|
| Included in group(s) [R/O] | *<i, j, k, … > where i, j, k are including groups' IDs, i ϵ j ϵ k* |




### *3.3.3.3 Highlighting optional group*

To observe the whole optional group with all its members user can highlight the group by selecting the related option from right click menu of head of group product.

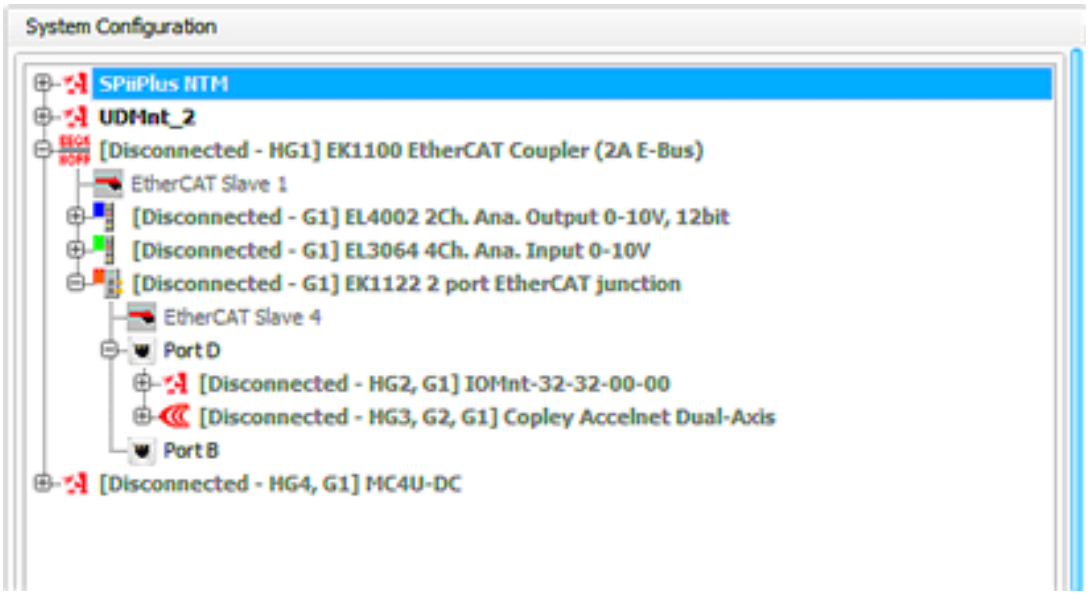



To stop highlighting the optional group user selects either “Undo highlighting optional group” from the right click menu, or highlights another optional group. Only one optional group can be highlighted at the same time.

### *3.3.3.4 View optional groups*

The maximal configuration is presented (including all optional groups). If a partial system is connected (one or more optional groups are disconnected) then all disconnected products are grayed and marked as disconnected by adding the “Disconnected – “word to the prefix of product name.

- Available from tasks
  - System Viewer and Diagnostics




### *3.3.4 Activating Ring Topology in the Controller*

The **Enhanced Network Failure Detection and Recovery** option in the controller affects both automatic and manual setup as follows:

**If Enhanced Network Failure Detection and Recovery is enabled in the controller:**

- Automatic Setup - If the master secondary port is connected, ring topology is activated in the controller; otherwise it is deactivated.
  If a configuration file does not support ring topology , the setup process cannot continue and an appropriate message appears.
- Manual Setup - in the Configuration Design Pad, the relevant check box can be set for activation or deactivation.
  If the check box is selected but the configuration file does not support ring topology, the setup process cannot continue and an appropriate message appears.

  The initial state of the check box depends on the task you are performing:

  - **Modify Configuration Stored on Controller** - the check box will be according to the controller current activation state.
  - **Modify Configuration from Database** and **Load Configuration from Database to Controller** - the check box will be according to EtherCAT configuration file of the selected configuration.

**If Enhanced Network Failure Detection and Recovery is not enabled in the controller:**

- Automatic Setup - the ring topology connection is deactivated in controller.
- Manual Setup - in the Configuration Design Pad, the relevant check box is not selected and cannot be changed.

## *3.4 Safety and Faults Configurator*

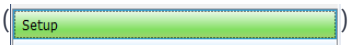
The Safety and Faults Configurator enables you to select specific faults that you want monitored during a program run. There are two types of faults:

> Axis Faults - These faults relate to motor motion.

> System Faults - These faults relate to the controller program that is running.

### *3.4.1 Activating Safety and Faults Configurator*

To activate Safety and Faults Configurator:

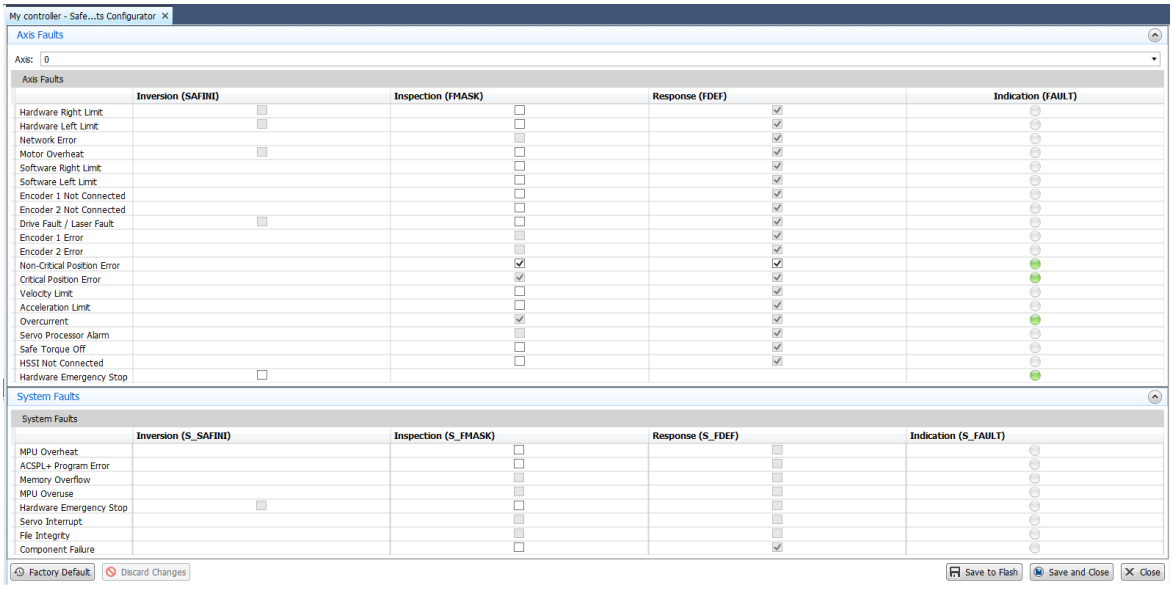
1. In the Toolbox click **Setup** ( 

) to display the Setup list of tools.

2. Click **Safety and Faults Configurator** in the Setup list of tools.

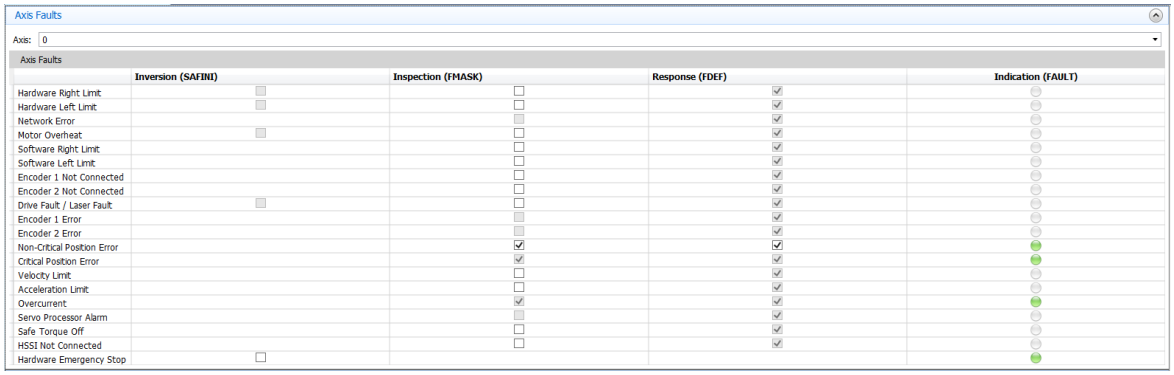
You can also activate the **Safety and Faults Configurator** using the right-click **Add Component** option of the Workspace Tree - see Controller Right-Click Options .

3. The Safety and Faults Configurator window is displayed:

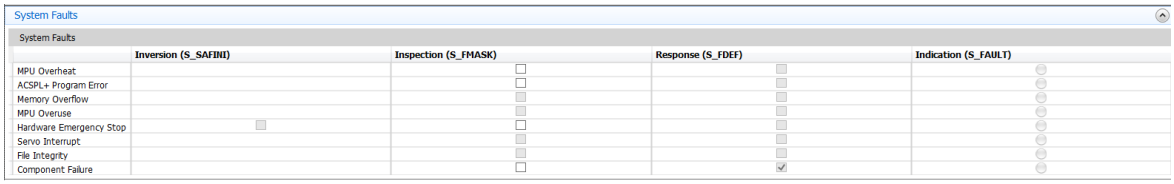



The System and Faults Configurator window is divided into two sections:

### Axis Faults




### System Faults




## *3.4.2 ACSPL+ Fault Variables*

There are two sets of ACSPL+ Fault variables, one for axis faults and one for system faults.

For details of all ACSPL+ variables see the ***SPiiPlus Command & Variable Reference Guide***.

### *3.4.2.1 Axis Fault Variables*

The axis fault variables are:

- SAFINI
- FMASK
- FDEF
- FAULT

#### *3.4.2.1.1 SAFINI*

**SAFINI** is a 64 member (one for each axis) integer array with a set of bits defining the active state of the axis safety input variable (**SAFIN**) specifying inversion of the signal input logic, if required.

When a **SAFINI** bit=0, the corresponding signal is not inverted and the high voltage state is considered active.

When a **SAFINI** bit=1, the bit is inverted and the low voltage state is considered active.

The **SAFINI** bits that can be inverted are:

| Bit | Bit Name | Description |
|---|---|---|
| 0 | #RL | Hardware Right Limit |

| Bit | Bit Name | Description |
|---|---|---|
| 1 | #LL | Hardware Left Limit |
| 4 | #HOT | Motor Overheat |
| 9 | #DRIVE | Drive Fault / Laser Fault |
| 28 | #ES | Hardware Emergency Stop<br>  Changing this bit will automatically change the corresponding bit of all axes belonging to the same device. |

*3.4.2.1.2 FMASK*

**FMASK** is an integer array with a set of bits (one set for each axis) used for enabling or disabling each axis fault bit.

The default value of each bit is 1 and causes the controller to check for the fault associated with that bit, as follows:

0 = the corresponding FAULT bit is disabled.

1 = the corresponding FAULT is enabled and examined each MPU cycle.

The **FMASK** bits are:

| Bit | Fault | Fault Description |
|---|---|---|
| 0 | #RL | Hardware Right Limit<br>1 = Right limit switch is activated. |
| 1 | #LL | Hardware Left Limit<br>1 = Left limit switch is activated. |
| 2 | #NT | Network Error<br>1 = Network error is activated. This applies on to SPiiPlus NT Motion Controllers. |
| 4 | #HOT | Motor Overheat<br>1 = Motor's temperature sensor indicates overheat. |

| Bit | Fault | Fault Description |
|---|---|---|
| 5 | #SRL | Software Right Limit<br>1 = Axis reference position (**RPOS**) is greater than the software right limit margin (**SRLIMIT**). |
| 6 | #SLL | Software Left Limit<br>1 = Axis reference position (**RPOS**) is less than the software left limit margin (**SLLIMIT**). |
| 7 | #ENCNC | Encoder Not Connected<br>1 = Primary encoder (for digital encoder type only) is not connected. |
| 8 | #ENC2NC | Encoder 2 Not Connected<br>1 = Secondary encoder (for digital encoder type only) is not connected. |
| 9 | #DRIVE | Drive Fault / Laser Fault<br>1 = Signal from the drive reports a failure. |
| 10 | #ENC | Encoder Error<br>1 = Primary encoder miscounts. |
| 11 | #ENC2 | Encoder 2 Error<br>1 = Secondary encoder miscounts. |
| 12 | #PE | Non-Critical Position Error<br>1 = Non-critical position error has occurred.<br>PE is defined by the following variables:<br>**ERRI -** Maximum position error while the axis is idle<br>**ERRV** - Maximum position error while the axis is moving with constant velocity<br>**ERRA** - Maximum position error while the axis is accelerating or decelerating<br>**DELI** - Delay on transition from **ERRA** to **ERRI**<br>**DELV** - Delay on transition from **ERRRA** to **ERRV** |

| Bit | Fault | Fault Description |
|---|---|---|
| 13 | #CPE | Critical Position Error<br>1 = Position error exceeds the value of the critical limit.<br>#CPE errors occur outside normal range of operation and #CPE > #PE.<br>The critical limit depends on the axis state and is defined by the following variables:<br>**CERRI** if the axis is idle (not moving)<br>**CERRV** if the axis is moving with constant velocity<br>**CERRA** if the axis is accelerating or decelerating<br>**DELI** - Delay on transition from **ERRA** to **ERRI**<br>**DELV** - Delay on transition from **ERRRA** to **ERRV** |
| 14 | #VL | Velocity Limit<br>1 = Absolute value of the reference velocity (**RVEL**) exceeds the limit defined by the **XVEL** parameter. |
| 15 | #AL | Acceleration Limit<br>1 = Absolute value of the reference acceleration (**RACC**) exceeds the limit defined by the **XACC** parameter. |
| 16 | #CL | Current Limit<br>1 = **RMS** current calculated in the Servo Processor exceeds the limit value defined by the **XRMSD** or **XRMSM** parameter. |
| 17 | #SP | Servo Processor Alarm<br>1 = Axis Servo Processor loses its synchronization with the MPU. The fault indicates a fatal problem in the controller. |
| 18 | #STO | Safe Torque Off<br>1 =STO is activated. |
| 20 | #HSSINC | HSSI Not Connected<br>1 = HSSI module is not connected. |

*3.4.2.1.3 FDEF*

**FDEF** is a 64 member (one for each axis) integer array with a set of bits used for setting default responses to an axis faults.

The **FDEF** bits are:

| Bit | Fault | Fault Description | Default Response |
|---|---|---|---|
| 0 | #RL | Hardware Right Limit<br>1 = Right limit switch is activated. | The controller kills the violating axis.<br>As long as the fault is active, the controller kills any motion that tries to move the axis in the direction of the limit; however, motion within the permissible range is allowed. |
| 1 | #LL | Hardware Left Limit<br>1 = Left limit switch is activated. | Same as for #RL. |
| 2 | #NT | Network Error<br>1 = Network error is activated. | Disable all. Valid only for SPiiPlus NT Motion Controllers. |
| 4 | #HOT | Motor Overheat<br>1 = Motor's temperature sensor indicates overheat. | None. |
| 5 | #SRL | Software Right Limit<br>1 = Axis reference position (**RPOS**) is greater than the software right limit margin (SRLIMIT). | The controller kills the violating axis. As long as the fault is active, the controller kills any motion that tries to move the axis in the direction of the limit. Motion in the direction out of the limit is allowed. |
| 6 | #SLL | Software Left Limit<br>1 = Axis reference position (**RPOS**) is less than the software left limit margin (SLLIMIT). | Same as #SRL. |
| 7 | #ENCNC | Encoder Not Connected<br>1 = Primary encoder (for digital encoder type only) is not connected. | The controller disables the violating axis. |
| 8 | #ENC2NC | Encoder 2 Not Connected | No default response. |

| Bit | Fault | Fault Description | Default Response |
|---|---|---|---|
| | | 1 = Secondary encoder (for digital encoder type only) is not connected. | |
| 9 | #DRIVE | Drive Fault / Laser Fault<br>1 = Signal from the drive reports a failure. | The controller disables the violating axis.<br>This fault is only detected when the axis is enabled. To catch this fault in an ACSPL+ program, write an autoroutine. |
| 10 | #ENC | Encoder Error<br>1 = Primary encoder miscounts. | The controller disables the violating axis. The faults remain active until the user resolves the problems and enables the axis again or executes **FCLEAR**. |
| 11 | #ENC2 | Encoder 2 Error<br>1 = Secondary encoder miscounts. | Same as #ENC. |
| 12 | #PE | Non-Critical Position Error<br>1 = Non-critical position error has occurred.<br><br>PE is defined by the following variables:<br>**ERRI -** Maximum position error while the axis is idle<br>**ERRV** - Maximum position error while the axis is moving with constant velocity<br>**ERRA** - Maximum position error while the axis is accelerating or decelerating<br>**DELI** - Delay on transition from **ERRA** to **ERRI** | None. |

| Bit | Fault | Fault Description | Default Response |
|---|---|---|---|
| | | **DELV** - Delay on transition from **ERRA** to **ERRV** | |
| 13 | #CPE | Critical Position Error<br>1 = Position error exceeds the value of the critical limit.<br>#CPE errors occur outside normal range of operation and #CPE > #PE.<br>The critical limit depends on the axis state and is defined by the following variables:<br>**CERRI** if the axis is idle (not moving)<br>**CERRV** if the axis is moving with constant velocity<br>**CERRA** if the axis is accelerating or decelerating<br>**DELI** - Delay on transition from **ERRA** to **ERRI**<br>**DELV** - Delay on transition from **ERRA** to **ERRV** | The controller disables the violating axis. |
| 14 | #VL | Velocity Limit<br>1 = Absolute value of the reference velocity (**RVEL**) exceeds the limit defined by the **XVEL** parameter. | The controller kills the violating axis. |

| Bit | Fault | Fault Description | Default Response |
|---|---|---|---|
| 15 | #AL | Acceleration Limit<br>1 = Absolute value of the reference acceleration (**RACC**) exceeds the limit defined by the **XACC** parameter. | The controller kills the violating axis. |
| 16 | #CL | Current Limit<br>1 = **RMS** current calculated in the Servo Processor exceeds the limit value defined by the **XRMSD** or **XRMSM** parameter. | The controller disables the violating axis. |
| 17 | #SP | Servo Processor Alarm<br>1 = Axis Servo Processor loses its synchronization with the MPU. The fault indicates a fatal problem in the controller. | The controller disables the violating axis and kills the motion that involves the axis. |
| 18 | #STO | Safe Torque Off<br>1 = STO is activated. | Blocks the PWM signals to the power stage of the drive. |
| 20 | #HSSINC | HSSI Not Connected<br>1 = HSSI module is not connected. | None. |

*3.4.2.1.4 FAULT*

**FAULT** is a 64 member (one for each axis) integer array with a set of bits that stores axis-related fault bits.

**FAULT** indicates axis-related fault bits as detected by the safety mechanism. When each of the faults is active (such as Left Limit), the corresponding fault bit becomes = 1 while the fault is active, and automatically reverts to 0 when the fault is no longer active.

Each fault can be masked by FMASK.

The logic of some faults can be inverted by SAFINI.

The default response of each fault can be disabled by FDEF. In this case, any customized default response can be implemented by autoroutines.

The **FAULT** bits are:

| Bit | Fault | Fault Description |
|---|---|---|
| 0 | #RL | Hardware Right Limit.<br>1 = Right limit switch is activated. |
| 1 | #LL | Hardware Left Limit.<br>1 = Left limit switch is activated. |
| 2 | #NT | Network Error<br>1 = Network error is activated. Valid only for SPiiPlus NT Motion Controllers. |
| 4 | #HOT | Motor Overheat<br>1 = Motor's temperature sensor indicates overheat. |
| 5 | #SRL | Software Right Limit<br>1 = Axis reference position (**RPOS**) is greater than the software right limit margin (**SRLIMIT**). |
| 6 | #SLL | Software Left Limit<br>1 = Axis reference position (**RPOS**) is less than the software left limit margin (**SLLIMIT**). |
| 7 | #ENCNC | Encoder Not Connected<br>1 = Primary encoder (for digital encoder type only) is not connected. |
| 8 | #ENC2NC | Encoder 2 Not Connected<br>1 = Secondary encoder (for digital encoder type only) is not connected. |
| 9 | #DRIVE | Drive Fault / Laser Fault<br>1 = Signal from the drive reports a failure. |
| 10 | #ENC | Encoder Error<br>1 = Primary encoder miscounts. |
| 11 | #ENC2 | Encoder 2 Error<br>1 = Secondary encoder miscounts. |
| 12 | #PE | Non-Critical Position Error<br>1 = Non-critical position error has occurred. |

| Bit | Fault | Fault Description |
|---|---|---|
| | | PE is defined by the following variables:<br>**ERRI -** Maximum position error while the axis is idle<br>**ERRV** - Maximum position error while the axis is moving with constant velocity<br>**ERRA** - Maximum position error while the axis is accelerating or decelerating<br>**DELI** - Delay on transition from **ERRA** to **ERRI**<br>**DELV** - Delay on transition from **ERRA** to **ERRV** |
| 13 | #CPE | Critical Position Error<br>1 = Position error exceeds the value of the critical limit.<br>#CPE errors occur outside normal range of operation and #CPE > #PE.<br>The critical limit depends on the axis state and is defined by the following variables:<br>**CERRI** if the axis is idle (not moving)<br>**CERRV** if the axis is moving with constant velocity<br>**CERRA** if the axis is accelerating or decelerating<br>**DELI** - Delay on transition from **ERRA** to **ERRI**<br>**DELV** - Delay on transition from **ERRA** to **ERRV** |
| 14 | #VL | Velocity Limit<br>1 = Absolute value of the reference velocity (**RVEL**) exceeds the limit defined by the **XVEL** parameter. |
| 15 | #AL | Acceleration Limit<br>1 = Absolute value of the reference acceleration (**RACC**) exceeds the limit defined by the **XACC** parameter. |
| 16 | #CL | Current Limit<br>1 = **RMS** current calculated in the Servo Processor exceeds the limit value defined by the **XRMSD** or **XRMSM** parameter. |
| 17 | #SP | Servo Processor Alarm<br>1 = Axis Servo Processor loses its synchronization with the MPU. The fault indicates a fatal problem in the controller. |

| Bit | Fault | Fault Description |
|---|---|---|
| 18 | #STO | Safe Torque Off<br>1 = STO is active. |
| 20 | #HSSINC | HSSI Not Connected<br>1 = HSSI module is not connected. |

### *3.4.2.2 System Fault Variables*

The system fault variables are:

- S_SAFINI
- S_FMASK
- S_FDEF
- S_FAULT

#### *3.4.2.2.1 S_SAFINI*

**S_SAFINI** is a scalar integer variable used for defining the active state of the system safety input variable (**S_SAFIN**) specifying inversion of the signal input logic, if required.

When an **S_SAFINI** bit=0, the corresponding signal is not inverted and the high voltage state is considered active.

When an **S_SAFINI** bit=1, the bit is inverted and the low voltage state is considered active.

There is only one **S_SAFINI** bit that can be inverted:

| Bit | Fault | Fault Description |
|---|---|---|
| 28 | #ES | Hardware Emergency Stop<br>1 = **ES** signal is activated. |

#### *3.4.2.2.2 S_FMASK*

**S_FMASK** is scalar integer variable consisting of a set of bits for enabling or disabling the system faults contained in S_FAULT. **S_FMASK** is connected to **S_FAULT** in the same way that FMASK is connected withFAULT.

The **S_FMASK** bit default values are 1 which cause the controller to check for the fault associated with that bit, as follows:

- 0: The corresponding **S_FAULT** bit is disabled
- 1: The corresponding **S_FAULT** is enabled and examined each MPU cycle.

The **S_FMASK** bits are:

| Bit | Fault | Fault Description |
|---|---|---|
| 25 | #PROG | Program Fault<br>1 = Run time error occurs in one of the executing ACSPL+ programs. |
| 26 | #MEM | Memory Overflow<br>1 = User application requires too much memory. |
| 27 | #TIME | MPU Overuse<br>1 = User application consumes too much time in the controller cycle. |
| 28 | #ES | Hardware Emergency Stop<br>1 = ES signal is activated. |
| 29 | #INT | Servo Interrupt<br>1 = The servo interrupt that defines the controller cycle is not generated. The fault indicates a fatal controller problem. |
| 30 | #INTGR | File Integrity<br>1 = The integrity of the user application in controller RAM is checked by the controller at power-up and whenever an **#IR** command is issued. |
| 31 | #FAILURE | Component Failure<br>1 = An MC4U hardware component other than the drive, such as the Power Supply, I/O card, or encoder card, has failed. |

*3.4.2.2.3 S_FDEF*

**S_FDEF** is a scalar integer variable consisting of a set of bits for defining the default response for the system faults contained in S_FAULT. **S_FDEF** is connected to **S_FAULT** in the same way that FDEF is connected with FAULT.

The default value for all **S_FDEF** bits is 1, which enables the default response. If an **S_FDEF** bit = 0, the default response is disabled.

The **S_FDEF** bits are:

| Bit | Fault | Fault Description | Default Response |
|---|---|---|---|
| 25 | #PROG | Program Fault<br>1 = Run time error occurs in one of the executing ACSPL+ programs. | The controller kills all axes. |
| 26 | #MEM | Memory Fault<br>1 = User application requires too much memory. | The controller kills all axes. |

| Bit | Fault | Fault Description | Default Response |
|---|---|---|---|
| 27 | #TIME | MPU Overuse<br>1 = User application consumes too much time in the controller cycle. | No default response. |
| 28 | #ES | Hardware Emergency Stop<br>1 = ES signal is activated. | The controller disables all axes.<br>It does not stop the program buffers. |
| 29 | #INT | Servo Interrupt<br>1 = The servo interrupt that defines the controller cycle is not generated. The fault indicates a fatal controller problem. | The controller disables all axes. |
| 30 | #INTGR | File Integrity<br>1 = The integrity of the user application in controller RAM is checked by the controller at power-up and whenever an **#IR** command is issued. | No default response |
| 31 | #FAILURE | Component Failure<br>1 = An MC4U hardware component other than the drive, such as the Power Supply, I/O card, or encoder card, has failed. | No default response<br>You are responsible for incorporating a user-defined response into your program. |

*3.4.2.2.4 S_FAULT*

**S_FAULT** is a scalar integer variable consisting of a set of bits equating to the occurrence of System Faults (faults that are not related to any specific axis).

The **S_FAULT** bits are:

The **S_FAULT** variable contains bits related to Axis Faults, but only the System Fault bits are relevant to the Safety and Faults Configurator.

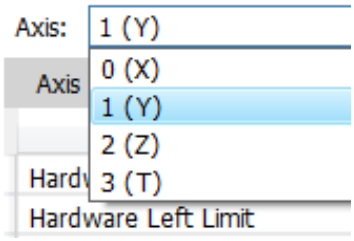
| Bit | Fault | Fault Description |
|---|---|---|
| 25 | #PROG | Program Fault<br>1 = Run time error occurs in one of the executing ACSPL+ programs. |
| 26 | #MEM | Memory Fault<br>1 = User application requires too much memory. |
| 27 | #TIME | MPU Overuse<br>1 = User application consumes too much time in the controller cycle. |
| 28 | #ES | Hardware Emergency Stop<br>1 = ES signal is activated. |
| 29 | #INT | Servo Interrupt<br>1 = The servo interrupt that defines the controller cycle is not generated. The fault indicates a fatal controller problem. |
| 30 | #INTGR | File Integrity<br>1 = The integrity of the user application in controller RAM is checked by the controller at power-up and whenever an **#IR** command is issued. |
| 31 | #FAILURE | Component Failure<br>1 = An MC4U hardware component other than the drive, such as the Power Supply, I/O card, or encoder card, has failed. |

### *3.4.3 Working with Safety and Faults Configurator*

**Select Axis**

You can select as many axes as you have or are interested in tracking. To select an axis:

1. Click the **Axis** down arrow to display the list of axes.




2. Click the axis that you want to configure.

**Enabling/Disabling Fault Bits**

Only those fault variable bits with a checkbox in the column are available for selection. To enable a bit you click it.

The bit selection symbols are:

| | |
|---|---|
| ☐ | Bit has not been selected. |
| ☑ | Bit has been selected. |
| ☐ | Bit is not available for selection. |

Some bit selections are dependent on your selecting other bits. For example, there is a connection between FMASK and FDEF, when you select a bit in **FMASK**, this makes the corresponding bit in **FDEF** available for selection.

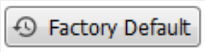
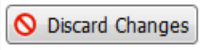
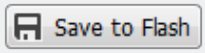
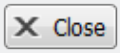
### Indication

The symbols in the **Indication** column act like LEDs and indicate:

| | |
|---|---|
|  | No fault. |
|  | Fault detected. |
|  | Not applicable. |

### Action Buttons

The Safety and Faults Configurator has the following action buttons:

| | |
|---|---|
|  | Sets all parameters to factory defaults. |
|  | Removes changes and restores previous selections. |
|  | Saves the data to the controller's flash memory. |
|  | Saves the data to the controller's flash memory and closes the window. |
|  | Closes the window without saving the data. |

# 4. Application Development

Application Development contains the following tools for use in programming your SPiiPlus Motion Controller:

> Application Wizard - enables you to manage the controller program files, such as loading applications parameters, saving the parameters, copying files to the controller, and the like.

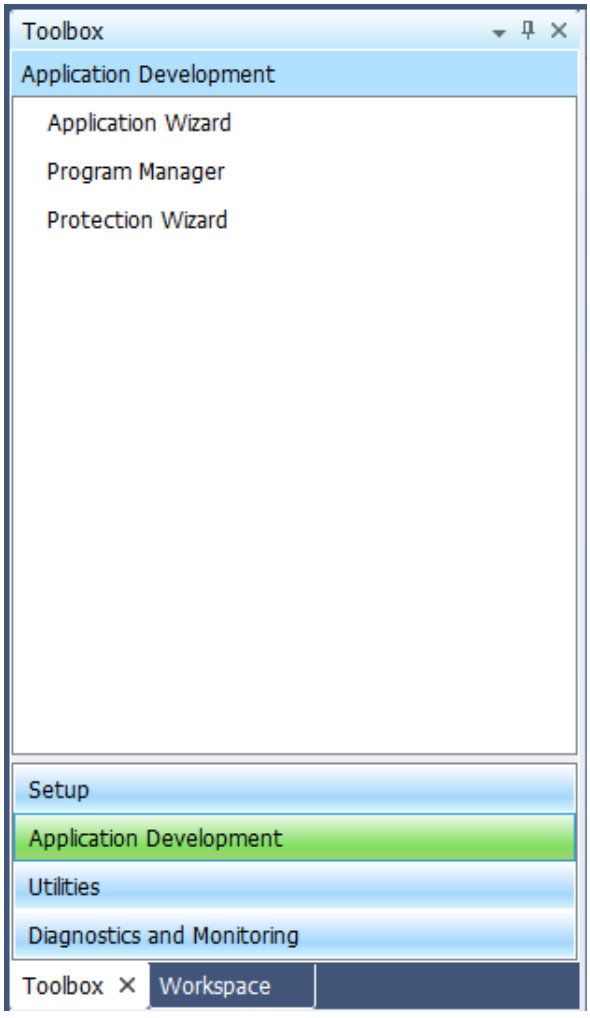
> Program Manager - your principal tool for programming your SPiiPlus Motion Controller.

> Protection Wizard - enables you to define the controller protection mode parameters, modify the protection mode parameters, and view the protection status.

## 4.1 Accessing Application Development Tools

To access the Application Development tools:

1. If the Toolbox is not displayed, click **Toolbox**.
2. Click **Application Development** . The Application Development list of tools appears:




3. Click the required tool in the list to open it.

## 4.2 Application Wizard

The **Application Wizard** is a tool for loading or saving system and axis parameters from or to the disk files, changing settings, managing the user files, exporting/importing arrays, copying any files to the controller's flash and deleting any files from the controller's flash.

**Application Wizard** leads you through the performance of the following tasks:

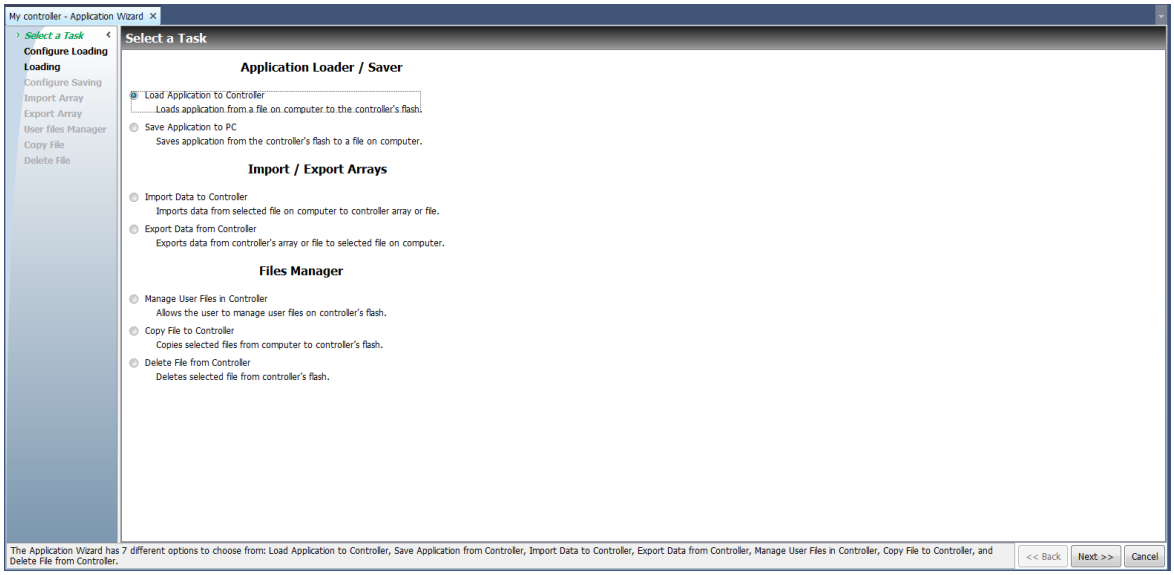
- Application Loader
- Application Saver
- User File Manager
- Import Data to Controller
- Export Data from Controller
- Copy File to Controller
- Delete File from Controller

### 4.2.1 Activating Application Wizard

1. From the Toolbox click **Application Development** .
2. Click **Application Wizard**.

You can also activate the **Application Wizard** using the right-click **Add Component** option of the Workspace Tree - see Controller Right-Click Options.

The **Application Wizard** Task window appears.




3. Select the required task, then click **Next**.

### 4.2.2 Application Loader

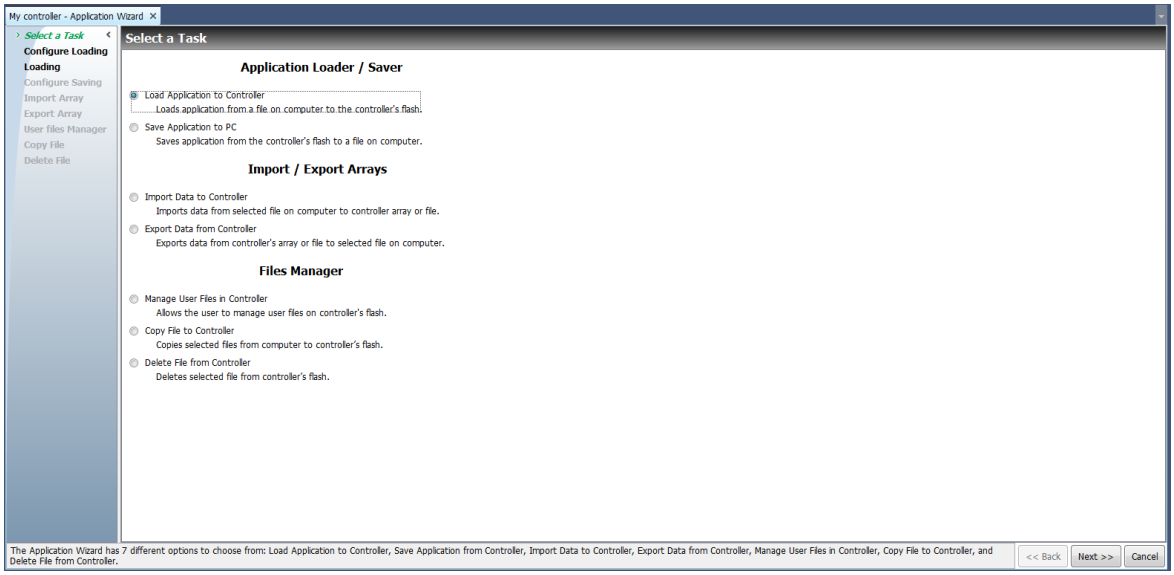
Application Loader is used:

> To load an application from the disk file

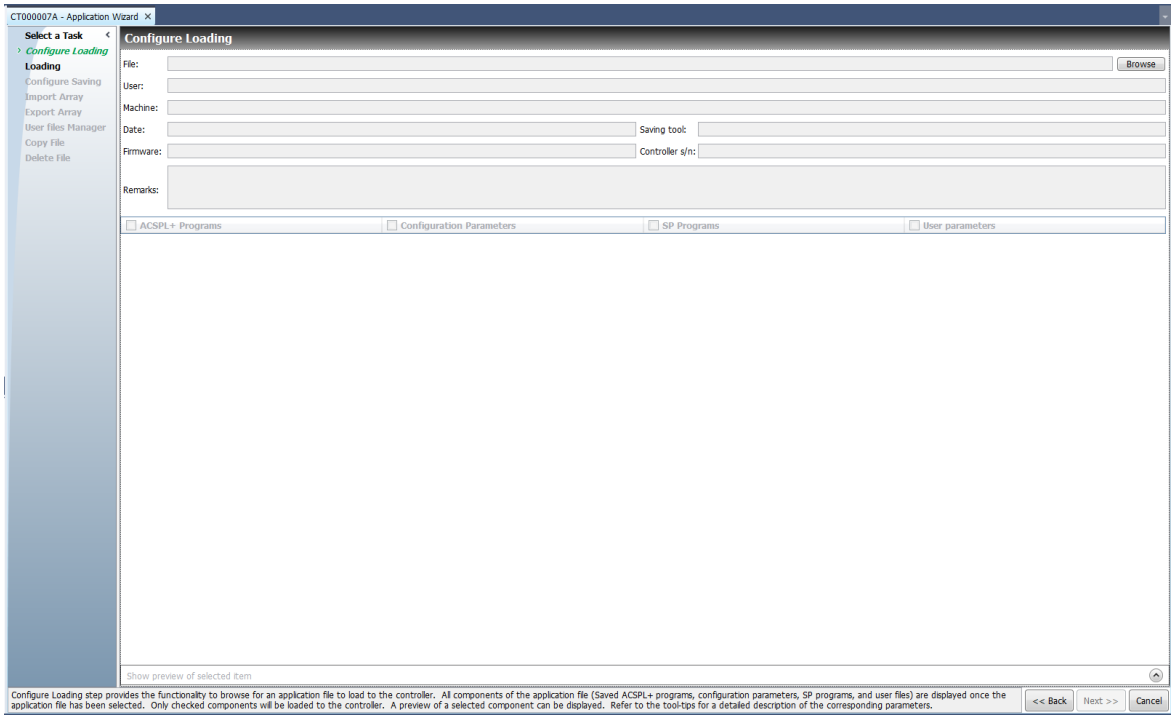
> To change settings of the loaded application.

To load an application from a disk file:

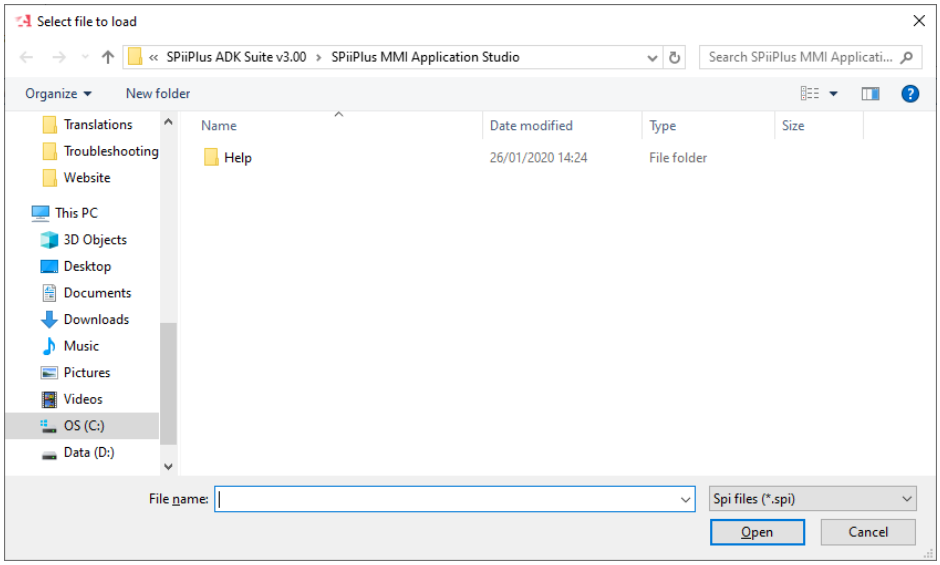
1. Select **Load Application to the Controller** under Application Saver/Loader in Select a Task.




2. Click **Next**. The Configure Loading window appears.

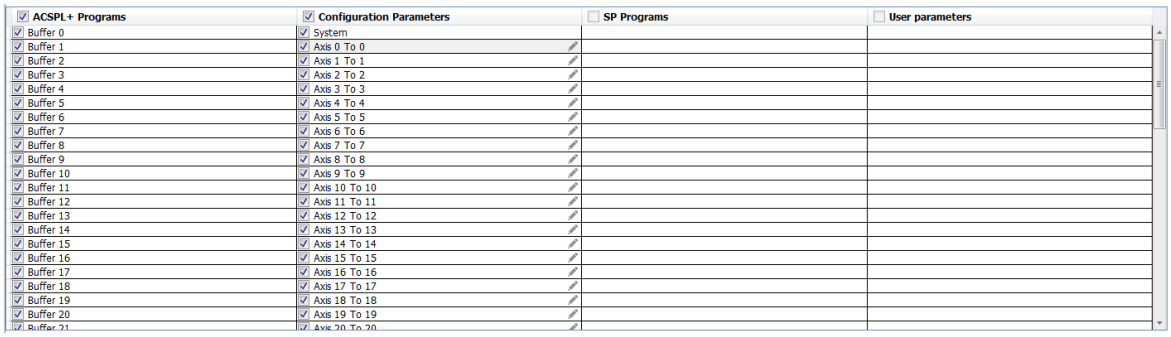



3. Click **Browse** to select the application file to be loaded.




Application files have an **.spi** extension.

4. Select the file and click **Open**.
   The common information about the application appears and the option grid is filled with the values from the file. You can choose the setting to be loaded.




   In the **Configuration Parameters** column you can see default axis assignment (which axis in the loaded application is assigned each axis in the controller).

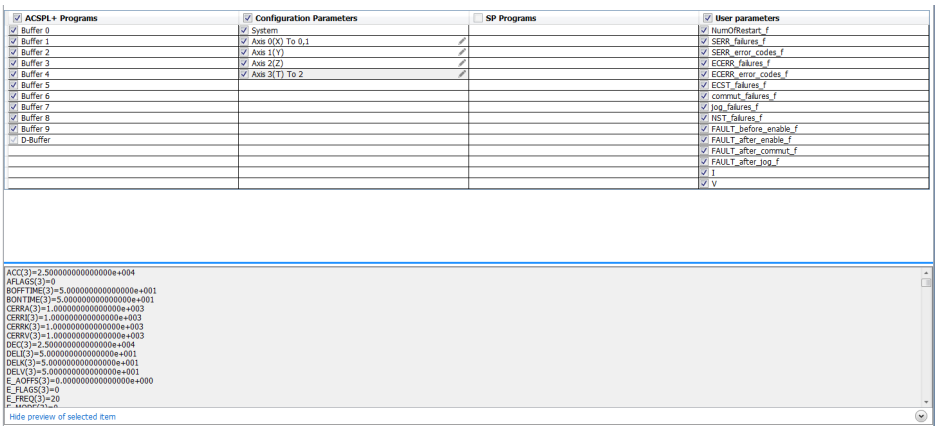
5. To change the default assignment, click  next to the required axis or axes and clear the destination axis.
   This lets you assign the destination axis with a different source axis.


6. Click **OK**.

7. Click next to any axis and assign the axis that you unassigned in the previous step.


In this way, you can assign any axis from the loaded application to any **axis or axes** in the controller.

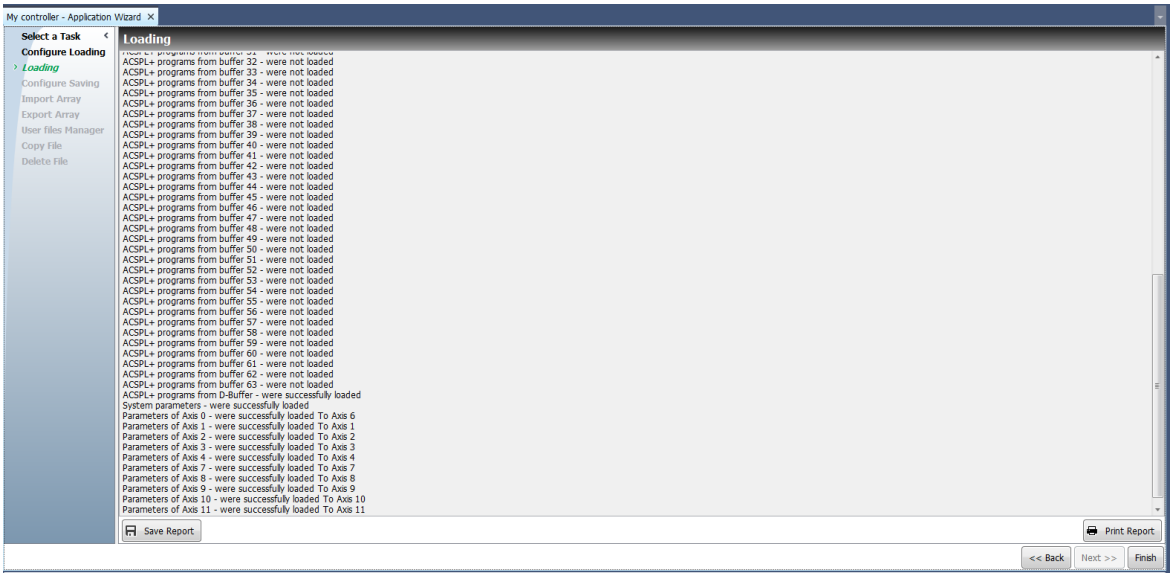
8. You can preview the data loading from the file. Select **Show preview of selected item**.



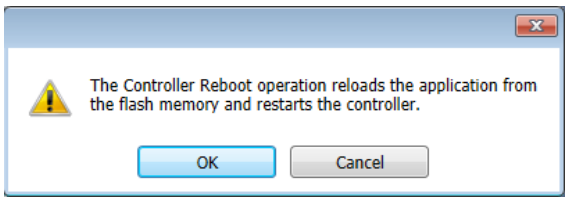

9. To check or uncheck all options in the grid column, click the checkbox in the column header.

SP Programs will appear only if they have been previously saved.

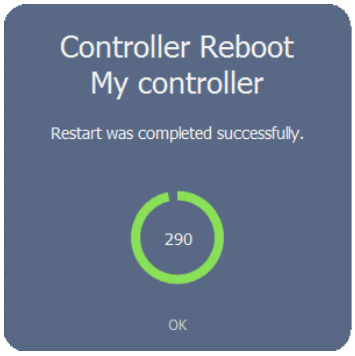
10. Click **Next** to start loading process.
The Loading window is displayed listing the process steps that are performed:



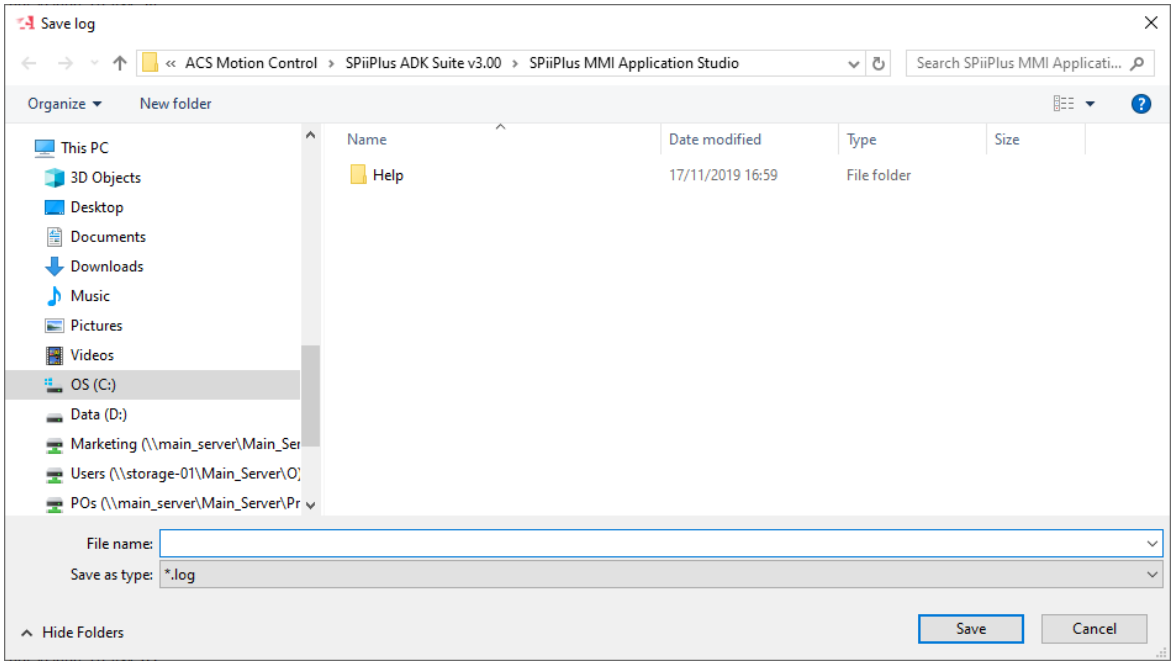

The controller is rebooted:




11. Click **OK**.
12. Wait for notification of a successful reboot:




13. Click **OK**. You are returned to the Loading window.
14. To save the report click **Save Report**. The browser window is displayed:




15. Select a directory and enter a filename. Click **Save**.

The file extension for all reports is **.log**.

16. To print the report click **Print Report**.
17. Click **Finish** to exit the Application Wizard.

### *4.2.3 Application Saver*

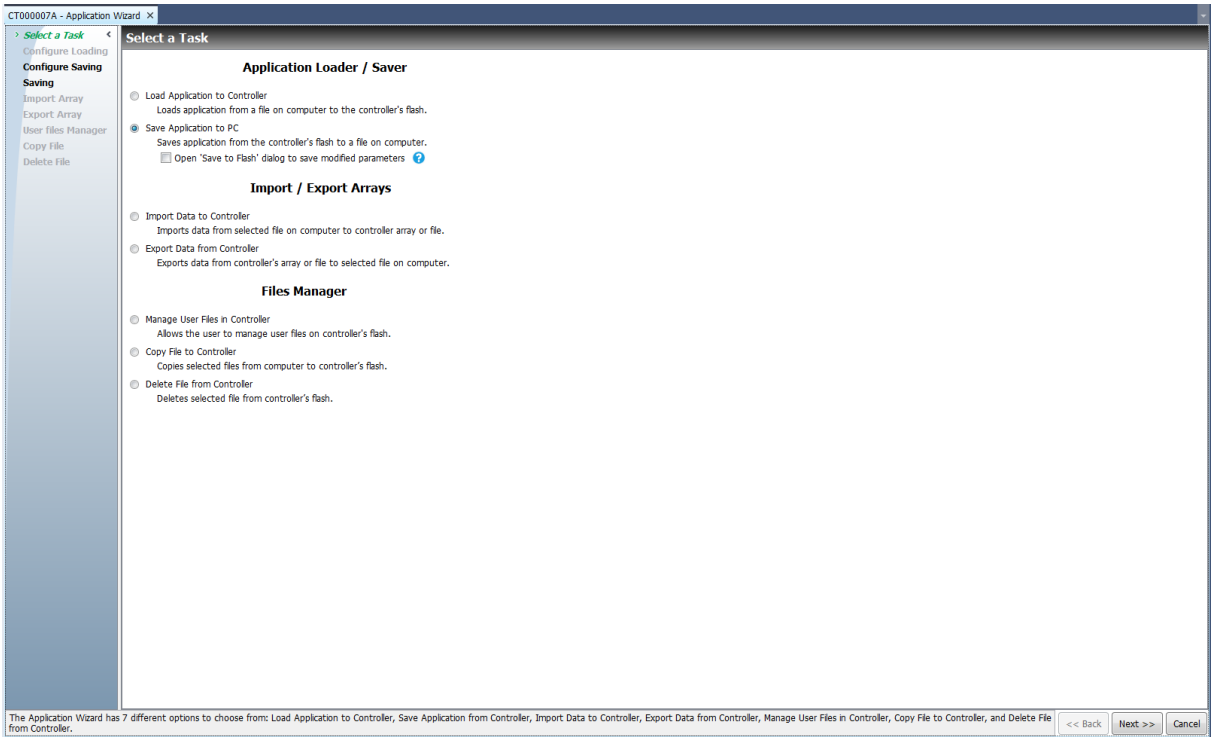
Application Saver is used:

> To save the application from the controller to the disk file.

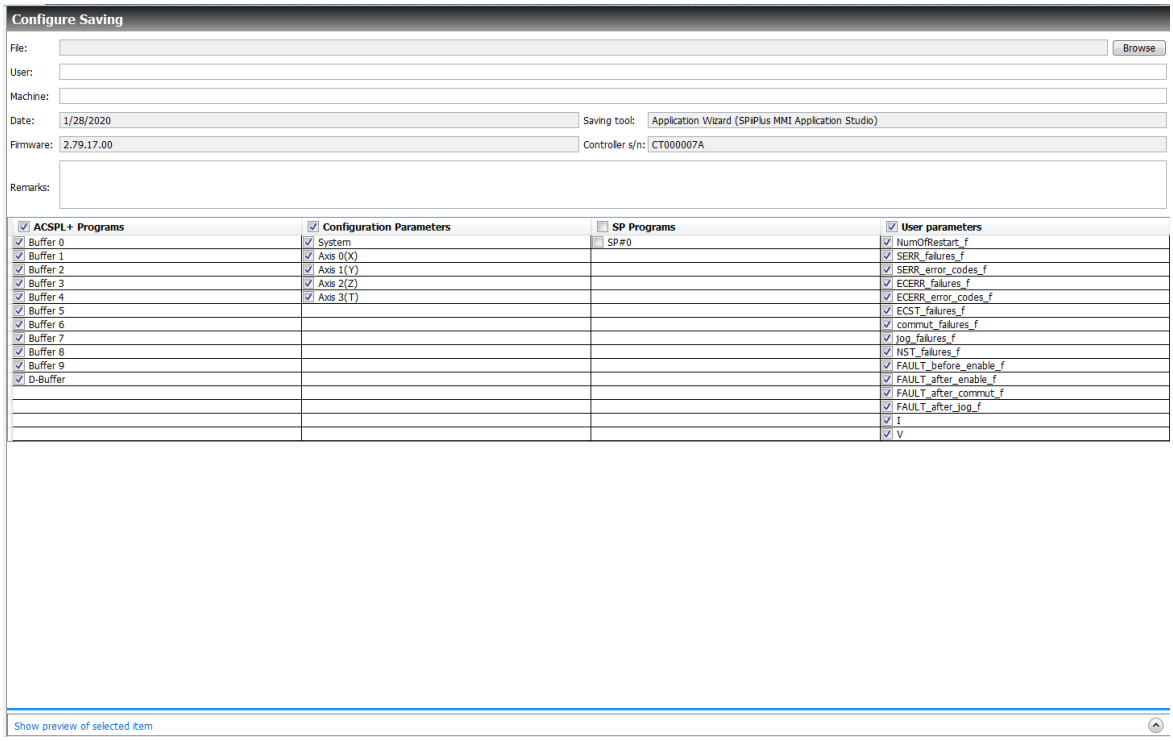
> To view and, where needed, change the settings of the axes, system parameters and variables in the controller application.

To save an application from the controller to a disk file:

1. Select **Save Application from Controller** from **SELECT A TASK**.




2. Click **Next**. The Save to Flash window is displayed:



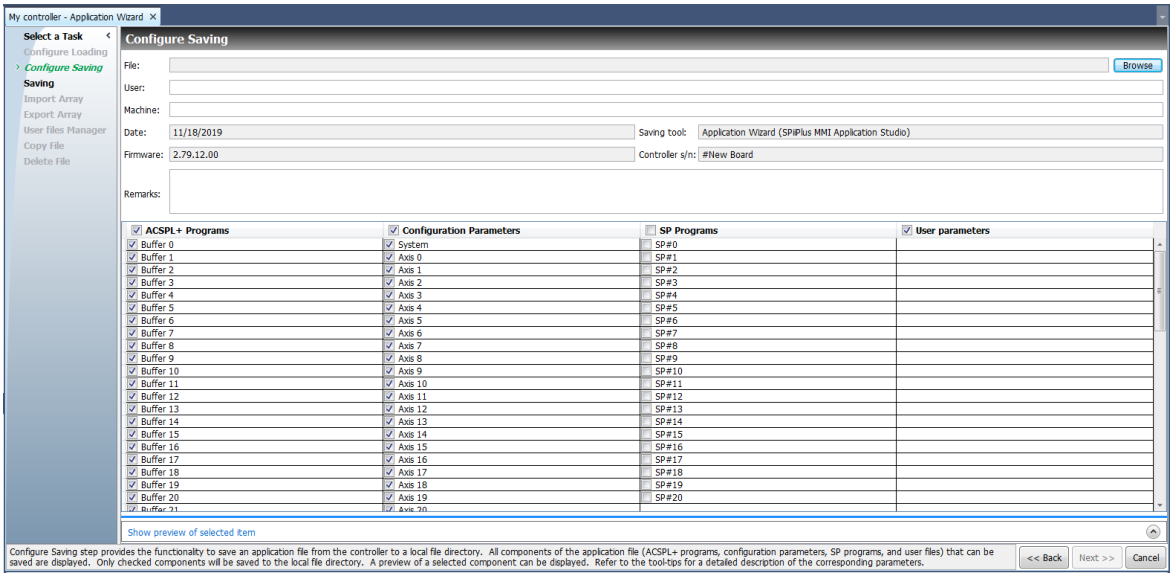

The save to file process is as follows:

> The wizard first saves the configuration data to the controller's flash.

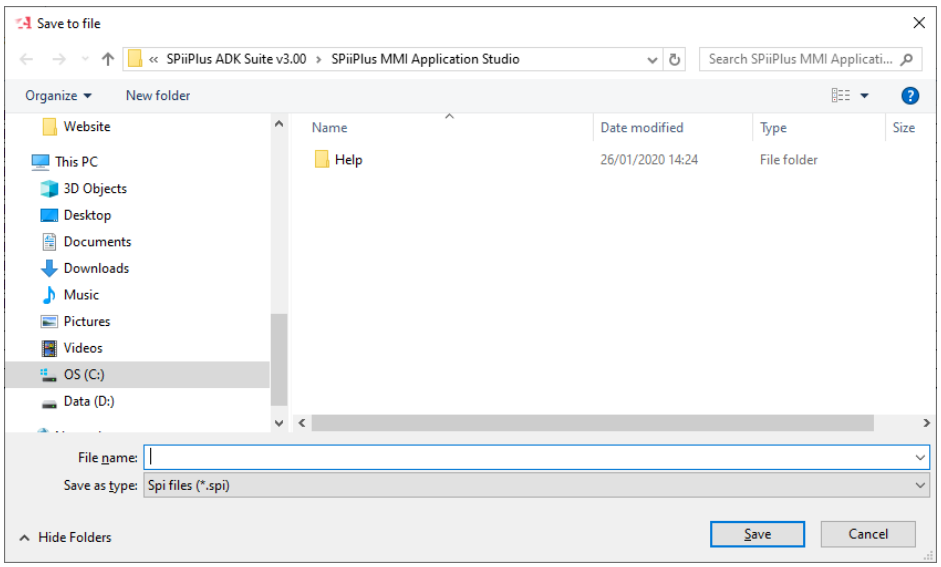
> It then saves the data from the controller's flash to the selected file.

You have the option of saving some or all of the application data by selecting or deselecting the checkbox ( ) in the column header.

3. Click **Save**. The Configure Saving window is displayed:




4. Click **Browse** to select where the application data is to be saved. A browse window is displayed:

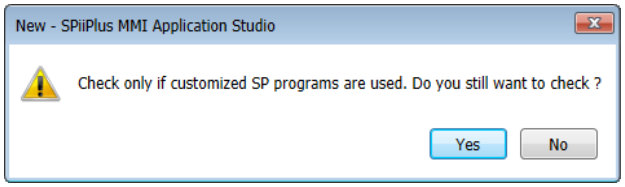



Application files have an **.spi** extension.

5. Select the directory, and enter a file name in the **File name** field. Click **Open**. You are returned to the Configure Saving window.

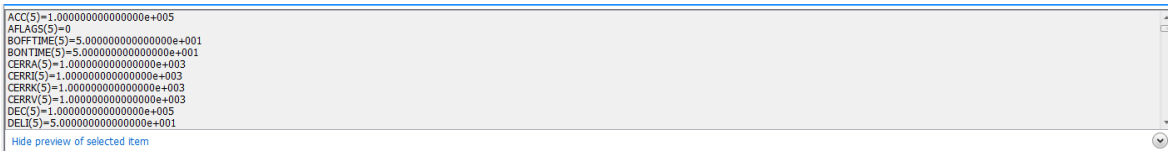
6. You can determine what data to be saved by selecting or deselecting the checkbox ( ) in the column header.

By default the **SP** column is not selected. If you want to save the SP data as well, select the column. You are advised:

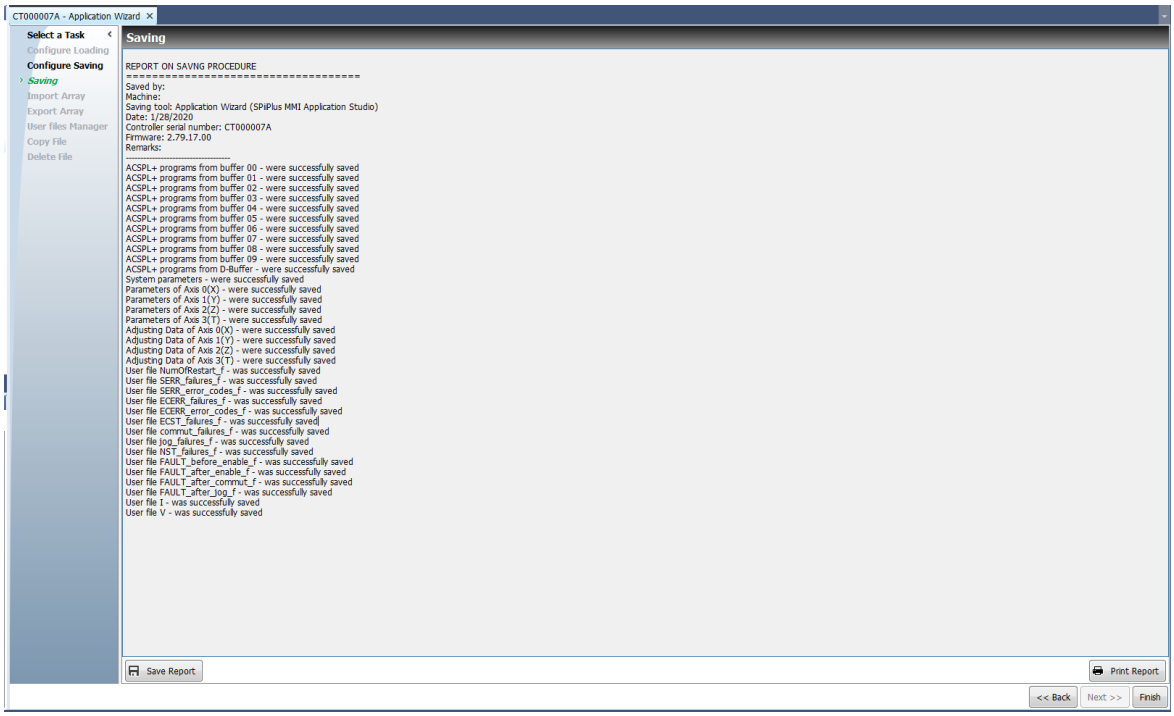



Normally you will want to save SP data only if the SP programs have been customized; however, you can save it in any case by clicking **Yes**.

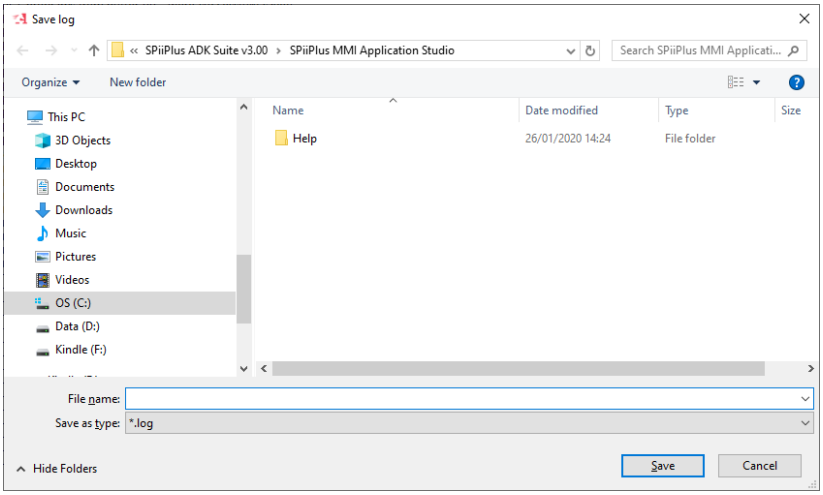
7. You can view any data in the controller application by selecting the item and clicking **Show preview of selected item**, for example:




8. Click **Next**. The Saving window is displayed.




9. To save the report click **Save Report**. The browser window is displayed:




10. Select a directory and enter a filename. Click **Save**.

The file extension for all reports is **.log**.

11. To print the report click **Print Report**.
12. Click **Finish** to exit the Application Wizard.

### *4.2.4 Import Data to Controller*

Import Data to Controller is used:

> To import arrays and user files to the controller.

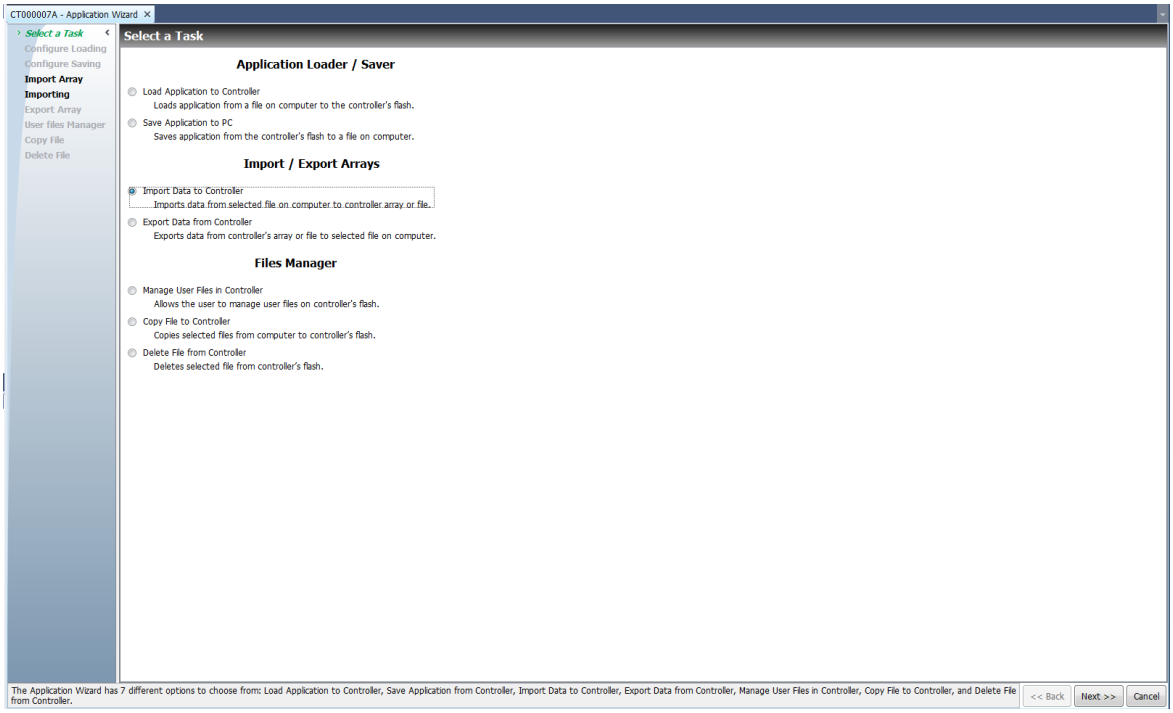
> To preview the contents of the array or the user file .

The imported data must be in tabular form (an array of rows and columns), with the columns separated by either a tab or space.

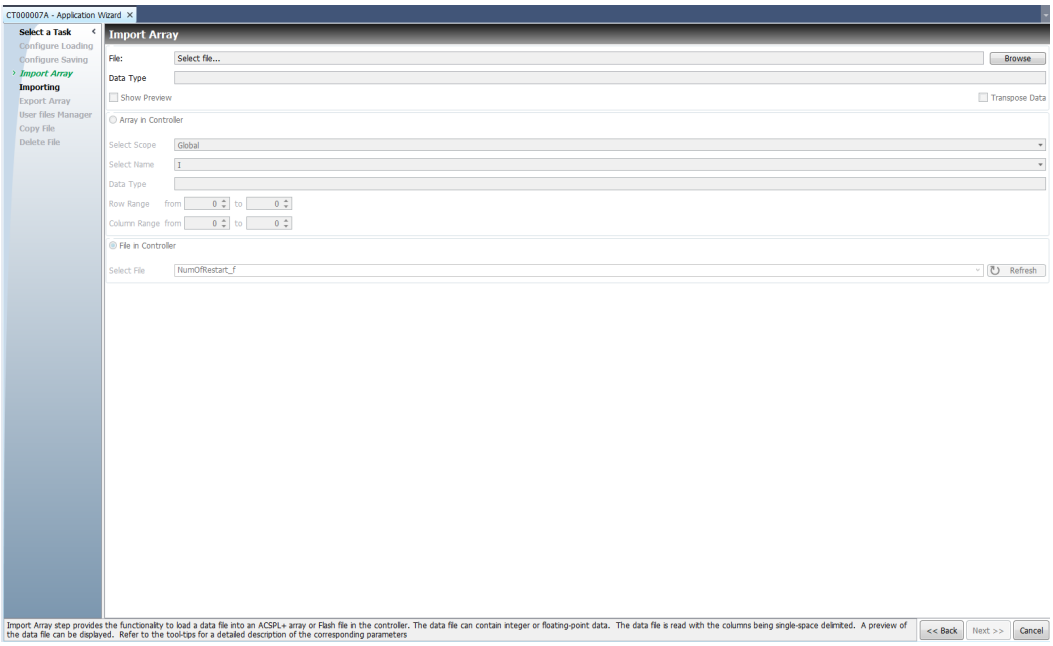
If you are importing data from an Excel file:

1. Ensure the data is organized in tabular form (rows and columns).
2. Depending on the Excel version you have, click **FILE** >**SAVE AS**.
3. Select the folder **PROGRAM ARRAYS**, and in the **FILE NAME** box, type a name for the file.
4. In the **SAVE AS TYPE** list, click either **TEXT (TAB DELIMITED) (*.TXT)**or **FORMATTED TEXT (SPACE DELIMITED) (*.PRN)**.
5. Click **SAVE**.

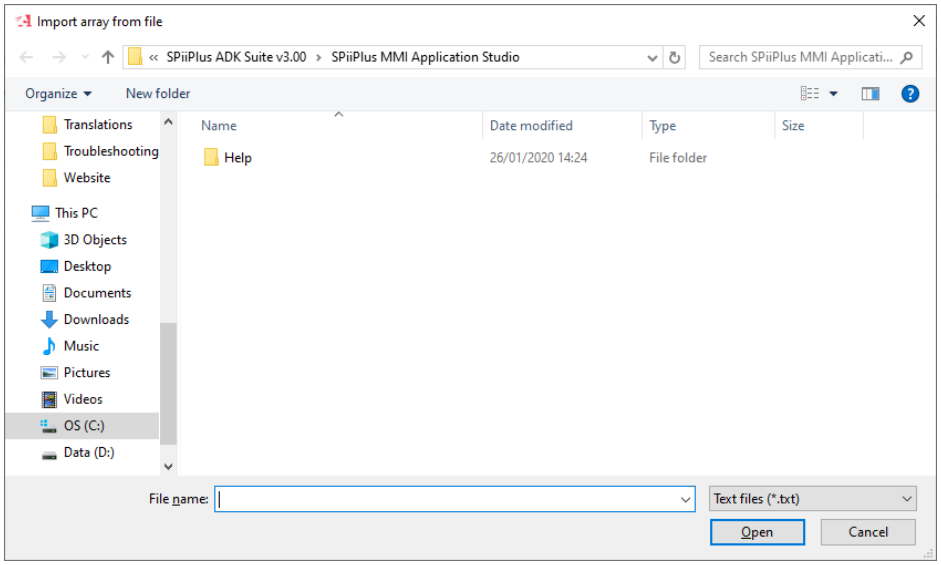
1. Select **Import Data to Controller** from Select a Task.



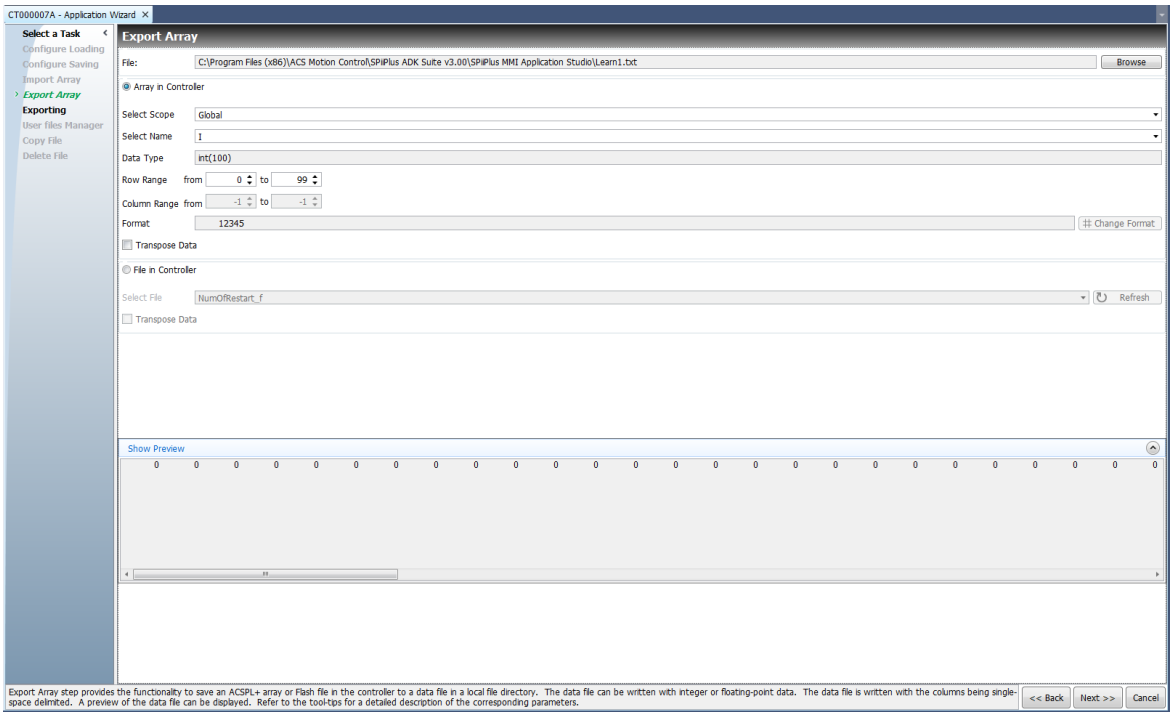

2. Click **Next**. The Import Array window is displayed.




3. Click **Browse** and select the file for importing from the list that is displayed. Then click **Open**.




4. You may preview the data of the selected array file by selecting **Show preview**.




You can transpose the array by clicking **Transpose**.

You have two options:

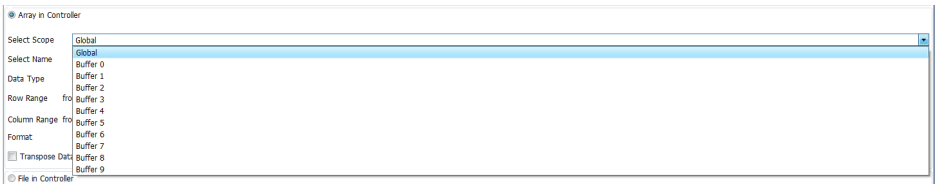
> Importing into an ASCPL+ array (or matrix) variable

> Importing into a user-defined array (or matrix) variable

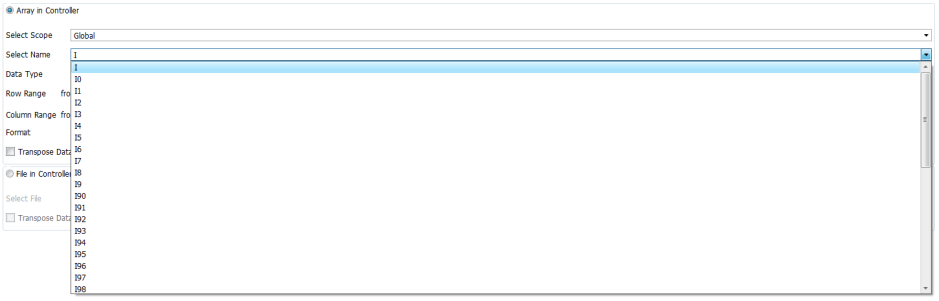
1. For importing into an ASCPL+ array:

> Select **Array in Controller**.

> You can select from Global variables, or narrow the selection of variables to a particular buffer by selecting the buffer from the **Select Scope** dropdown list:




> Select the variable name from the dropdown list.




Once you have entered the variable name, the data type is automatically displayed.

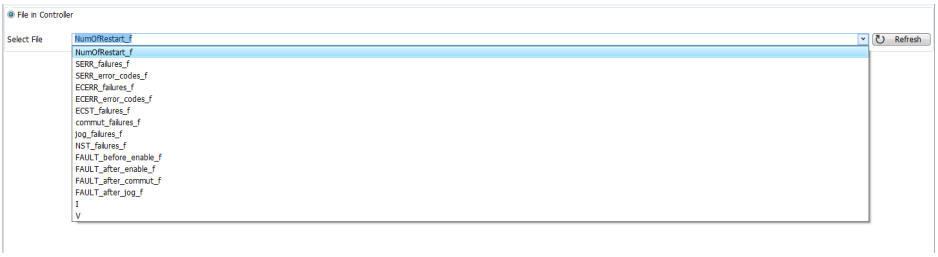
You have the option of selecting only a portion of the array by entering the range using **Row Range** and **ColumnRange**, as appropriate.


> Click **Next** to start the exporting process.

2. For importing into a file in the controller's flash:

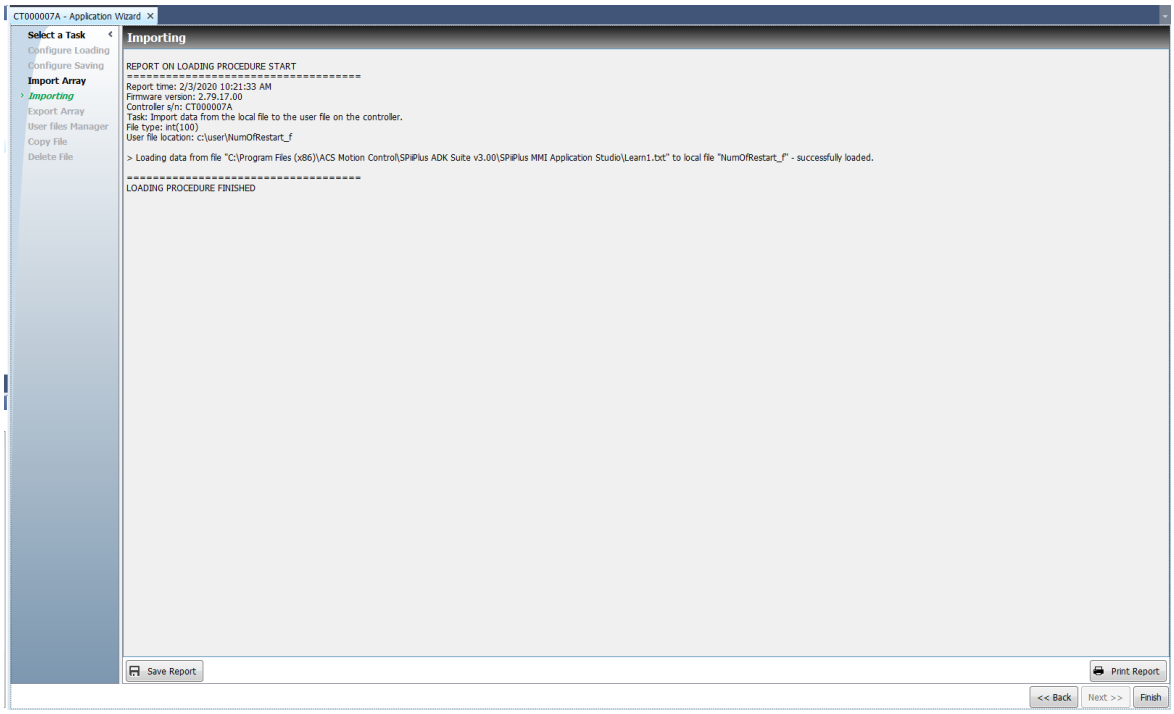
> Select **File in Controller**.

> Either type in a name or select the required file from the dropdown list.

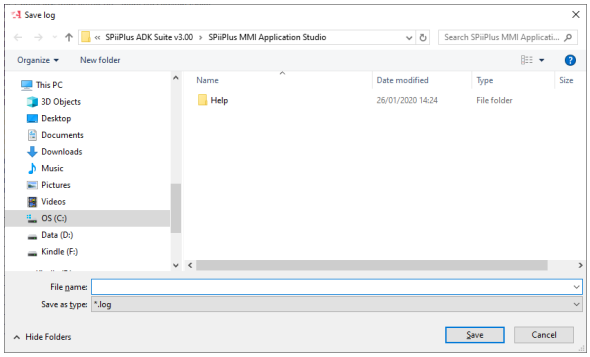



> Click **Next** to start the exporting process.

The Importing window is displayed with a running log of the process:




3. You have the option of saving the log to a file by clicking **Save Report** which opens a dialog window:




4. Select a directory and enter a filename. Click **Save**.

The file extension for all reports is **.log**

5. To print the report click **Print Report**.
6. Click **Finish** to exit the Application Wizard.

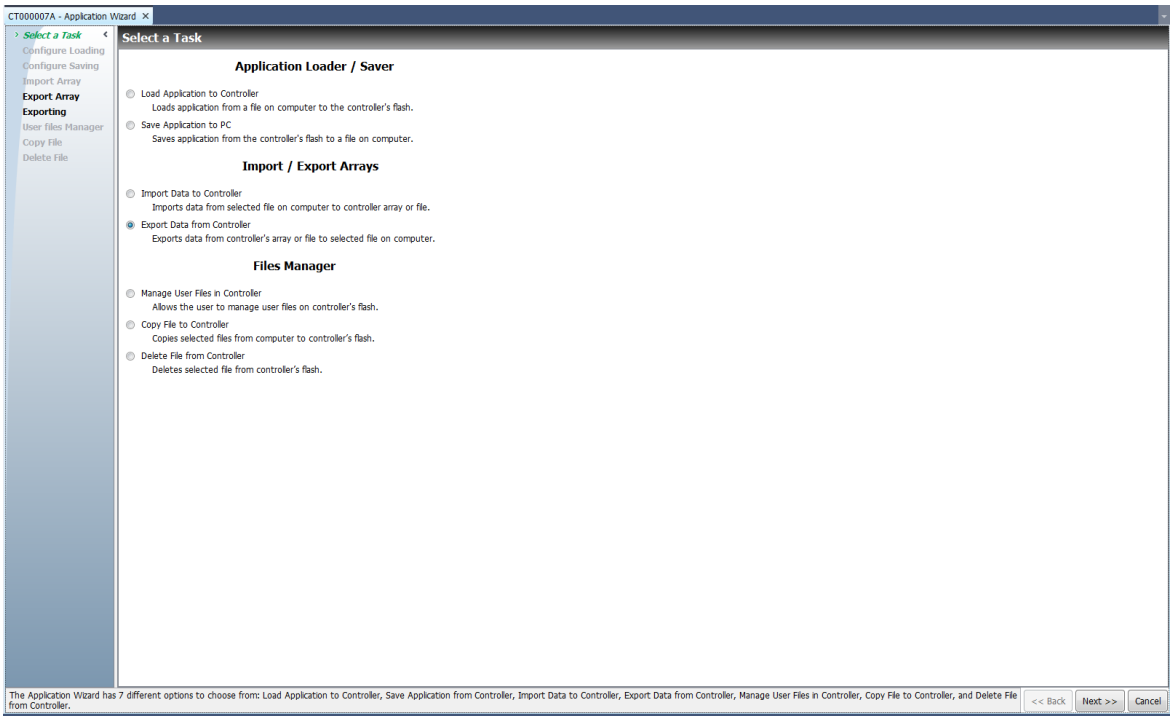
### 4.2.5 Export Data from Controller

Export Data from Controller is used:

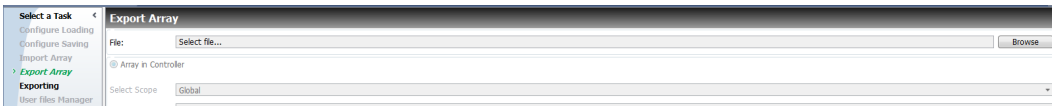
> To export arrays and user files from the controller flash to files on the disk

> To preview the content of the array or the user file from the controller

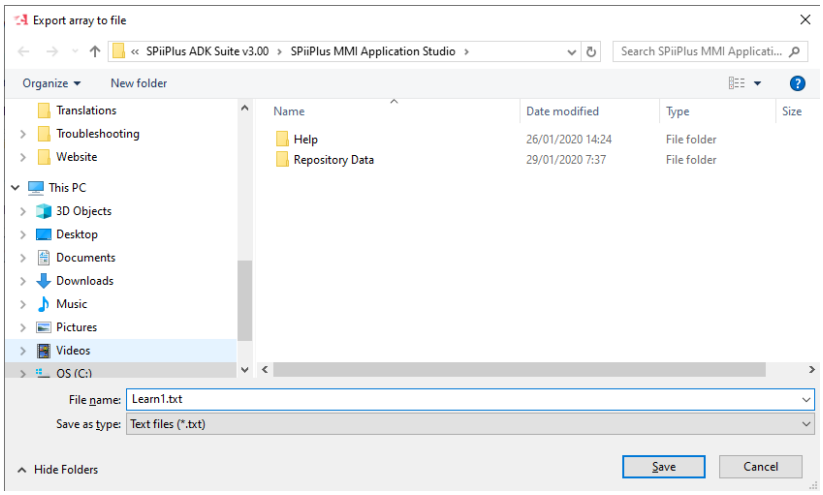
1. Select **Export Data from Controller** from Select a Task.




2. Click **Next**. The Export Array to the File window is displayed.
3. Click **Browse** in the **Select File** field, the browse window is displayed:



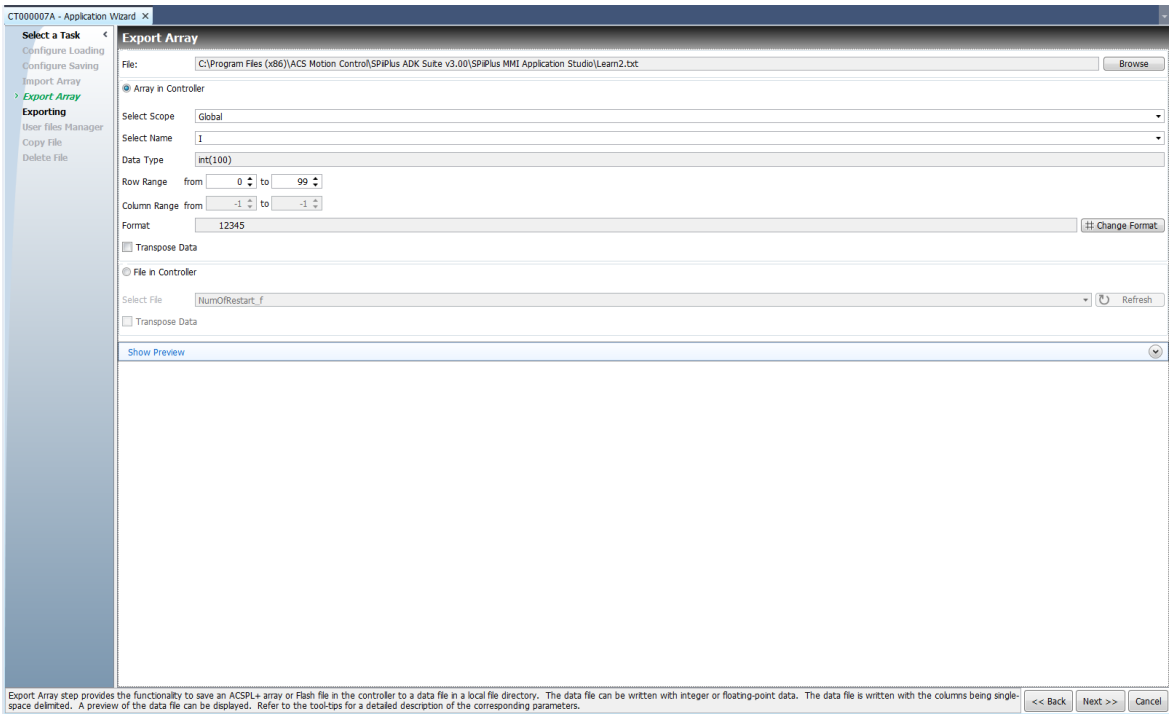

4. Use the browse window to select the directory, and, if applicable, a file name.




5. Click **Save**.

You can type a name in the **File name** field to create a new file.

The Export Array to the File window is displayed:




You have two options:

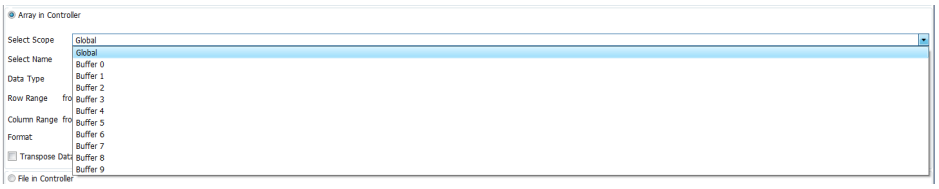
> Exporting an ASCPL+ array (or matrix) variable

> Exporting a user-defined array (or matrix) variable

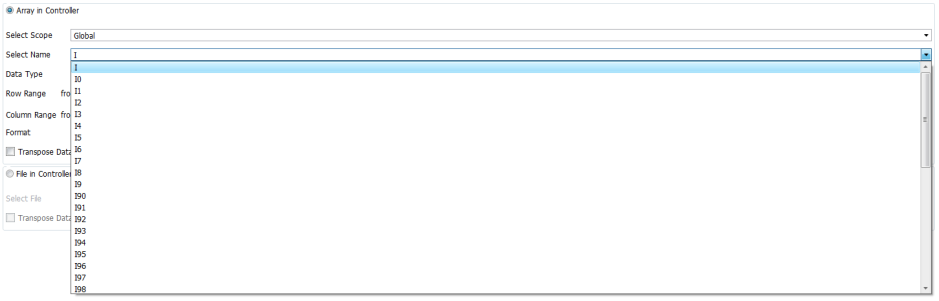
1. For exporting an ASCPL+ array:

> Select **Array in Controller**.

> You can select from Global variables, or narrow the selection of variables to a particular buffer by selecting the buffer from the **Select Scope** dropdown list:

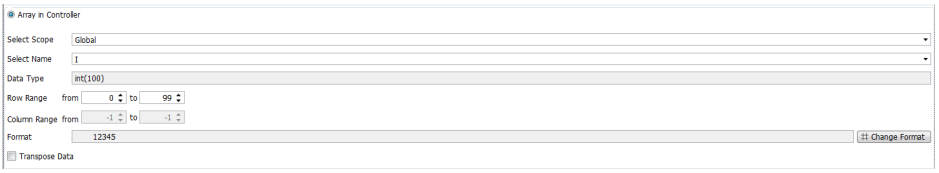



> Select the variable name from the dropdown list.

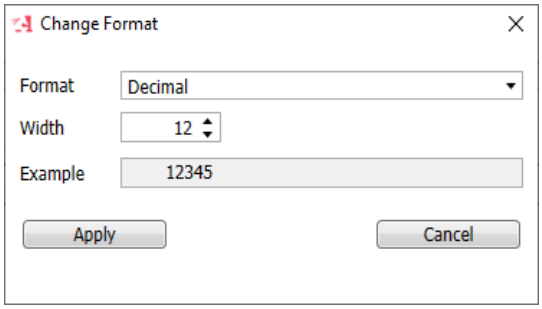



Once you have entered the variable name, the data type is automatically displayed.

> You have the option of selecting only a portion of the array by entering the range using **Row Range** and **ColumnRange**, as appropriate.



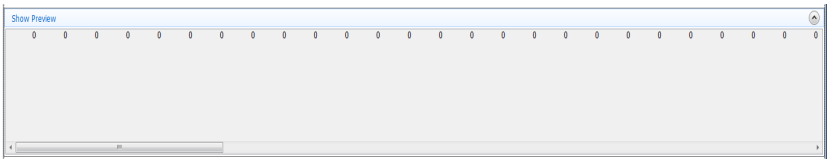

> The current format of the output data is displayed in the Format field. You have the option to change output format by clicking **Change Format**.




> Select the required notational format from the **Format** dropdown list. Where needed, change the **Width** and/or **Precision**. Click **Apply**.

You can transpose the array elements by selecting **Transpose**, for example, if a matrix has m rows and n columns, selecting Transpose causes it to be saved as a matrix having n rows and m columns.

You also have the option of viewing the current values of the array elements by clicking **Show Preview**.

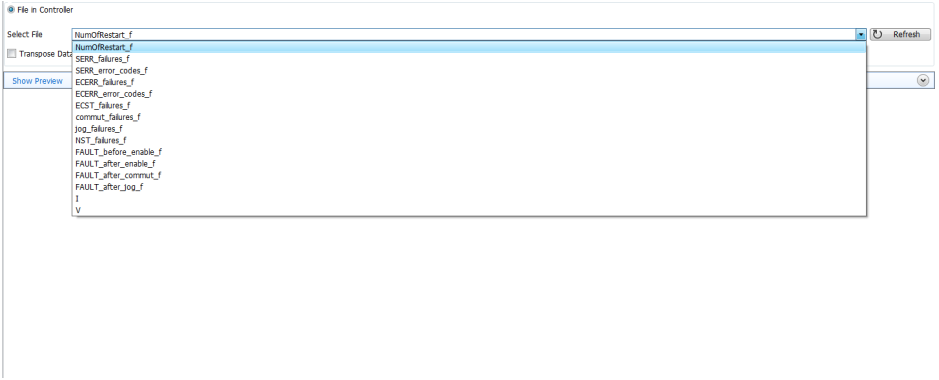



> Click **Next** to start the exporting process.

2. For exporting a file in the controller's flash:

> Select **File in Controller**.

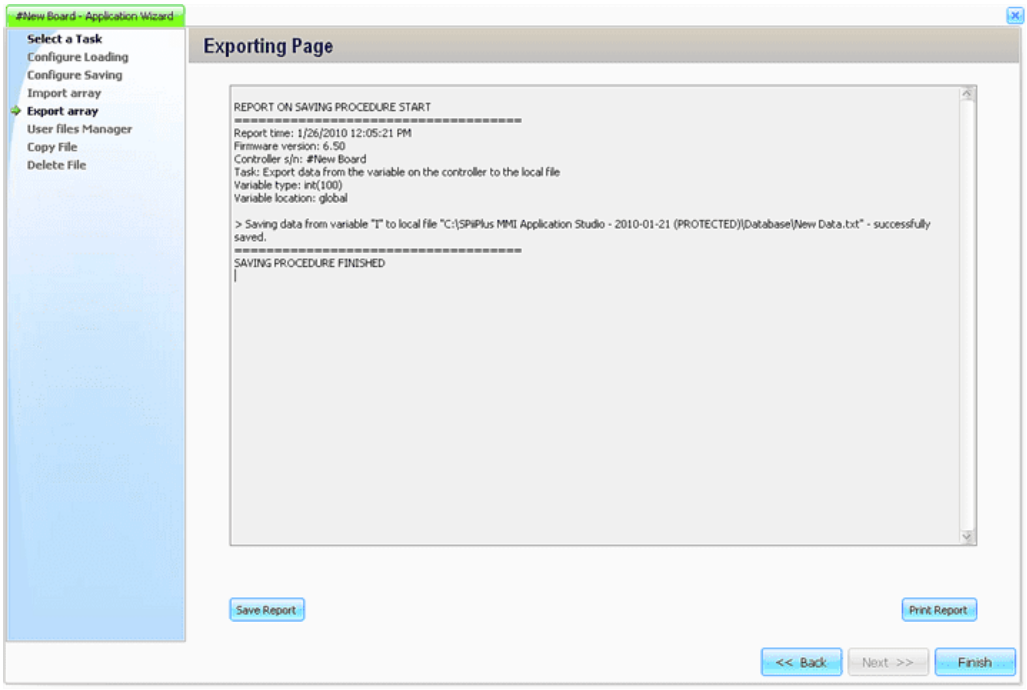
> Select the required file from the dropdown list.



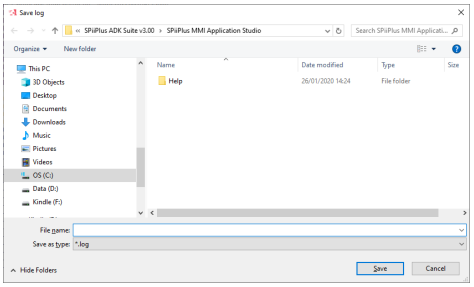

You can transpose the array by clicking **Transpose.**
You can preview the data by clicking **Show Preview.**

> Click **Next** to start the exporting process.

The Exporting window is displayed with a running log of the process:




3. You have the option of saving the log to a file by clicking **Save Report** which opens a dialog window:




4. Select a directory and enter a filename. Click **Save**.

The file extension for all reports is **.log**.

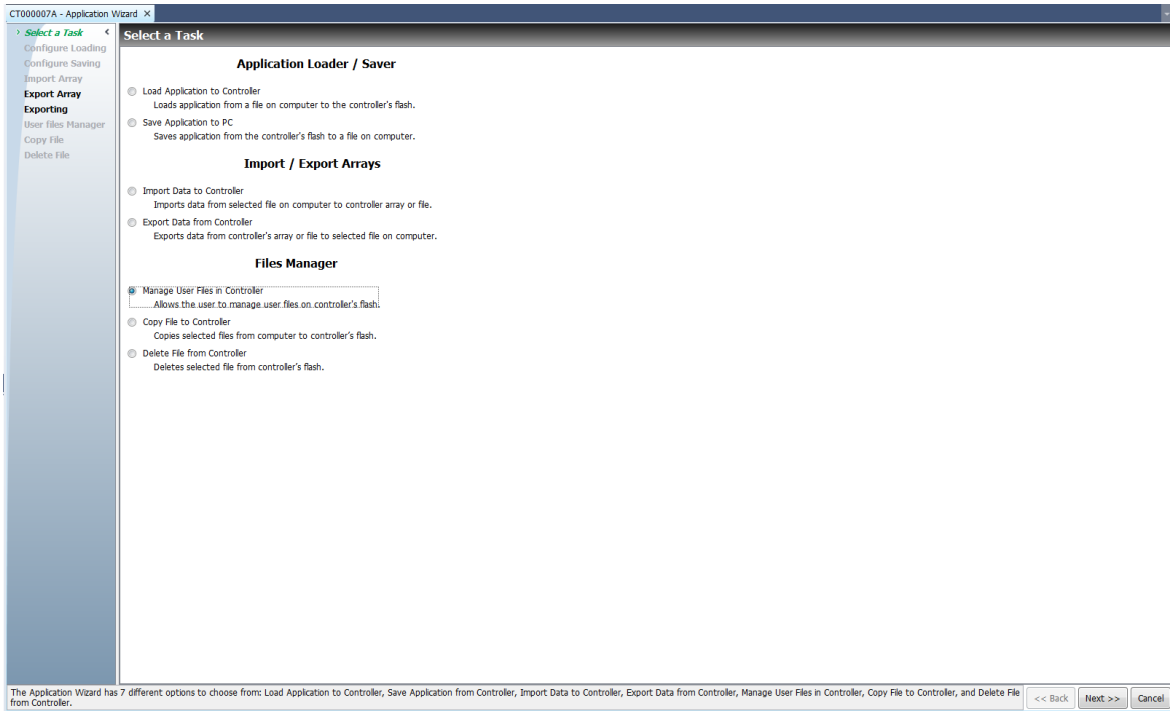
5. To print the report click **Print Report**.
6. Click **Finish** to exit the Application Wizard.

### *4.2.6 User File Manager*

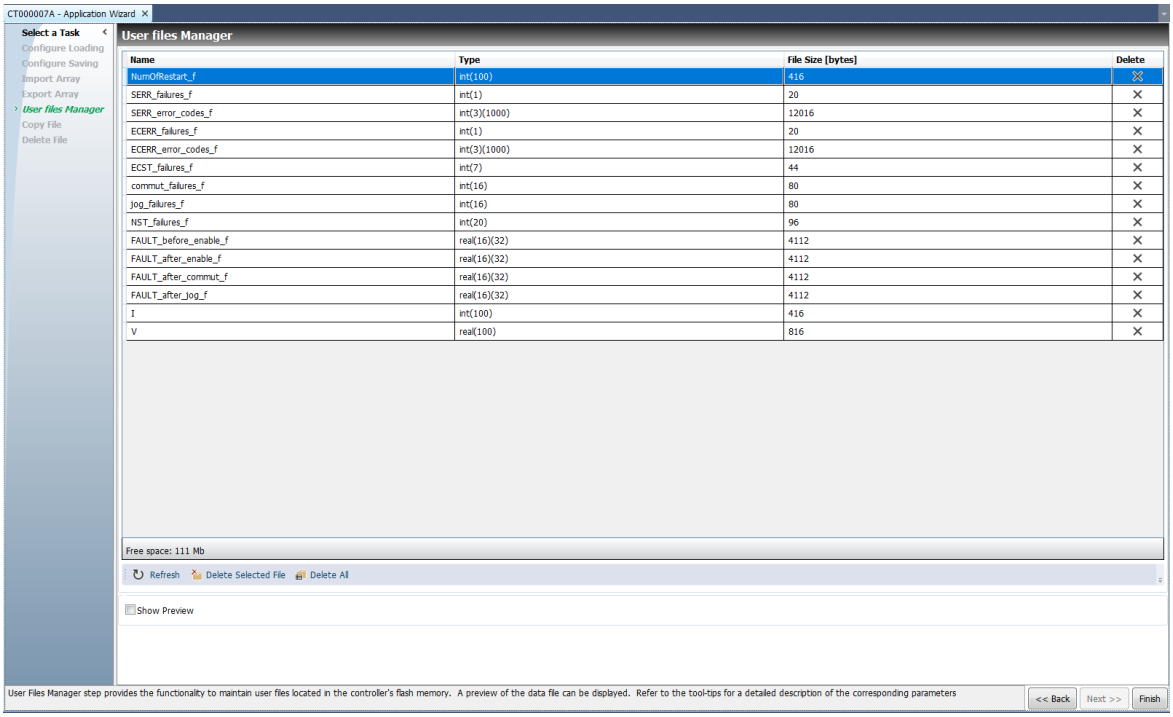
User File Manager is used:

> To view the list of the user files stored in the controller flash
> To preview the contents of the user file
> To delete some or all user files

1. Select **Manage User Files in Controller** from Select a Task.




2. Click **Next**. The Manage User Files in Controller window is displayed:



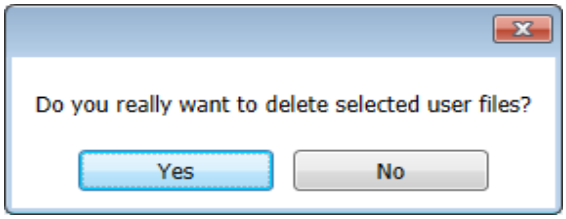

3. You select the required file(s) either by pressing **Ctrl** and clicking the files for selecting individual files, or select a file and press **Shift** to select a range of files.

You can view the contents of a file by selecting the file and then clicking **Show Preview**.

4. To delete selected files, click **Delete Selected**. Or click **Delete All** to remove all of the files in the controller flash.

You can also delete a file by clicking ✕ under the delete column in the file's row.

5. You are prompted by:



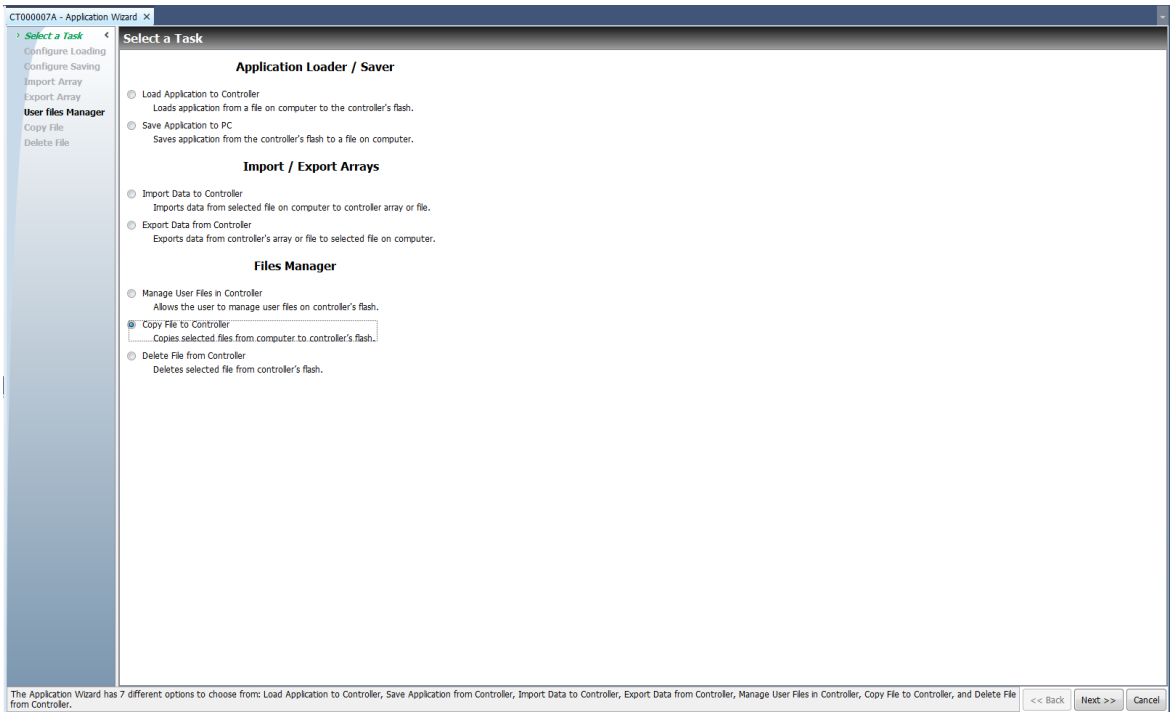

Click **Yes** to delete the file.

6. Click **Finish** to exit the Application Wizard.

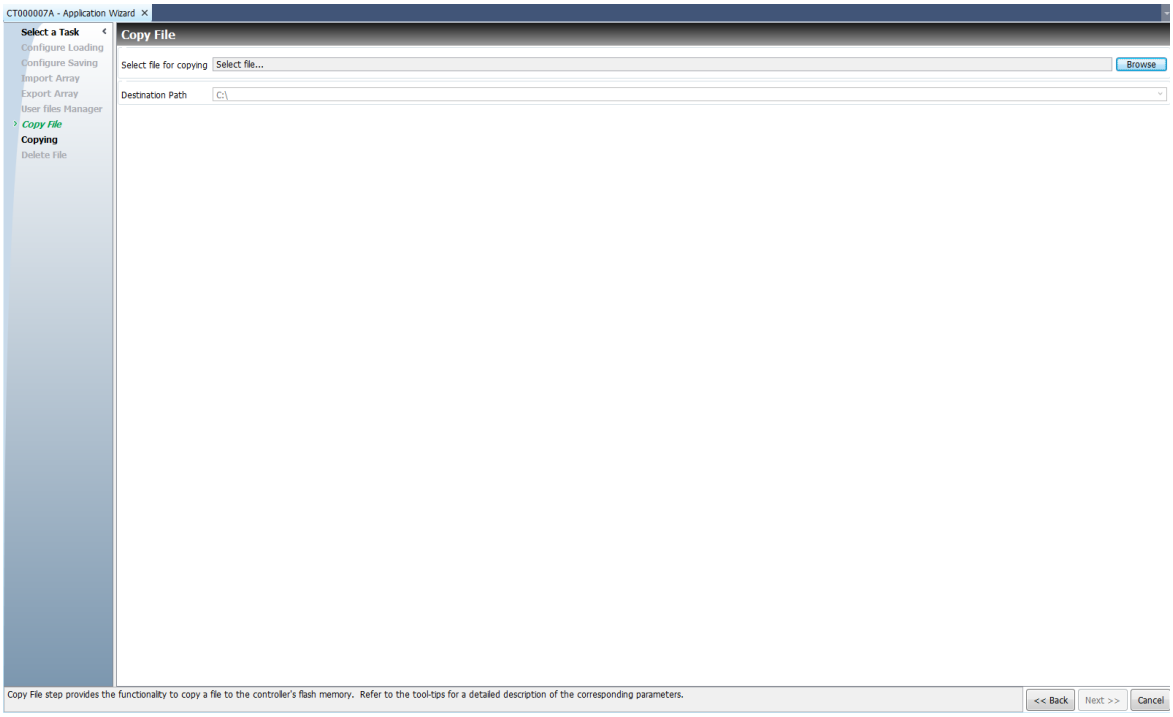
### *4.2.7 Copy File to Controller*

Copy File to Controller is used for copying files from the hard disk to the controller.

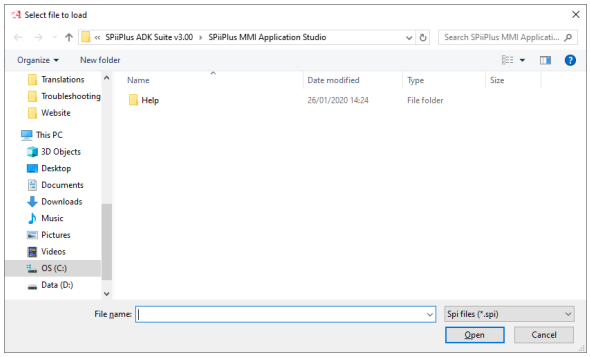
1. Select **Copy File to Controller** from Select a Task.



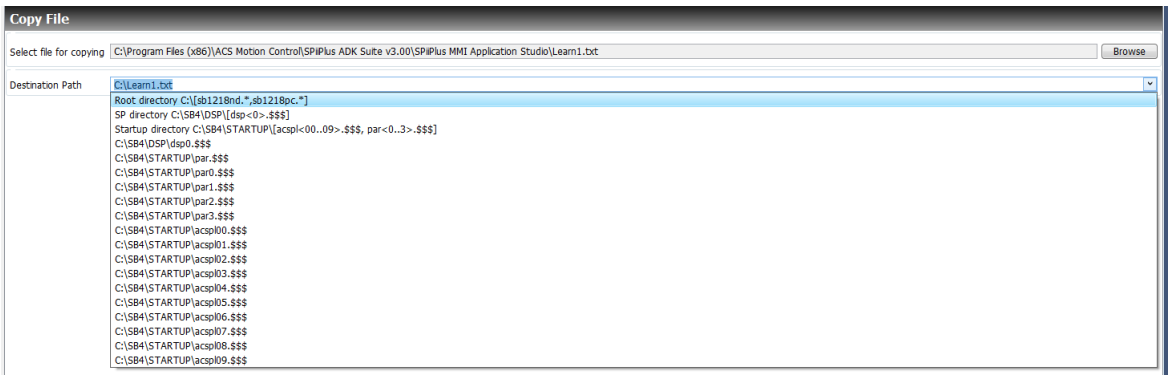

2. Click **Next**. The Copy File to Controller window is displayed.



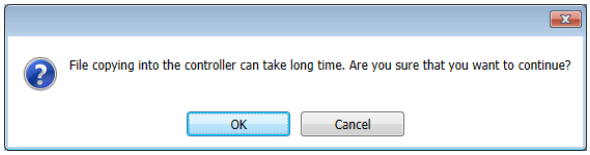

3. Click **Browse** and select the file for copying from the list that is displayed. Then click **Open**.



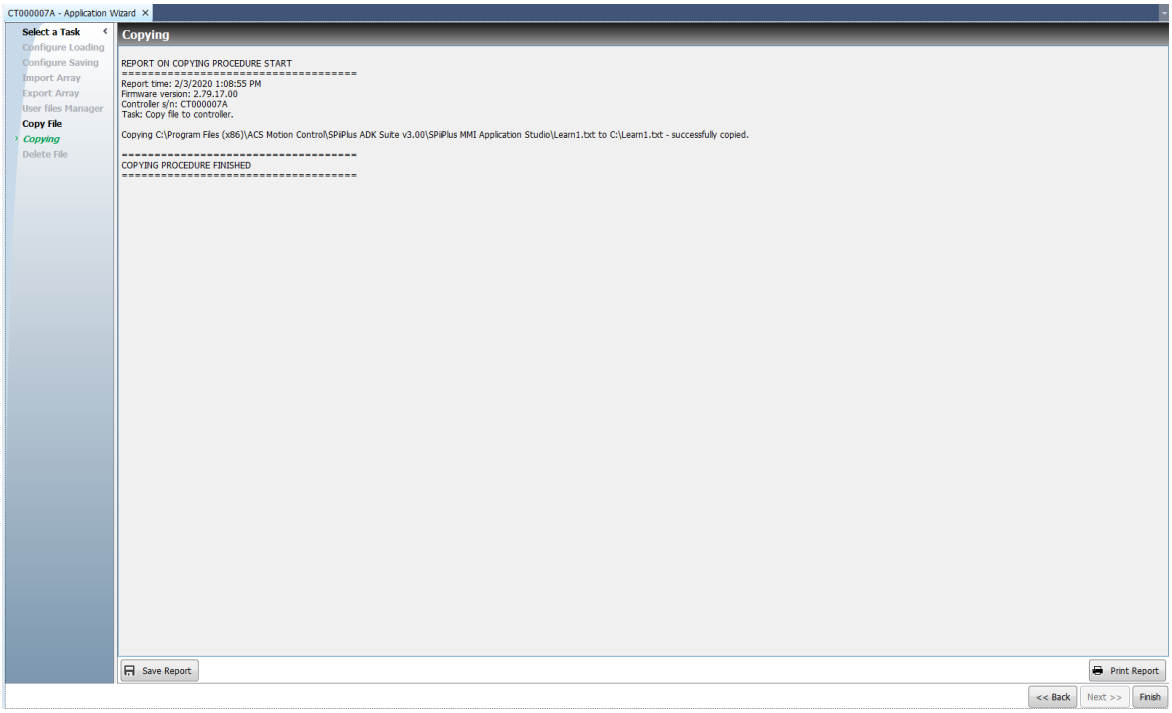

4. Select the controller's path from the Destination Path dropdown list.



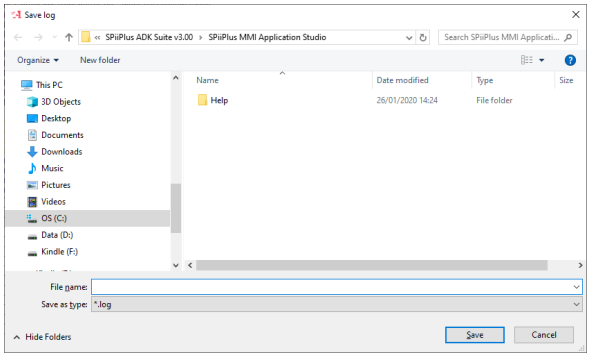

5. Click **Next** to start the copying process. You are prompted by:




6. Click **OK** to start the copying process and wait for the conclusion of the process as indicated in the Copying window:




7. You have the option of saving the log to a file by clicking **Save Report** which opens a dialog window:




8. Select a directory and enter a filename. Click **Save**.

The file extension for all reports is **.log**.

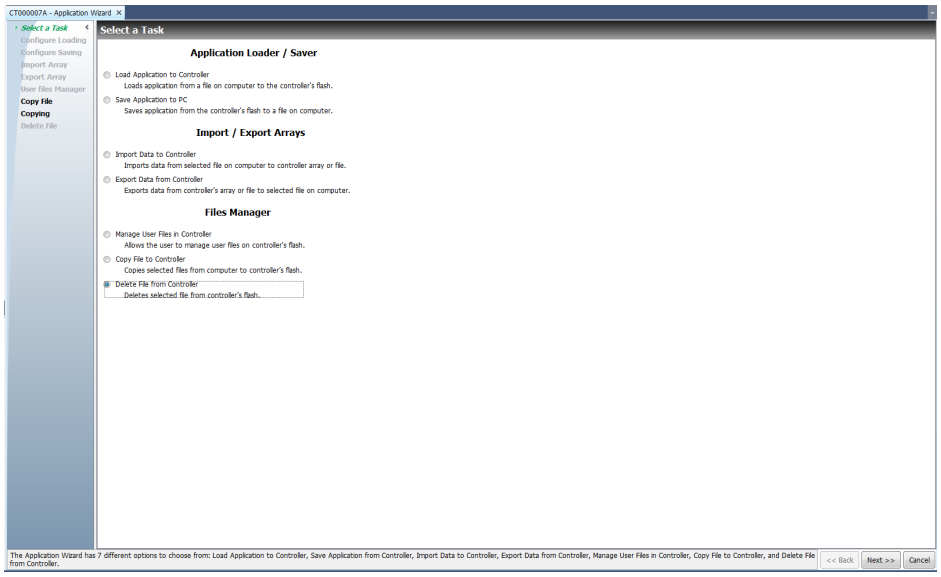
9. To print the report click **Print Report**.

10. Click **Finish** to exit the Application Wizard.

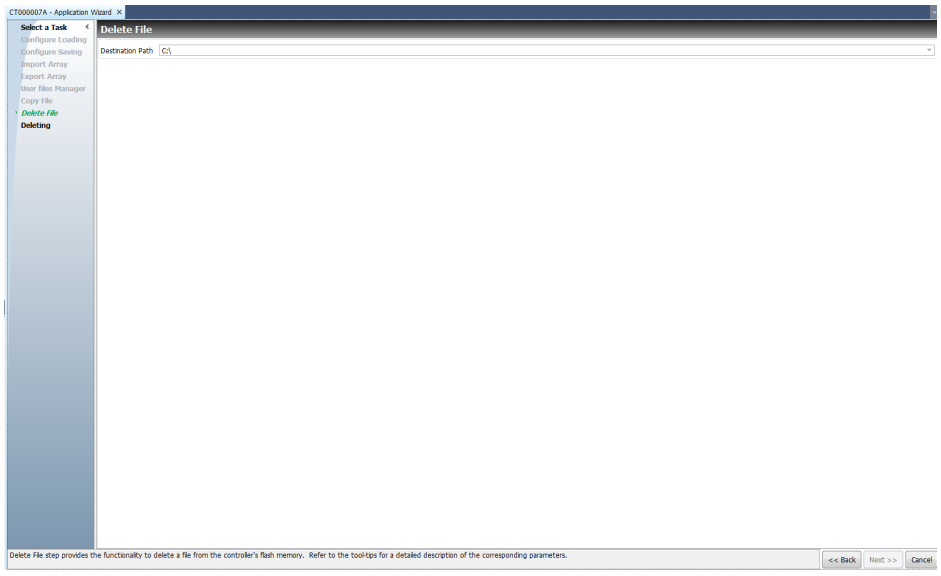
### *4.2.8 Delete File from Controller*

Delete File from Controller is used for deleting a file from the controller.

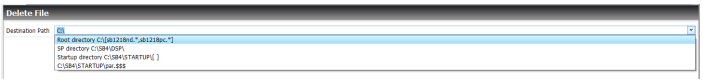
1. Select **Delete File** from Controller from Select a Task.



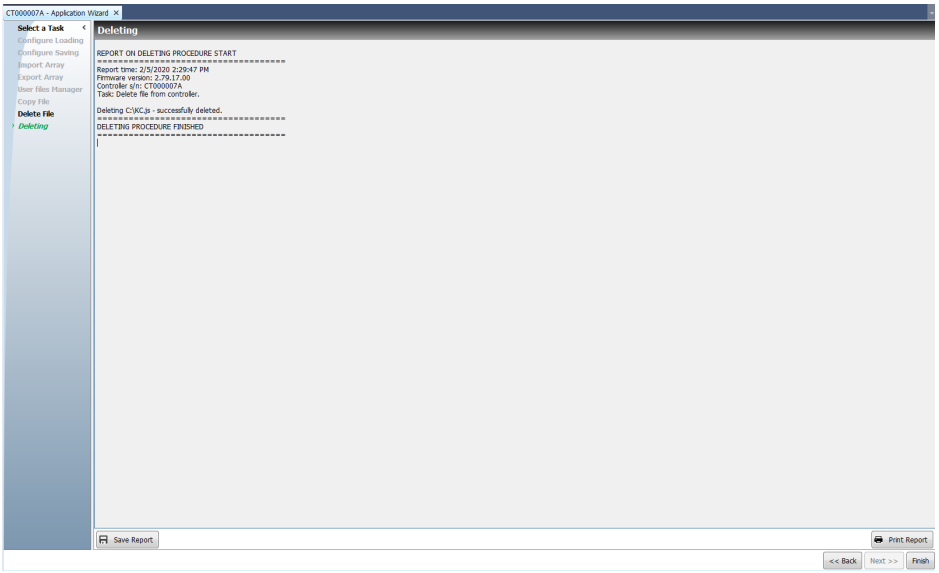

2. Click **Next**. The Delete File from Controller window is displayed.



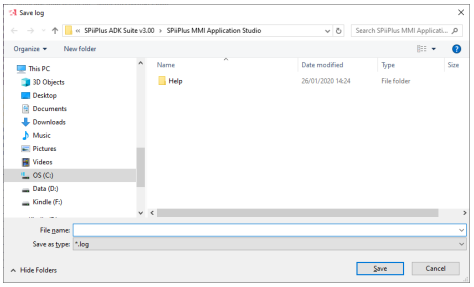

3. Select the **Destination Path** of the file you want to delete from the dropdown list.




4. Type the name of the file.
5. Click **Next** to start the deleting process. When the process is complete, the deleting log is displayed.




6. You have the option of saving the log to a file by clicking **Save Report** which opens a dialog window:




7. Select a directory and enter a filename. Click **Save**.

The file extension for all reports is **.log**.

8. To print the report click **Print Report**.
9. Click **Finish** to exit the Application Wizard.

## 4.3 Program Manager

**Program Manager** is your principal tool for programming your SPiiPlus Motion Controller. Using **Program Manager** you can:

> Load your programs into the controller's buffers.
> Edit the programs in the buffers.
> Run the programs.
> Debug the programs.

Once your programs are running properly, it is highly recommended that you save the contents of the buffers to the controller's flash memory (see Saving to Flash). In the event of a loss of electricity, your programs will be restored to the controller's buffers upon restart. Not doing this will result in the loss of your programs.

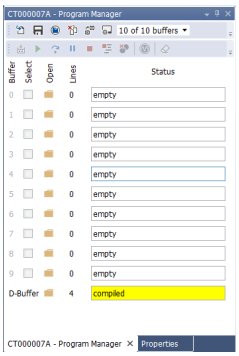
### 4.3.1 Activating Program Manager

By default, when you initially enter SPiiPlus MMI Application Studio, **Program Manager** is automatically displayed. If you have closed it, you can reactivate it by either clicking it in the Workspace Tree or by:

1. From the Toolbox click **Application Development** (Application Development) to display the Application Development tools.
2. Click **Program Manager** in the Application Development list of tools.

You can also activate the **Program Manager** using the right-click **Add Component** option of the Workspace Tree - see Controller Right-Click Options .

The Program Manager window is displayed.

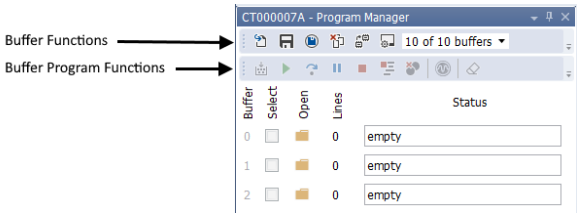



### 4.3.2 Program Manager Window Overview

The **Program Manager** window has 5 columns:

- Buffer - the number of the buffer
- Select - indicates if the buffer has been selected
- Open - indicates if the buffer has been opened
- Lines - gives the number of program lines in the buffer
- Status - provides the status of the buffer

There are also two toolbars:




- Buffer Functions - these functions are associated with various buffer actions.
- Buffer Program Functions - these functions are associated with various buffer program actions.

### *4.3.2.1 Select*

The **Select** column indicates:

| | |
|---|---|
| | **Selected** - the buffer has been selected. |
| | **Not selected** - the buffer has not been selected. |
| | **Unused** - the buffer is not being used. |

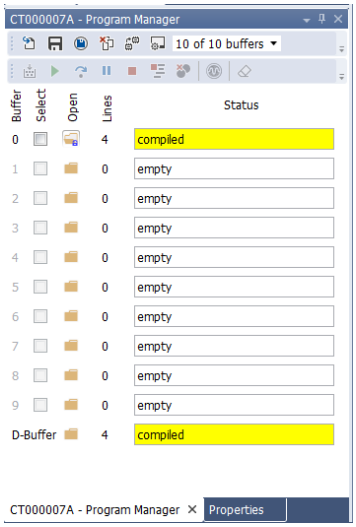
You select a buffer by clicking the icon. You can select any and all buffers with the exception of Unused buffers.

### *4.3.2.2 Open*

The icons in the **Open** column indicate if a buffer is open or not:

| | |
|---|---|
| | Buffer is open. |
| | Buffer is closed. |

They also indicate if the buffer is in the Protected mode, for example:




There are two icons associated with the Protected mode.

| | |
|---|---|
|  | **Hidden** - the program in the buffer cannot be viewed. |
|  | **Read-Only** - the program in the buffer can be viewed, but cannot be changed. |

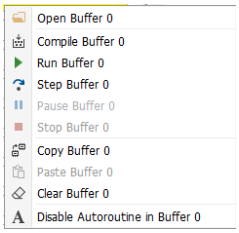
### *4.3.2.3 Status*

The Buffer Status column displays the current status of the buffer. There are four indications associated with the Status:

| | |
|---|---|
|  | The buffer is empty. |
|  | The program in the buffer has been successfully compiled but is not running. |
|  | The program in the buffer is running (the current line in the program that is being executed is displayed as the program runs). |
|  | An error has been detected in the program in the buffer (the line number shows the first error to be detected is displayed along with an error message). |

**Status Right-Click Menu**

Right-clicking the Status displays the following menu:




Clicking one of the options activates it. Your options are:

| | |
|---|---|
| | **Open Buffer** - opens the buffer. |
| | **Compile Buffer -** compiles the program in the buffer**.** |
| | **Run Buffer -** runs the program in the selected buffer**.**<br>When you click this option, the motors start to move. Also, some of the limits and safety features may be temporarily disabled. To avoid personal injury or damage to the equipment, check the following before clicking the button:<br>> Ensure that nothing (people, cables, or other obstacles) is in the path of the motor or objects connected to the motor.<br>> Ensure that the motor is securely anchored and that proper safety barriers, stops, and limits are installed. |
| | **Step Buffer -** executes the command where the cursor is currently sitting. Once executed, the cursor moves down one line**.** |
| | **Pause Buffer -** temporarily halts program running in the buffer. The program can be restarted by**Run Buffer.** |
| | **Stop Buffer -** halts the program running in the buffer. The program can be rerun by**Run Buffer.** |
| | **Copy Buffer** - puts contents of the buffer into the clipboard so they can be copied into another buffer. |
| | **Paste Buffer** - copies contents in the clipboard into the buffer. |
| | **Clear Buffer** - removes the contents from the buffer |
| A | **Disable Autorountine** - disables any autoroutine that is in the buffer. When selected it changes to **Enable Autoroutine in Buffer**, which you can select to reenable the autoroutine in the buffer. |

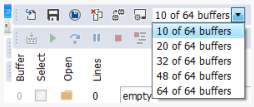
The D-Buffer has a different right-click menu:


Your options are:

| | |
|---|---|
| | **Open D-Buffer** - opens the D-Buffer. |
| | **Apply changes -** applies any changes you have made in the D-Buffer**.** |

#### *4.3.2.4 Buffer Functions Toolbar*

The Buffer Functions toolbar contains:

| | |
|---|---|
| | **Load from File** - loads the contents of a program file that has been previously saved into the buffer. |
| | **Save to File** - saves the contents of a buffer to a file on the computer. |
| | **Clear Buffer** - deletes the contents of a buffer. |
| | **Copy Buffer** - copies the contents of a buffer to another buffer. |
| | **Program Buffer Parameters** - sets the **PRATE** and **ONRATE** buffer parameters. |
| 10 of 10 buffers | **Number of Buffers Selector** - select the number of buffers to be displayed:<br>  <br>The total number of buffers that you can select depends on the controller. Once you have selected the number, it is stored as part of the Workspace data. |



#### *4.3.2.5 Buffer Program Functions Toolbar*

The Buffer Program Functions toolbar contains:

| | |
|---|---|
| | **Compile Buffer(s) -** compiles the program in the selected buffers |
| | **Run Buffer(s) -** runs the program in the selected buffers. |

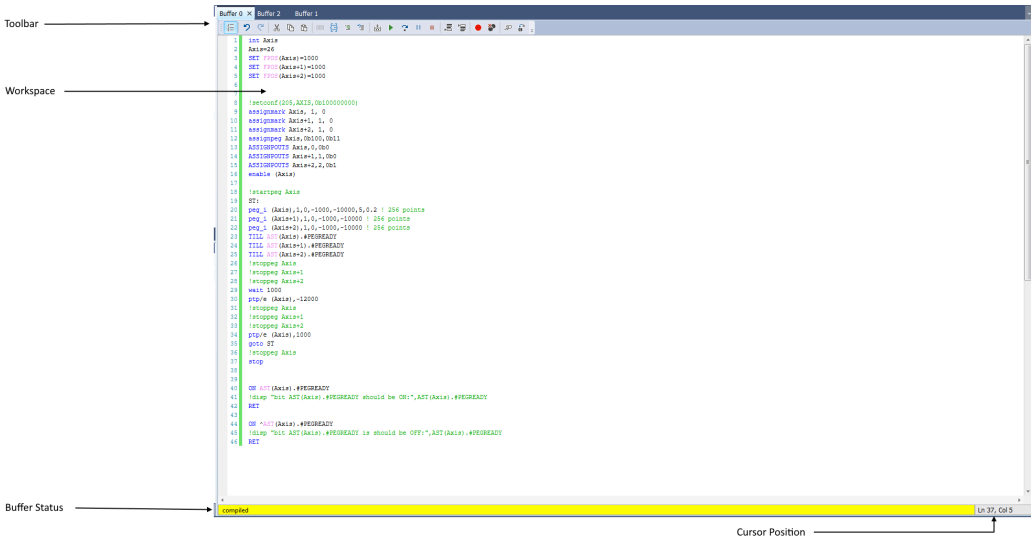
| | |
|---|---|
| | **Warning**<br>When you click this button, the motors start to move. Also, some of the limits and safety features may be temporarily disabled. To avoid personal injury or damage to the equipment, check the following before clicking the button:<br>• Ensure that nothing (people, cables, or other obstacles) is in the path of the motor or objects connected to the motor.<br>• Ensure that the motor is securely anchored and that proper safety barriers, stops, and limits are installed. |
|  | **Step Buffer(s) -** executes the command where the cursor is currently sitting for the selected buffers. Once executed, the cursor moves down one line**.** |
|  | **Pause Buffer(s) -** temporarily halts programs running in selected buffers. The program(s) can be restarted by clicking **Run Buffer.** |
|  | **Stop Buffer(s) -** halts program(s) running in the selected buffers. The program(s) can be rerun by clicking **Run Buffer(s).** |
|  | **Stop All Buffer(s) -** halts all running programs. The programs can be rerun by clicking **Run Buffer(s).** |
|  | **Remove All Breakpoints in Buffer(s) -** removes all breakpoints that have been set in the selected buffers. |
|  | **Trigger Scope -** triggers the display of the signals from the running program. |
|  | **Clear Buffer(s) -** removes the contents from the selected buffers. |

### *4.3.3 Programming a Buffer*

Clicking **Open Buffer** () opens the Buffer Editor window in the Workspace which enables you to:

> Enter a program into the buffer manually.
> Edit an existing program (if it is not Protected).
> View a program in the buffer (if it is not Hidden).
> Run the program and watch the progress and status of each program command.
> Debug the program (steps, breakpoints, start from line).

## Buffer Editor Window




## Toolbar

The Buffer Editor toolbar contains the following:

| | |
|---|---|
|  | **Line Numbers** - toggles the display of line numbers on and off. |
|  | **Undo** - undoes any changes made and restores the previous contents. Same as pressing **Ctrl+Z**. |
|  | **Redo** - reverses preceding undo actions. Same as pressing **Ctrl+Z**. |
|  | **Cut** - deletes the selected contents and places it in the clipboard. Same as pressing **Ctrl+X**. |
|  | **Copy** - places the selected contents into the clipboard. Same as pressing **Ctrl+C**. |
|  | **Paste** - pastes the contents of the clipboard into the buffer at the cursor position. Same as pressing **Ctrl+V.** |
|  | **PARAMETER INFO** - displays function parameter info. Same as pressing **Ctrl+Shift+Space** |
|  | **FORMAT** - formats the contents of the buffer. Same as pressing **Ctrl+I** |
|  | **COMMENT** - formats the contents of the selected lines. Same as pressing **Ctrl+R** |

| | |
|---|---|
| | **UNCOMMENT** - removes commenting of the selected lines. Same as pressing **Ctrl+T** |
| | **Compile** - compiles the program in the buffer. Same as pressing **Ctrl+F5.** |
| | **RUN** runs the program in the buffer. Same as pressing **F5**.<br>  When you click this option, the motors start to move. Also, some of the limits and safety features may be temporarily disabled. To avoid personal injury or damage to the equipment, check the following before clicking the button:<br>> Ensure that nothing (people, cables, or other obstacles) is in the path of the motor or objects connected to the motor.<br>> Ensure that the motor is securely anchored and that proper safety barriers, stops, and limits are installed. |
| | **Step -** executes the command where the cursor is currently sitting. Once executed, the cursor moves down one line. Same as pressing **F9.** |
| | **Pause -** temporarily halts a running program. The program is restarted by clicking **Run.** Same as pressing **Ctrl+P.** |
| | **Stop -** halts a running program. Same as pressing **Shift+F5.** |
| | **Run to Cursor -** executes the program from the start of the program up to the line where the cursor sits. |
| | **Start From -** executes the program from the line where the cursor is sitting**.** |
| | **Set/Remove Breakpoint** - sets a breakpoint at the line where the cursor is sitting. If there is a breakpoint there, it removes the breakpoint. Same as pressing **Ctrl+B.** |
| | **Remove All Breakpoints -** removes all breakpoints that have been set**.** |
| | **Find -** activates a search for a user-entered string. Same as pressing **Ctrl+F.** |
| | **Find & Replace -** activates a search for a user-entered string, and replaces it with another user-entered string**.** |

### Workspace

Used for typing in the program commands.

**Status Line**

The Status Line provides the status of the program in the buffer. There are four colors associated with the Status:

- White - the buffer is empty.
- Yellow - the program in the buffer has been successfully compiled but is not running.
- Green - the program in the buffer is running (the current line in the program that is being executed is displayed as the program runs).
- Red - an error has been detected in the program in the buffer (the error number, line number, and a brief description of the error of the first error to be detected is displayed).

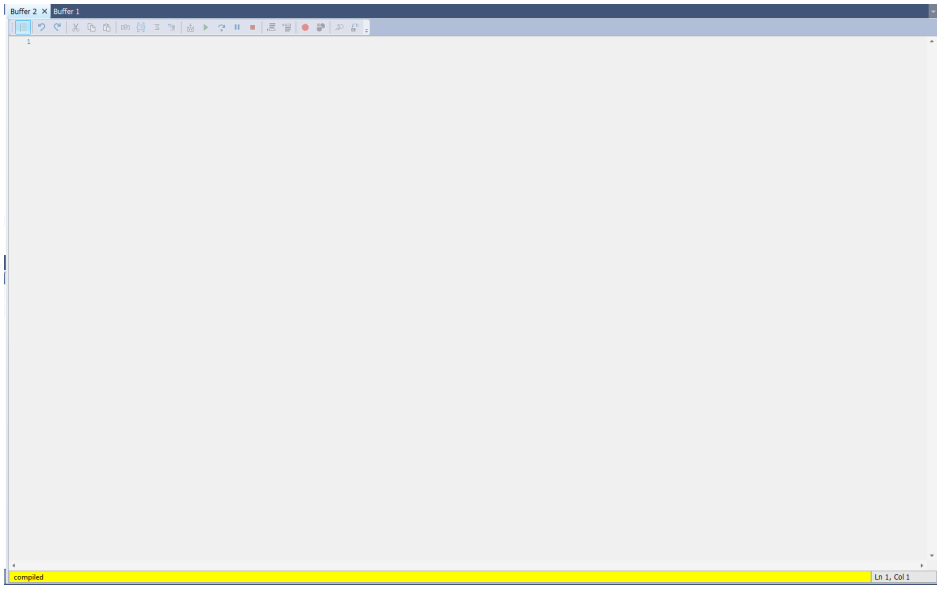
**Cursor Position**

Informs you of exactly where in the program the cursor sits.

### *4.3.3.1 Entering a Program*

To enter a new program into a Buffer or modify an existing program:

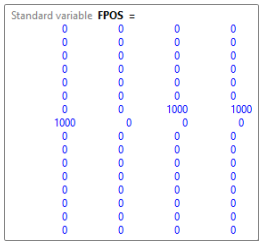
1. In **Program Manager** click **Open Buffer** ( ) of the buffer (or click its Status Line). This opens the Buffer Editor Window.




2. Type in your program.

You can view the value of a variable in the program by letting the mouse pointer linger on the variable. Variables of any type - ACSPL+, user-defined, scalar, arrays (the first 30 elements) - can be viewed.

The current value is displayed in popup window by the mouse pointer.




3. Click **Compile** ( ) or press **Ctrl+F5**. Check the Status Line that the program compiled successfully.

Any work made within a buffer window is stored locally within the MMI workspace. It is only written to the controller upon compilation (even if compilation fails).

If there is an error in your program, the Status Line provides the error number, the line number in which the error is detected, and a brief description of the error. Refer to Chapter 6 of the *SPiiPlus Command & Variable Reference Guide* for details of errors.

#### *4.3.3.2 Programming the D-Buffer*

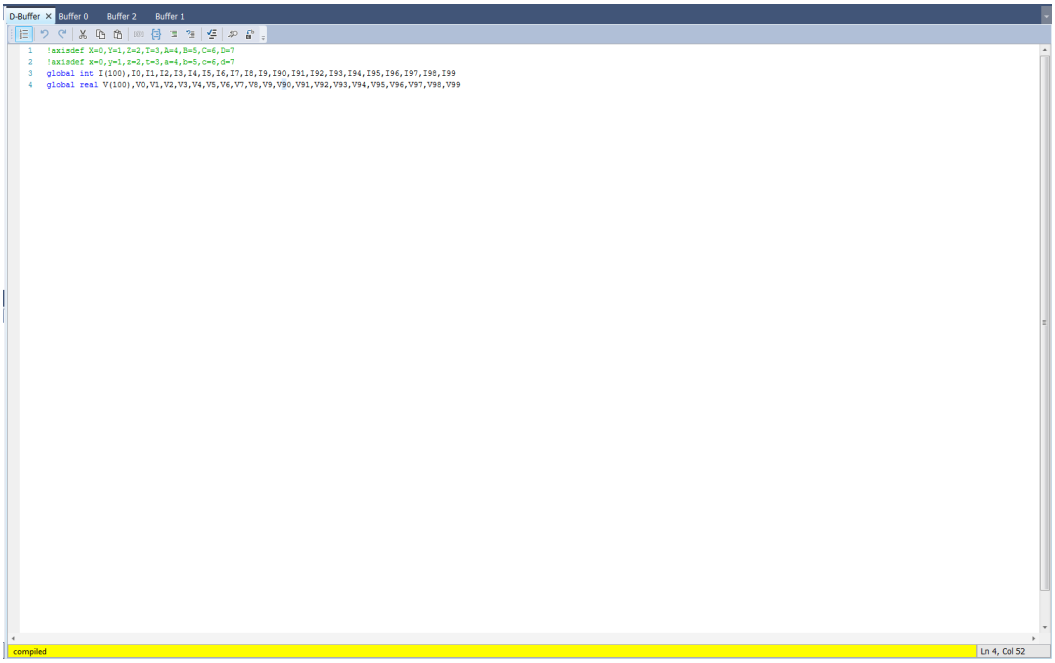
The D-Buffer is an additional special buffer that provides a place for the definitions of axis names and global variables. At start-up, the controller loads the D-Buffer before any other buffers.

Executing programs and autoroutines is not supported in D-Buffer.

Axes and global variables defined in D-Buffer are not required to be defined in other buffers before use. However, such redefinition is not an error, if all the attributes of the definitions are identical.

To program D-Buffer:

1. In **Program Manager** click **Open Buffer** ( ) of D-Buffer (or click its Status Line). The D-Buffer is displayed:




D-Buffer differs from other buffers in that it contains default contents. Other buffers are empty, by default, but the default contents of D-Buffer contain a set of definitions that provides compatibility with previous Firmware versions.

2. Enter your variable definitions.
You have the following functions on the D-Buffer editor toolbar:

| | |
|---|---|
|  | **Line Numbers** - toggles the display of line numbers on and off. |
|  | **Undo** - undoes any changes made and restores the previous contents. Same as pressing **Ctrl+Z**. |
|  | **Redo** - reverses preceding undo actions. Same as pressing **Ctrl+Z**. |
|  | **Cut** - deletes the selected contents and places it in the clipboard. Same as pressing **Ctrl+X**. |
|  | **Copy** - places the selected contents into the clipboard. Same as pressing **Ctrl+C**. |
|  | **Paste** - pastes the contents of the clipboard into the buffer at the cursor position. Same as pressing **Ctrl+V.** |
|  | **PARAMETER INFO** - displays function parameter info. Same as pressing **Ctrl+Shift+Space** |

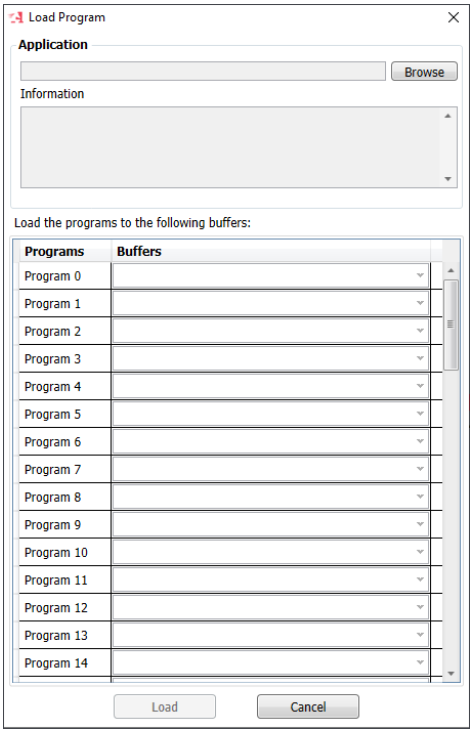
| | |
|---|---|
| | **FORMAT** - formats the contents of the buffer. Same as pressing **Ctrl+I** |
| | **COMMENT** - formats the contents of the selected lines. Same as pressing **Ctrl+R** |
| | **UNCOMMENT** - removes commenting of the selected lines. Same as pressing **Ctrl+T** |
| | **APPLY CHANGES** - writes changes to the current buffer. |
| | **Find -** activates a search for a user-entered string. Same as pressing **Ctrl+F.** |
| | **Find & Replace -** activates a search for a user-entered string, and replaces it with another user-entered string**.** |

3. Click **Apply changes** ( ) and close the window.

#### *4.3.3.3 Loading Programs from a PC File*

To load programs from a PC file:

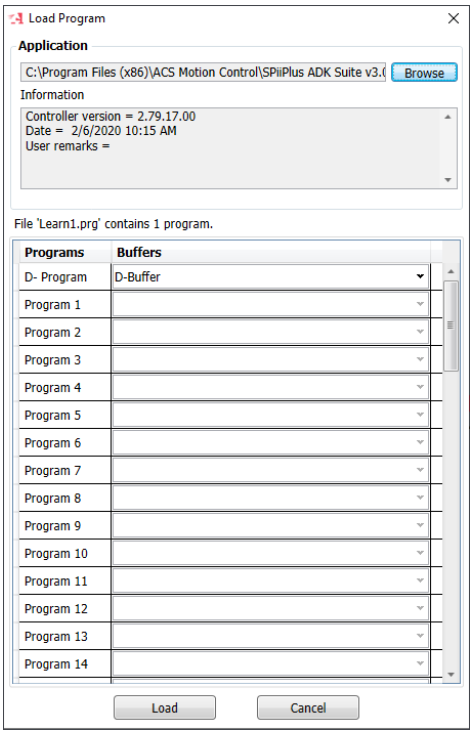
1. In **Program Manager** click **Load from File** ( ), the Load Program dialog window is displayed:



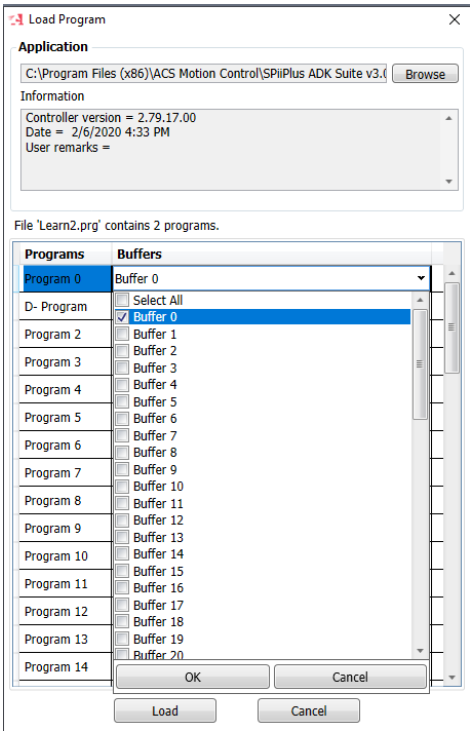

2. Click **Browse** to search for the file (when found, select the file).

Program files have a **prg** file extension.

3. Click **OK**, the Load Program dialog window displays the default buffer location of each program contained in the selected file, for example:




You can reassign the buffer location by clicking the **Buffers** field of the program and selecting the buffer from the list:




You can load the same program into more than one buffer.

For controllers set to the **Protected** mode, if a buffer is tagged with (Hidden buffer) or (Read-Only buffer), you cannot load a program into it.

You can load a program for the D-Buffer only into the D-Buffer.

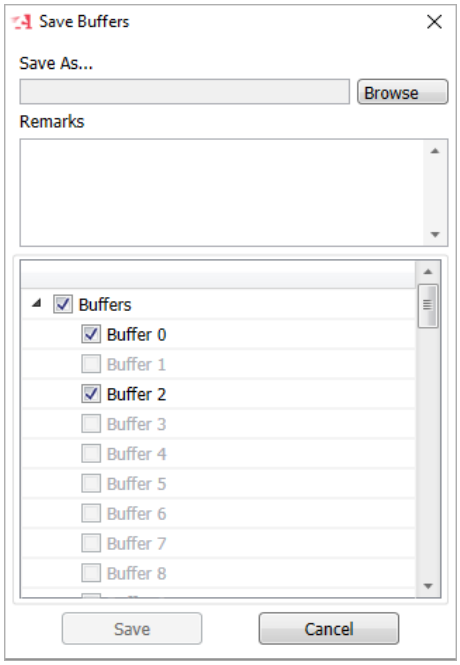
4. Click **OK**.

### *4.3.3.4 Saving Programs to PC File*

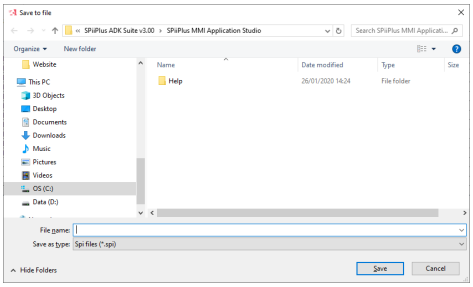
Any program in a buffer window can only be saved after compilation.

To save programs from the buffers to a PC file:

1. In **Program Manager** click **Save to File** ( ), the Save Buffers dialog window is displayed:




2. Click **Browse** to open the browser.



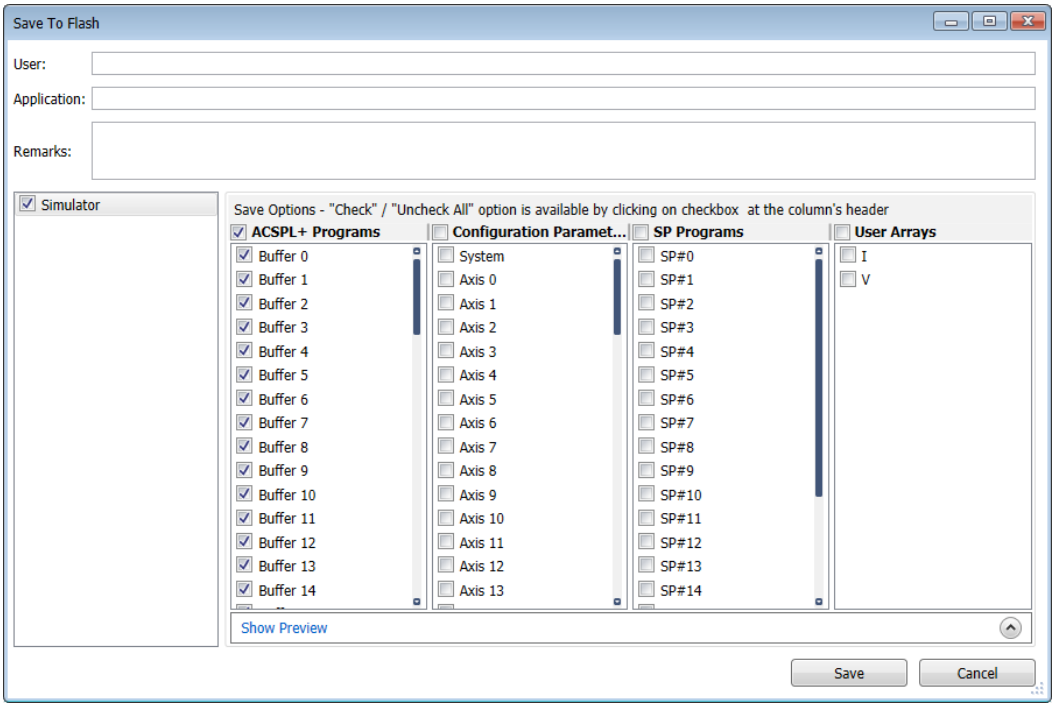
3. Drill down to the directory into within which you want to save your program files, and either select an existing file (for overwriting), or enter a new file in **File name** and click **Save**. You return to the Save Buffers dialog window.
4. Select the buffers you want to save in Buffer Tree.

For controllers set to the **Protected** mode, if a buffer is tagged with (Hidden buffer), you cannot save it to a file.

5. Click **Save**.

### *4.3.4 Saving Programs to Flash Memory*

1. In the **Program Manager**, click **Save to Flash** ( ); the Save to Flash dialog window will open.

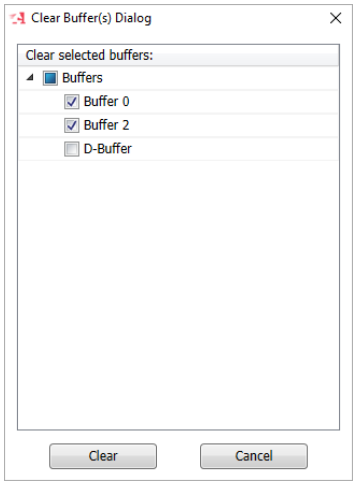



2. Fill in User, Application, and Remark text boxes as appropriate.
3. Click **Save**.

### *4.3.4.1 Clearing Buffers*

You can clear the contents of any or all buffers by:

1. In **Program Manager** click **Clear buffers** (), the Clear Buffers dialog window is displayed:




2. Select buffer(s) to be cleared.

For controllers set to the **Protected** mode, if a buffer is tagged with (Hidden buffer) or (Read-Only buffer), you cannot clear a program from it.

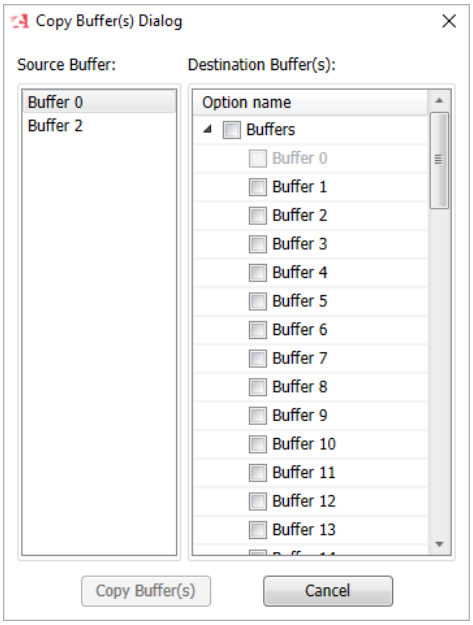
3. Click **Clear**.

You can also make use of the popup menu that appears when you right-click the buffer Status Line (see Status Right-Click Menu).

### *4.3.4.2 Copying Buffers*

To copy programs from some buffer to others:

1. In **Program Manager** click **Copy buffers** (  ), the Copy Buffers dialog window is displayed:




Note that a buffer cannot be copied into itself.

Select the buffer from which to copy.

For controllers set to the **Protected** mode, if a buffer is tagged with  (Hidden buffer), you cannot copy the file. However if the buffer is tagged with  (Read-Only buffer), it can be copied into another buffer so long as the destination buffer is not Read-Only.

2. Select the buffer(s) into which to copy.
3. Click **Copy**.

You can also make use of the popup menu that appears when you right-click the buffer Status Line (see Status Right-Click Menu).

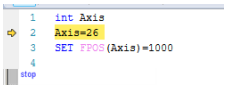
### 4.3.5 Stepping Buffer Program

To aid you in debugging your program or programs you can use the **Program ManagerStep** function to run programs step-by-step. You have the option of using this function on a single buffer or a number of buffers.

#### 4.3.5.1 Stepping Single Program

To run a single buffer program step-by-step:

1. If the buffer is not open, click **Open Buffer** ( ) of the required buffer in the **Program Manager**.
2. In the Buffer Editor Window click **Step Buffer** ( ) or press **F9**.
   Note that the Status Line in the **Program Manager** turns green (indicating that the program is running) and the line number of the command being executed is displayed.
   In addition an arrow appears indicating the line being executed:




   The Status Line in the Buffer Editor Window indicates **Suspend**. If an error is detected, it turns red.
3. Continue clicking **Step Buffer** until you have reached the end of the program.

You can stop the program at any point and reset it by clicking **Stop** ( ) or pressing **Shift+F5**.

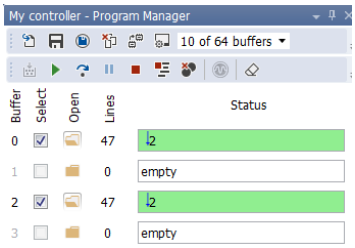
#### 4.3.5.2 Stepping Multiple Programs

To run several buffer programs step-by-step simultaneously:

1. In **Program Manager** select the buffers in the **Select** column.
2. Click **Step Buffer(s)** ( ) .

You can also make use of the popup menu that appears when you right-click the buffer Status Line (see Status Right-Click Menu).

The Status for each selected buffer turns green (indicating that the program is running in them) and the line number of the next command is displayed.




If an error is detected in any of the buffers, the buffer's status turns red and the program line number of the error is displayed.

3. Continue clicking **Step Buffer(s)** until you have reached the end of the programs.

You can stop the programs at any point and reset them by clicking **Stop Buffer(s)** ( ).

### *4.3.6 Setting/Removing Breakpoints*

A breakpoint is an intentional stopping or pausing place in your program, put in place for debugging purposes. More generally, a breakpoint is a means of acquiring knowledge about the buffer program during its execution. During the interruption, you can inspect the test environment (logs, memory, files, etc.) to find out if the program functions as expected.

#### Setting Breakpoints

To set a breakpoint:

1. In **Program Manager** click **Open Buffer** ( ).
2. Scroll down to the line at which you want to set the breakpoint.
3. Click **Set Breakpoint** ( ). A red ball is displayed to the left of the line indicating that a breakpoint has been set.

```
5   SET FPOS(Axis+1)=1000
6   SET FPOS(Axis+2)=1000
7
8
9   !setconf(205,AXIS,0b100000000)
10  assignmark Axis, 1, 0
11  assignmark Axis+1, 1, 0
12  assignmark Axis+2, 1, 0
13  assignpeg Axis,0b100,0b11
14  ASSIGNPOUTS Axis,0,0b0
15  ASSIGNPOUTS Axis+1,1,0b0
16  ASSIGNPOUTS Axis+2,2,0b1
17  enable (Axis)
18
```

You can set as many breakpoints as you want. Once you have set the breakpoints, you can run the program (see Running/Stopping Buffer Programs). It will run up to the line containing the breakpoint, and then pause. When you execute it again, it will run from this line to the next breakpoint.

#### Removing Breakpoints

To remove breakpoint:

4. In **Program Manager** click **Open Buffer** ( ).

5. Scroll down to the line where the breakpoint is set (indicated by the red ball).

Click **Remove Breakpoint** ( ). The red ball indicated that a breakpoint has been set for line is no longer displayed.

You can remove all of the breakpoints by clicking **Remove All Breakpoints in Buffer** ( ).

### *4.3.7 Disable/Enable Autoroutines*

Autoroutines are similar to hardware interrupts. In distinction to routines that must be explicitly executed (by way of the ACSPL+ **CALL** command), the autoroutine is automatically executed when a specific condition is satisfied. The routine interrupts the currently executing program, executes the commands specified in the autoroutine body, and then returns control to the interrupted program.

**To disable autoroutines in single buffer:**

1. Right-click the buffer.
2. Select **Disable Autoroutines in Buffer** ( ) from the popup-menu.

Note that the option changes to **Enable Autoroutines in Buffer** .

**To reenable the autoroutines in the buffer:**

1. Right-click the buffer.
2. Select **Enable Autoroutines in Buffer** from the popup-menu.

### *4.3.8 Setting Program Buffer Parameters*

There are two program buffer parameters:

> PRATE - defines the program execution rate in buffer. The execution rate is PRATE lines per each MPU cycle. Its value ranges between 1 (default) and 10 (maximum).

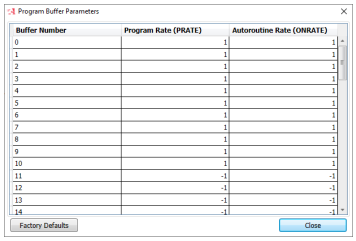
> ONRATE - When an autoroutine executes in the program buffer, the execution rate is ONRATE lines per each MPU cycle. Its value ranges between 1 (default) and 10 (maximum).

**PRATE** is used only if no autoroutine is activated in the buffer. While an autoroutine is executed, **ONRATE** defines execution rate.

For example, if **PRATE** is set to 1 for a buffer, but **ONRATE** is set to 4, the program in the buffer will be executed one line per one controller cycle, and any autoroutine specified in buffer that interrupts the program will be executed four lines per one controller cycle. When the **RET** command that terminates the autoroutine is executed, the controller switches back to the rate of one line per one cycle.

To set the program buffer parameters:

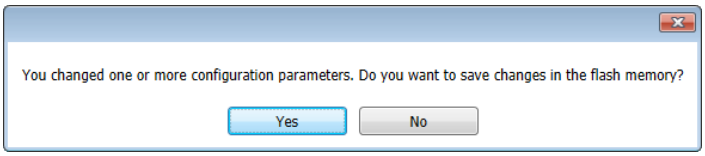
1. In **Program Manager** click **Program Buffer Parameters** ( ). The Program Buffer Parameters window is displayed:



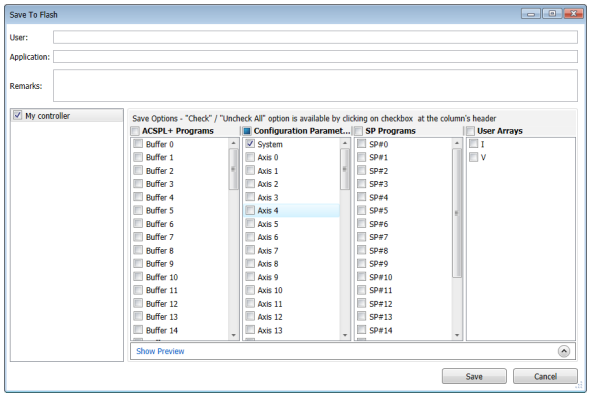

2. For each buffer enter the required **PRATE** and **ONRATE** values, as appropriate, and click **Close**.

To set the value to the default, click **Default**.

If you have changed any of the values, the following prompt appears:




If you want to save the changes to the SPiiPlus Motion Controller flash (recommended), click **Yes**. The Save to Flash window is displayed (note that **System** is selected):




3. Enter:
   - > Your name in the **Name** field (optional).
   - > Application name in the **Application** field (optional).
   - > Appropriate remarks in the **Remarks** field (optional).
4. Click **Save**.

### *4.3.9 Running/Stopping Buffer Programs*

#### Running Buffer Programs

To run the buffer program(s):

1. In **Program Manager** select the buffer (or buffers) in the **Select** column.

2. Click **Run Buffer(s)** ( ).

Run Buffer(s) causes the motors to move. Also, some of the limits and safety features may be temporarily disabled. To avoid personal injury or damage to the equipment, check the following before running the program:

Ensure that nothing (people, cables, or other obstacles) is in the path of the motor or objects connected to the motor.

Ensure that the motor is securely anchored and that proper safety barriers, stops, and or limits are installed.

If you are working in the Buffer Editor, you can use **Run** or press **F5** to run the program.

**Pausing/Stopping Buffer Programs**

To pause programs while running:

In **Program Manager** select the buffers in the **Select** column.

Click **Pause Buffer(s)** ( ).

You can restart the programs by clicking **Run Buffer(s)**

To stop programs:

1. In **Program Manager** select the buffer in the **Select** column
2. Click **Stop Buffer(s)** ( ).

For **Pause Buffer** and **Stop** you can also make use of the popup menu that appears when you right-click the buffer Status Line (see Status Right-Click Menu).

To stop all programs that are running click **Stop All Buffer(s)** ( ).

If you are working in the Buffer Editor, you can use **Pause** or press **Ctrl+P** to pause the program, and **Stop** or press **Shift+F5** to stop the program.

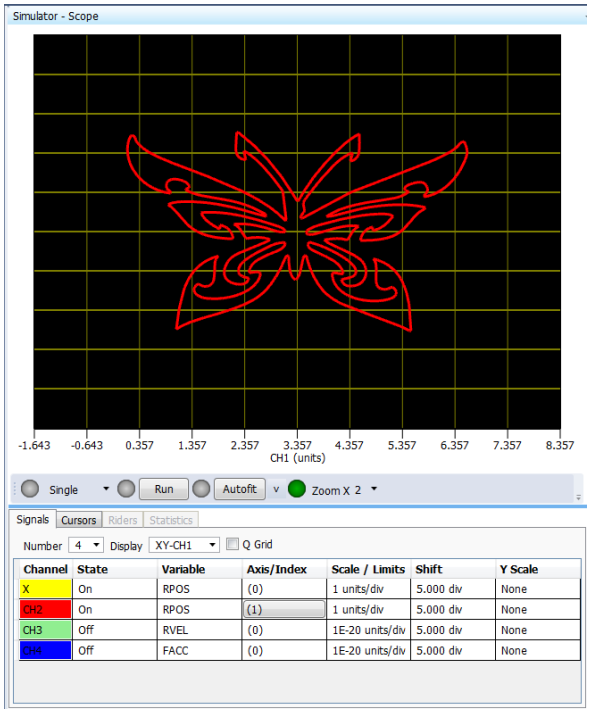
### *4.3.10 Trigger Scope*

Once the programs are running, you can analyze the motion using the scope.

To make use of the Scope when running your programs:

1. Activate the Scope (see Activating the Scope).
2. Set the Scope display to **Single** and **Autofit**.

3. In **Program Manager** click **Trigger Scope** ( ).
4. In the **Select** column select the buffer you want to run and click **Run Buffer(s)** ( ).
   You get a display like this:




## 4.4 Protection Wizard

The controller can be in one of two modes:

> **Protected Mode** - certain restrictions are placed on the programs in the controller's Program buffers. Usually this mode is used to ensure that the programs cannot be edited.

> **Not Protected Mode** - no restrictions are placed on the Program buffers.

The **Protection Wizard** is used for setting and unsetting the Protection Mode of the controller. It leads you through the performance of the following tasks:

> Define Protection

> Update Protection

> Remove Protection

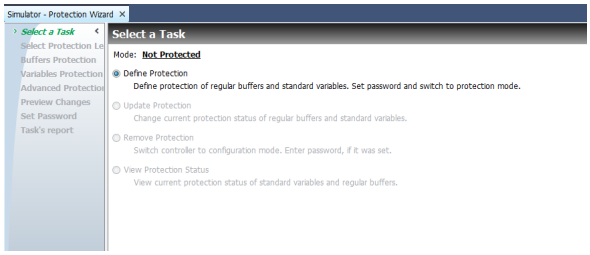
> View Protection

### 4.4.1 Starting Protection Wizard

1. From the Toolbox click **Application Development** ( Application Development ) to display the Application Development tools.
2. Click **Protection Wizard** in the Application Development list of tools.

You can also activate the **Protection Wizard** using the right-click **Add Component** option of the Workspace Tree - see Controller Right-Click Options.

The **Protection Wizard** Task window is displayed in the workspace.




3. Select the task you want to perform by clicking the appropriate button and then click **Next**.

If the controller is in the **Not Protected** mode, the only task available is **Define Protection**.

If the controller is in the **Protected** mode, the tasks available are: **Update Protection**, **Remove Protection**, and **View Protection Status**.

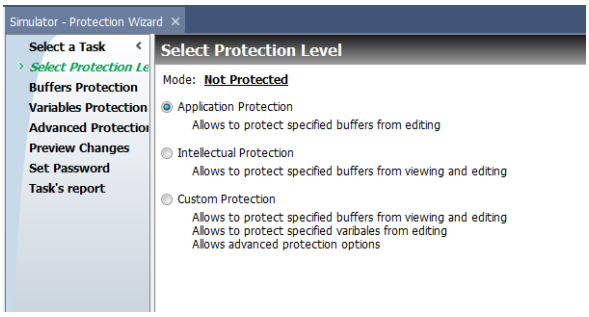
### *4.4.2 Define Protection*

The Define Protection task enables you to put the controller into the Protected Mode and to set certain restrictions on the programs stored in the controller's buffers.

To define Protection:

This task, and this task only, is available if the controller is in the **Not Protected** mode.

1. Since this is the only task available, click **Next**. The Set Protection Level window is displayed.



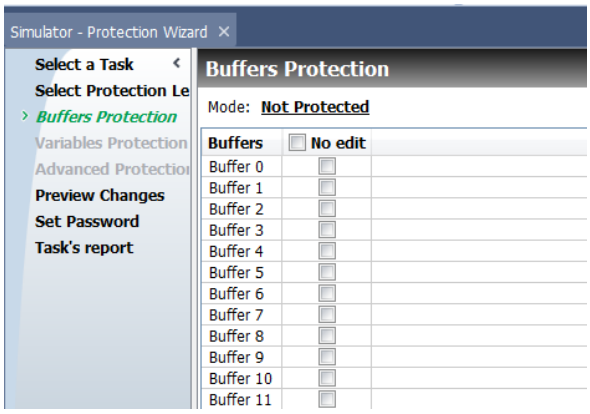

There are three tasks that can be performed from the Select Protection Level window:

> Application Protection

- You perform this task to set the program in the specified buffers to Read Only, that is, it can be viewed but cannot be modified.
- Intellectual Property Protection
- You perform this task to set the program in the specified buffers such that its contents cannot be modified and cannot be seen.
- Custom Protection
- There are three tasks that can be performed from the Custom Protection window:
  - You can set the program in the specified buffers such that its contents cannot be modified and/or cannot be seen.
  - You can set specified variables to Read Only, that is, it can be viewed but cannot be modified.
  - You allow the system to be reconfigured.

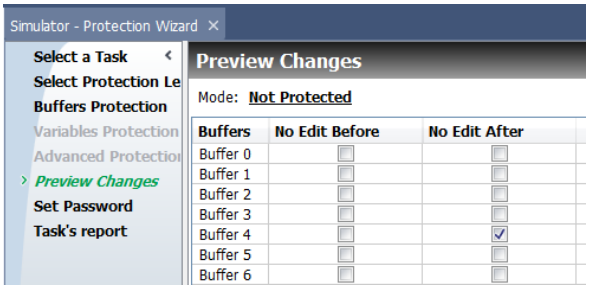
#### *4.4.2.1 Setting Application Protection*

1. To set Application Protection, select the button next to **Application Protection**. Then click **Next**. The Buffers Protection task is displayed.




By selecting the **Prevent from editing** checkbox you set the program in the buffer to Read Only, that is, it can be viewed but cannot be modified.

2. Click **Next**. The Preview Changes window is displayed:

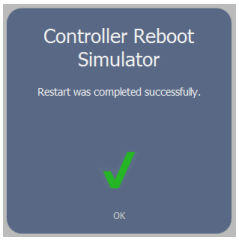



3. Check that the proper Protections are set. If they are, select **Accept Changes**. Click **Next**. The Set Password window is displayed.


4. If you want to set a user password to prevent unauthorized personnel from removing Protection, type it into the **Enter password** and **Confirm password** fields.

A password is not required.

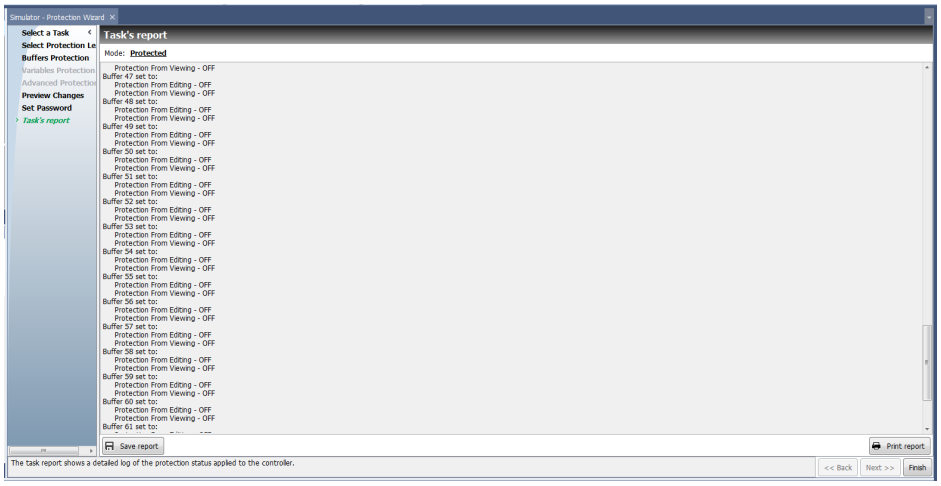
5. Click **Next**. The controller is rebooted. Wait for the notification of a successful restart.



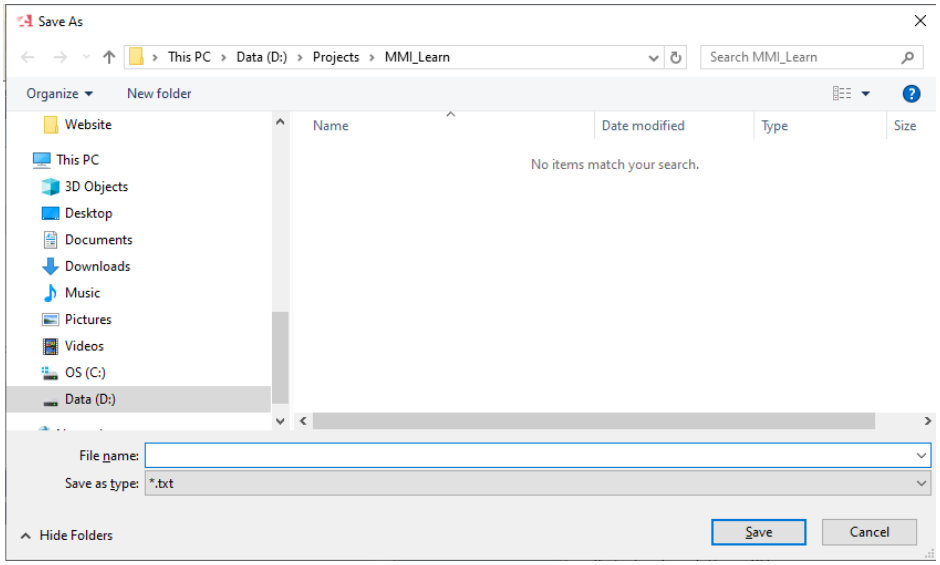

The controller must be restarted, otherwise the protected programs can become corrupted.

6. Click **OK**.

The Report window is displayed.




7. To save the report click **Save report**. The browser window is displayed:



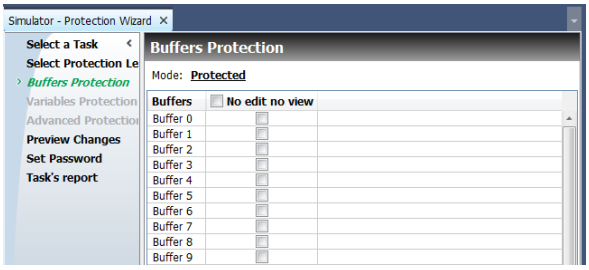

8. Select a directory and enter a filename. Click **Save**.

The file extension for all reports is **.log**.

9. To print the report click **Print report**.
10. Click **Finish** to exit the Protection Wizard.

### *4.4.2.2 Setting Intellectual Property Protection*

1. To set Intellectual Property Protection, select the button next to **Intellectual Property Protection**. Then click **Next**. The Buffers Protection task is displayed.



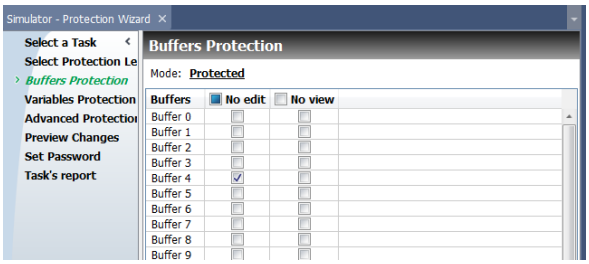

By selecting the **Prevent from editing and viewing** checkbox, you set the program in the specified buffers such that its contents cannot be modified and cannot be seen.

2. Click **Next** to preview changes and follow step 3 through step 10 in Setting Application Protection to complete setting protection and to exit the Protection Wizard.

### *4.4.2.3 Setting Custom Protection*

#### *4.4.2.3.1 Custom buffer protection*

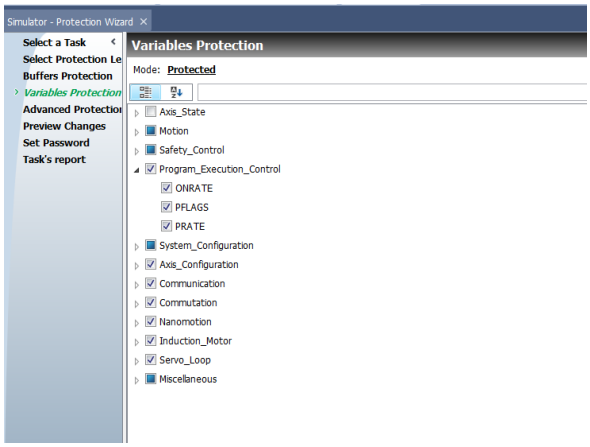
1. To set Custom Protection, select the button next to **Custom Protection**. Then click **Next**. The Buffers Protection task is displayed.




By selecting the **Custom Protection** checkbox, you set the program in the specified buffers such that its contents cannot be modified and/or cannot be seen.

2. Click **Next**. The Variables Protection Task is displayed.

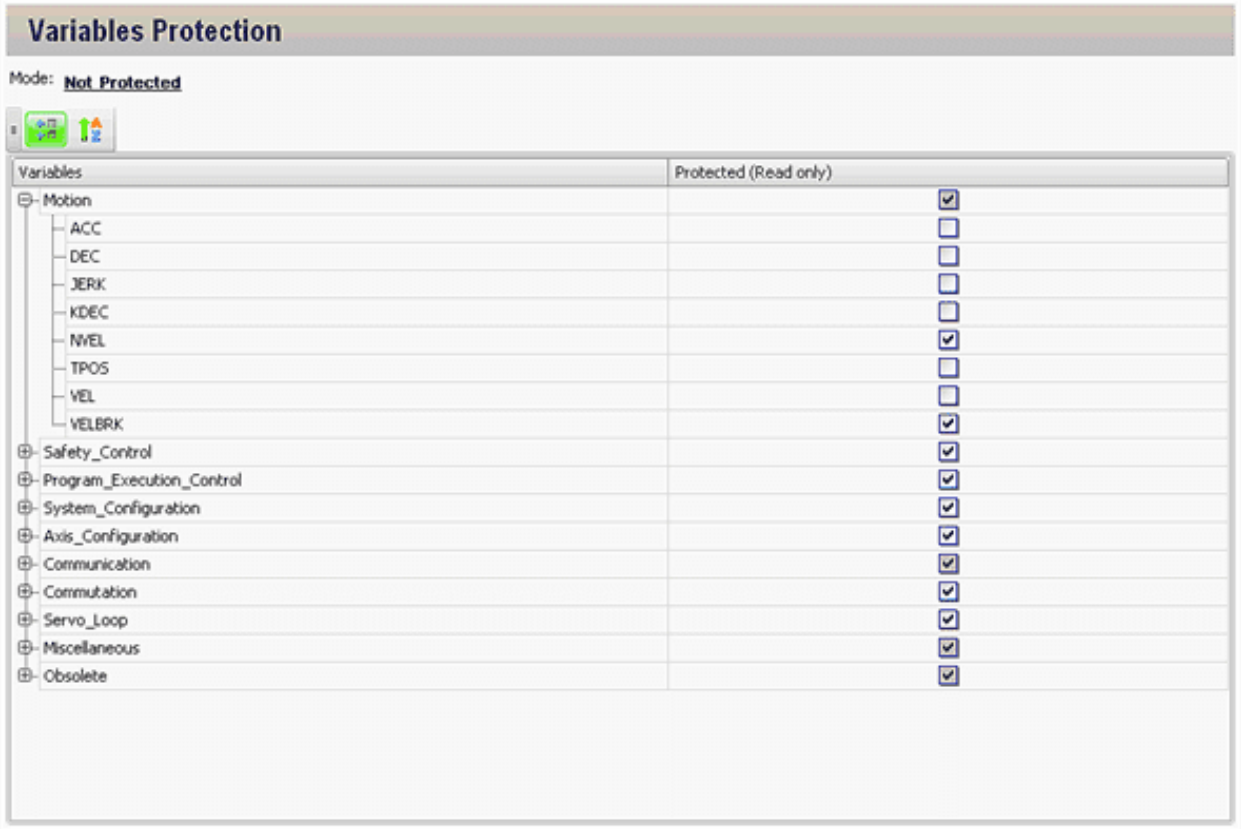
#### *4.4.2.3.2 Custom variable protection*




You can click  to sort the variables by groups, or  to sort alphabetically.

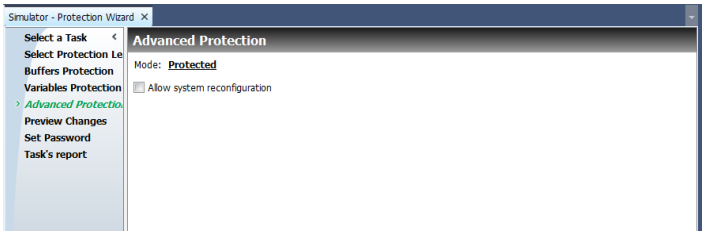
1. Select the appropriate **Read only** box to prevent the variables from being edited.

   You can either set all the variables in a group by selecting the **Read only** box, or expand the groups by clicking the tree expansion button:  and marking individual variables for read only.




2. Click **Next**. The Advanced Protection Task is displayed.

#### *4.4.2.3.3 Advanced protection options*




1. To allow system configuration, select the checkbox.
2. Click **Next** to preview changes and follow step 3 through step 10 in Setting Application Protection to complete setting protection and to exit the Protection Wizard.

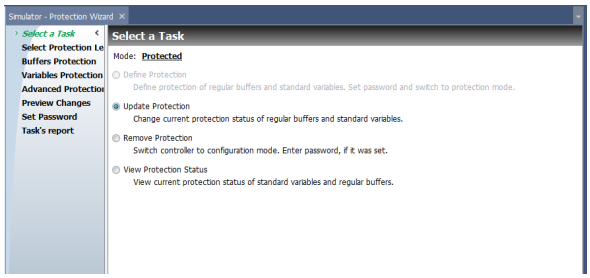
### *4.4.3 Update Protection*

The Update Protection task enables you to make changes in Protection settings.

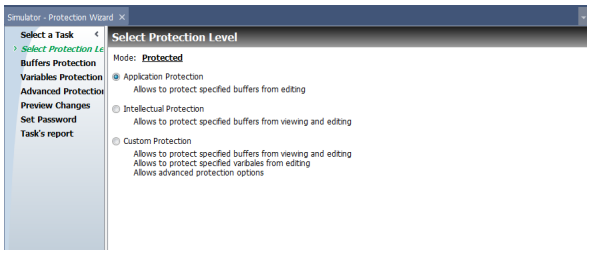
This task is available only if the controller is in the **Protected** mode.

To update Protection data:

1. Select **Update Protection** in the Select a Task window.




2. Click **Next**. The Set Protection Level window is displayed.




3. You have the option of changing the Protection by selecting the appropriate buttons:

> To change Application Protection settings, select the button next to **Application Protection**. Then click **Next** and follow the procedure described in Setting Application Protection.

> To change Intellectual Property Protection settings, select the button next to **Intellectual Property Protection**. Then click **Next** and follow the procedure described in Setting Intellectual Property Protection.

> To change Custom Protection settings, select the button next to **Custom Protection**. Then click **Next** and follow the procedure described in Setting Custom Protection.

In the selected protection task, you enable protection or cancel an existing Protection by clicking the selected checkbox (  ) to toggle it on or off.

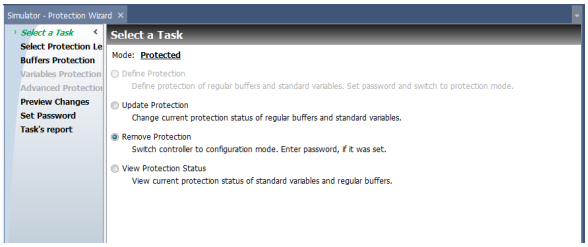
### *4.4.4 Remove Protection*

The Remove Protection task enables you to disable the **Protection** mode.

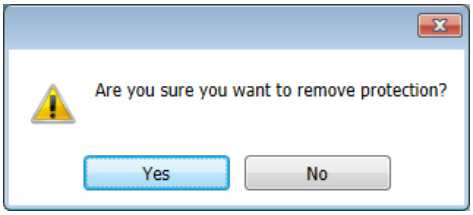
This task is available only if the controller is in the **Protected** mode.

To remove Protection:

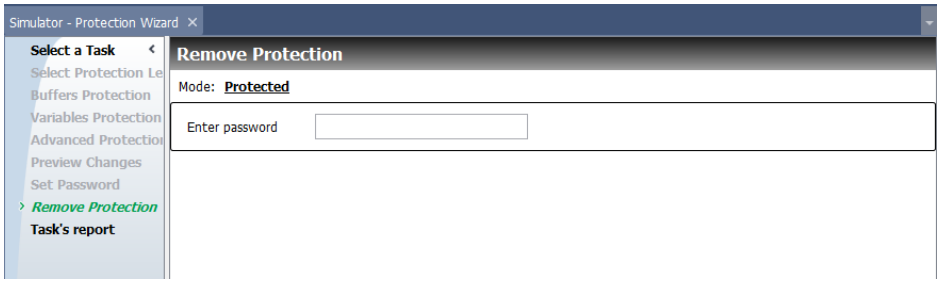
1. Select **Remove Protection** in the Select a Task window.



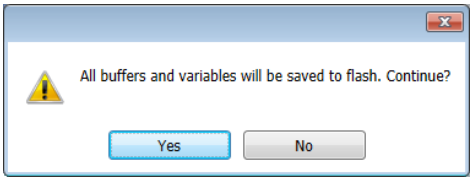

2. Click **Next**. The following notification appears:



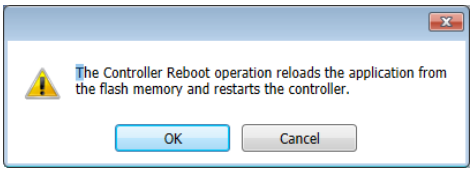

3. The Enter Password dialog is presented.




4. Enter the password and click **Next**. The following dialog appears

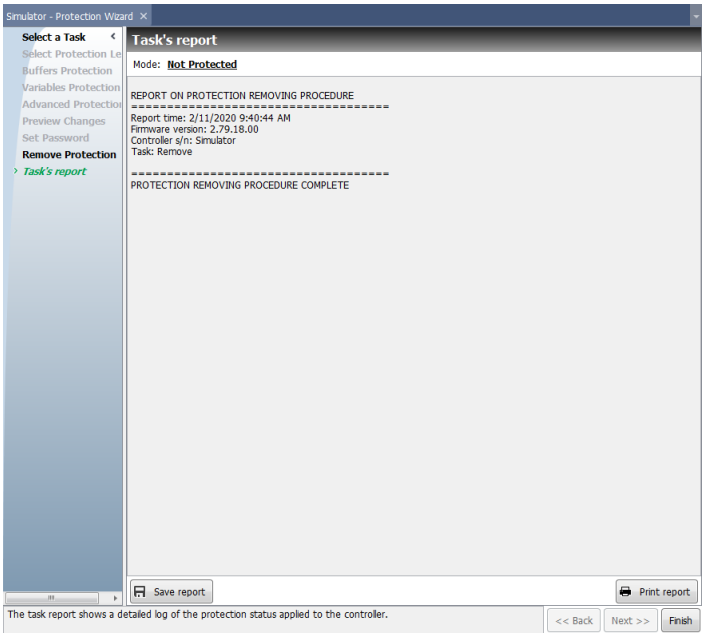



5. Click **Yes**. Another confirmation dialog appears.

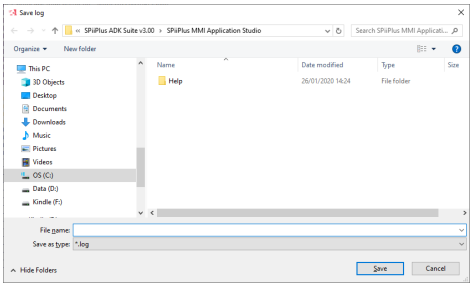



6. Click **OK**
7. Wait for the appearance of a successful restart message:


8. Click **OK**. The Report window is displayed:




9. To save the report click **Save report**. The browser window is displayed:




Select a directory and enter a filename. Click **Save**.

The file extension for all reports is **.log**.

10. To print the report click **Print log**.
11. Click **Finish** to exit the Protection Wizard.

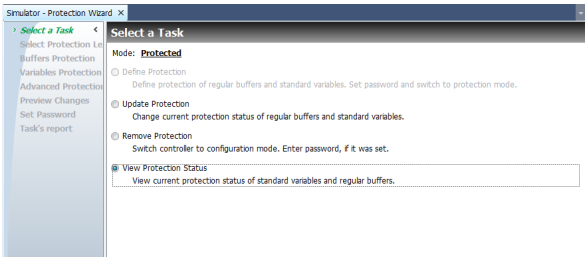
### 4.4.5 View Protection

The View Protection task enables you to view the Protection mode settings of the controller.

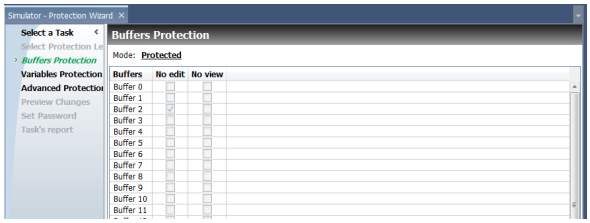
This task is available only if the controller is in the **Protected** mode.

To view the Protection mode settings:

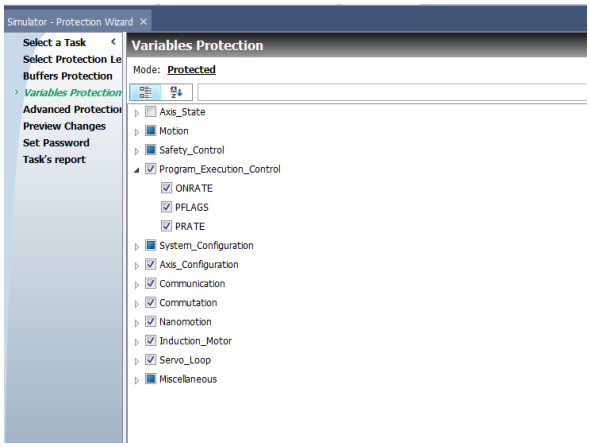
1. Select **View Projection** in the Select a Task window.




2. Click **Next**. The Buffers Protection task is displayed.

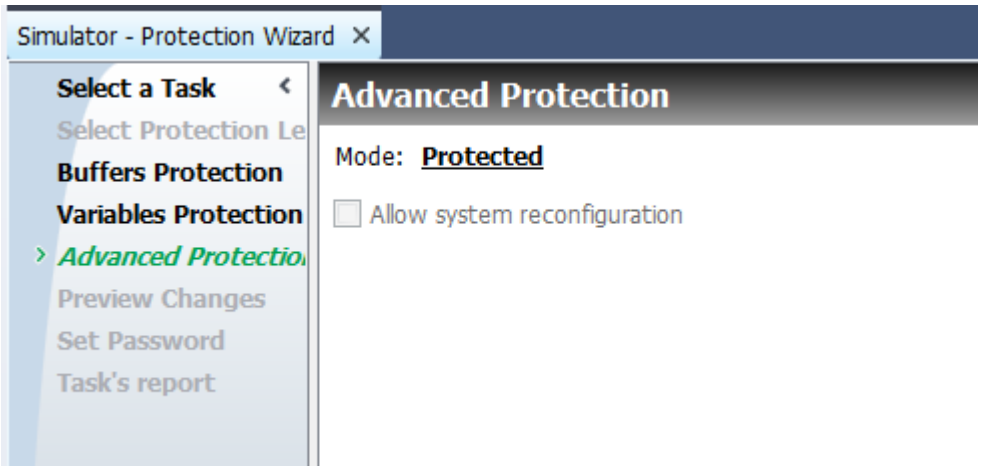



3. Click **Next**. The Variables Protection is displayed.




You can view the settings of the individual variables by clicking the tree expansion button: ⊞ of the groups.

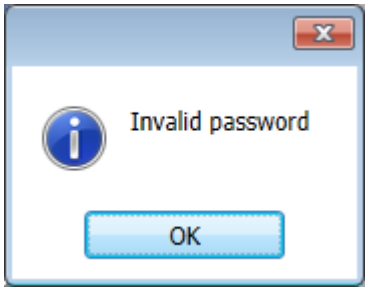
4. Click **Next**. The Advanced Protection is displayed.



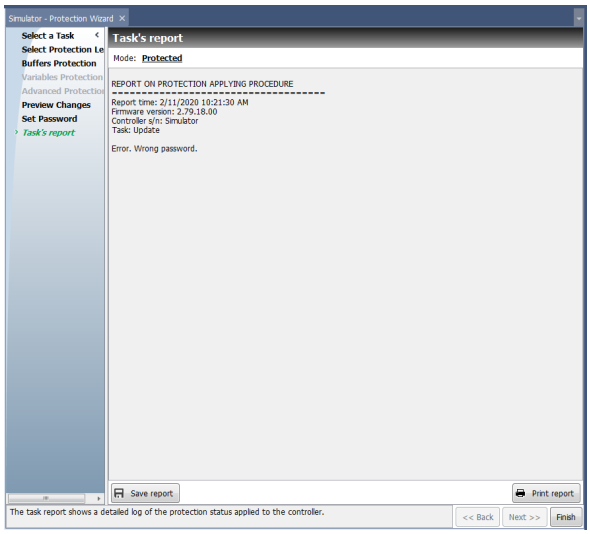

5. Click **Finish** to exit the Protection Wizard.

### *4.4.6 Entering Incorrect Password*

If the password that you enter is incorrect, the **Protection Wizard** displays the following message:




1. Acknowledge the message by clicking **Continue**. The task report informs the user that the task has failed, for example:




2. Your options are:
   - > Click **Back** to return to the window with the password field and enter the correct password, or
   - > Click **Finish** to exit the wizard.

# 5. Utilities

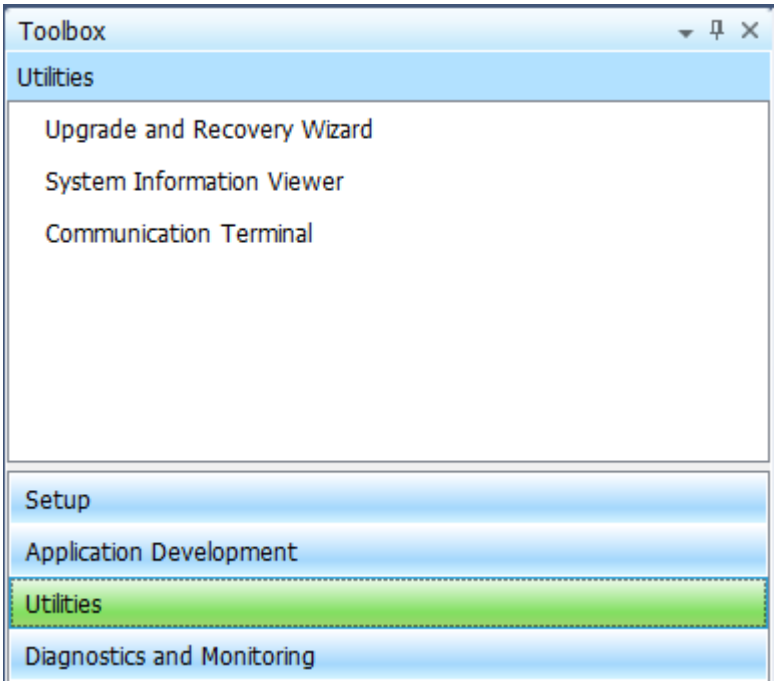
Utilities include the following tools:

- Upgrade and Recovery Wizard - upgrades your SPiiPlus Motion Controller software and firmware.
- System Information Viewer - consolidates all the pertinent data about your system and displays it in a single window.
- Communication Terminal - enables you to enter commands directly to your controller. These commands are executed as soon as they are received through one of the communication channels. The results of the command, once executed, appear in the Communication Terminal window Output panel.

## 5.1 Accessing Utilities

To access Utilities:

1. If the Toolbox is not displayed, click **Toolbox**.
2. To display the Utilities list click **Utilities**.




3. Click the required utility.

## 5.2 Upgrade and Recovery Wizard

You use the **Upgrade and Recovery Wizard** to:

- Upgrade the SPiiPlus Motion Controller firmware and user applications or downgrade to revert to an earlier firmware version.
- Recover a SPiiPlus Motion Controller that has totally stopped functioning.
- Provides the option to set a shortcut for automatically installing updates.

> Backup specific controller (machine) information to a location that you specify and create MMC file for Smarter Motion Tools.

> Duplicate by copying the contents in the source directory created by the "Backup Machine" feature to a SPiiPlus Motion controller.

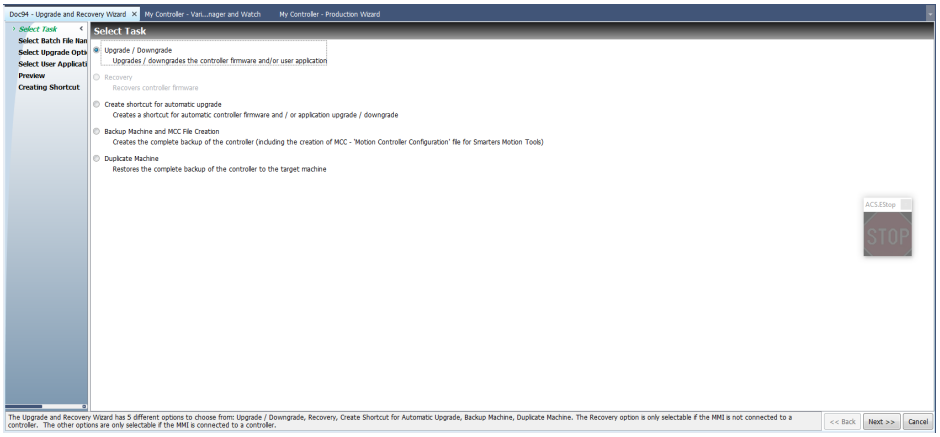
### 5.2.1 Starting Upgrade and Recovery Wizard

To start the **Upgrade and Recovery Wizard**:

1. From the Toolbox click **Utilities**.
2. Click **Upgrade and Recovery Wizard**.

You can also activate the **Upgrade and Recovery Wizard** using the right-click **Add Component** option of the Workspace Tree - see Controller Right-Click Options.

The Upgrade and Recovery Wizard Task window appears.




Available tasks are:

> Upgrade Controller Task

> Recovery Task (available only if the controller is disconnected)

> Create Shortcut for Automatic Upgrade Task

> Backup Machine

> Duplicate Machine

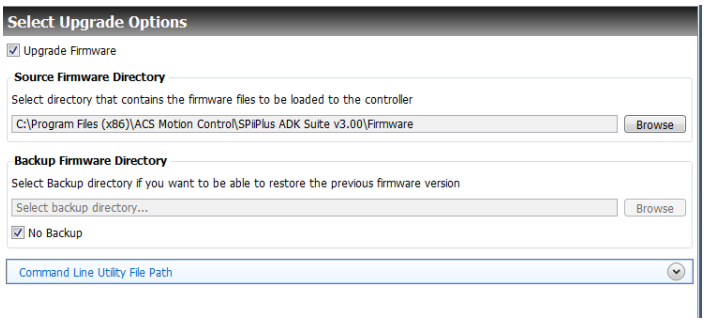
### 5.2.2 Upgrade Controller Task

Upgrading the controller firmware is a dangerous operation that may destroy the controller if executed improperly. It is strongly recommended that you adhere strictly to the directions presented by the wizard.

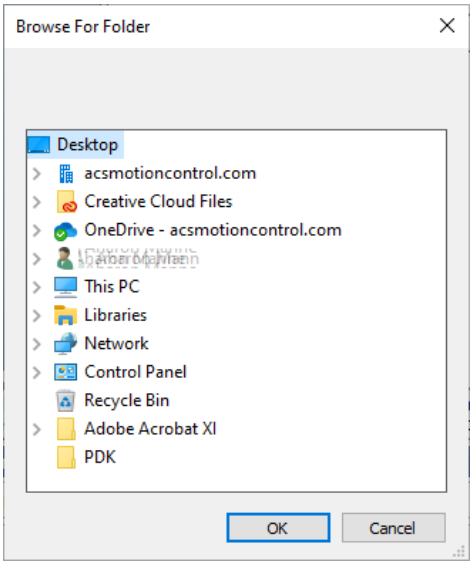
Upgrading from SPiiPlus NT v1.0 to SPiiPlus NT v2.0 is done through the Recovery Task and not the Upgrade Controller Task.

To upgrade the controller firmware and user applications:

1. Select **Upgrade/Downgrade** and click **Next**. The Select Upgrade Options window is displayed.




2. Enter the pathname of the directory in which the firmware files are located if different from what the Upgrade and Recovery has automatically entered, or click **Browse** and search for the directory:



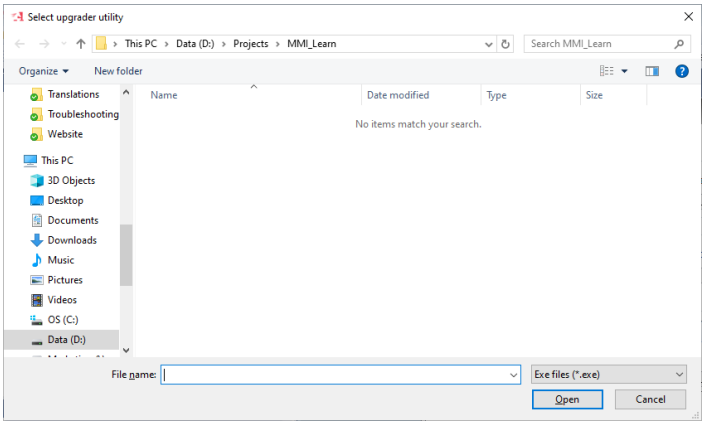

3. Click **OK**.

Normally when you install the firmware on the computer from the ACS Motion Control Installation CD, by default, it is copied into the **Firmware** subdirectory of the same directory in which the SPiiPlus MMI Application Studio is located. However, it may be stored anywhere.

4. You can select to backup the old firmware by deselecting the **No backup** checkbox and entering the pathname of the directory into which the data is to be saved in the **Backup**

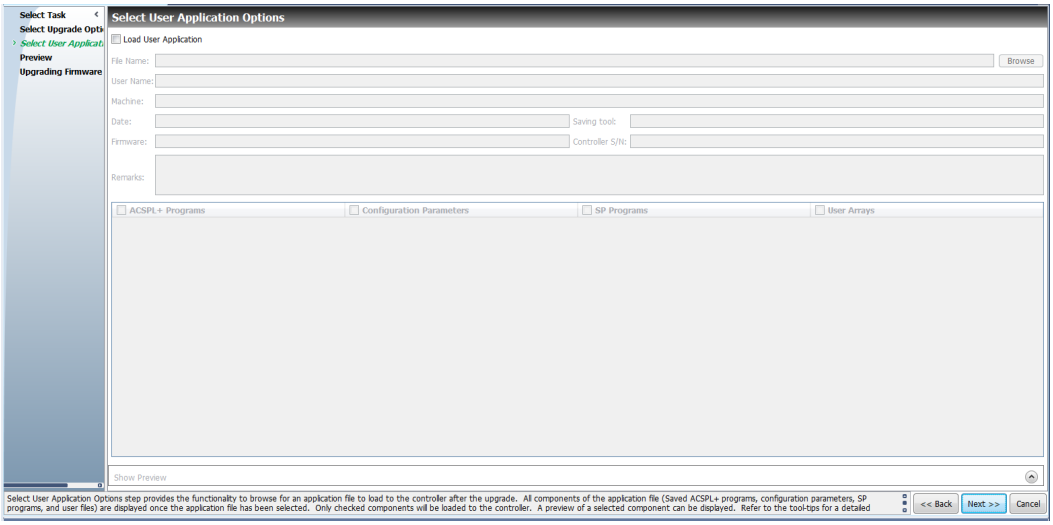
**directory** field (or use **Browse** to find it).
The default is not to backup the old firmware.

5. Enter the path to the **Upgrader.exe** utility if different from what the Upgrade and Recovery has automatically entered, or use **Browse** to find it.

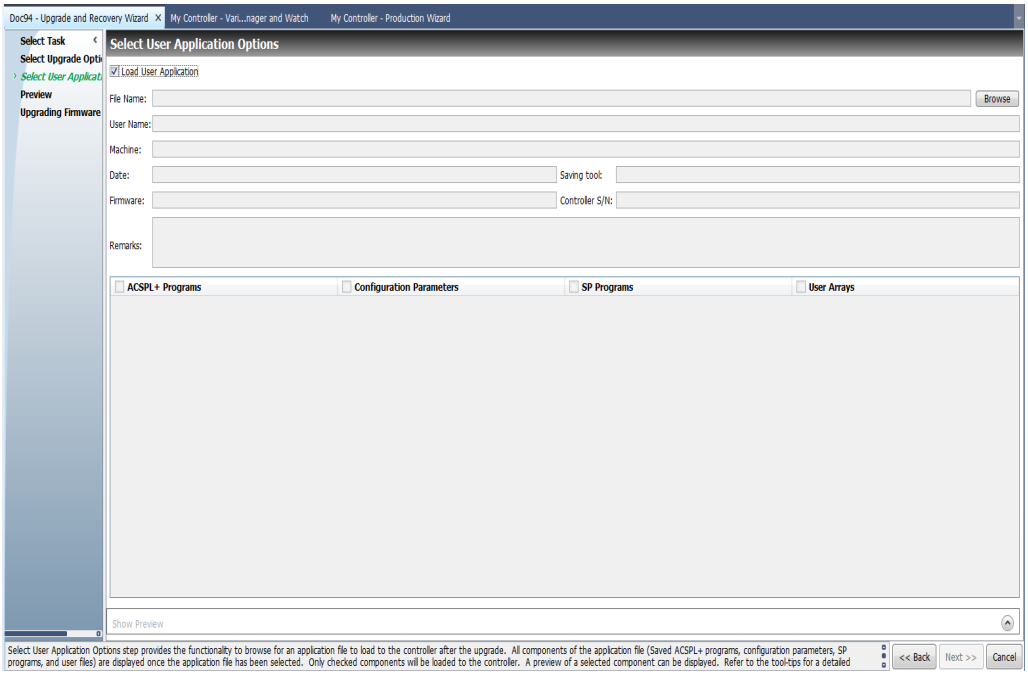



6. Click **OK**.

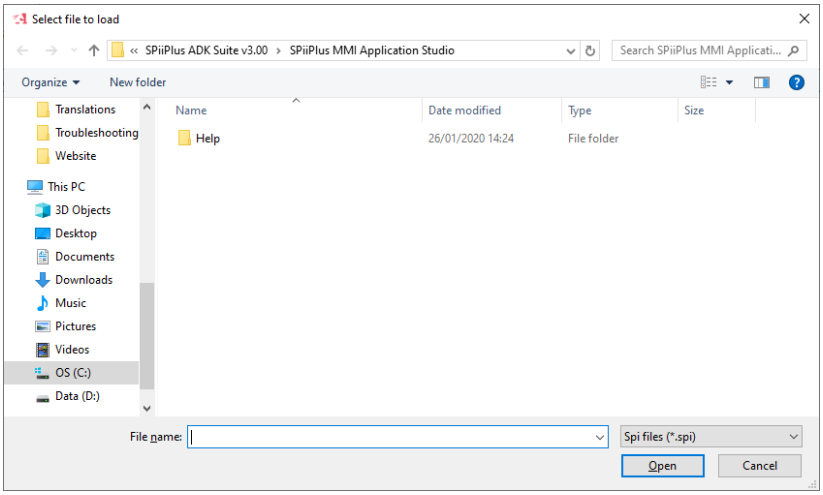
7. Click **Next**. The Select User Application Options window is displayed:




8. You can either skip this task by clicking **Next** and go to Step 8, or, if you have saved your application to the computer (see Application Saver), you can load it by selecting the **Load User Application checkbox.**

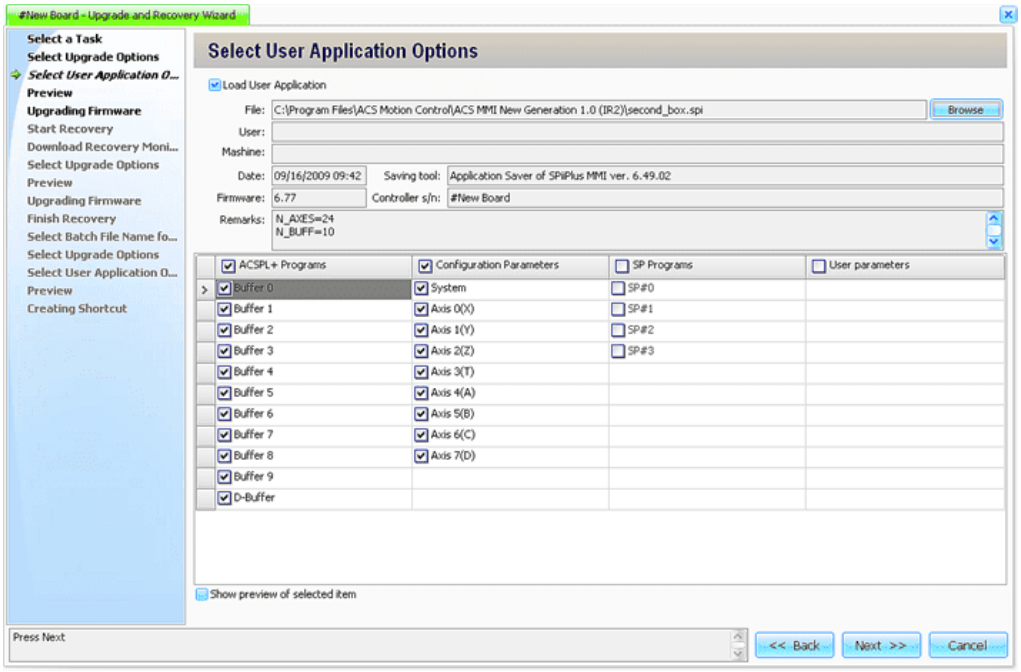



9. Use the **Browse** button to locate and select the application data file:

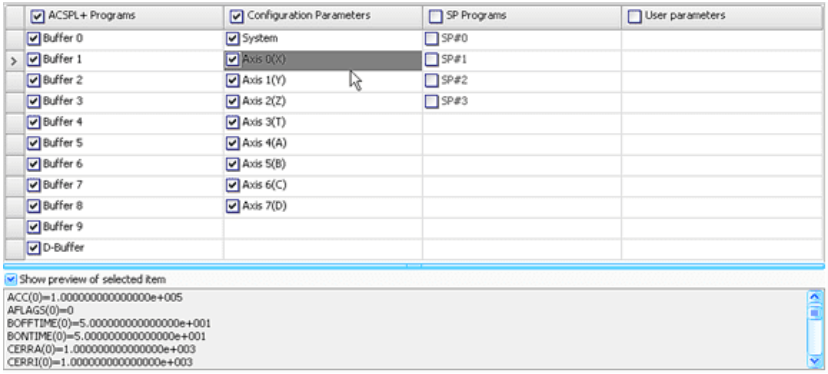



Application data is saved in files having an **spi** extension.

10. Click **Load**. The application data is loaded into the window.

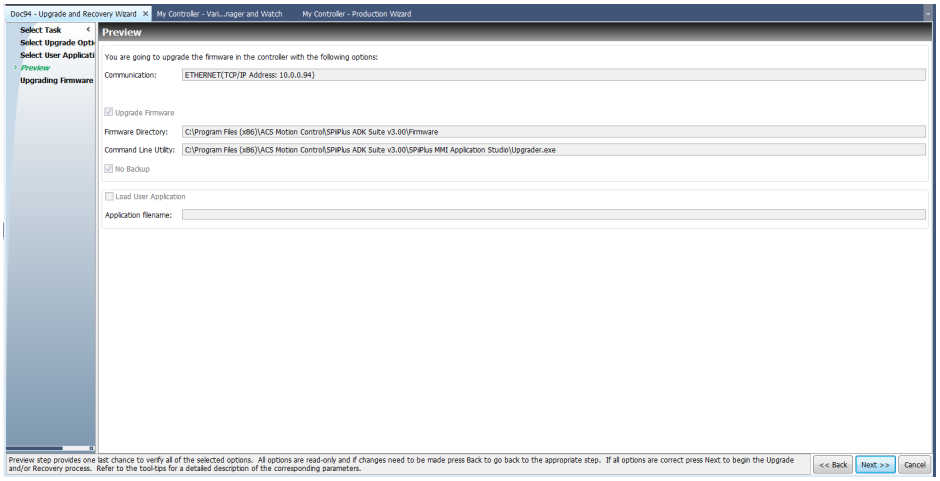



You can see the contents of any buffer program or variable by selecting it and clicking **Show preview of selected item** checkbox. For example:




You can control what is loaded from the application data by unselecting the checkbox of those items you do not want loaded.

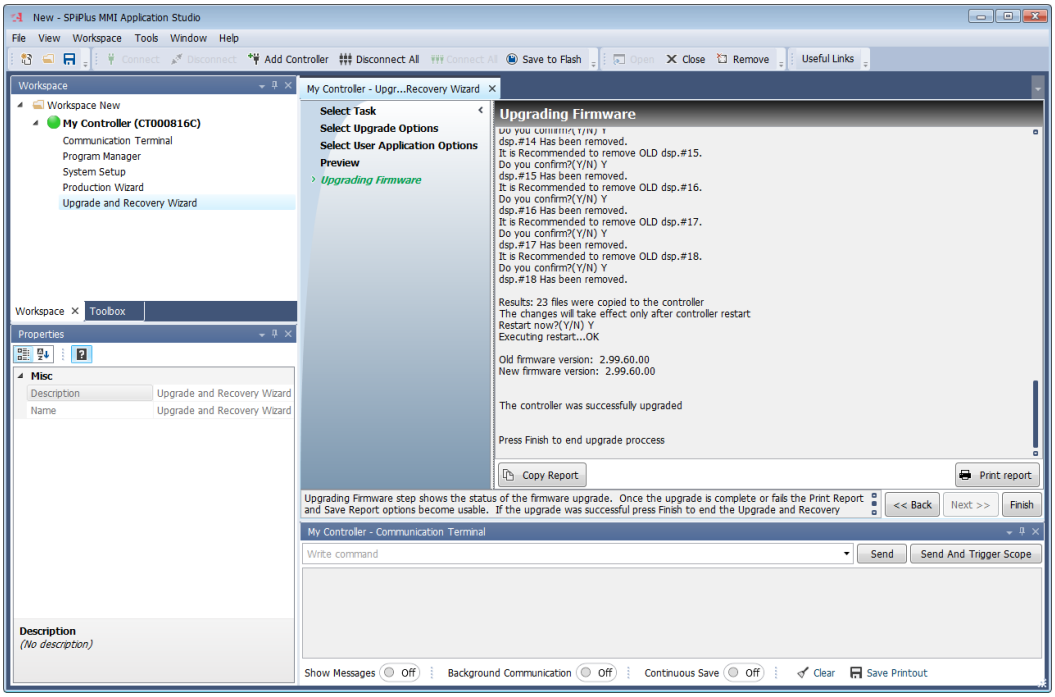
11. Click **Next**. The Preview task window is displayed.




12. Review the Preview window contents, and if it is what you want to load, click **Next**.

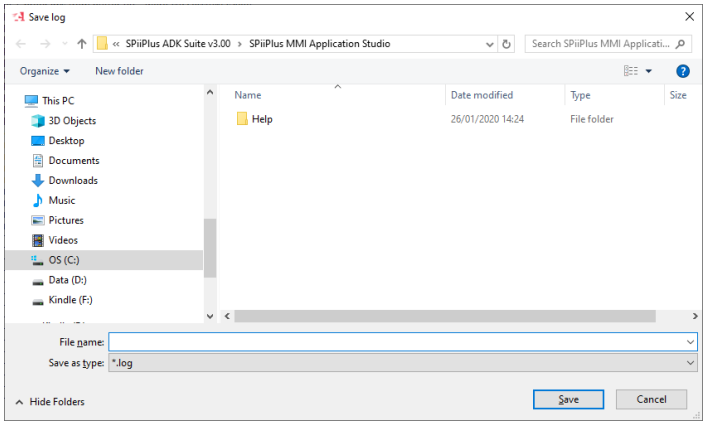
> The **Preview** window is for information only. In order to make any changes, you have to use the **Back** button to return to previous windows and make your changes there.

The Upgrade Firmware window is displayed while the wizard loads the data.




13. When the upgrading is complete, you have the options of:

    > Saving the Upgrade Report to a log file by clicking **Copy Report**. A browser window is displayed:




Drill down to the directory in which you want to save the report and enter a name in the **Filename** field.

14. Click **Save**.

All log files have the **log** file extension.

> Print the Upgrade Report by clicking **PRINT REPORT**, a Print Preview window is displayed - seePrint Report Preview Window.

15. Click **Finish** to exit the Upgrade and Recovery wizard.

### *5.2.3 Recovery Task*

The Recovery task is a routine to restore a SPiiPlus motion controller that has totally stopped functioning.

To run the Recovery task:

1. Select the controller in the Workspace Tree, and click **Disconnect** or press **Alt+D**.

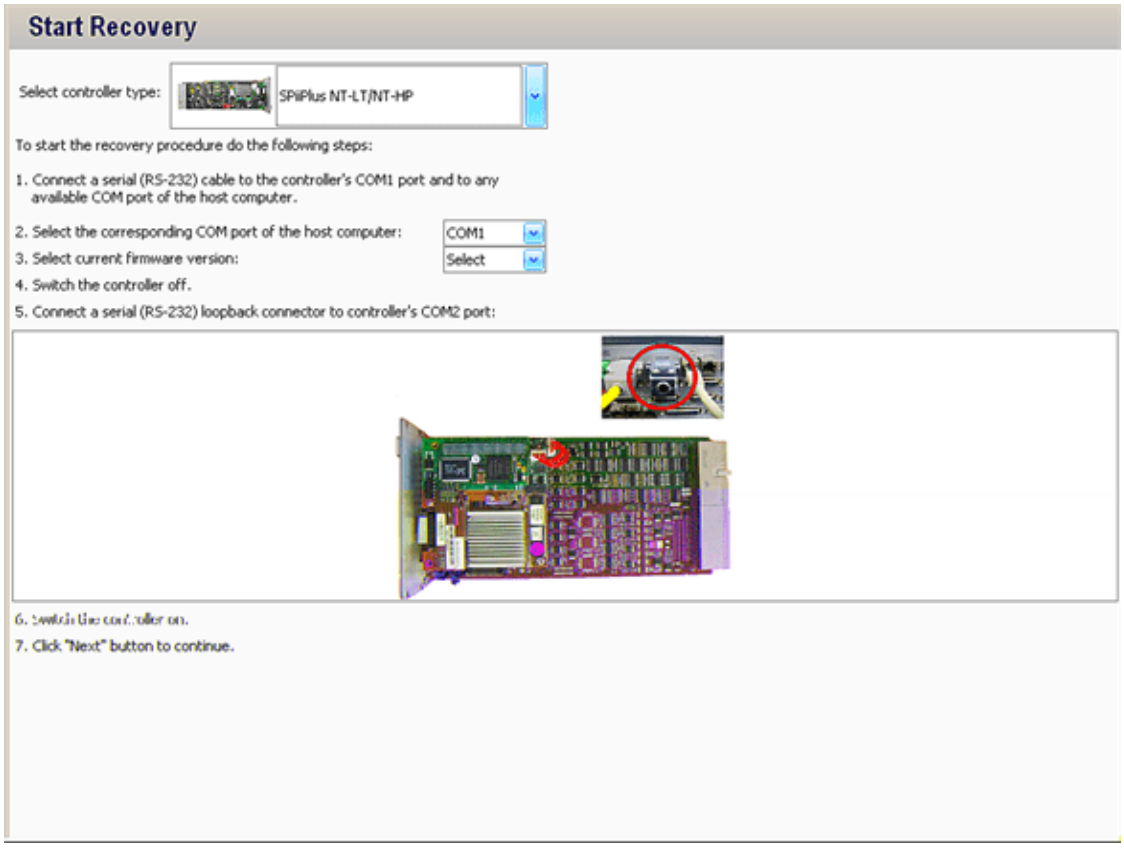
The controller has to be disconnected before the **Recovery** task can be run.

2. Activate the Upgrade and Recovery wizard (see Starting Upgrade and Recovery Wizard).
3. Click **Next**. The Start Recovery window is displayed:


4. Select the SPiiPlus motion controller type from the dropdown list, for example:


SPiiPlusEC
SPiiPlusEC
SPiiPlusES
MC4U-LT/HP/LD/NP
SPiiPlusCMhp/CMba
SPiiPlusCMnt
SPiiPlusCMhv

The window display changes depending on the type of controller you have selected:

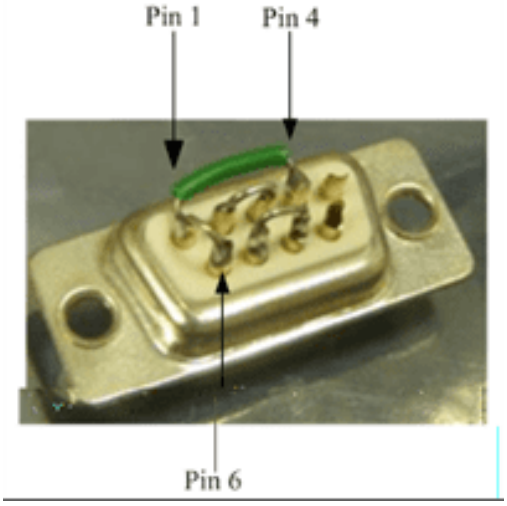



Perform the following steps appearing in the window:

It is extremely important for the success of this task that you adhere to the instructions and perform each step in its proper sequence.

- Connect a serial (RS-232) communication cable to the controller's COM1 port and to any available COM port on the host computer.
- Select the host's COM port from the dropdown list.
- Select the current firmware version from the dropdown list.
- Switch the controller off.
- Connect a serial (RS-232) port loop-back adapter to the controller's COM2 port (as shown in the picture in the window).

With the serial port loop-back adapter installed in COM2, you then install the loop back plug (shown below) on the adapter.



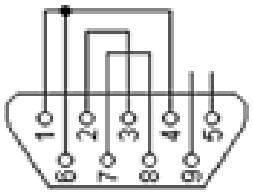

To make the loop back plug, the following pins need to be connected together:

Data Carrier Detect, Data Set Ready and Data Terminal Ready (Pins 1, 6 & 4)

Receive Data and Transmit Data (Pins 2 & 3)

Request To Send and Clear To Send (Pins 7 & 8)

as follows:



> Turn the controller back on.

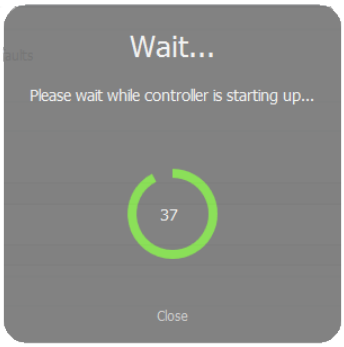
Perform the following steps appearing in the window:

It is extremely important for the success of this task that you adhere to the instructions and perform each step in its proper sequence.

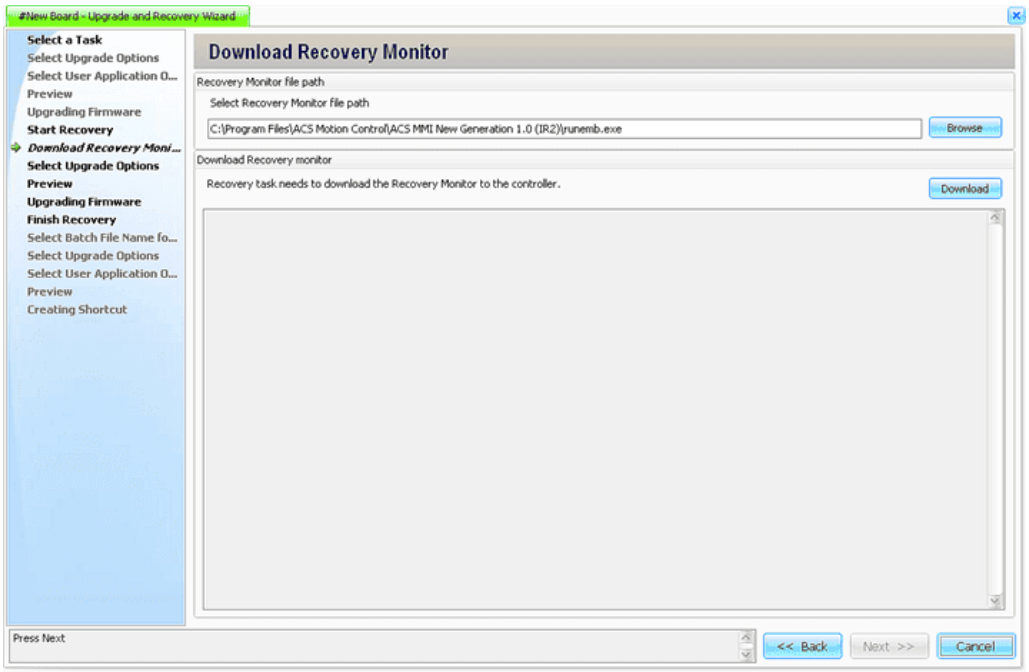
1. Connect a serial (RS-232) communication cable to the controller's COM1 port and to any available COM port on the host computer.
2. Select host's COM port from the dropdown list.
3. Turn the power off.

Power must be turned off to prevent the possibility of electric shock to personnel or damage to the controller card when extracting the controller to install the jumper.

4. Extract the controller and install the jumper. Then reinstall the controller.
   Note that the picture showing the jumper location has changed to match the controller that you have selected.
5. Turn the power on.
6. Click **Next**. The controller is restarted:

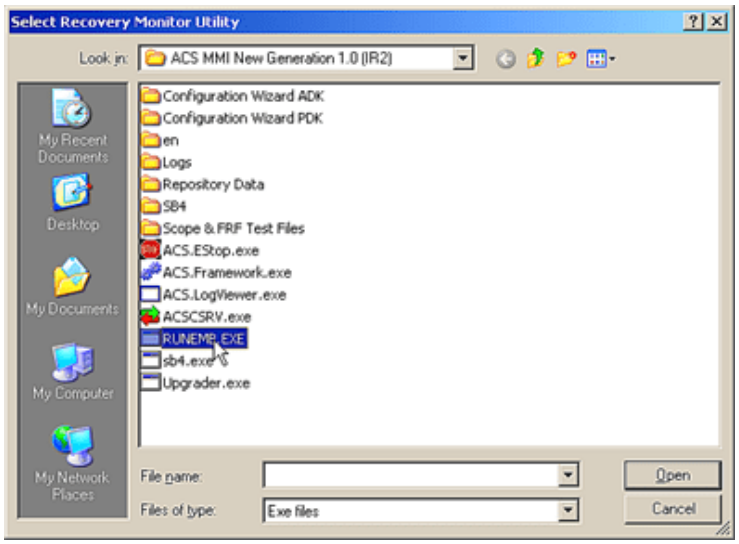



   Wait until the controller has finished rebooting.
7. If you are using SPiiPlus v2.0 or earlier, the Download Recovery Monitor appears. Otherwise, skip to The Select Upgrade Options window appears:.




The default path and filename (**RUNEMB.exe**) of the Recovery Monitor is in the same directory as the SPiiPlus MMI Application Studio.

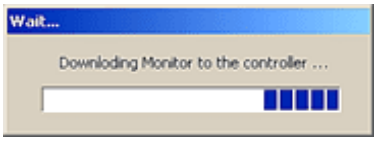
8. If for some reason it is somewhere else, click **Browse** and in the browser window look for the file: **RUNEMB.exe**.




and click **Open**.

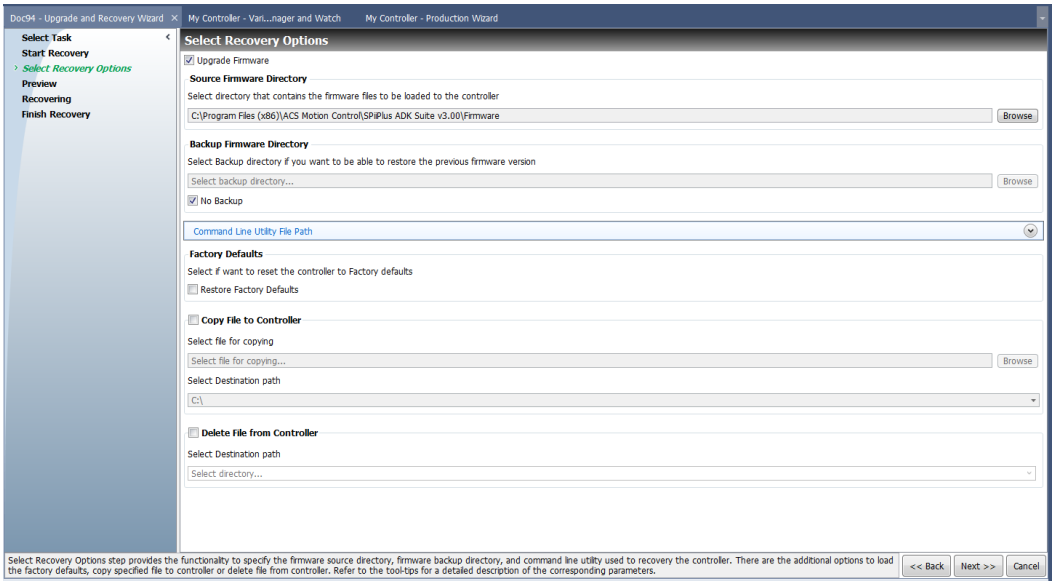
9. Click **Download** ( 

), the Recovery Monitor is downloaded to the controller.




Wait for the completion of the downloading.

10. The **Select Upgrade Options** window appears:

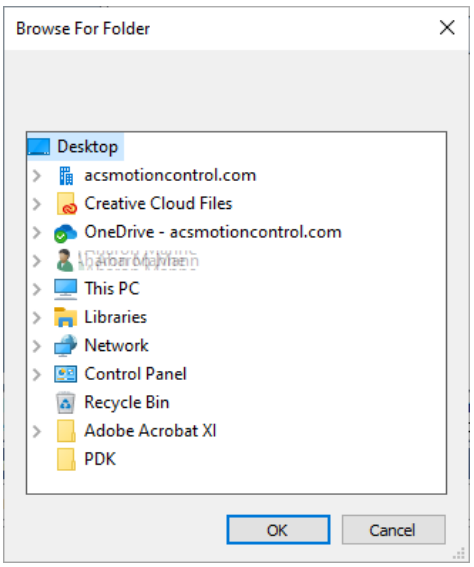



You can select one or more operations from the following list:

- Upgrade Firmware (with option to save the existing Firmware files in the backup folder)
- Restore Factory Defaults
- Copy File to Controller
- Delete File from Controller

11. Normally, the firmware files reside in the **Firmware** subdirectory under the same directory as the SPiiPlus MMI Application Studio.

    If for some reason they are somewhere else, click **Browse** and search for the directory:

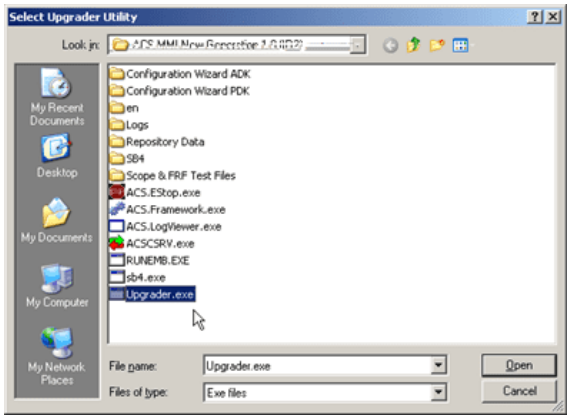



    and click **OK**.

12. You can select to backup the old firmware by deselecting the **No backup** checkbox and entering the pathname of the directory into which the data is to be saved in the **Backup directory** field (or use **Browse** to find it).

    The default is not to backup the old firmware.

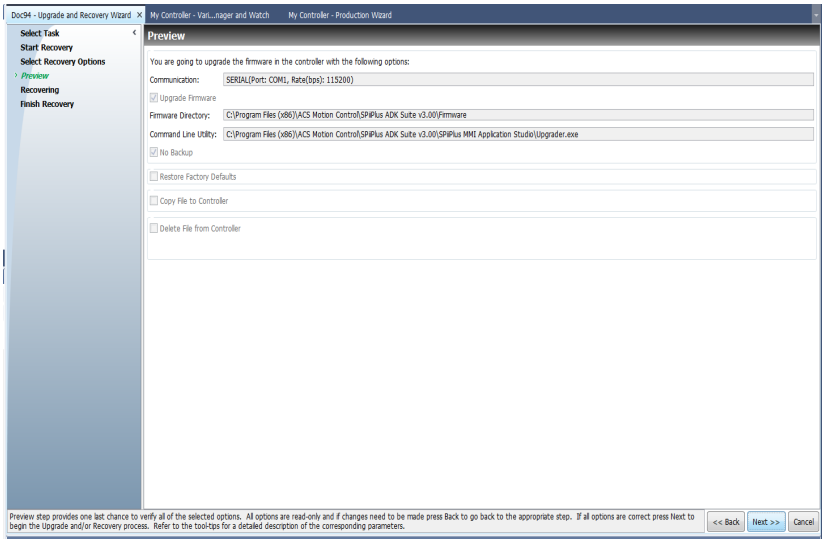
13. Enter the path to the **Upgrader.exe** utility or use **Browse** to find it.




    and click **OK**.

You also have the option of loading the factory default values by selecting the **Restore Factory Defaults** checkbox.

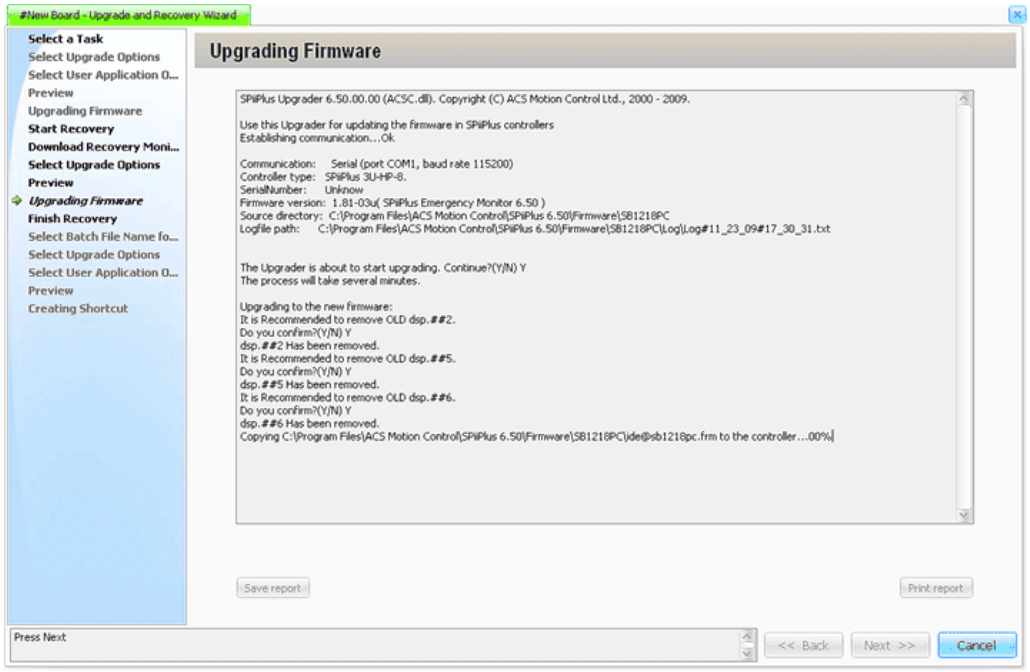
14. The Preview window is displayed:




Review the contents of the Preview window, and if it is what you want to load to the controller, click **Next**.

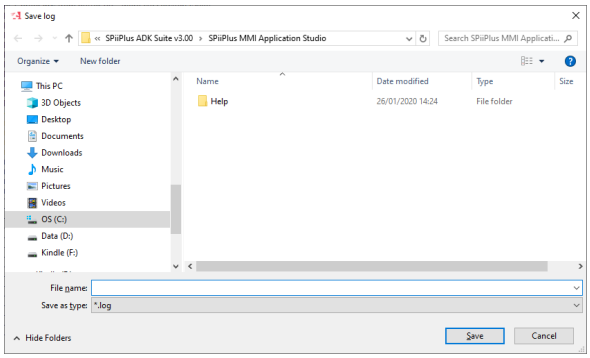
The **Preview** window is for information only. In order to make any changes, you have to use the **Back** button to return to previous windows and make your changes there.

15. The Upgrading Firmware window is displayed showing the progress of the Recovery.




16. Wait for the completion of the process. On completion you have two options.

    > Saving the Upload Firmware Report to a log file by clicking **Save report**. A browser window is displayed:




    Drill down to the directory in which you want to save the report and enter a name in the **Filename** field. Then click **Save**.

All log files have the **log** file extension.

    > Printing the Upload Firmware Report by clicking **PRINT REPORT**. A **PRINT PREVIEW** window is displayed. See Print Report Preview Window

17. Click **Finish**.

In order for the changes to take effect you have to reboot the controller - see Controller Reboot.

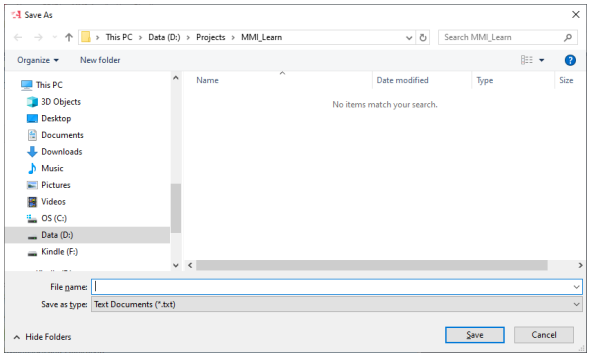
### 5.2.4 Create Shortcut for Automatic Upgrade Task

This task enables you to create a batch file that can be used to automatically upgrade your SPiiPlus Motion Controller.

To create the automatic upgrade shortcut:

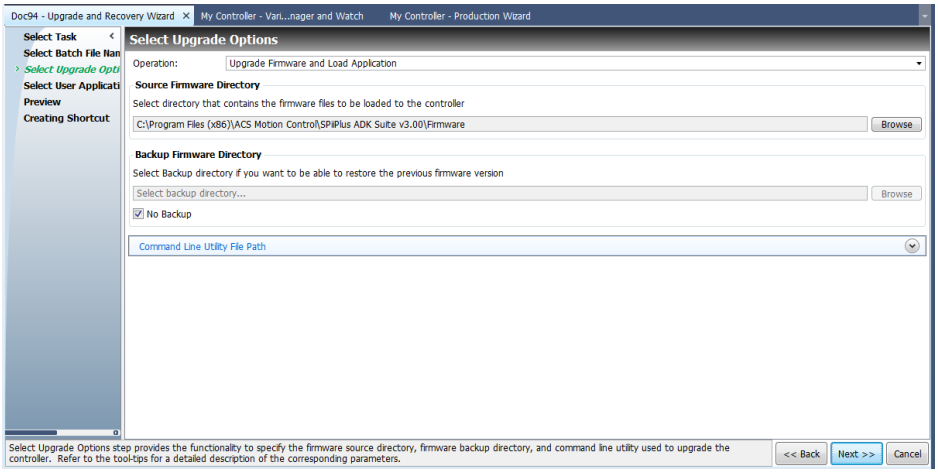
1. Select **Create shortcut for automatic upgrade** and click **Next**. The Select Batch File Name for Automatic Upgrade window is displayed.


2. Click **Browse** to enter a path and filename where the upgrade routine will be saved.

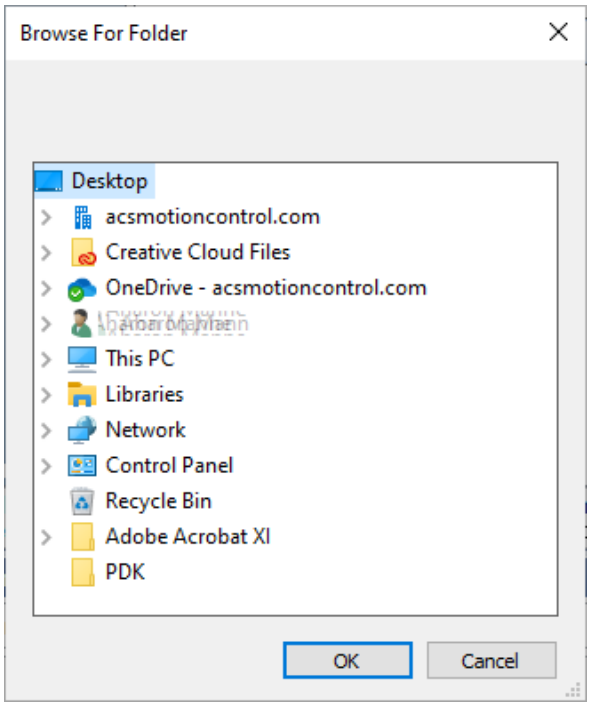



Drill down to the directory in which you want to save the batch file, and enter a name in the **Filename** field. Then click **Save**.

3. Click **Next**. The Select Upgrade Options window is displayed.



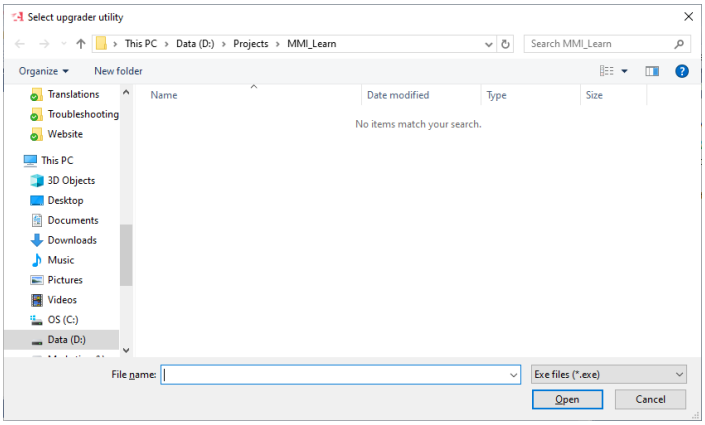

4. Enter the pathname of the directory in which the firmware files are located, or click **Browse** and search for the directory:




and click **OK**.

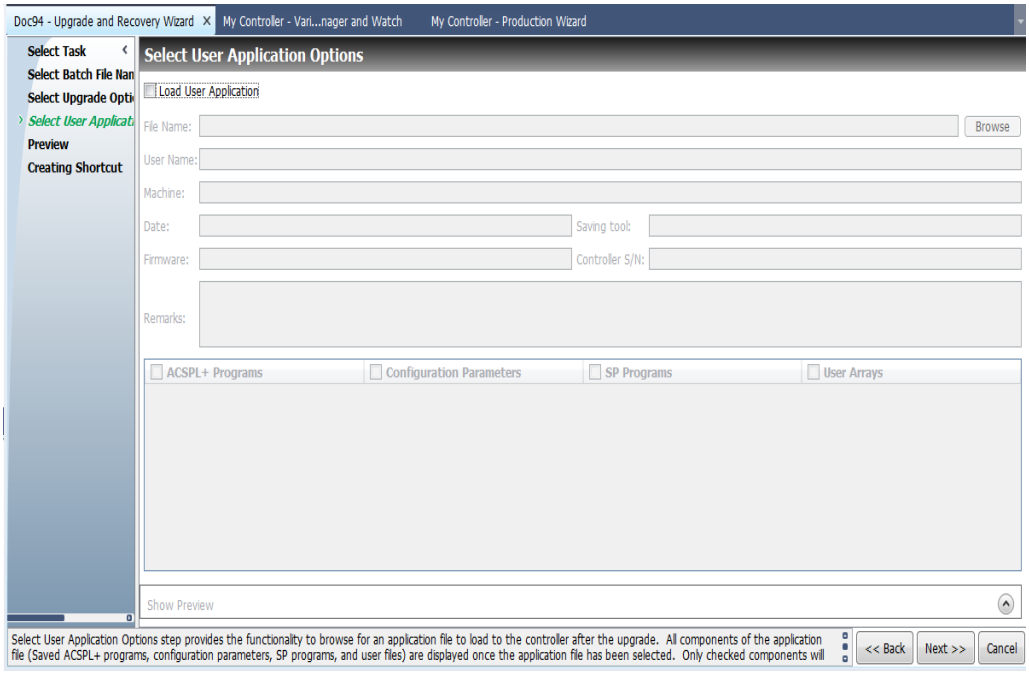
Normally when you install the firmware on the computer from the ACS Motion Control Installation CD, by default it is copied into the **Firmware** subdirectory of the SPiiPlus <version number> directory as shown above. However, it may be stored anywhere.

5. You can select to backup the old firmware by deselecting the **No backup** checkbox and entering the pathname of the directory into which the data is to be saved in the **Backup directory** field (or use **Browse** to find it).
   The default is not to backup the old firmware.
6. Enter the path to the **Upgrader.exe** utility or use **Browse** to find it.

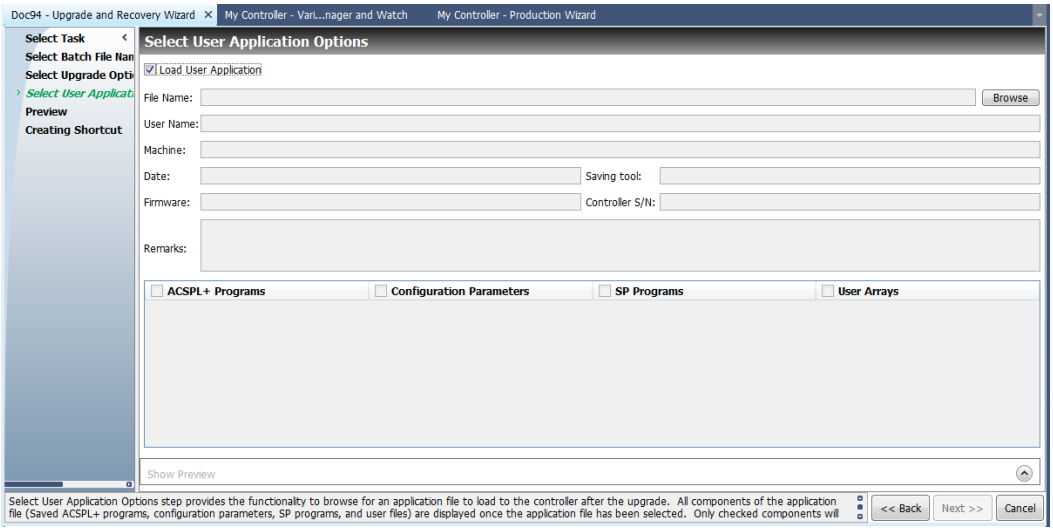



and click **OK**.

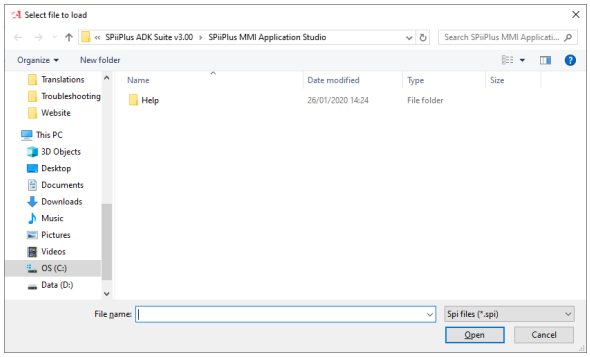
7. Click **Next**. The Select User Application Options window is displayed:




8. You can either skip this task by clicking **Next** and go to Step 10, or, if you have saved your application to the computer (see Application Saver), you can load it by selecting the **LOAD USER APPLICATION** checkbox.

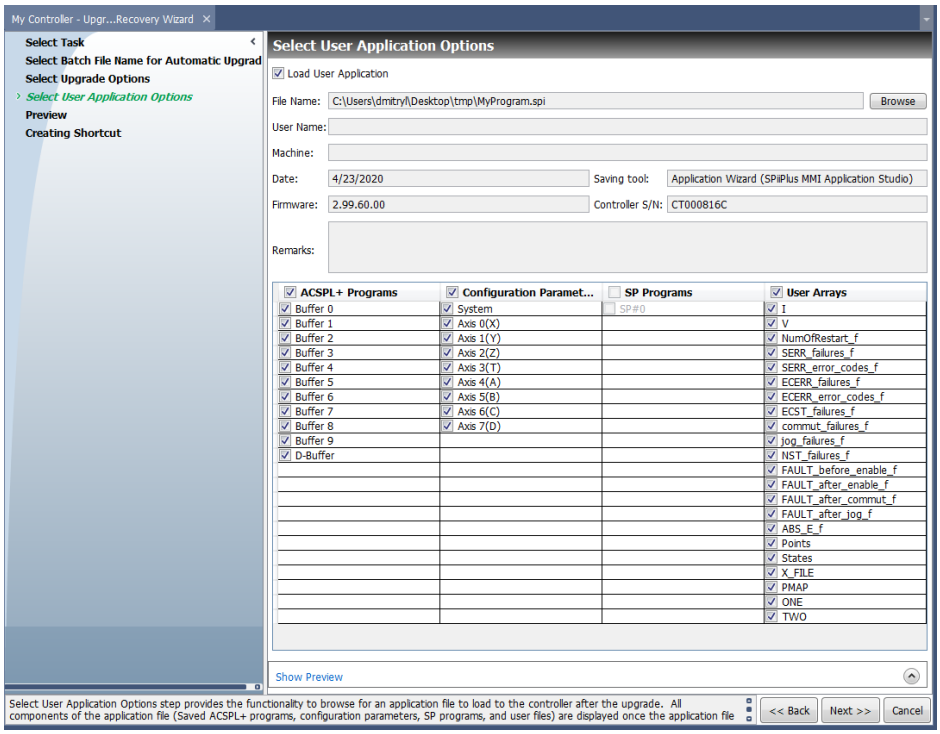



9. Use the **Browse** button to locate and select the application data file:

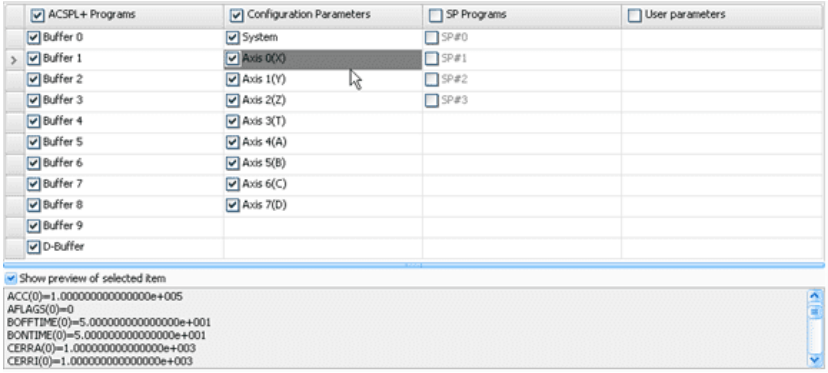



Application data is saved in files having an **spi** extension.

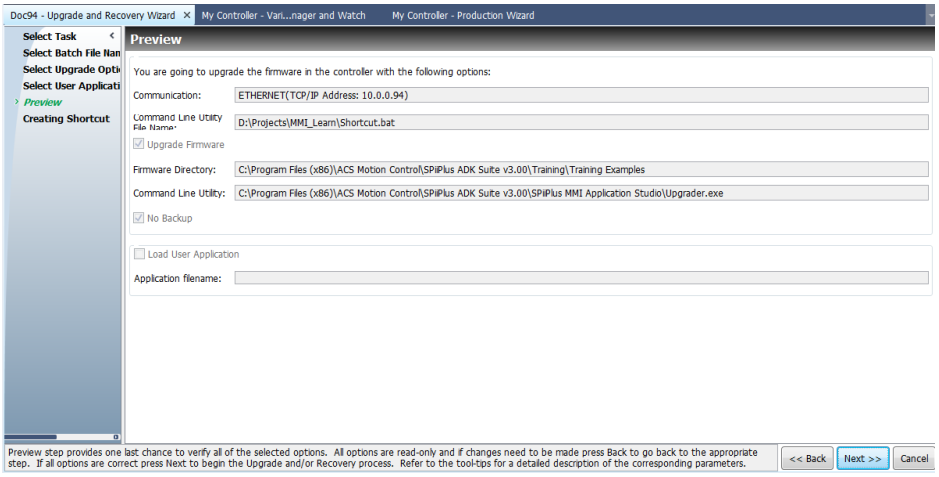
Click **Load**. The application data is loaded into the window.




You can see the contents of any buffer program or variable by selecting it and clicking **Show preview of selected item** checkbox. For example:



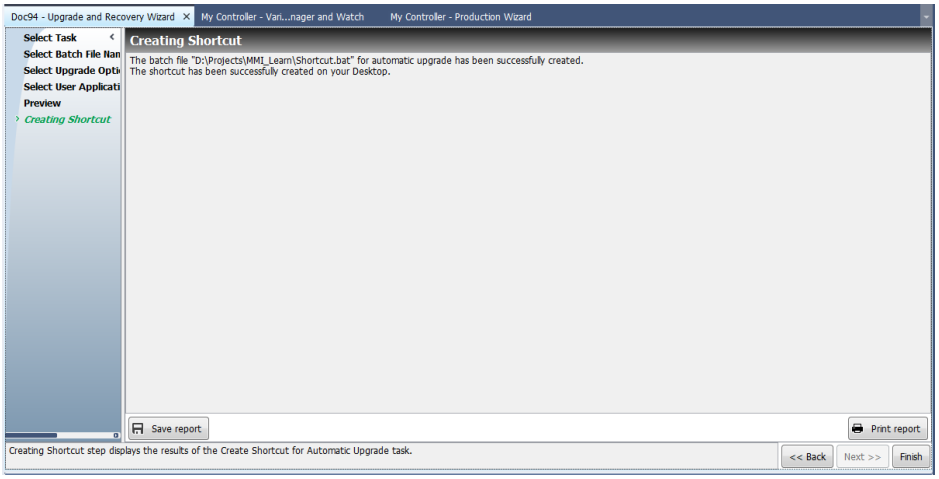

You can control what is loaded from the application data by unselecting the checkbox of those items you do not want loaded. Click **Next**. The Preview task window is displayed.




10. Review the contents of the Preview window, and if it is what you want to load, click **Next**.

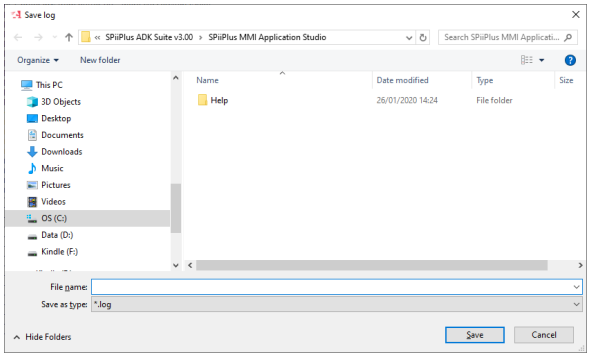
> The **Preview** window is for information only. In order to make any changes, you have to use the **Back** button to return to previous windows and make your changes there.

The Creating Shortcut is displayed while the wizard creates the batch file.




11. When the creation of the batch job is complete, you have the options of:

    > Saving the Shortcut Creation Report to a log file by clicking **Save report** ( Save report ). A browser window is displayed:




Drill down to the directory in which you want to save the report and enter a name in the **Filename** field. Then click **Save**.

All log files have the **log** file extension.

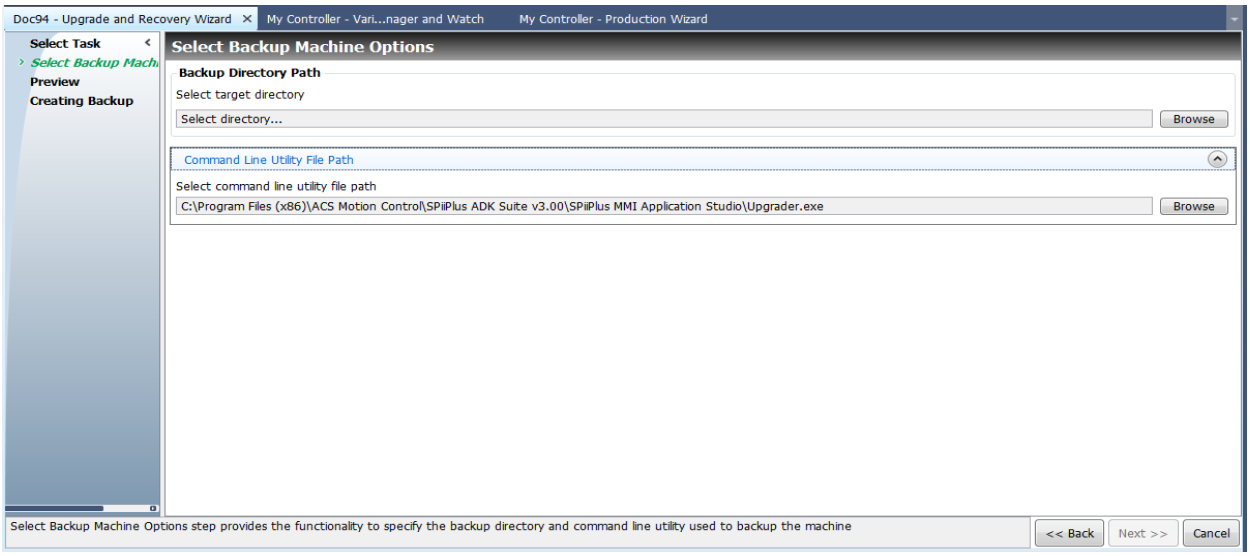
> Clicking **PRINT REPORT**. A **PRINT PREVIEW** window is displayed. See Print Report Preview Window

12. Click **Finish**.

### 5.2.5 Backup Machine

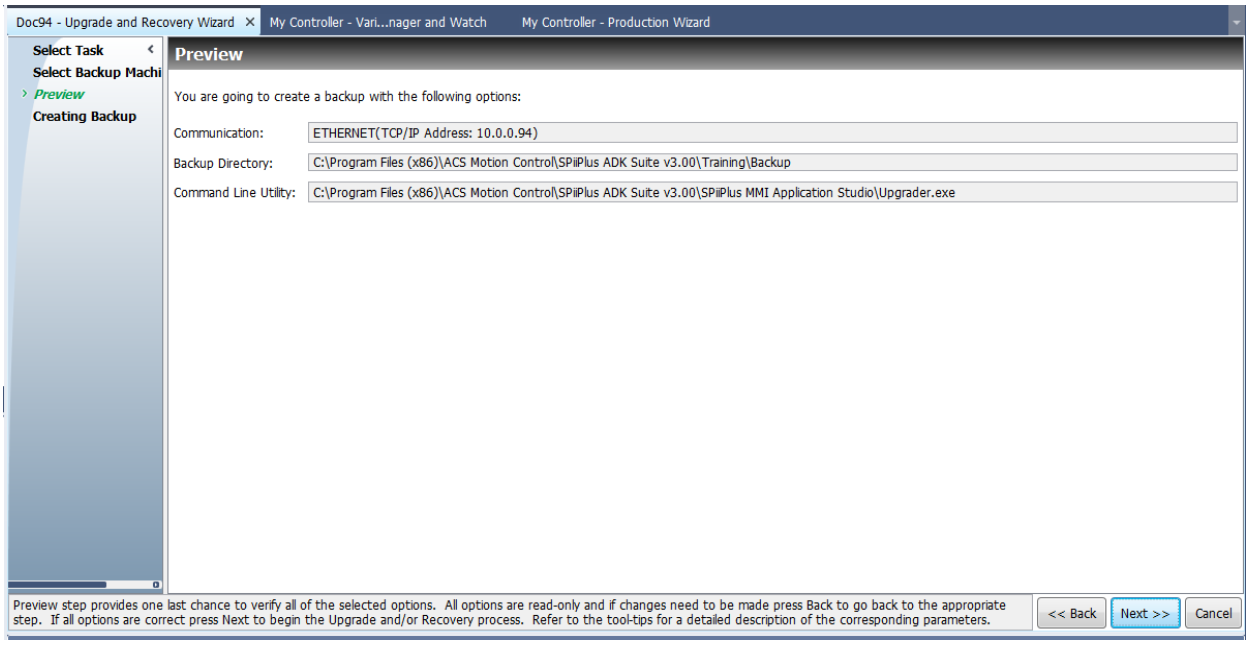
The Backup Machine function allows the user to store all application level information (axes, system parameters, user defined parameters) as well as system configuration (Firmware, DSPs). The backup location is selected by the user on the host computer. This information can later be used as input for "duplicate machine" functionality.

1. Select **Backup Machine** and click **Next**. The Select Backup Machine Options window is displayed.

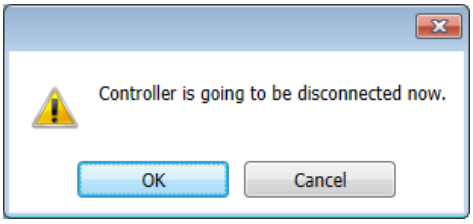



2. Enter the backup directory path to save the backup machine files, or click **Browse** and search for the directory.

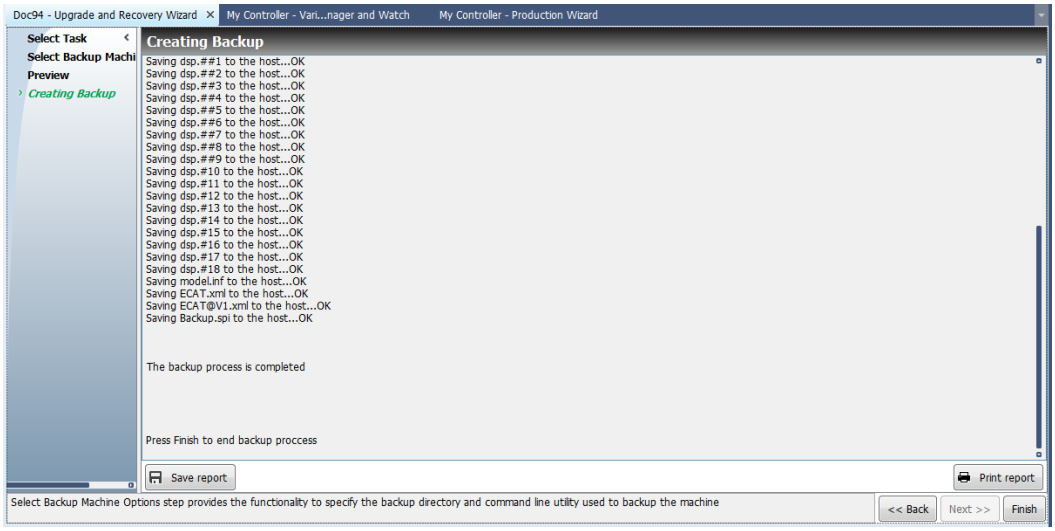
3. Enter the path to the **Upgrader.exe** utility if different from what the Upgrade and Recovery has automatically entered, or click **Browse** and search for the directory.
4. Click **Next**. The Preview window is displayed.




5. Veryify the Backup Directory and Upgrader Utility locations and make any necessary changes, then Click **Next**. A warning message is displayed.




6. Click **OK**. The Creating Backup window is displayed.

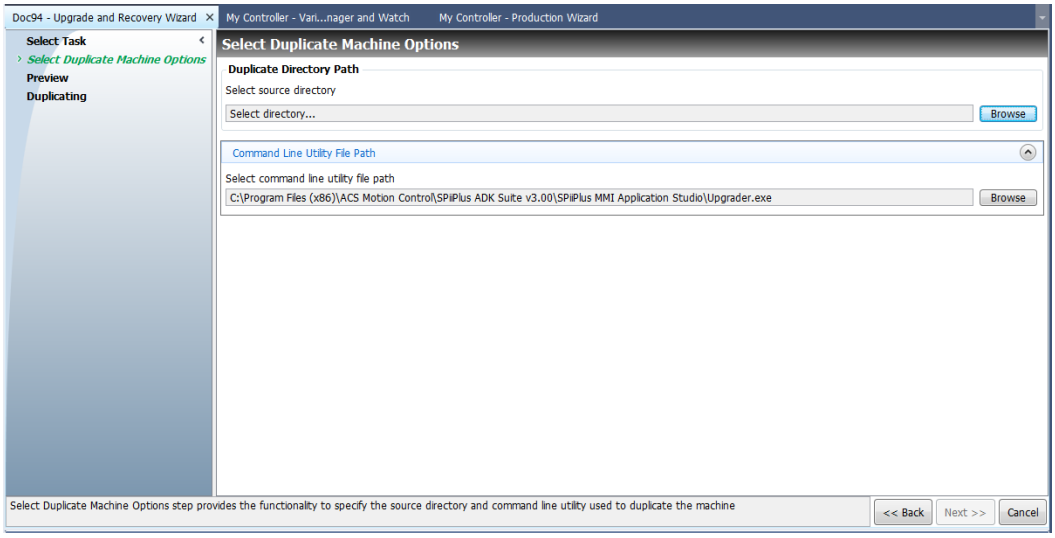



7. If desired, click **Save Report** or **Print Report**.
8. Click **Finish**.

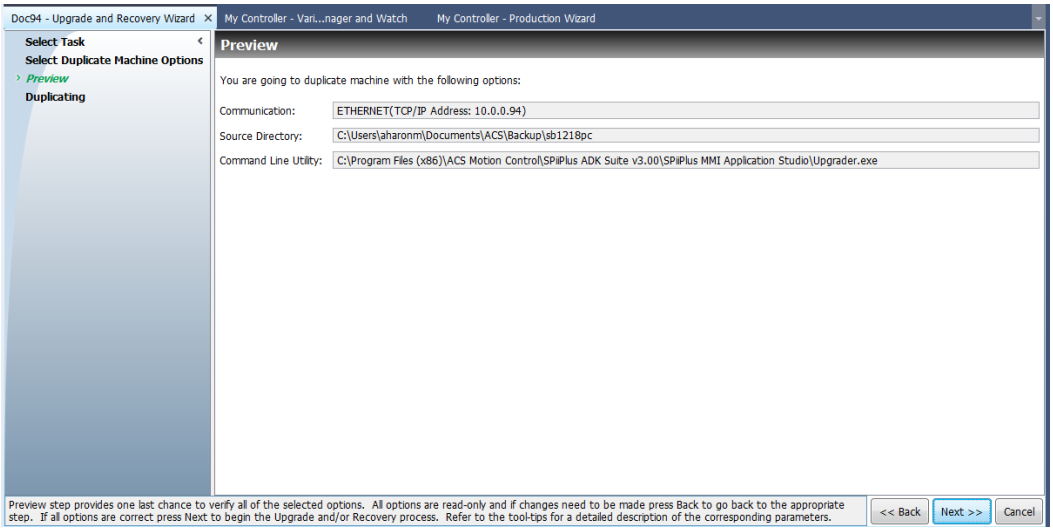
### 5.2.6 Duplicate Machine

This functionality is based on "Backup Machine" feature output. User selects the source directory which was created by "Backup Machine" feature. The contents of the selected folder are copied to the controller.

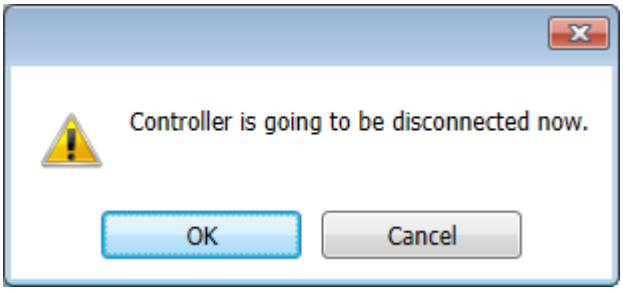
1. Select **Duplicate Machine** and click **Next**. The Select Duplicate Machine Options window is displayed.



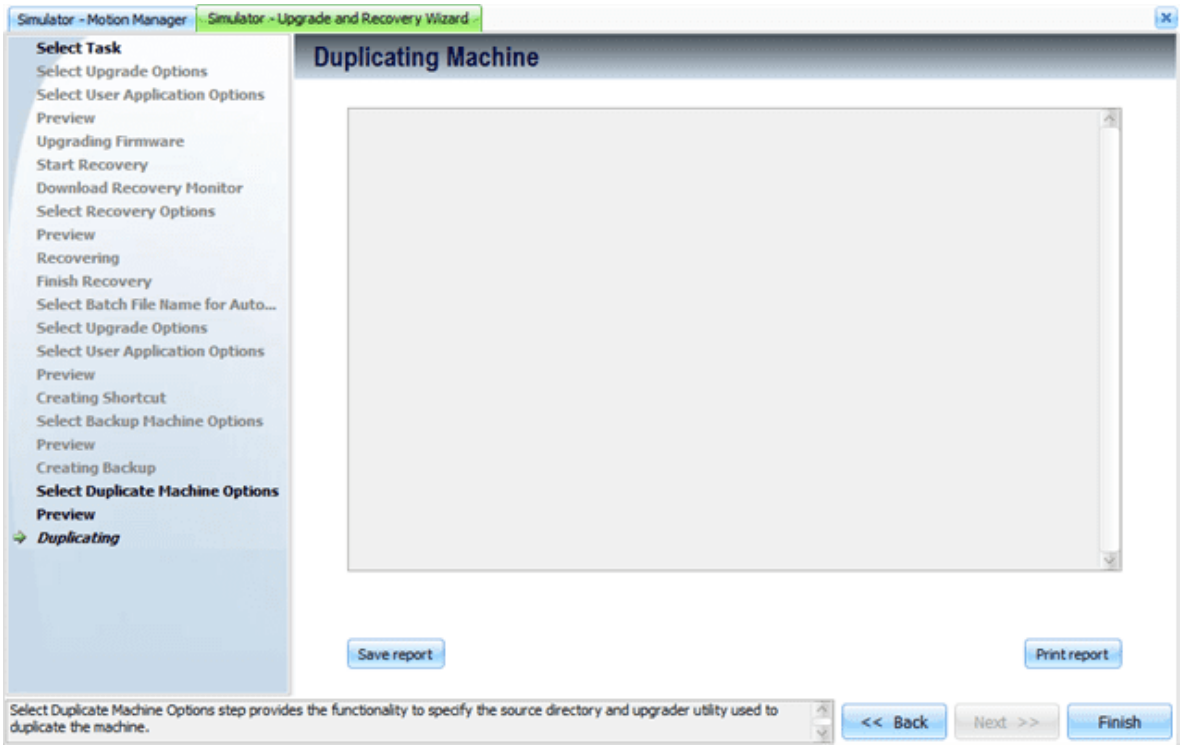

2. Enter the pathname of the source directory in which to save the duplicate machine files, or click **Browse** and search for the directory.
3. Enter the path to the **Upgrader.exe** utility if different from what the Upgrade and Recovery has automatically entered, or click **Browse** and search for the file path.
4. Click **Next**. The Preview window is displayed.




5. Verify the Source Directory and Upgrader Utility locations and make any necessary changes, then Click **Next**. A warning message is displayed.




6. Click **OK**. The Duplicating Machine window is displayed.




7. If desired, click **Save Report** or **Print Report**.
8. Click **Finish**.

## 5.3 System Information Viewer

The **System Information Viewer** consolidates all of the system information and enables you to view it in one place. It is extremely convenient in that it relieves you of having to activate numerous SPiiPlus MMI Application Studio tools to get the same information.

### 5.3.1 Activating the System Information Viewer

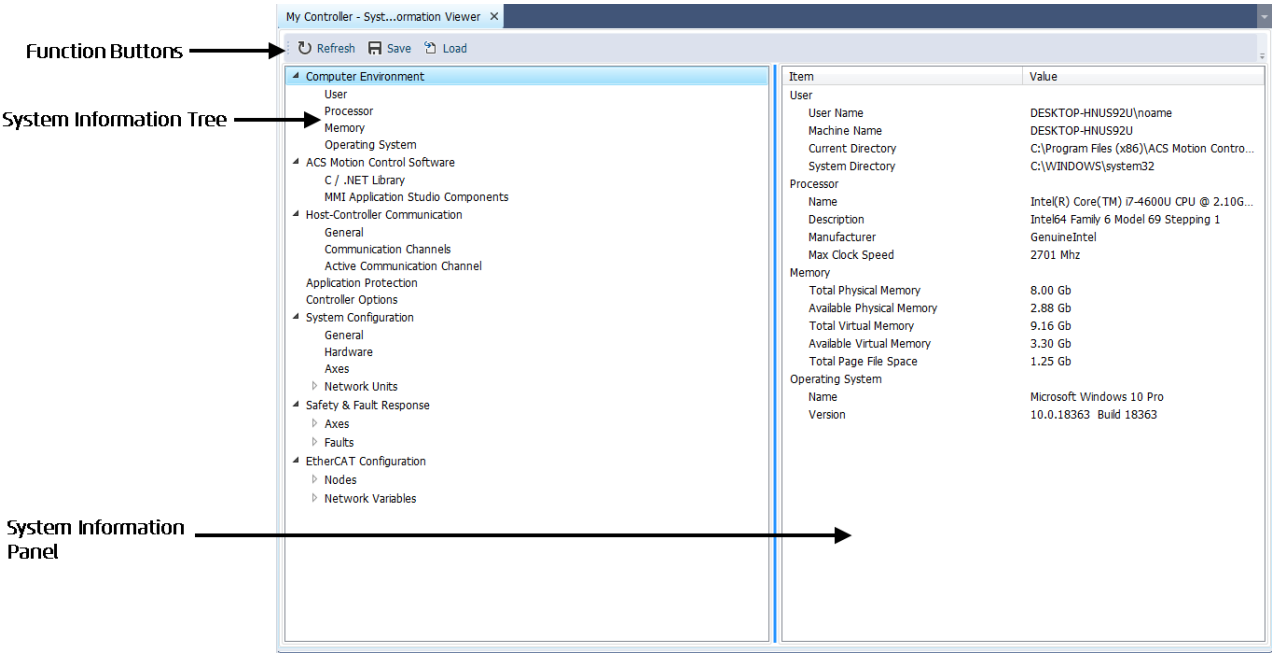
To activate the System Information Viewer:

1. From the Toolbox click **Utilities** to display the list of utilities.
2. Click **System Information Viewer** in the list of utilities.

You can also activate the **System Information Viewer** using the right-click **Add Component** option of the Workspace Tree.

The System Information Viewer window is displayed in the workspace.

### 5.3.2 System Information Viewer Window




The components of the System Information Viewer window are:

#### Functions

The System Information Viewer has the following function buttons:

| | |
|---|---|
| **Refresh** | Causes the System Information Viewer to re-read your system and update, if necessary, the information that is displayed.<br>There are two cases when you use this:<br>A change has been made to one or more of the components during operation and you want an updated version of the information.<br>You have loaded a previously saved snapshot of the System Information and you want to restore the correct information. |
| **Save** | Enables you to save a snapshot of the System Information to a file on the computer. |
| **Load** | Enables you to load the System Information snapshot that has been previously saved on your computer into the System Information Viewer. |

#### System Information Tree

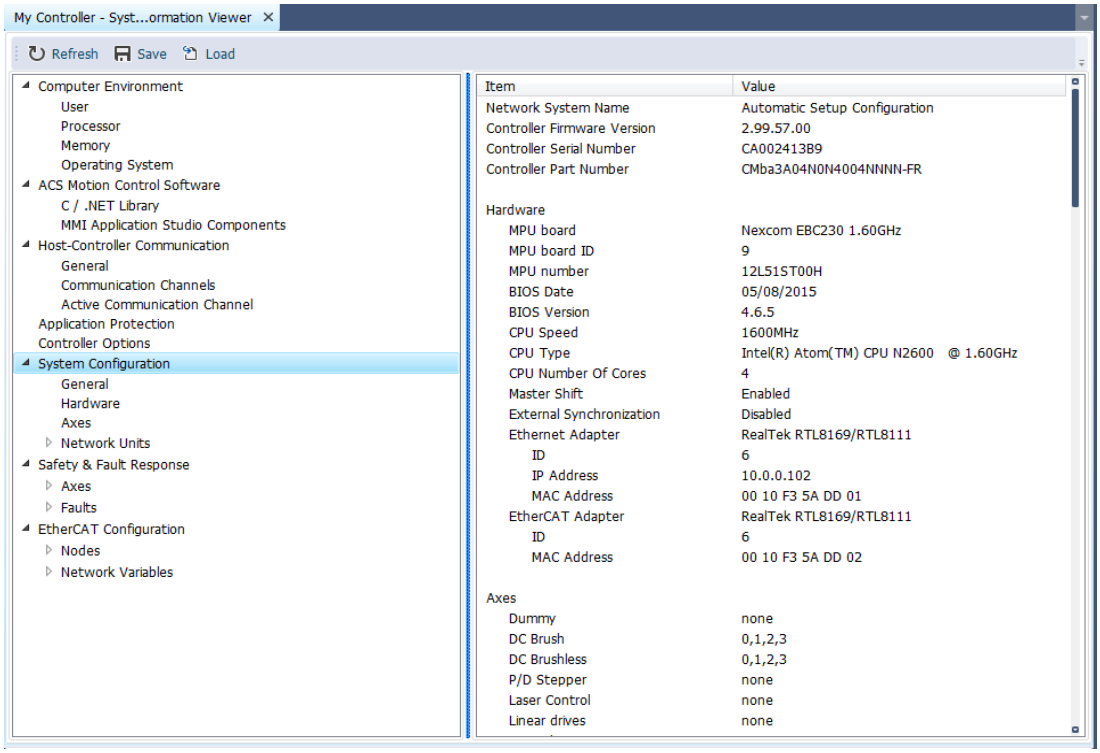
There are two different structures of the System Information Tree:

- EtherCAT Network System Information Tree - reflects a system that consists network elements.
- Non-EtherCAT Network System Information Tree - This structure reflects a system that is not network-based.

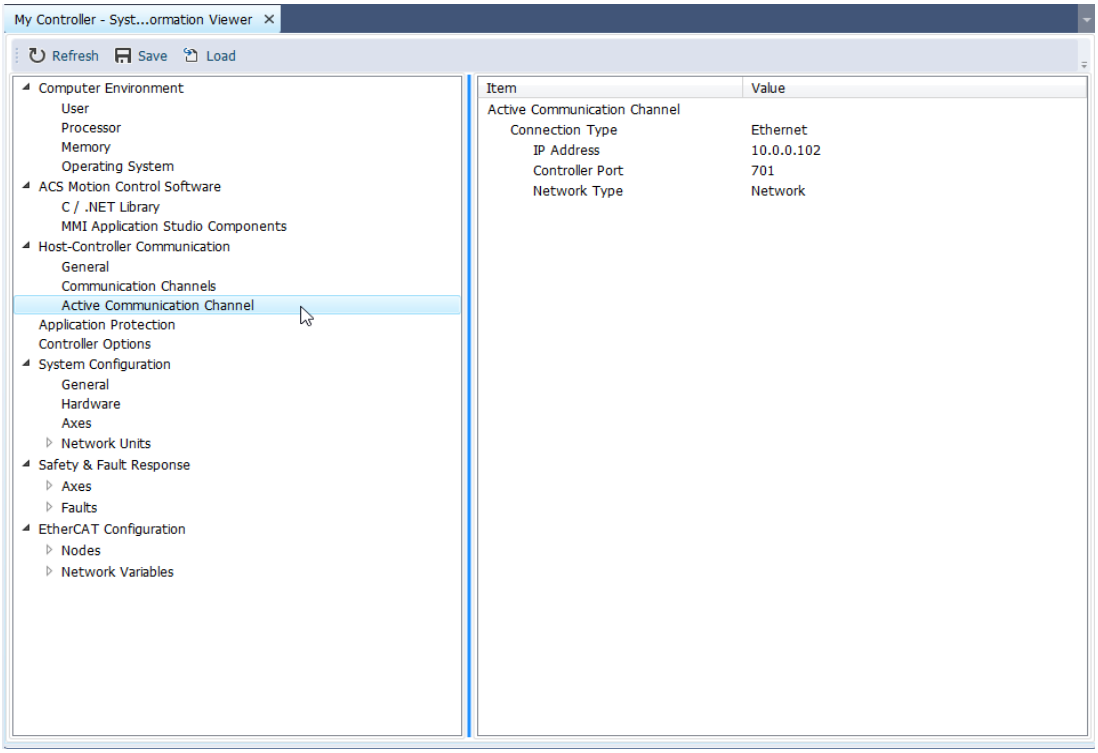
### System Information Panel

The System Information Panel displays the information for the selected item in the System Information Tree.

When you select a main branch of the System Information Tree, the System Information Viewer displays all the information for the branch, for example:




When you select a sub-branch, the System Information Viewer displays only the information for the sub-branch, for example:




### *5.3.3 EtherCAT Network System Information Tree*

An EtherCAT network-based System Information Tree has the following items:

- Computer Environment
- ACS Motion Control Software
- Host-Control Communication
- Application Protection
- Controller Options
- System Configuration

- Safety & Fault Response
- EtherCAT Configuration

#### *5.3.3.1 Computer Environment*

Computer Environment displays the host computer data, including:

- User - Name of the user currently logged into the system
- Processor - Displays the technical information of the host computer's processor
- Memory - Displays the amount used and amount available of the physical and virtual memories.
- Operating System - Displays the name of the operating system and version number

#### *5.3.3.2 ACS Motion Control Software*

ACS Motion Control Software displays:

- C/.NET Library - the most current versions of the C and .NETLibraries.
- MMI Application Studio Components - each tool and application making up the SPiiPlus MMI Application Studio.

#### *5.3.3.3 Host-Control Communication*

Host-Control Communication includes:

- General - displays the timeout setting and remote connection status
- Communication Channels - displays all of the available communication channels by channel number, type and mode
- Active Communication Channel - displays the currently active communication channel, including connection type, IP address, controller port number and network type

#### *5.3.3.4 Application Protection*

An application can have one of two states: Protected, that is, some variables have been protected from change, or Unprotected. See Protection Wizard for details.

#### *5.3.3.5 Controller Options*

Controller Options displays the status of various controller options that may exist, such as total number of axes, number of SIN-COS encoders, whether or not Input Shaping is on, number of non-ACS drivers are connected, and for each node that is connected, it displays the Unit ID and DIP switch settings (if exists).

#### *5.3.3.6 System Configuration*

System Configuration includes:

- General - Displays the name of the network, firmware version of the controller, serial and part number of the controller
- Hardware - Displays data of the hardware, including such things as the MPU, Ethernet adapter, and EtherCAT adapter.

- Axes - Displays a consolidated listing of all the axes that have been defined in the system, including their numbers, types of drives, operating current (continuous and peak), etc.
- Network Units

  For each unit in the network the following is provided:

  - General - Displays the unit's ID and DIP switch settings (if exists), part number, vendor ID, controller part number, vendor ID, network system name ID, revision, controller serial number, hardware ID, FPGA ID
  - Axes Assignments - Displays the axes numbers as assigned to the unit
  - Inputs/Outputs Assignments - Displays the values of the IN and OUT (Digital I/O), AIN and AOUT (Analog I/O), and EXTIN and EXTOUT (External I/O) variables along with the number of HSSI channels
  - Integrated Components

This appears only if the unit is one of the **MC4U** products.

  - Lists all the components that make up the unit, such as power supply(s), controller(s), and drivers, and displays the information for each component
  - Servo Processors - Displays the Servo Processors (SP) contained in the unit

#### *5.3.3.7 Safety & Fault Response*

This displays the following for each axis:

- The FAULT value for the selected axis.
- A list of the faults and the value of the selected fault for each axis.

#### *5.3.3.8 EtherCAT Configuration*

This provides the same information as running **#ETHERCAT** in the System Commands command in the **Communication Terminal**. It divides the information between the Node and the Input/Output Network Variables.

### *5.3.4 Non-EtherCAT Network System Information Tree*

A non-EtherCAT network-based System Information Tree has the following items:

- Computer Environment
- ACS Motion Control Software
- Host-Control Communication
- Application Protection
- Controller Options
- System Configuration
- Safety & Fault Response

#### 5.3.4.1 Computer Environment

Computer Environment displays the host computer data, including:

- User - Name of the user currently logged into the system
- Processor - Displays the technical information of the host computer's processor
- Memory - Displays the amount used and amount available of the physical and virtual memories.
- Operating System - Displays the name of the operating system and version number

#### 5.3.4.2 ACS Motion Control Software

ACS Motion Control Software displays:

- C/.NET Library - the most current versions of the C and .NET Libraries.
- MMI Application Studio Components - each tool and application making up the SPiiPlus MMI Application Studio.

#### 5.3.4.3 Host-Control Communication

Host-Control Communication includes:

- General - Displays the timeout setting and remote connection status
- Communication Channels - Displays all of the available communication channels by channel number, type and mode
- Active Communication Channel - Displays the currently active communication channel, including connection type, IP address, controller port number and network type

#### 5.3.4.4 Application Protection

An application can have one of two states: Protected, that is, some variables have been protected from change, or Unprotected. See Protection Wizard for details.

#### 5.3.4.5 Controller Options

Controller Options displays the status of various controller options that may exist, such as, number of SIN-COS encoders, whether or not Input Shaping is on, Servo Features mask, etc.

#### 5.3.4.6 System Configuration

System Configuration includes:

- General - Displays the name of the network if the controller is connected through a LAN, firmware version of the controller, serial and part number of the controller
- Hardware - Displays data of the hardware, including such things as the MPU, Ethernet adapter(s), controller card revision and number of Servo Processors.
- Axes - Displays the list of axes controlled by the controller, the type of drives that those axes can control, the operating current (continuous and peak) for integrated axes, and the PEG status that have been defined for the axes.
- Inputs/Outputs - Displays what axes have been defined for the IN and OUT (Digital I/O), and AIN and AOUT (Analog I/O) variables along with the number of HSSI channels

> Integrated Modules - Lists all the components that make up the unit, such as power supply (s), controller, and drivers, and displays the information for each component, including such things as type, axes associated with the component, voltage and amperage values, etc.

This appears only if the unit is one of the **MC4U** products.

> Servo Processors - Displays the Servo Processors (SP) contained in the unit with a brief description of the functionality of each SP.

#### *5.3.4.7 Safety & Fault Response*

This displays the following for each axis:

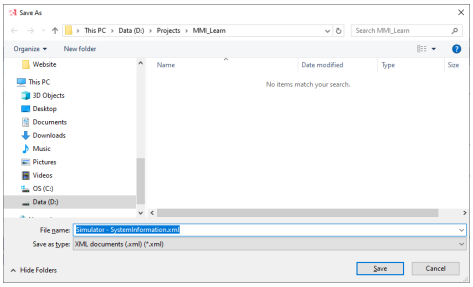
> The FAULT value for the selected axis

> A list of the faults and the value of the selected fault for each axis

### *5.3.5 Saving System Information*

To save the System Information to a file on your computer:

1. Click **Save** (). The browser is displayed.



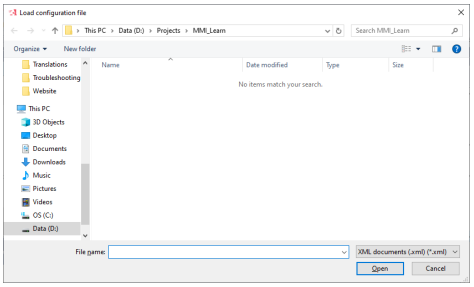

2. Browse to the location where you want to save the information and click **Save**.

The System Information data is saved in a file having an **.ini** file extension.

### *5.3.6 Loading System Information*

To load System Information that has been previously saved on your computer:

1. Click **Load**. The browser is displayed.




2. Browse to the location where you saved the information and click **Open**. The System Information stored in the file is then displayed in the System Information Viewer.

## 5.4 Communication Terminal

**Communication Terminal** enables you to enter commands directly to the controller.

There are two types of commands that can be entered through Communication Terminal:

> ASCPL+ Commands - valid ASCPL+ command or a string of ASCPL+ commands.

Only certain ACSPL+ commands may be strung together. The commands that you can enter in this way must be fairly simple. You cannot, for example, enter ACSPL+ loops or conditional commands.

> Terminal Commands - commands that are specific to the Communication Terminal and are not part of the ACSPL+ commands and variables set.

### 5.4.1 Accessing Communication Terminal

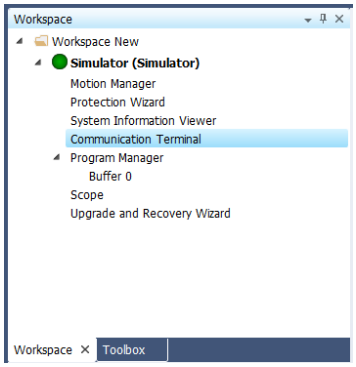
By default Communication Terminal is loaded when SPiiPlus MMI Application Studio is first activated. However, if you have closed the Communication Terminal window, or removed Terminal from the Workspace Tree, you can activate it again by:

> Access Communication Terminal from Workspace Tree

> Access Communication Terminal from Utilities

#### *5.4.1.1 Access Communication Terminal from Workspace Tree*

1. Display **Workspace**.

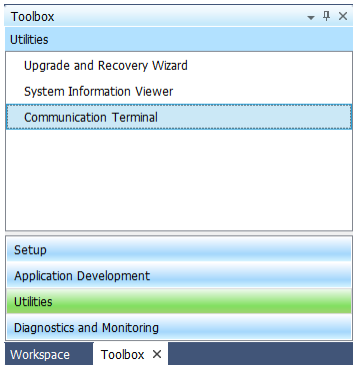



2. Click **Communication Terminal**.

### *5.4.1.2 Access Communication Terminal from Utilities*

Since the Communication Terminal is one of the SPiiPlus MMI Application Studio Utilities, it can be accessed by:

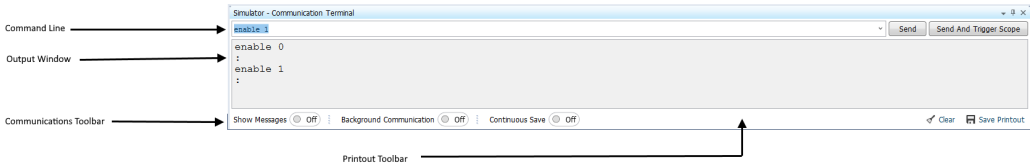
1. From the Toolbox click **Utilities**.
2. Click **Communication Terminal**.




You can also activate **Communication Terminal** using the right-click options of the Workspace Tree (see Controller Right-Click Options).

## *5.4.2 The Communication Terminal Window*

The Communication Terminal window consists of:




### Command Line

The Command Line is used for entering the commands. You can enter ASCPL+ commands or Communication Terminal Commands.

Certain ACSPL+ commands may be strung together using the semicolon (;) as a delimiter, for example:

**SET FPOS0 = 0;ENABLE X;PTP X, 10000**

In the example, the initial position of the X axis is set to 0, then the axis is enabled, and a point-to-point command is executed. If you were to enter the Communication Terminal query command, **?FPOS0** after sending the string, you would get a response of **10000**.

The commands that you can enter in this way must be fairly simple. You cannot, for example, use loops or conditionals.

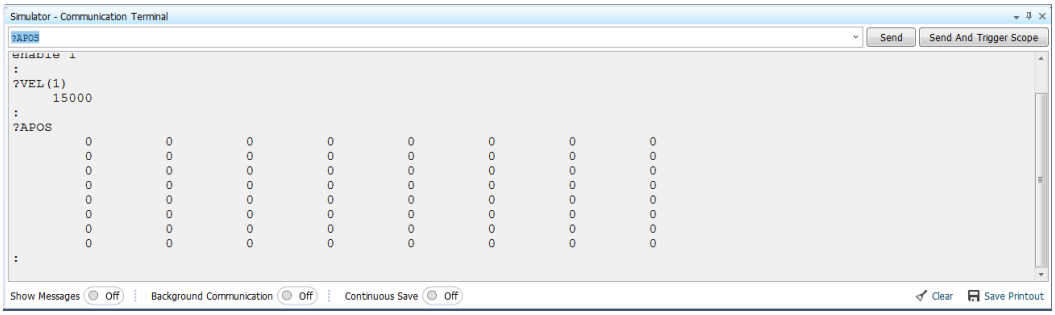
Communication Terminal commands, however, cannot be strung together; they must be entered individually.

Once you have typed in the command(s), press **Enter** to send the command, or click **Send**.

You also have the option of sending the command(s) and seeing the system reaction on the by clicking **Send And Trigger Scope**.

### Output Window

The Output Window displays what you have sent and reactions to Communication Terminal commands. For example;




### Communications Tools Bar

You have the following Communication options available to you.

- **Show Messages** - a program that is running in the controller may have display commands (such as **DISP**) that you might want to see, in which case you click this and the messages are displayed.
- **Background Communication** - selecting this option causes the display of the ongoing communication checks that Communication Terminal conducts with the controller.

Currently **Exclusive Communication** is not supported.

**Printout Tools Bar**

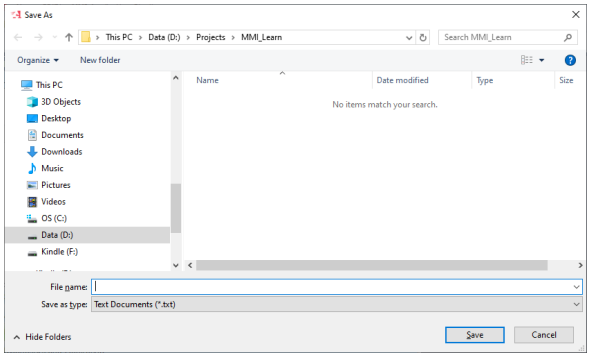
You have the following Printout options available to you.

- **Clear** - selecting this clears the Output Window.
- **Save Printout** - selecting this enables you to save the contents of the Output Window to a text file.
- **Continuous Save** - selecting this enables you to open a text file and Communication Terminal will save the running contents of the Output Window until you close Communication Terminal.

### *5.4.3 Saving Communication Terminal Command Output to File*

To save the output of a Communication Terminal command:

1. Run the Communication Terminal command.
2. Click **Save Printout**, a browser window is displayed:




3. Use the browser to select where you want to store the file, enter a **File Name** and click **Save**.

If you select **Continuous Save** instead of **Save**, Communication Terminal displays the same browser. Once you have designated a file to which Terminal is to write the command output, Communication Terminal appends the output of each command you enter to the file. When you close Communication Terminal, the file is closed.

### *5.4.4 Communication Terminal Commands*

Communication Terminal commands are divided into:

- Query Commands
- Program Management Commands
- System Commands

#### *5.4.4.1 Query Commands*

Query commands are designated by the question mark (**?**).

| Command & Syntax | Description |
|---|---|
| **?**[ACSPL+ variable name] | Returns the current value of a ACSPL+ variable. The ACSPL+ variable name must be entered in all CAPs. |
| **?**[ACSPL+ variable name], [ACSPL+ variable name]... | Returns the current value of listed standard variables. The ACSPL+ variable name must be entered in all CAPs. |
| **??**[ACSPL+ variable name] | Returns a brief description of the ACSPL+ variable. The ACSPL+ variable name must be entered in all CAPs. |
| **?**[buffer number]: [local user-defined variable] | Returns the current values of a local user variable or array defined in a program buffer. The variable name must be entered exactly as it appears in the program. |
| **?**[array_variable (index)] | Returns the current value of a specific element in the given array. The brackets enclosing the index are optional. If it is an ACSPL+ array variable, it must be entered in all CAPs. |
| **?**[global user array_variable [(index)]] | Returns the current values of a global user variable or array.<br>Or the value of an element in the array if index is included. |
| **?**[matrix_variable (row_index) (col_ index)] | Returns the current value(s) of an element in a two dimensional matrix.<br>The indices may be entered as a range, e.g., (0,4) is the first through fifth, inclusively. |
| **?**[buffer number] | Returns the current status and information about a program buffer |
| **?#** | Returns the current status of all program buffers. |
| **?$** | Returns the current status of all motors. |
| **?VR** | Returns the firmware version |
| **?SN** | Returns the controller serial number and hardware version - indicated by a letter. |
| **??**[error code] | Returns the error code number. If the error code is included in the query, returns error description. |

By adding tags you can determine the format of the response to the Query.

| Format | Description |
|---|---|
| ?D/ | Decimal format.<br>This format is identical to the default format for integer variables. When applied to a state variable, the format displays the decimal presentation of the variable. |
| ?X/ | Hexadecimal format.<br>When applied to an integer variable, this format displays the hexadecimal presentation of the variable. |
| ?B/ | Binary format.<br>This format is identical to the default format for the **IN** and **OUT** variables. When applied to an integer variable, the format displays the binary presentation of the variable. |
| ?E/ | Extended format.<br>This format is useful for very large or small real values, when the default format produces ambiguous results because the default does not provide enough positions to display very large or very small numbers. When applied to a real variable, the format displays each value in 20 positions. |

You also have the option specifying a format using C notation. The specification is placed just before the variable name in curly brackets. The specification applies only to the value of the name that is included in the command. If an array is queried, the specification applies to each element of the array.

C notation provides an unlimited number of possible formats. Some examples are:

| Format | Description |
|---|---|
| ?{%12.3f}FPOS | The motor feedback values for all axes are displayed in 12 digits, fixed decimal point, 3 digits after the point. The same format applies to all 8 values. |
| ?{%8.0f}X_FPOS | 8 digits, no decimal point, no fraction digits. |
| ?{XFPOS = %8.0f}X_FPOS | The response will look like **XFPOS = 1234.** |
| ?{%08X} X_MST | 8 digits, hexadecimal format with leading zeros |

### *5.4.4.2 Program Management Commands*

Program Management commands are used for:

> Buffer Handling

- Program Variables Handling
- Program Handling
- Debug Handling

The Program Management commands are designated by the pound (**#**) character.

In those commands including a line designation, the line designation can appear in one of the following forms:

- Single number - Specifies only one line.
- Two numbers separated by comma - Specifies a range of lines. If the second number is larger than a total number of lines in the program, the list range spans the last program line.
- Label preceded by a slash (/) character - Specifies the line containing the designated label.
- Label preceded by a slash (/) character, then a comma and number - Specifies a range of lines starting from the line with the designated label.

Items enclosed in square brackets, for example, **[buffer_no]**, are optional.

*5.4.4.2.1 Buffer Handling*

These commands enable you to open buffers for entering program commands, searching the buffer contents, deleting contents from the buffer, and the like. The buffer commands are:

All Terminal commands are case-sensitive and must be entered as listed.

- # - Open/Close buffer
- D - Delete
- F - Find
- FI - Case-sensitive Find

| Command | Syntax | Description |
|---|---|---|
| **#** | #buffer_no [I][line_no] | The **#** (Open/Close Buffer) command is used to open and close a buffer for the purpose of entering code. Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10.<br>> **I -** Optional, if included, opens the buffer for insertion of code. If omitted, the command closes the buffer.<br>> **line_no** - Optional, if included, the command opens the buffer specified by **buffer_no** and sets the insert line before the line specified by **line_no**. |
| **D** | #buffer_ noDline_no [,line_no] | The **D** (Delete) command deletes the specified lines in the buffer. Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10.<br>> **line_no** - Specifies the line to delete. If a second **line_no** (separated by a comma) is included, it specifies a range of lines. If the second number is larger than the total number of lines in the program, the delete range includes the last program line.<br>If a buffer is open, the **D** command that addresses the buffer shifts the insert line to before the first undeleted line. |

| Command | Syntax | Description |
|---|---|---|
| **F**<br><br>**FI** | #buffer_noF/search_string [,line_no]<br><br>#buffer_noFI/search_string [,line_no] | The **F** and **FI** (Find/Find Case-sensitive) commands are used to search for specific text in a specified buffer or in all buffers. Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10. If **#** is used, the search is conducted throughout all buffers.<br>> **search_string** - The text being sought.<br>> **line_no** - Optional, if included, **line_no** defines the start line for the search. Otherwise, the search starts from the first line.<br>**search_string** in search commands can be a label or a label and number separated by comma. The label in the **search_string** is not, however, limited to labels in the program, any name may be specified. Therefore, the command can be used to search for any variable, constant, label or keyword.<br>The search terminates when the first entry of the specified text is found, or the buffer end is reached. The command reports the line that contains the text, or an error message if the text was not found.<br>To find the next entry, you have to execute the command again, specifying the new start line number of the reported line plus one. |

*5.4.4.2.2 Program Variables Handling*

There are three types of variables:

- ACSPL+ Standard Variables – variables built into ACSPL+
- SP Variables – variables built into the controller
- User-Defined Variables – variables that you have declared in your program

These commands enable you to list variables. The commands are:

All Communication Terminal commands are case-sensitive and must be entered as listed.

- VS - Display ACSPL+ variable names
- VSG - Display ACSPL+ variable names within ACSPL+ variable name categories
- VSF - Display of ACSPL+ variable names and attributes
- VSGF - Display of ACSPL+ variable names and attributes
- VSP - Display SP variables

- VGR - Display categories within which ACSPL+ variables are grouped
- VSD - Display brief description of ACSPL+ variables
- VG - Display list of all global variables
- VGF - Display all global variables and attributes
- VL - Display local variables
- VLF - Display local variables and attributes
- V - Display user-defined global and local variables
- VF - Display of user-defined global and local variables and attributes
- L - List program
- VST - Display ACSPL+ variables which can be protected using the Protection Wizard
- VSGT - Display ACSPL+ variables for ACSPL+ variable categories which can be protected using the Protection Wizard
- VSTF - Display ACSPL+ variables which can be protected using the Protection Wizard and their attributes
- VSGTF - Display ACSPL+ variables for ACSPL+ variable categories which can be protected using the Protection Wizard and their attributes
- VSDT - Display ACSPL+ variables for ACSPL+ variable categories which can be protected using the Protection Wizard and their attributes
- RESET - Reset program variables to factory defaults
- VGV - Remove global variables set through **Terminal**

| Command | Syntax | Description |
|---|---|---|
| **VS**<br>**VSG** | #VS<br>#VSG [group_ name] | Both the **VS** and **VSG** commands display the names of the ACSPL+ standard variables.<br>**group_name** is the name of the category within which ACSPL+ variables are grouped (for the categories see VGR). When **group_name** is added to the **VSG** command, the names of the ACSPL+ standard variables within the specified category are listed. **group_name** is case-sensitive and must be entered exactly as given in the list displayed by **VGR**, for example, **#VSG Axis_State**. |
| **VSF**<br>**VSGF** | #VSF<br>#VSGF [group_ name] | Both the **VSF** and **VSGF** commands, in addition to the variable names, display the variable type, the number of elements (for arrays only), address of the variable in the controller memory and the step between array elements (for arrays only). |

| Command | Syntax | Description |
|---|---|---|
| | | **group_name** is the name of the category within which ACSPL+ variables are grouped (for the categories see #VGR. When **group_name** is included in the **VSGF** command, the names of the ACSPL+ standard variables within the specified category and their details are listed. **group_name** is case-sensitive and must be entered exactly as given in the list displayed by **VGR**, for example, **#VSGF Axis_State**. |
| **VSP** | #VSPservo_number | The **VSP** (List Servo Processor Variables) provides a list of SP variables that are defined in the program in the specified SP.<br>**servo_number** is an integer number, 0, 1, 2, or 3, specifying the SP.<br>Each variable name in the list is accompanied by an SP address of the variable. |
| **VGR** | #VGR [group_name] | The **VGR** command, without **group_name**, lists the categories within which the ACSPL+ variables are grouped. These are:<br>> Axis_State<br>> Monitoring<br>> Motion<br>> Safety_Control<br>> Inputs_Outputs<br>> Program_Execution_Control<br>> System_Configuration<br>> Axis_Configuration<br>> Communication<br>> Commutation<br>> Data_Collection<br>> Servo_Loop<br>> Miscellaneous<br>> Obsolete<br>**group_name**, when included, is the category name and must be one of the above. It is case-sensitive and must be entered exactly as given in the list, for example, **#VGR Axis_State**. When included in the command, all ACSPL+ variables within the category is listed. |

| Command | Syntax | Description |
|---|---|---|
| **VSD** | #VSD [group_ name] | The **VSD** command, without **group_name**, lists all ACSPL+ variables with a short description.<br>**group_name** is the name of the category within which ACSPL+ variables are grouped (for the categories see VGR). When **group_name** is added to the **VSD** command, the ACSPL+ variables within the specified category and a brief description of each variable is listed. **group_name** is case-sensitive and must be entered exactly as given in the list displayed by **VGR**, for example, **#VSD Axis_State** |
| **VG**<br>**VGF** | #[buffer_ no]VG<br>#[buffer_ no]VGF [variable_ name] | The **VG** command lists all global variable names in the system.<br>If **buffer_no** (a number between 0 and 10) is included, **VG** lists all the global variables in the specified buffer.<br>The **VGF** command, in addition to the global variable names, lists the variable type, the number of elements (for arrays only), address of the variable in the controller memory and the step between array elements (for arrays only).<br>If **buffer_no** (a number between 0 and 10) is included, **VGF** lists the details for all the global variables in the specified buffer.<br>If **variable_name** specifying one of the global variables is included, **VGF** lists the details just for the specified variable. |
| **VL**<br>**VLF** | #[buffer_ no]VL<br>#[buffer_ no]VLF [variable_ name] | The **VL** command lists all local variable names in the system. If **buffer_no** (a number between 0 and 10) is included, **VL** lists all the local variable names in the specified buffer.<br>The **VLF** command, in addition to the local variable names, lists the variable type, the number of elements (for arrays only), address of the variable in the controller memory and the step between array elements (for arrays only).<br>If **buffer_no** (a number between 0 and 10) is included, **VLF** lists the details for all the local variables in the specified buffer.<br>If **variable_name** specifying one of the local variables is included, **VLF** lists the details just for the specified variable. |
| **V**<br>**VF** | #[buffer_ no]V<br>#[buffer_ no]VF | The **V** command lists all user-defined variable names in the system.<br>If **buffer_no** (a number between 0 and 10) is included, **V** lists all the user-defined variables in the specified buffer. |

| Command | Syntax | Description |
|---|---|---|
| | | The **VF** command, in addition to the variable names, lists the variable type, the number of elements (for arrays only), address of the variable in the controller memory and the step between array elements (for arrays only).<br>If **buffer_no** is included, **VF** lists the details of all the user-defined variables in the specified buffer. |
| **L** | #buffer_noL[line_no] | The **L** (List) command is used for displaying a program listing.<br>The listing contains all program lines preceded by line numbers. Each line appears exactly as it was inserted. No automatic formatting is provided.<br>To address all buffers the **#** character is used instead of the buffer number, for example, the command **##L** provides a listing of all programs in all buffers.<br>If **line_no** is included, **L** lists the contents of the specified line. If a buffer does not contain the specified line number, only the buffer number is listed.<br>If the buffer is empty, the list includes the buffer designation followed by the first line (0) which is blank. |
| **VST**<br>**VSGT** | #VST<br>#VSGT [group_name] | Both the **VST** and **VSGT** commands display a list of ACSPL+ variables to which **Program Variables Handling can be applied.**<br>However, when **group_name**, the name of the category within which ACSPL+ variables are grouped (for the categories see VGR), is included with the **VSGT** command, the ACSPL+ variables within the specified category are listed. **group_name** is case-sensitive and must be entered exactly as given in the list displayed by **VGR**, for example, **#VSGT Axis_State**. |

| Command | Syntax | Description |
|---|---|---|
| **VSTF**<br><br>**VSGTF**<br><br>**VSDT** | #VSTF<br>#VSGTF [group_ name]<br>#VSDT [group_ name] | The **VSTF**, **VSGTF**, and **VSDT** commands all list the variable names, the variable type, the number of elements (for arrays only), address of the variable in the controller memory and the step between array elements (for arrays only) of those ACSPL+ variables to which **Program Variables Handling can be applied.**<br>However, when **group_name**, the name of the category within which ACSPL+ variables are grouped (for the categories see VGR), is included with the **VSGTF** or **VSDT** command, the ACSPL+ variables within the specified category are listed. **group_name** is case-sensitive and must be entered exactly as given in the list displayed by **VGR**, for example, **#VSGTF Axis_State**. |
| **RESET** | #RESET | The **RESET** command is used to reset the controller to factory default state.<br>The **RESET** command can be issued even if the application is in the Protected mode (see Program Variables Handling), in which case the password, if included, is not needed. |
| **VGV** | #VGV [global_var] | The **VGV** command is used to remove global variables that have been set via **Terminal**. If **global_var** (the name of the variable) is included, it removes only this variable; otherwise it removes all of them. |

*5.4.4.2.3 Program Handling*

These commands are used for compiling, executing, pausing and halting the program. The commands are:

All Communication Terminal commands are case-sensitive and must be entered as listed.

- C - Compile program
- X - Execute program
- S - Stop program
- SR - Stop and reset program
- P - Pause program

| Command | Syntax | Description |
|---|---|---|
| **C** | #[buffer_no]C | The **C** (Compile) command compiles a program in the buffer or all programs in all buffers, depending on the buffer qualifier.<br>Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10.<br>The **C** command must not include a line qualifier and is prohibited when the buffer is in the Run or Suspended states.<br>The **C** command is not obligatory in order to execute a program. When the X (Execute) command is issued, the controller automatically compiles the program if it was not previously compiled. However, a separate compilation step is required in the following cases:<br>> To check the program correctness without executing it.<br>> The program is not intended for direct starting, but contains autoroutines. The autoroutines are ready for execution only after compilation.<br>> The program is intended for starting from another program by the START command. The program started by the START command must be compiled before the START command can be executed.<br>If the program is successfully compiled, the Terminal displays a short report of how many lines were compiled. If an error was encountered, Terminal reports the error code and the line number in which the error was found. |

| Command | Syntax | Description |
|---|---|---|
| **X** | #buffer_noX [line_no] | The **X** (Execute) command starts a program in a specific buffer, and can be executed in any program state except the Run state. Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10. It must specify one buffer only.<br>> **line_no** - Can be a line number or a label. Execution starts from the specified line. If included, **line_no** defines a specific line to be listed. Otherwise the program starts from the first line.<br>If the state of the program is Not Compiled, the controller first compiles the program and then starts it. If an error was encountered during compilation, the program does not start.<br>If the state of the program is Suspended, the **X** command resumes the program execution. In this case the command must not contain **line_no** because execution resumes from the point where the program was suspended. |
| **S**<br>**SR** | #[buffer_no]S<br>#[buffer_no]SR | The **S** (Stop) command terminates program execution in a buffer or execution of all programs in all buffers.<br>The **SR** (Stop and Reset) command terminates program execution in a buffer or execution of all programs in all buffers, and resets the buffer or all buffers to the Not Compiled state. The command provides the decompile function, which is useful if the program contains autoroutines that are ready to start when the buffer is in Compiled state. Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10. It must specify one buffer only.<br>If **buffer_no** is omitted, this will stop, or stop and reset all programs in all buffers, or the **#** character may be used as the **buffer_no**, for example, **##S**, which will do the same.<br>Program termination commands must not include line qualifiers.<br>The **S** command can be issued in any program state.<br>The issuance of the **SR** command effectively prevents the activation of autoroutines. |

| Command | Syntax | Description |
|---|---|---|
| **P** | #buffer_noP | The **P** (Pause) command suspends program execution in a buffer. Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10. It must specify one buffer only.<br>The # character may be used instead of a buffer number, for example, **##P**, in which case the execution of all programs in all buffers is suspended.<br>The **P** command must not include a line qualifier.<br>**P** commands are allowed in any program state, but in all states, other than Run, the command has no effect.<br>If the program is in the Suspended state, the **X** command resumes execution. The S command transfers the buffer to the Compiled state. The SR command transfers the buffer to the Not Compiled state. |

*5.4.4.2.4 Debug Handling*

The Debug commands enable you to run your program in the Debug Mode, set breakpoints and to run sections of the program. The commands are:

All Communication Terminal commands are case-sensitive and must be entered as listed.

> XS - Execute one step
> XD - Execute program in Debug mode
> BS - Set breakpoint
> BR - Reset breakpoint

| Command | Syntax | Description |
|---|---|---|
| **XS** | #buffer_noXSline_no | The **XS** (Execute One Step) command executes one program line. Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10. It must specify one buffer only.<br>> **line_no** - Can be a line number or a label. Indicates what line to run.<br>After executing the specified **line_no**, the buffer automatically enters the Suspended state. |
| **XD** | #buffer_noXD | The **XD** (Execute in Debug Mode) command executes the program up to the next breakpoint. Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10. It must specify one buffer only. |
| **XD (CONTINUED)** | | The command is similar to the **X** command. The difference is that the **X** command ignores breakpoints in the program. If the program is started by the **XD** command, it will stop when it reaches a breakpoint. At the breakpoint the program transfers to the Suspended state and can be started again by the **X**, **XS** or **XD** commands. |
| **BS** | #buffer_noBS [line_no] | The **BS** (Set Breakpoint) command sets a breakpoint at the specified line. Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10. It must specify one buffer only.<br>> **line_no** - Can be a line number or a label. Indicates in what line to set the breakpoint.<br>You can set any number of breakpoints in your program; however, for breakpoints to be active, you have to start the program with the **XD** command. |

| Command | Syntax | Description |
|---|---|---|
| **BR** | #buffer_noBR [line_no] | The **BR** (Reset Breakpoint) command resets the breakpoint at the specified line or all breakpoints. Where:<br>> **buffer_no** - Specifies the buffer, a number between 0 and 10. It must specify one buffer only.<br>> **line_no** - Can be a line number or a label. Indicates in what line to reset the breakpoint. If it is omitted, the command resets all breakpoints in the buffer.<br>If the buffer qualifier is specified as **#**, for example, **##BR**, and **line_no** is omitted, the command resets all breakpoints in all buffers. |

### *5.4.4.3 System Commands*

The System commands provide you with information contained in the system. The commands are:

All Communication Terminal commands are case-sensitive and must be entered as listed.

> IR - Integrity Report
> U - System Usage Report
> TD - User-defined Variables in System
> SC - Safety Control Report
> SI - System Information
> SIR - System Information Report
> ETHERCAT - Network Information

| Command | Syntax | Description |
|---|---|---|
| **IR** | #IR | The **IR** (Integrity Report) command activates integrity validation and provides a report of current integrity state. The report displays a list of files. Each list entry displays a file name, expected file size and checksum of the file and actual file size and checksum.<br>If any integrity problem is detected, the command sets **S_FAULT.#INTGR** fault bit. You can view this by using the Query Commands. |

| Command | Syntax | Description |
|---|---|---|
| **U** | #U | The **U** (Usage) command is used for monitoring MPU usage. It returns the maximum, average, and minimum values as a percent.<br><br>The controller continuously measures the time taken by real-time tasks. When the **U** command is received, the controller analyzes the measured times during the last 50 controller cycles and calculates minimal, maximal and average time. The results are reported as a percentage. |
| **TD** | #TD | The **TD** command returns the names of all user-defined variables and arrays that are in the controller flash memory. |
| **SC** | #SC | The **SC** (Safety Control) command reports the current safety system configuration.<br>The controller response includes the following:<br>Active safety groups<br>The configuration of each fault for each motor |
| **SI** | #SI | The **SI** (System Information) command returns system information about the SPiiPlus controller including serial number, firmware version, configuration, name and SP programs. |

| Command | Syntax | Description |
|---|---|---|
| **SIR** | #SIR/section/key/ | The **SIR** (System Information Report) command is an expanded version of the **SI** command and returns system information broken down into various categories (**section**) and subjects (**key**) within the categories.<br>**section**:<br>> **Hardware** - This section contains information about the controller's hardware<br>> **Options** - This section contains information about the controller's options, such as, total number of axes, encoder support, number of non-ACS unit servo axes, etc.<br>> **Network** - This section contains information about the controller's EtherCAT channels.<br>> **Axes_support** - This section contains information about the features that each axis has.<br>> **UNIT#** - There is a UNIT section for each unit in the system, they are numbered from 0 up to the number of units minus 1, for example, UNIT0 is the first unit in the system. Each UNIT section contains information specifically about the unit.<br>> **key** depends on the category. The format of the return is:<br>[Section]<br>Key = value<br>To see all sections and keys enter:<br>#SIR/ALL/ALL/ |
| **ETHERCAT** | #ETHERCAT | The **ETHERCAT** command is used for obtaining complete information about the devices connected in the network.<br>The information it displays is:<br>> Slave number<br>> Vendor ID<br>> Product ID<br>> Revision |

| Command | Syntax | Description |
|---|---|---|
| | | > Serial number<br>> EtherCAT physical address<br>> DC support<br>> Mailbox support<br>This is followed by a list of network variables associated with the device. Each variable is described with:<br>> Name (as in XML)<br>> Offset inside the telegram (magic number that is used for mapping)<br>> IN or OUT description<br>> Data size<br>> PdoIndex |

# 6. Diagnostics and Monitoring

Diagnostics and Monitoring tools enable you to view the status of various elements in the controller's program and aid you in analyzing the motion.

You can use the following tools for viewing the values and status of critical variables:

- Inputs/Outputs Manager - The Inputs/Outputs Manager tool displays the status of the digital input and output variables of the SPiiPlus controller as well as those of the HSSI, if defined.
- Safety and Faults Monitor - displays the status of the SPiiPlus controller faults as defined through the Safety and Faults Configurator.
- Variables Manager and Watch - displays the values currently stored in variables of a program that is running.
- HSSI Manager - used for monitoring the status of HSSI channels. The manager displays the quantity and state of the HSSI channels.

The following windows provide graphic representations:

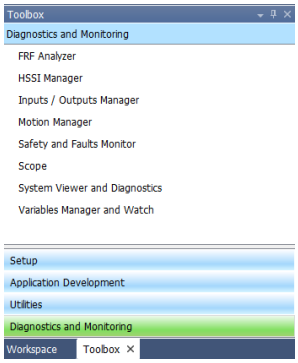
- Scope - presents a real time display of the selected motion or variable. You can set the Scope to display any parameters you like.
- FRF Analyzer - is the steady-state transfer function of a dynamic system describing the relation between an input and an output as a function of frequency in terms of gain and phase. You have the option of displaying the data as either a Nyquist Diagram or a Bode Diagram.
- Sin Cos Encoder Compensation and Monitoring - displays a Sin-Cos Lissajous curve graph of the wave form and enables you to make adjustments to controller's gain, phase and offsets compensation parameters (**SCGAIN**, **SCPHASE**, **SCSOFFS**, and **SCCOFFS**). Sin Cos Encoder Compensation automatically calculates best fits for the parameters.

Diagnostics and Monitoring also provides you with Motion Manager to setup the motion for each axis and monitor motion status.

## 6.1 Accessing Diagnostics and Monitoring Tools

To access the Diagnostics and Monitoring tools:

1. If the Toolbox is not displayed, click **Toolbox**.
2. Click **Diagnostic and Monitoring**.




3. Click the required tool.

## 6.2 FRF Analyzer

The **FRF** (Frequency Response Function) **Analyzer** provides you with a powerful tool for servo analysis of the ACS Motion Control SPiiPlus motion controllers. It allows mechanical characterization of the machine, and is a valuable aid in tuning your machine.

- With this tool you can:
- Obtain the mechanical signatures of the machine.
- Identify and compensate for mechanical resonances.
- Expand machine bandwidth and improve performance.
- Improve machine stability.
- Compare between different machines.

FRF measurements can be made under the following conditions:

- Brushless motor should be commutated. The FRF Analyzer does not allow FRF measurement for non-commutated brushless motor or stepper motors.
- Axis must be enabled. In the axis is disabled it will be automatically enabled when FRF measurement begins.
- Axis is not in open loop. You should not be confused with the fact that Open-Loop FRF and Plant FRF are measured as well. All FRF measurements are performed in a closed loop. For this purpose the axis must be adjusted before performing the measurement.

It is recommended using the Adjuster Wizard to make preliminarily adjustments to the axis with a relatively low bandwidth. After the FRF measurements are taken, you can more knowledgeably set the servo parameters and adjust the filters to maximize the performance and the bandwidth, while remaining within adequate stability margins.

Do not perform FRF Measurement when the servo loop is not stable or with very marginal tuning!

Sometimes it is preferable to do the measurement while the axis is moving. This is typically the case, when it is required to overcome the non-linear effect of coulomb friction. In this case it is recommended to write an ACSPL+ program in one of the buffers that takes care of the required motion. The following types of motion can be used during FRF measurement:

> Axis is idle, no motion is executed

> Jog motion – when the travel is not limited

> PTP motion – when the travel is limited (i.e., linear motor)

In case of PTP or JOG, it is recommended moving at very low velocity, which is enough to overcome the coulomb friction.

A disturbance in the FRFs may appear if PTP is used while the axis is changing its direction. To minimize this effect, it is recommended performing cyclic motion between two points without any dwell time between motions.

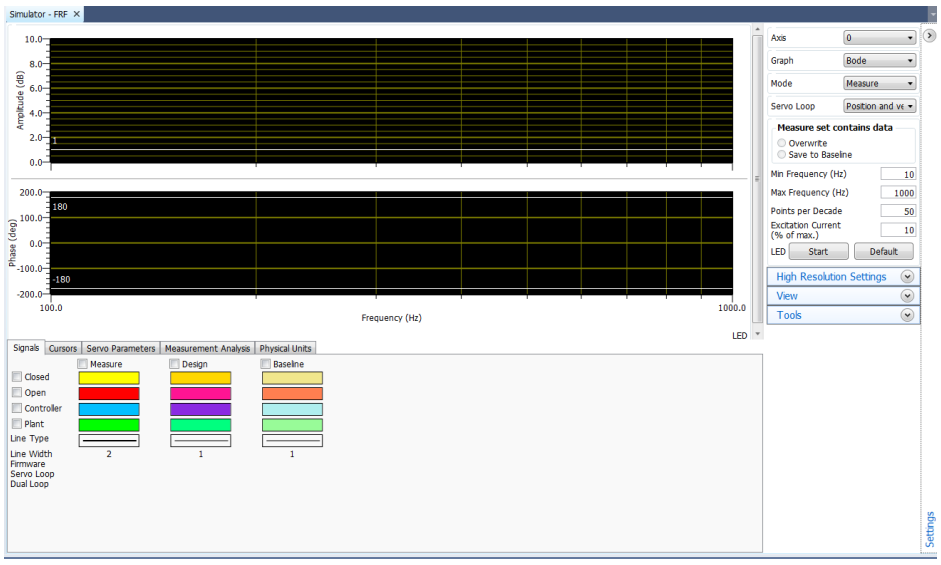
### 6.2.1 Activating FRF Analyzer

To activate the FRF Analyzer:

1. From the Toolbox click **Diagnostics and Monitoring** to display the Diagnostics and Monitoring tools list.
2. Click **FRF Analyzer**.

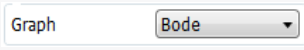
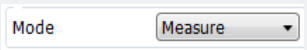
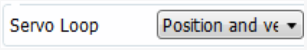
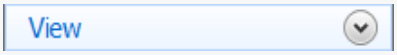
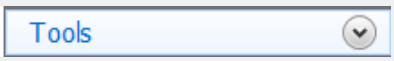
You may use **Add Component** from the right-click menu of the controller in the Workspace Tree (Controller Right-Click Options ) to activate the **FRF Analyzer**.

The FRF Analyzer window appears.

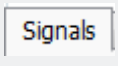
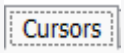



### 6.2.2 Working with FRF Analyzer

The right side of the window contains the following option controls:

| | |
|---|---|
|  | Used for selecting the axis to be measured. See Selecting Axis. |
|  | Used for selecting the FRF Analyzer display format. See Selecting Graph Format. |
|  | Used for setting the FRF Analyzer mode. The FRF Analyzer has two modes:<br>> Measure (see Measure Mode)<br>> Design (see Design Mode) |
|  | Used for setting the Servo Loop to:<br>> Position & Velocity<br>> Current<br>> Velocity<br>> Position<br>> Open |
|  | |
|  | Used for selecting:<br>> Markers (see Markers)<br>> Phase Modulo<br>> Amplitude<br>for display. See Setting View Options. |
|  | Used for:<br>> Saving FRF data to a file<br>> Retrieving FRF data from a file<br>> Clearing FRF data from the display<br>See Tools. |

Underneath the FRF graph there is a section for selecting what is displayed and for viewing the measured values. This section is divided into the following tabs:

| | |
|---|---|
|  | This tab enables the user to select the signals you want displayed. See Signals Tab. |
|  | This tab enables the user to set cursors in the graph in order to view the values at points of your choosing. See Cursors Tab. |

| | |
|---|---|
| Servo Parameters | This tab enables the user to view the values of servo parameters. In the Design mode change the user can set the values of Servo Parameters. See Servo Parameters Tab. |
| Measurement Analysis | This tab enables the user to perform measurement analysis. See Measurement Analysis. |
| Physical Units | This tab enables the user to set the physical measurements used in the analysis. See FRF Physical Units. |

### High Resolution Settings

This section allows the user to control the display resolution. Checking the "Enable" checkbox activates the parameter input controls. The start frequency can be set using the text box or the slider. Text boxes also control the number of points per decade and the excitation current. The "Default" button resets all parameters to their default values.

### Selecting Axis

You select the axis to be measured from the **Axis** dropdown list. The list of axes is extracted from the controller.

To display the list click .

### Selecting Graph Format

There are two graphic formats from available to you:

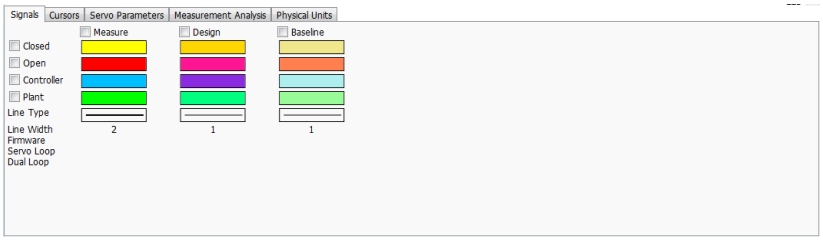
- Bode Diagram
- Nyquist Diagram

You select the graphic format from the **Graph** dropdown list which you display by clicking .

## *6.2.3 Signals Tab*

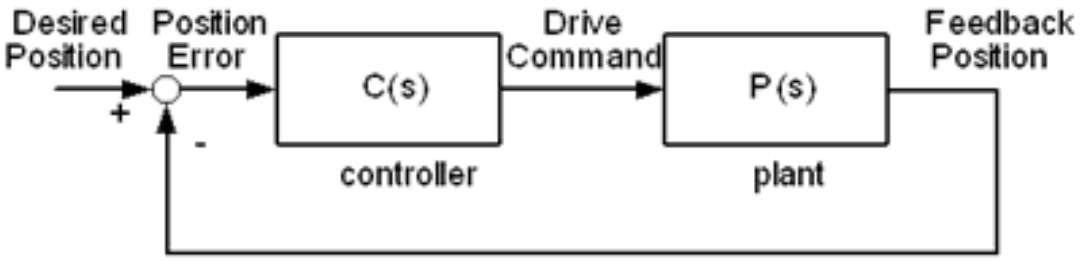
The Signals tab is used for setting what FRFs are to be measured.

If the Signals tab is not displayed, click Signals.




### FRF Signals

The FRF signals are based on the following diagram:




- **Closed** - the relation between the required position and the feedback position, when the loop is closed, CL(s)= C(s)*P(s)/(1+C(s)*P(s)).
- **Open** - this is the loop transfer function, OL(s) = C(s)*P(s). This function has a major role in a stability analysis of a system.
- **Controller** - the relation between the position error and the drive command, C(s). This is the control algorithm that is implemented in the SPiiPlus controller.
- **Plant** - the relation between the drive command and the feedback position, P(s). The plant constitutes of the mechanical stage, the motor, the feedback device and the drive.

The FRF gain is shown in absolute units or decibels (dB). A value in dB corresponds to 20log10 (value). The phase is typically shown in degrees.

**Display Options**

You can select which FRF signals (Closed, Open, Controller, Plant) to display by clicking the relevant check boxes .

You can also select which graph sets (Measure, Design, Baseline) to display by clicking the relevant check boxes.

To set the line color, type and width for each graph, see Setting FRF Analyzer Properties.

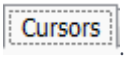
**Controller Information**

The following information is displayed for reference:

- **Firmware** - controller software version.
- **Servo Loop** - servo loop setting used during analysis.
- **Dual Loop** - state (yes/no) of DUALLOOP CONTROL variable for each axis. If state is yes, the **Position and Velocity** option cannot be selected for **Servo Loop**.

### *6.2.4 Cursors Tab*

The FRF Analyzer provides cursors that you can use to see the exact values at any point in the graphs.

To display the Cursors tab click 

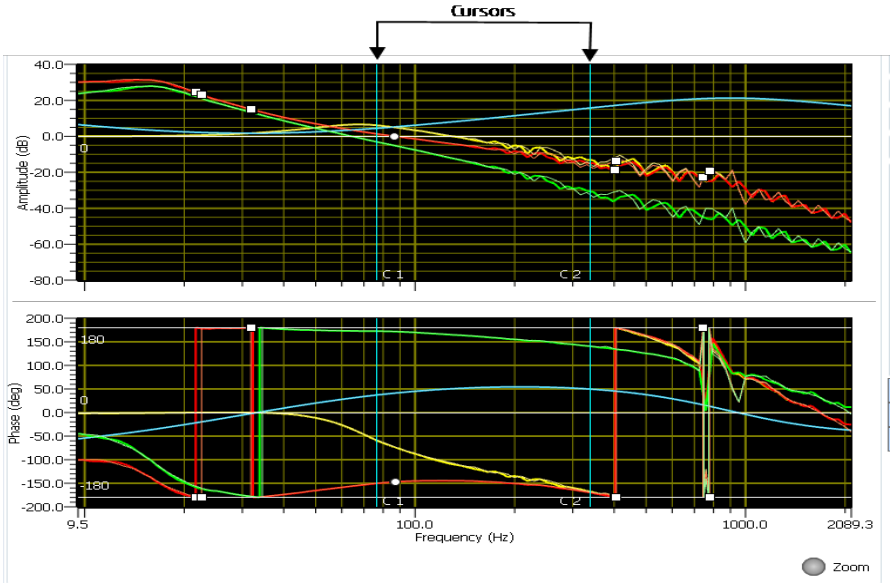
.

The cursor options depend on what graph format you are using:

- Bode Diagram Cursors
- Nyquist Diagram Cursors

### *6.2.4.1 Bode Diagram Cursors*

In the Bode diagram there are two cursors, as shown, giving you the ability to compare values at two different positions.




You move the cursors by dragging them left or right. At each point you will notice that the values of the graph at the cursor point are displayed in table in the tab.

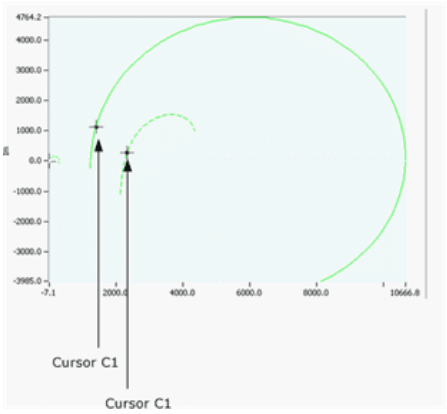
Signals | Cursors | Servo Parameters | Measurement Analysis | Physical Units

| Mode | Graph | Freq1 | Amp1 | Phase1 | Freq2 | Amp2 | Phase2 | Amp2 - Amp1 | Phase2 - Phase1 |
|---|---|---|---|---|---|---|---|---|---|
| Closed | Measure | 76.19 | 6.14593 | -58.36 | 337.41 | -13.22436 | -167.35 | -19.37030 | -108.98 |
| Closed | Baseline | | 6.05966 | -57.90 | | -15.23343 | -167.29 | -21.29309 | -109.39 |
| Open | Measure | | 1.37944 | -150.52 | | -14.90698 | -169.60 | -16.28642 | -19.08 |
| Open | Baseline | | 1.42466 | -150.20 | | -16.61265 | -169.19 | -18.03732 | -18.99 |
| Controller | Measure | | 4.42427 | 36.98 | | 15.71786 | 49.82 | 11.29359 | 12.84 |
| Controller | Baseline | | 4.42427 | 36.98 | | 15.71786 | 49.82 | 11.29359 | 12.84 |
| Plant | Measure | | -3.02662 | 172.50 | | -30.62503 | 140.58 | -27.59841 | -31.93 |
| Plant | Baseline | | -2.98473 | 172.83 | | -32.34220 | 140.99 | -29.35747 | -31.84 |

**Freq1**, **Amp1**, and **Phase1** are the values of **C1.**
**Freq2**, **Amp2**, and **Phase2** are the values of **C2**.
Their units depend on your selections made in **View** (see Setting View Options).

### *6.2.4.2 Nyquist Diagram Cursors*

In the Nyquist diagram due to the nature of the graph there is only one cursor, **C1**. It is distributed along the points of the graph.




You move the cursor either by dragging it to a new point on the graph, or just click the point you are interested in and the cursor moves to the point. The values are displayed in the fields of the tab.

Signals | Cursors | Servo Parameters | Measurement Analysis | Physical Units

| Mode | Graph | Freq1 | Amp1 | Phase1 | Freq2 | Amp2 | Phase2 | Amp2 - Amp1 | Phase2 - Phase1 |
|---|---|---|---|---|---|---|---|---|---|
| Closed | Measure | 76.19 | 6.14593 | -58.36 | 337.41 | -13.22436 | -167.35 | -19.37030 | -108.98 |
| Closed | Baseline | | 6.05966 | -57.90 | | -15.23343 | -167.29 | -21.29309 | -109.39 |
| Open | Measure | | 1.37944 | -150.52 | | -14.90698 | -169.60 | -16.28642 | -19.08 |
| Open | Baseline | | 1.42466 | -150.20 | | -16.61265 | -169.19 | -18.03732 | -18.99 |
| Controller | Measure | | 4.42427 | 36.98 | | 15.71786 | 49.82 | 11.29359 | 12.84 |
| Controller | Baseline | | 4.42427 | 36.98 | | 15.71786 | 49.82 | 11.29359 | 12.84 |
| Plant | Measure | | -3.02662 | 172.50 | | -30.62503 | 140.58 | -27.59841 | -31.93 |
| Plant | Baseline | | -2.98473 | 172.83 | | -32.34220 | 140.99 | -29.35747 | -31.84 |

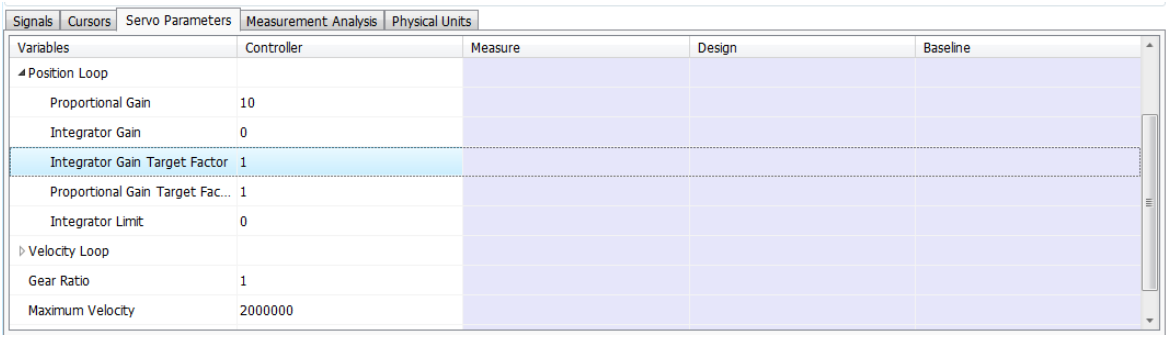
For each signal the values displayed are:

- **Freq** - the frequency of the signal at the cursor point
- **Amplitude (abs)** - the absolute amplitude of the signal at the cursor point
- **Phase** - the phase of the signal at the cursor point
- **Real** - the value of the signal on the X axis
- **Imaginary** - the value of the signal on the Y axis

### *6.2.5 Servo Parameters Tab*

The Servo Parameters tab serves primarily as a tool for Design. It enables you to fine-tune servo parameters and see the system reaction to your changes.

To display the Servo Parameters tab click Servo Parameters.




The tab is divided into five columns:

> **Variables**
This column lists all the servo-loop variables.

For the meanings of the variables refer to the *SPiiPlus Command & Variable Reference Guide*.

> **Controller**
This column displays the status of the servo-loop variable (ON/OFF) and its value(s) stored in the motion controller. It is for information only and the values cannot be changed.

> **Measure**
This column displays the status of the servo-loop variable (ON/OFF) and its value(s) stored in the Measure set. It is for information only and the values cannot be changed.

> **Design**
This column displays the status of the servo-loop variable (ON/OFF) and its value(s) stored in the Design set.
When in the **Design** mode, if a value in this column is not greyed-out, it can be changed.

> **Baseline**
This column displays the status of the servo-loop variable (ON/OFF) and its value(s) stored in the Baseline set. It is for information only and the values cannot be changed.

When working in the Design Mode, once you have found the proper values, you can then use the Adjuster Wizard to set the values into the motion controller.

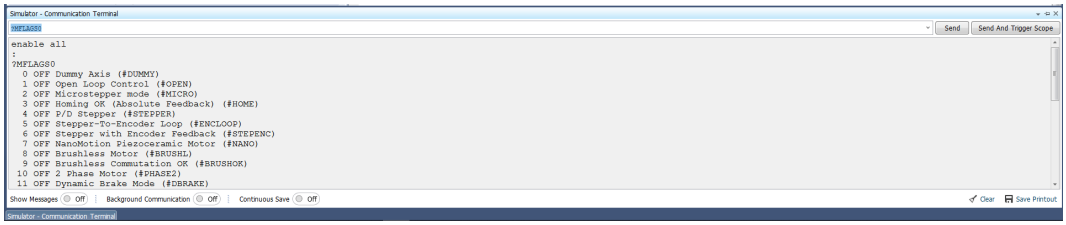
### Flag Variable Status

The status of the flag variable is indicated a checkbox.

| | |
|---|---|
| ☑ | This indicates that the flag variable is ON. |
| ☐ | This indicates that the flag variable is OFF. |

If you are in the Design Mode, you can change the status, for example, from ON to OFF, by clicking the button.

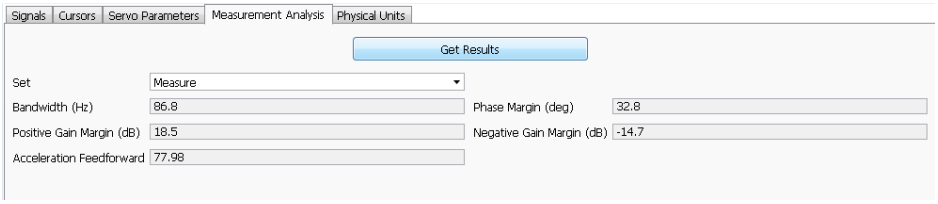
You can check the meaning of a flag variable by using and the **?** command, for example:




## 6.2.6 Measurement Analysis Tab

### Using Measurement Analysis to Adjust System Identification Parameters

To display the measurement analysis page click Measurement Analysis




You can use Measurement Analysis to adjust system identification parameters through the Adjuster Wizard as follows:

1. Open the **Adjuster Wizard**.
2. Click **Axis Setup and Tuning**.
3. Click **Position and Velocity Loops**.
4. Under **System Identification**, click **Start Identification** and if it is safe to proceed, click **Yes** in the **Adjuster Wizard** dialog box.
5. Wait for the message "System identification succeeded" to appear in the System Identification results text box.
6. By using the **Measure** and **Design** sets, you can use the information displayed on the **Measurement Analysis** tab to fine tune the Advanced Tuning parameters received during the **Axis Tuning and Setup**.

The following information is displayed:

> **Set** - click **Measure**, **Design** or **Baseline** to display the required set.

> **Bandwidth** - displays the system bandwidth.

> **Positive Gain Margin** - how much the gain can be raised before instability results. The value is the amount by which the open loop gain is higher than 0dB at the phase crossover frequency.

> **Acceleration Feedforward** - reduces position errors during acceleration and deceleration.

- **Phase Margin** - how much delay can be added to the loop before instability results. The value is the amount by which the open loop phase exceeds –180° at the gain crossover frequencies.
- **Negative Gain Margin** - how much the gain can be reduced before instability results. The value is the amount by which the open loop gain is lower than 0dB at the phase crossover frequency.

## 6.2.7 FRF Analyzer Modes

The **FRF Analyzer** has two modes:

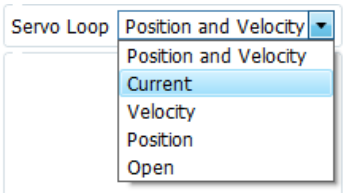
- Measure Mode - enables you to collect data for calculating the FRF signals of your system (Closed, Open, Controller and Plant - see Signals Tab).
- Design Mode - enables you to design a new Controller FRF and to view the effect on the Open Loop FRF and Closed Loop FRF, based on a previous system measurements. The Plant FRF does not change because of modifications by the controller FRF.

### 6.2.7.1 Measure Mode

The **Measure** mode enables you to perform FRF analysis on the actual motion of your system. You can save the analysis data to use for making comparisons in the **Design** mode.

**Measure Mode Settings**

Use the **Servo Loop** button to set the loop to be measured:



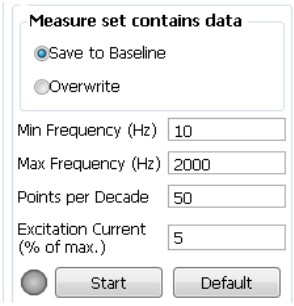

Your options are:

- **Position and Velocity** (default)
  The controller FRF acts as a PID equivalent. A current sine-sweep excitation is injected to the drive command (% of max current), and the total closed position loop FRF is measured (measurement takes into account the velocity FRF).
- **Current**
  This option measures the current loop closed and open loop FRF. During the measurement the position and velocity loops are open.
- **Velocity**
  This option measures a separate velocity loop. During the measurement the **FRF Analyzer** disables the position loop.
- **Position**
  A separate position loop FRF measurement is done. This measurement differs from all other measurements since the excitation is a velocity command rather than current. A velocity sweep excitation is injected to the velocity command (% of max velocity **XVEL**), and the total closed position loop is measured (without accounting for velocity FRF).

- **Open**
  Measures transmission function from current to motor velocity. The measurement is done in an open loop.

You select **Measure** from the **Mode** dropdown list.

Your measurement data save options are:




- **Save to Baseline** - When you start the measurement, if there is previous data in the Measure set, it is written to the Baseline prior to sampling new signals.
- **Overwrite** - When you start the measurement, whatever data was previously stored in the Measure set is overwritten with the new data. The previous data is not saved.

You can set the following parameters:

- **Minimal Frequency (Hz)** and **Maximal Frequency (Hz)** - These set the sine sweep, specified in Hz. These frequencies can range from 1 Hz to 5000 Hz. The FRF measurement itself is executed at 20 kHz. This provides a reliable measurement at the specified frequencies.
- **Frequencies per Decade** - Adjacent frequency steps are uniformly distributed on a logarithmic scale. You can set the required amount of points per one decade of frequencies. Using a larger number of points results in finer resolution of the data, and a longer measurement results. A smaller number of points results in coarser resolution of that data, and a shorter measurement.

  For each frequency, the FRF Analyzer waits 20 cycles for transients to decay and then measures for additional 20 cycles. As the frequency gets higher, the measurement time gets shorter.
- **Excitation Current** - The excitation current determines the amplitude of the sine sweep as percentage of the maximum drive command. The excitation current should be appropriately determined. It should be sufficient to provoke the system. If the excitation current is too low, measurement results may be unreliable.

  However, excessive excitation current is dangerous. If the current is too high, overcurrent may occur. In certain cases, high excitation current can cause saturation s and lead to unreliable measurement results.

Clicking **Default** inserts the factory-set default values. These are:

Minimal Frequency (Hz): 10

Maximal Frequency (Hz): 1000

Frequencies per Decade: 50

Excitation Current (% of max): 10.00

The **FRF Analyzer** does not allow performing FRF measurements if the excitation current is higher than the overcurrent parameter XRMSD, or XRMSM, or if it is higher than 80% of the current limit parameters XCURI or XCURV. It also recommended setting the excitation current lower than the integrator limit parameter, **SLVLI**.

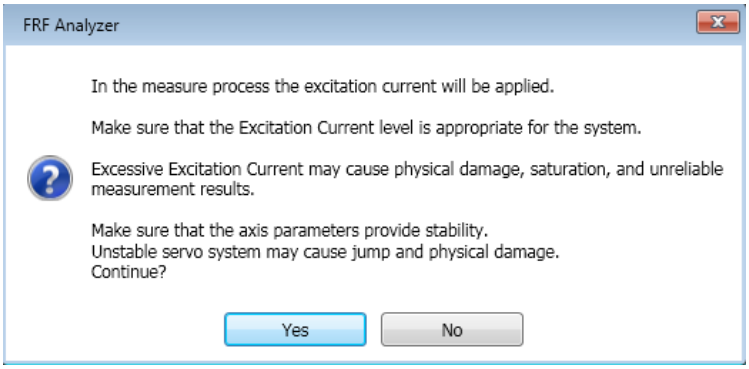
Prior to running the FRF measurements, you can set which signals you want to view from the Signals Tab.

### Running in Measure Mode

If the excitation current is excessively high, mechanical damage may occur.

1. Once you have entered the Measure mode settings, click **Start** ( 

) to initiate the FRF analysis.
2. The **FRF Analyzer** prompts you with:




3. Click **Yes** to start the analysis.

The **Start** button changes to **Stop**. Normally you do not need to click this button. The **FRF Analyzer** analyzes one cycle and then stops automatically. You use this button in the event you need to stop the motion before completion of the measurement.

To view particular values you can use the cursors (see Cursors Tab) and markers (see Markers).

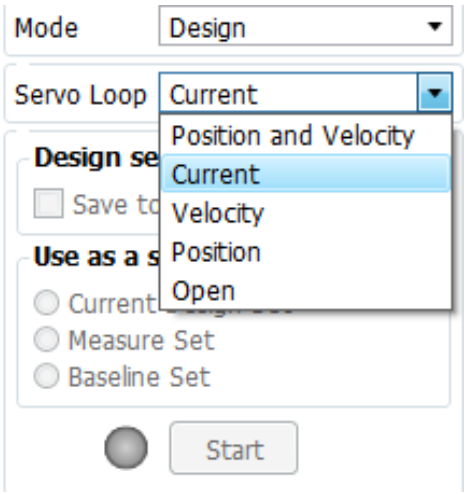
### *6.2.7.2 Design Mode*

The **Design** mode enables you to make changes to the values of the Servo Parameters (see Servo Parameters Tab) and run **the FRF Analyzer** to check the effect of these values.

Prior to running the **FRF ANALYZER** in **DESIGN** mode, you must first have run the analyzer in Measure Mode so that the analyzer has data to compare your changes with. Once you have run the analyzer in the **DESIGN** mode, you can use the data from this mode for further designs.
Alternately, you can load the values from a previously saved file (see Tools).

**Design Mode Settings**

Use the **Servo Loop** button to set the loop to be measured:




Your options are:

- **Position and Velocity** (default)
  The controller FRF acts as a PID equivalent. A current sine-sweep excitation is injected to the drive command (% of max current), and the total closed position loop FRF is measured (measurement takes into account the velocity FRF).
- **Current**
  This option measures the current loop closed and open loop FRF. During the measurement the position and velocity loops are open.
  In the Design mode the **FRF Analyzer** simulates changes in the current loop servo parameters: **SLIKP**,**SLIKI**. You should be aware that due to non-linearities, the results may differ for different excitation currents, depending on the motor parameters and used bus voltage.
- **Velocity**
  This option measures a separate velocity loop. During the measurement the **FRF Analyzer**

disables the position loop.
In the Design mode the FRF simulates changes in velocity loop servo parameters: **SLVKP**, **SLVKI**, low-pass, Notch and BiQuad filters (BiQuad in single loop mode only). A separate velocity loop measurement is required in dual loop systems, but not necessarily required for single loop systems (all this information exists in the combined position & velocity FRFs).

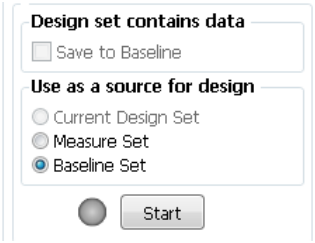
> **Position**
A separate position loop FRF measurement is done. This measurement differs from all other measurements since the excitation is a velocity command rather than current. A velocity sweep excitation is injected to the velocity command (% of max velocity **XVEL**), and the total closed position loop is measured (without accounting for velocity FRF).
In the Design mode, the **FRF Analyzer** simulates changes in position loop servo parameter **SLPKP** and position loop BiQuad filter parameters (dual loop only). A separate position loop measurement is required in dual loop systems, but not necessarily required for single loop systems (all this information exists in the combined position & velocity FRFs).

> **Open**
Measures transmission function from current to motor velocity. The measurement is done in an open loop.

Select **Design** from the **Mode** ( 

) dropdown list.

Your options for using the source data for your design are:




> **Current Design Set** - Usually selected as a follow-on when making adjustments to the design.

> **Measure Set** - The most commonly selected source.

> **Baseline Set** - Normally selected if you have saved your design to the Baseline.

To save previously collected data to the Baseline select the **Save to Baseline** checkbox.

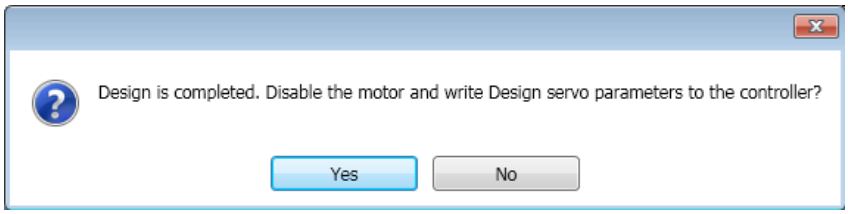
### Running in Design Mode

To run **FRF Analyzer** in the **Design** mode:

1. Click **Start**.

The **Start** button changes to **Finish** and the Servo Parameters Tab is displayed.

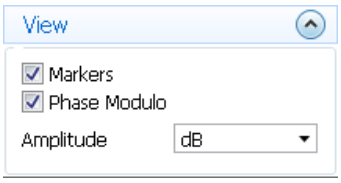
2. Enter the required Servo Loop parameter values in the **Design** column and press **Enter**. Note that the graph changes to reflect the new values.
   To view particular values you can use the cursors (see Cursors Tab) and markers (see Markers). If the values did not meet your requirements, you can set other values in the Servo Parameters and press **Enter** again.
3. Once you are satisfied with the values, click the **Finish** button. Your are prompted with:




4. Click **Yes**. The new values are loaded into the controller.

### 6.2.8 Setting View Options

The **View** options enable you to set certain graph display options. To display the options click .




Your options are:

- **Markers** - Displays the crossover markers (see Markers)
- **Phase Modulo** - Displays the phase wrapped back to 180º at a point where it reaches -180º.
- **Amplitude** - select displaying the amplitude:
  - **Absolute** - displays the amplitude in absolute units
  - **dB** - displays the amplitude in decibels

The **Phase Modulo** and **Amplitude** options are relevant only in Bode Diagrams.

### 6.2.9 Markers

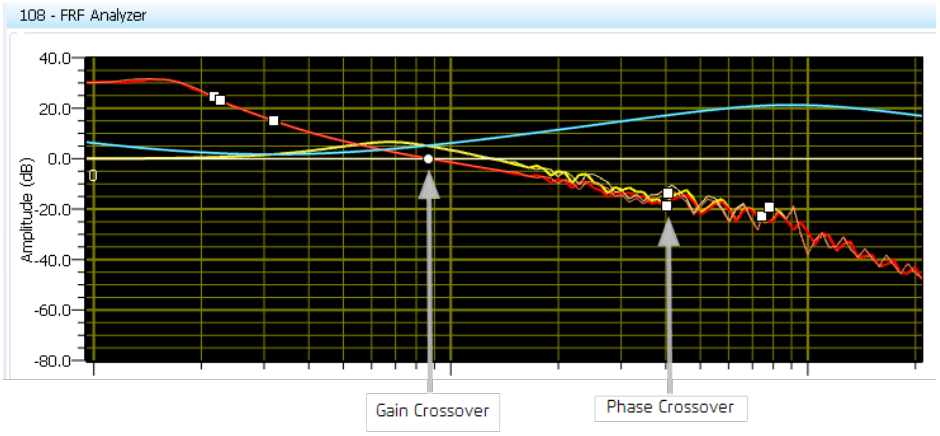
The level of stability is typically judged at certain frequencies, known as crossover frequencies:

- Gain crossover frequency - 0 dB crossing of the open-loop FRF gain.
- Phase crossover frequency - ±180º crossing of the open-loop FRF phase.

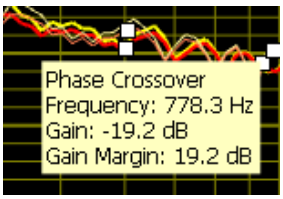
In order to quantify the level of stability, two criteria have been introduced:

- Gain margin (GM) is a measure of how much the gain can be raised or reduced before instability results. It is the amount by which the open loop gain is lower (negative GM) or higher (positive GM) than 0dB at the phase cross over frequency.
- Phase margin (PM) is a measure of how much delay can be added to the loop before instability results. It is the amount by which the open loop phase exceeds –180$^{0}$ at the gain cross over frequencies.

When you select **Markers**, phase crossover frequencies are marked by white squares, and gain crossover frequencies are marked by white circles:




Clicking one of the markers shows details of the frequency, the gain, the phase and the stability margin at that point, for example:



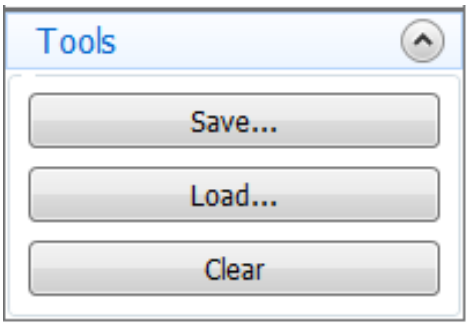

The markers are displayed in both the Bode and Nyquist Diagrams.

### *6.2.10 Tools*

The Tools options are:

- **SAVE** - for saving data to a file, see Saving FRF Data to File
- **LOAD** - for loading data from a file, see Loading FRF Data from File
- **CLEAR** - for clearing data from the display, see Clearing FRF Data from Display

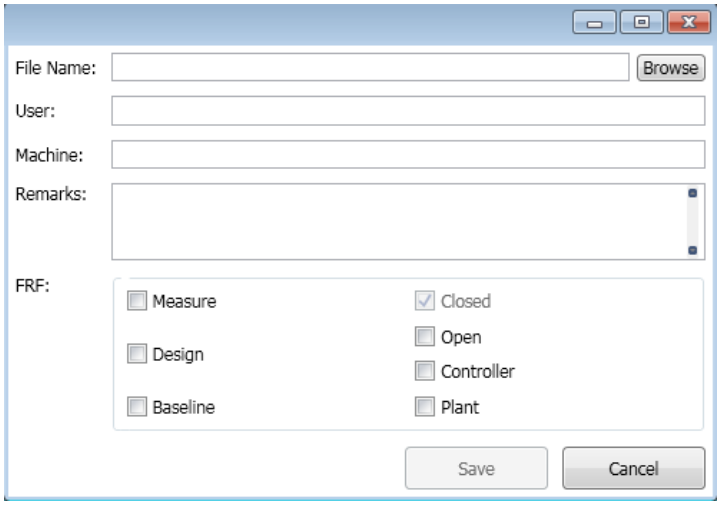
To display the Tools options click .




### *6.2.10.1 Saving FRF Data to File*

To save the FRF data to a file:

1. Click **Save**, the Save Signals dialog window is displayed:

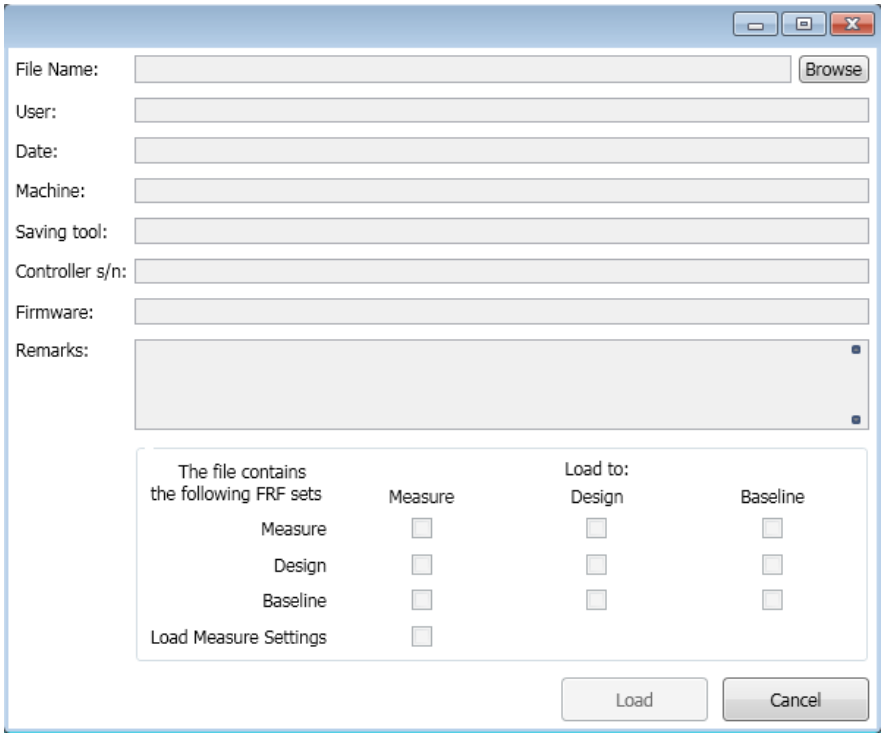



2. Enter the file path and filename in **File Name** (or click **Browse** to search for a directory, and filename if required).
3. Enter your name in the **User** field.
4. Enter the controller ID in the **Machine** field.
5. You can add optional remarks in the **Remarks** field.
6. Select or deselect, as required, the FRF data you want to save by clicking the appropriate checkbox.
7. Click **Save**.

### *6.2.10.2 Loading FRF Data from File*

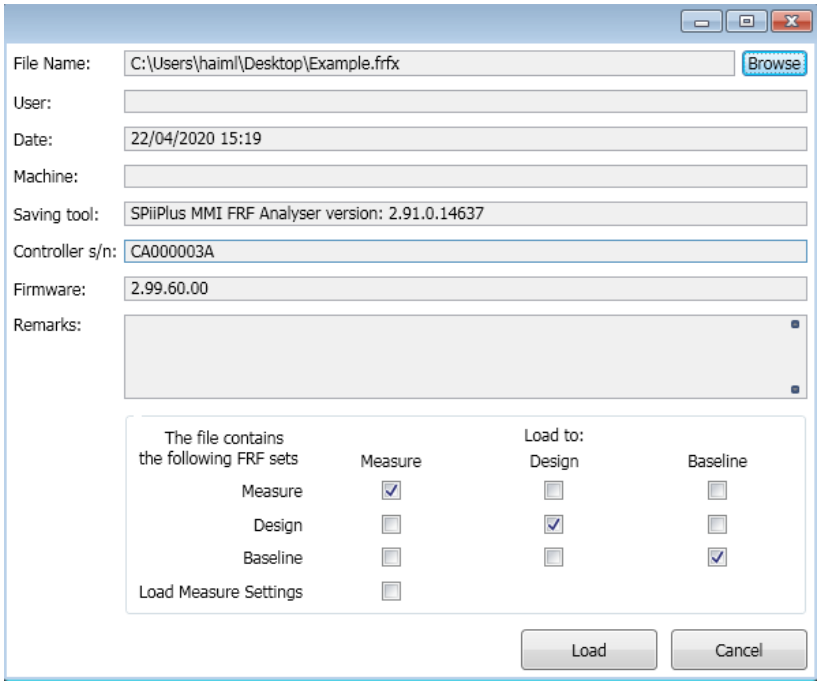
To load FRF data that has been saved to a file:

1. Click **Load**, the Load FRF Data dialog window is displayed:




2. Enter the path and filename in the **File Name** field (or click **Browse** to search for the file). Once you have entered a filename, the contents of the saved file are displayed.

**FRF Analyzer** signal data is stored in files having an **.frf** file extensions. File data that has been stored using an old version of the **FRF Analyzer** can be loaded.

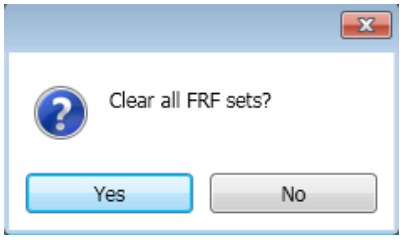



3. You can change into what subset the data is loaded (for example, loading the Design data into Measure) by clicking the appropriate checkbox.
4. Click **Load**.

#### *6.2.10.3 Clearing FRF Data from Display*

To clear the current FRF data from the display:

1. Click **Clear**, you are prompted with:

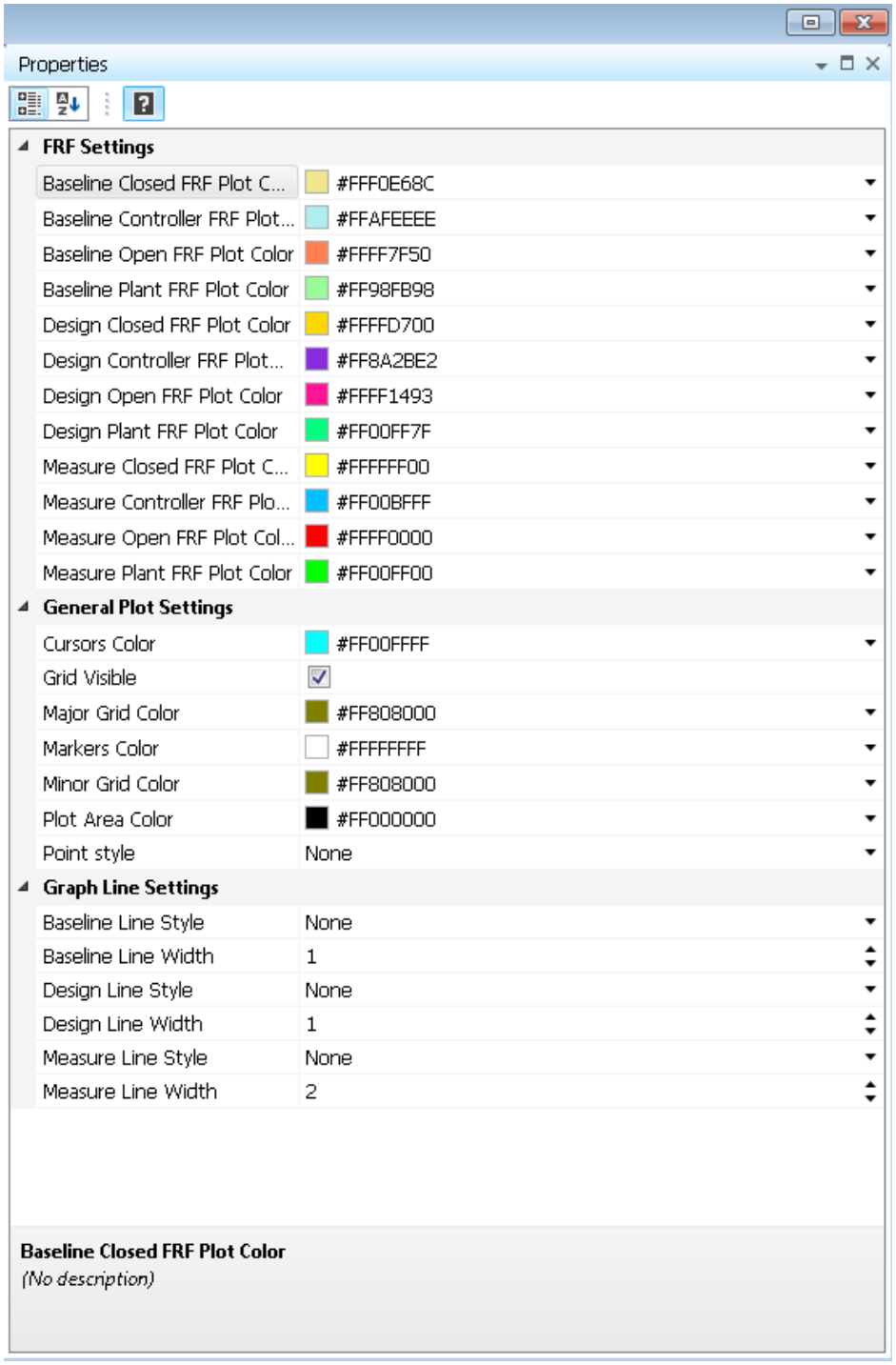



2. Click **Yes**. All data in the display is removed.

### *6.2.11 Setting FRF Analyzer Properties*

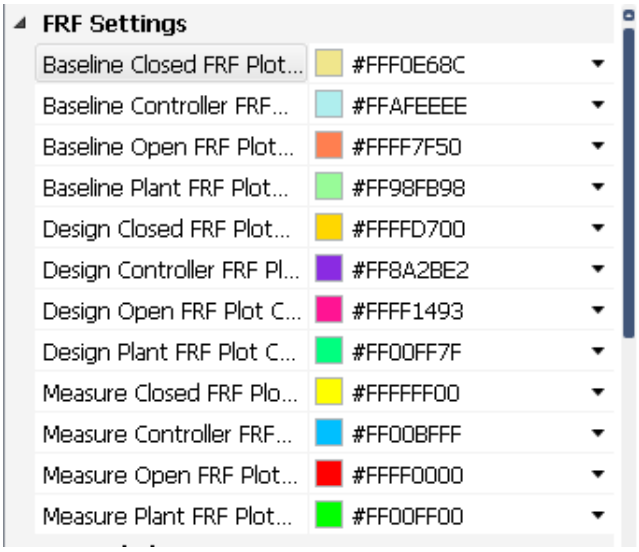
The FRF Analyzer plot properties, such as color, line width, line style, and the like, of each signal is set in the **FRF Analyzer Properties** window.

If the panel is not displayed, see Working with Properties Window for displaying the **FRF Analyzer Properties** panel.




The properties are divided into four groups, as explained following.

### *6.2.11.1 FRF Settings*

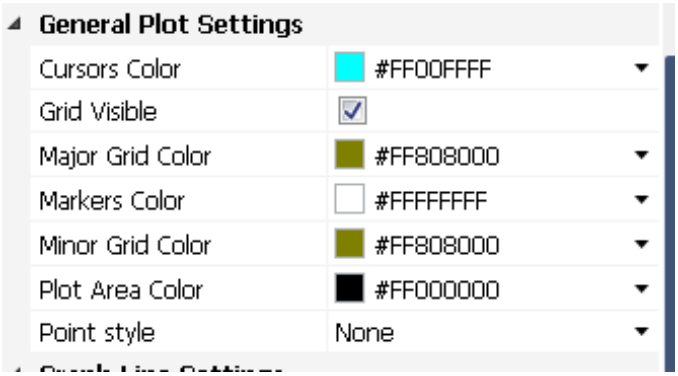



The FRF Settings are used to set the colors in which the four signals: Closed, Controller, Open and Plant are plotted for:

- Baseline
- Design
- Measure

To select a color click the color box and select a color from the dropdown list.

### *6.2.11.2 General Plot Settings*



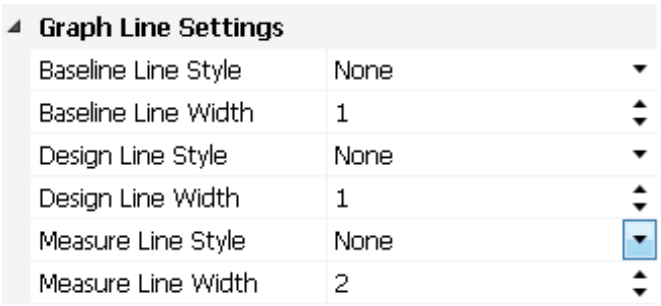

General Plot Settings are used to set the properties of the graph against which the data is to be plotted. These include:

- Cursors Color - the color of the cursors when displayed
- Grid Visible - toggles between displaying the grid (checkbox selected) or not (checkbox unselected)
- Major Grid Color - the color of the main grid, this marks the horizontal and vertical divisions of the graph
- Markers Color - the color of the markers when displayed

- Minor Grid Color - the color of the secondary grid, this marks the horizontal and vertical division is between the main grid points
- Plot Area Color - the background color of the graph
- Point Style - the style of the points being displayed

To select a color click the color box and select a color from the dropdown list.

#### *6.2.11.3 Graph Settings*



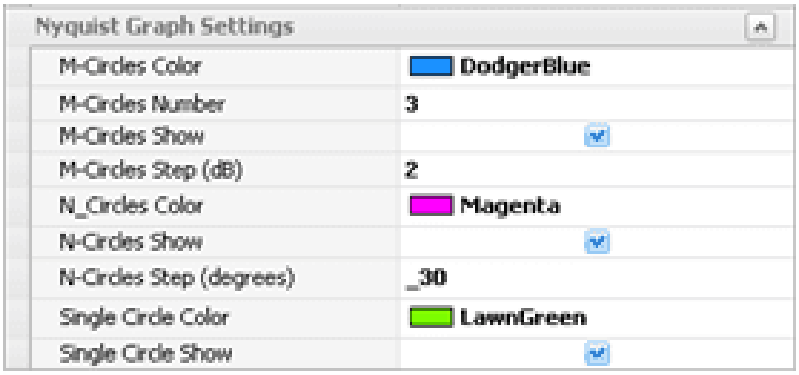

Graph Settings are used to set the line width and style of graphic display for:

- Baseline
- Design
- Measure

To set the width click the value and enter a number, then press **Enter**.

To set the style click the field and select from the dropdown list.

#### *6.2.11.4 Nyquist Graph Settings*




Nyquist Graph Settings are used for the Nyquist graph when it is displayed.

The settings are for M-Circles, N-Circles, and Single Circle, which are:

- **M-Circles –** loci for constant gains of the total closed-loop FRF
- **N-Circles -** loci for constant phases of the total closed-loop FRF
- **Single Circle** - unity loop

The properties in this group are:

- M-Circles Color - set the color of the M-Circles
- M-Circles Number - set the number of M-Circles to display
- M-Circles Show - toggles between displaying the M-Circles (checkbox selected) or not (checkbox unselected)
- M-Circles Step (dB) - sets the steps, in dB, between each M-Circle
- N-Circles Color - set the color of the N-Circles
- N-Circles Show - toggles between displaying the N-Circles (checkbox selected) or not (checkbox unselected)
- N-Circles Step (degrees) - sets the steps, in degrees, between each N-Circle
- Single Circle Color - set the color for the unity circle
- Single Circle Show - toggles between displaying the Single Circle (checkbox selected) or not (checkbox unselected)

### *6.2.12 Troubleshooting*

Because of non-linear effects in the system, for example, backlash, different results may be achieved for various scenarios:

- Problem: Different results when the axis is idle or moving.
  **Solution:** In case of static friction it is recommended to move the axis at low velocity during the measurement.
- Problem: A disturbance in the FRFs may appear if PTP is used, while the axis is changing its direction.
  **Solution:** To minimize this effect, it is recommended to perform cyclic motion between two points without any dwell time between motions.
- Problem: Different results for different excitation levels.
  **Solution:** Always verify that the excitation current level is appropriate.
- Problem: Non-linearities because of saturation s in the system, the drive or the control algorithm, such as, integrator windup, current limit, etc.
  **Solution:** Always verify that nothing in the control algorithm and the system saturates.
- Problem: Cross coupling between axes can affect the measurement results, for example, if one axis location affects the center of gravity of a second axis.
  **Solution:** Always record the axes state and position while taking a measurements.
- Problem: Dual loop topology usually has two encoders and the FRF of each loop has to be separately measured.
  **Solution:** This is not supported by the FRF Analyzer. Still, the total Close loop FRF will be measured correctly. Open-Loop and plant FRF will be only estimations.
- Problem: Resonances with very low attenuation (lower than 2%) may affect the coherence of the measurement and its reliability.
  **Solution:** After the resonance is identified, it is recommended to attenuate it using low-pass or Notch filter. Afterward you should re-measure the system.
- Problem: Noise may effect the measurement results, especially at high frequencies when the signal to noise ratio is low.

**Solution:** Make sure that the system is properly grounded and shielded, that is, ground the controller, use twisted-pair shielded cables, etc.

## 6.3 HSSI Manager

The **HSSI Manager** is used for monitoring the status of the HSSI (High Speed Serial Interface) channels. The manager displays the quantity and state of the HSSI channels.

Some strings will be connected if the device occupies more than one register.

The **HSSI Manager**

- Displays the status of the controller's HSSI channels.
- Provides a "Reconnect/Refresh" function for reinitializing communication between the controller and connected HSSI modules.

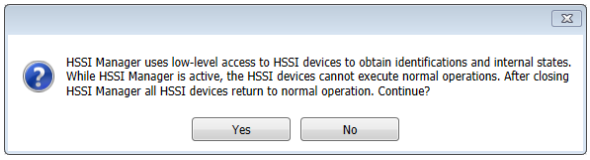
### 6.3.1 Activating HSSI Manager

To activate **HSSI Manager**:

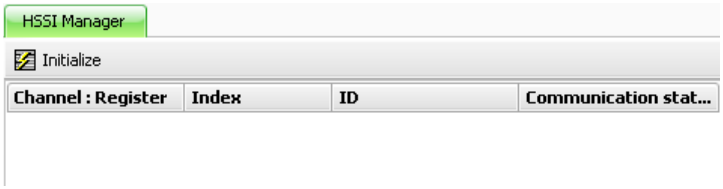
1. From the Toolbox click **Diagnostics and Monitoring** .
2. Click **HSSI Manager**.

You may use **Add Component** from the right-click menu of the controller in the Workspace Tree (Controller Right-Click Options ) to activate the **HSSI Manager**.

You are prompted with:



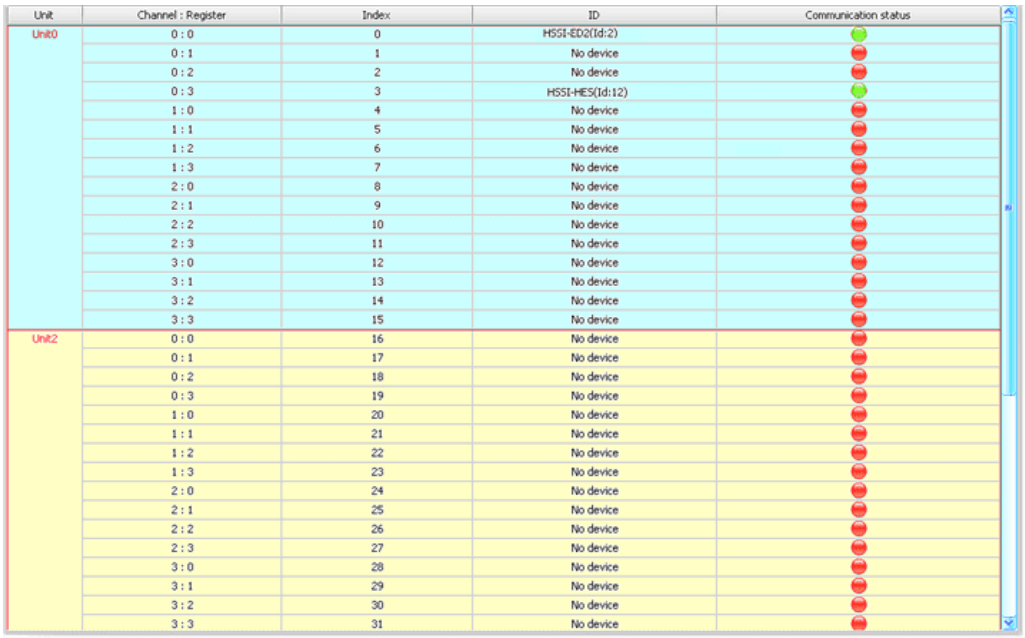

3. Click **Yes**. The HSSI Manager window is displayed.




4. Click **Initialize** to establish communication with the controller. The HSSI Manager reads the HSSI Channel data and displays it.

Should you close the window and then reopen it, communication is cut off with the controller. To reestablish communication, click **Reconnect**.

### 6.3.2 HSSI Manager Window



| Unit | Channel : Register | Index | ID | Communication status |
|---|---|---|---|---|
| Unit0 | 0 : 0 | 0 | HSSI-ED2(Id:2) | |
| | 0 : 1 | 1 | No device | |
| | 0 : 2 | 2 | No device | |
| | 0 : 3 | 3 | HSSI-HES(Id:12) | |
| | 1 : 0 | 4 | No device | |
| | 1 : 1 | 5 | No device | |
| | 1 : 2 | 6 | No device | |
| | 1 : 3 | 7 | No device | |
| | 2 : 0 | 8 | No device | |
| | 2 : 1 | 9 | No device | |
| | 2 : 2 | 10 | No device | |
| | 2 : 3 | 11 | No device | |
| | 3 : 0 | 12 | No device | |
| | 3 : 1 | 13 | No device | |
| | 3 : 2 | 14 | No device | |
| | 3 : 3 | 15 | No device | |
| Unit2 | 0 : 0 | 16 | No device | |
| | 0 : 1 | 17 | No device | |
| | 0 : 2 | 18 | No device | |
| | 0 : 3 | 19 | No device | |
| | 1 : 0 | 20 | No device | |
| | 1 : 1 | 21 | No device | |
| | 1 : 2 | 22 | No device | |
| | 1 : 3 | 23 | No device | |
| | 2 : 0 | 24 | No device | |
| | 2 : 1 | 25 | No device | |
| | 2 : 2 | 26 | No device | |
| | 2 : 3 | 27 | No device | |
| | 3 : 0 | 28 | No device | |
| | 3 : 1 | 29 | No device | |
| | 3 : 2 | 30 | No device | |
| | 3 : 3 | 31 | No device | |

The HSSI Manager Main Window has 5 columns:

- **Unit**
  In an NT configuration, several motion controllers can be connected together, each motion controller is treated separately
- **Channels:Register**
  This column displays the number of the channel and register number in the channel (separated by a colon).
- **Index**
  This column displays number of SPii index.
- **ID**
  This column displays the Device ID, if it exists.
- **Communication status**
  This column displays the channel status: Green for normal, Red for error.

## 6.4 Inputs/Outputs Manager

The **Inputs/Outputs Manager** tool displays the status of the digital input and output variables of the SPiiPlus controller as well as those of the HSSI, if defined.

Status of digital input is measured against the value of the associated **IN** variable, and that of digital output against the value of the associated **OUT** variable.

The **IN** variable is a scalar 16-bit integer and stores the current state of the General Purpose digital inputs represented by bits 0..15 of **IN**. Each bit reports the state of one General Purpose input.

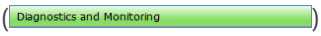
In a like manner **OUT** is a scalar 16-bit integer and stores the current state of the General Purpose digital outputs represented by bits 0..15 of **OUT**. Each bit reports the state of one General Purpose output.

HSSI input is reported through the **EXTIN** (Extended In) variable which is a 16 member integer array that reads the current state of the HSSI inputs. Each element of the array contains a 16 bit number, each bit of which representing the state of one input; therefore, up to 256 inputs can be read. The number of inputs depends on the HSSI module.

HSSI output is reported through the **EXTOUT** (Extended Out) variable which, like **EXTIN**, is a 16 member integer array that reads the current state of the HSSI outputs. Each element of the array contains a 16 bit number, each bit of which representing the state of one output; therefore, up to 256 outputs can be read. The number of outputs depends on the HSSI module.

### 6.4.1 Activating Inputs/Outputs Manager

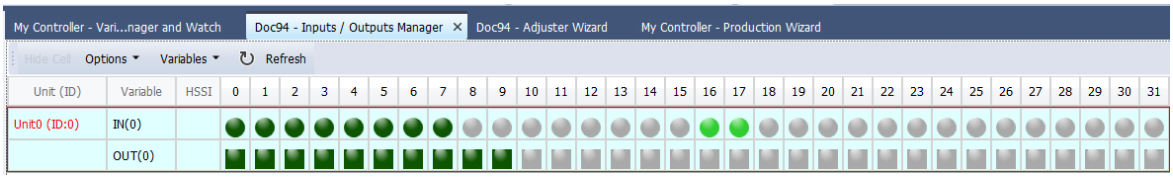
To activate the Inputs/Outputs Manager:

1. From the Toolbox click **Diagnostics and Monitoring** ( 

) to display the Diagnostics and Monitoring tools list.
2. Click **Inputs/Outputs Manager**.

You may use **Add Component** from the right-click menu of the controller in the Workspace Tree (Controller Right-Click Options ) to activate the Inputs/Outputs Manager.

The Inputs/Outputs Manager window is displayed:




Some controllers have only 8 digital inputs and outputs instead of 16. In this case the symbols in columns 8 through 31 are greyed-out indicating that they are not available.

An NT configuration may contain several controllers linked together. The **Inputs/Outputs Manager** displays each controller - **Unit** - separately.

### 6.4.2 Inputs/Outputs Manager Operations

The Inputs/Outputs Manager window employs two icons to show input and output:

| | |
|---|---|
|  | Input |
|  | Output |

They can be displayed in one of three colors:

| | |
|---|---|
|  | Green - indicates that it has been activated. |
|  | Dark Grey - indicates that it has not been activated. |
|  | Light Grey - indicates that it is not supported by the controller (for example, an external device that has been included in the system).<br><br>not been connected to the HSSI channel). |

The controller activates and deactivates the bits as it progresses through its program.

You can activate a bit manually by clicking it. To deactivate it manually you just click it again.

You can use the **Terminal t**o check the value or values of the bits, for example, if you enter **?IN(0)** in the Terminal command line, you will get the values for all 16 bits of the variable.

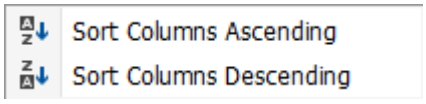
### *6.4.2.1 Display Options*

You have the following display options:

- Column Display
- Cell Display
- Selecting HSSI EXTIN/EXTOUT Variables

**Column Display**

If you right-click a column, you are given the following options:



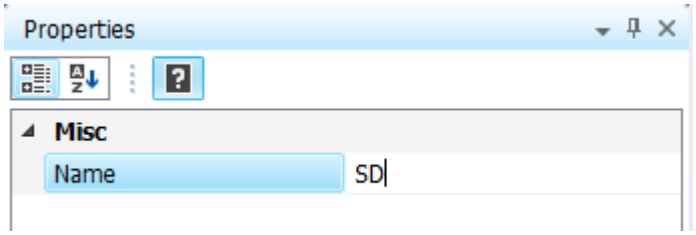

Using this menu you can sort the columns from left to right (ascending order, that is, 0 to 31), or from right to left (descending order - 31 to 0).

**Cell Display**

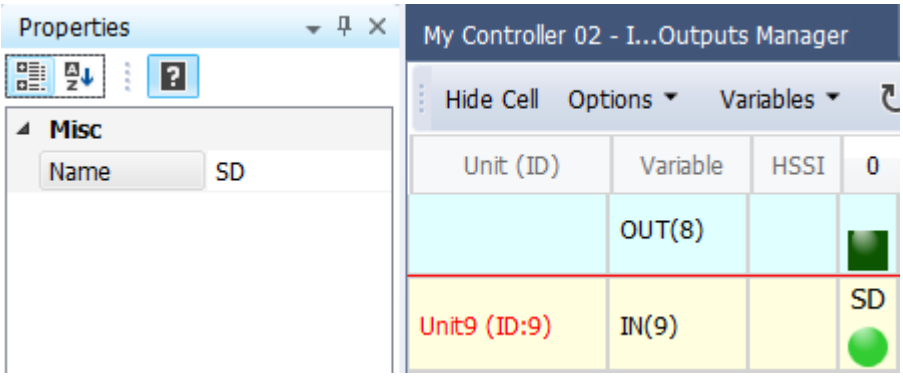
You hide the display of any cell by selecting it and clicking **Hide Cell**. To restore its display, select the hidden cell and click **Show Cell**.

As an aid in identifying the specific inputs and outputs, you can give cells names:

1. Right-click the cell and select **Properties**. The Properties panel for the cell is activated.




2. Enter the identifier you want in the **Name** field.
3. Select **Show Names** either from **Options**, or by right-clicking the cell. You can also activate this by pressing **Ctrl+C** and then pressing **N**. The name is displayed.




You are not constrained as to the length of the name, if it is long, you can expand the column width by dragging its boundaries to show the whole name:


Once the **Properties** panel is displayed, you can select it for each cell by selecting the cell, then entering its identifying name in the Properties **Name** field.

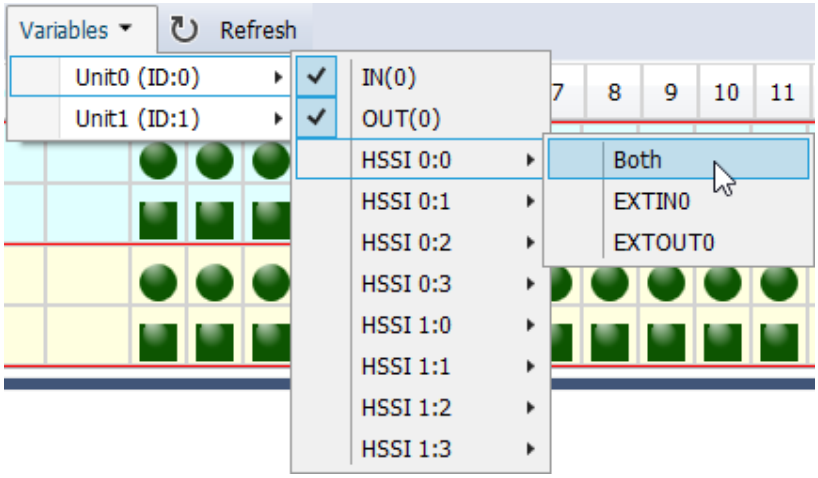
To hide the names select **Show Names** either from **Options**, or by right-clicking the cell. You can also toggle the display of names off by pressing **Ctrl+C** and then pressing **N**.

### *6.4.2.2 Selecting HSSI EXTIN/EXTOUT Variables*

If you have EXTIN and EXTOUT signals that you want to monitor:

1. Click **Variables**, and select the **Unit** from the dropdown list.
   A list of HSSI variables is displayed.
2. Select the HSSI variable, you have three choices:

- **EXTIN** for the selected variable
- **EXTOUT** for the selected variable
- **Both** (display both EXTIN and EXTOUT)




Once you have made your choice, additional rows are added to the selected Unit's row showing the status of the variables.

## 6.5 Motion Manager

**Motion Manager** is used for the following:

- Setting the motion parameters for each axes
- Initiating the motion
- Monitoring the progress of the motion
- Stopping the motion

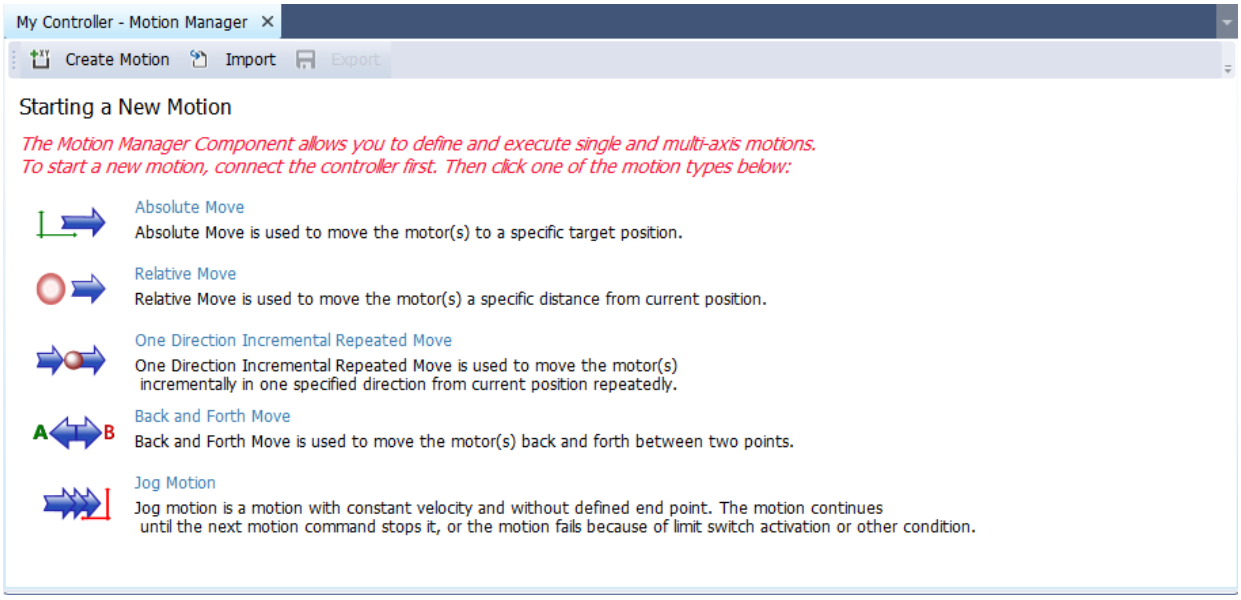
### 6.5.1 Activating Motion Manager

By default, when you initially enter SPiiPlus MMI Application Studio, **Motion Manager** is automatically displayed. If you have closed it, you can reactivate it by either clicking it in the Workspace Tree or by:

1. From the Toolbox click **Diagnostics and Monitoring** to display the Diagnostics and Monitoring tools.
2. Click **Motion Manager** in the Diagnostics and Monitoring list of tools.

You can also activate the **Motion Manager** using the right-click **Add Component** option of the Workspace Tree - see Controller Right-Click Options .

The Motion Manager main window appears:

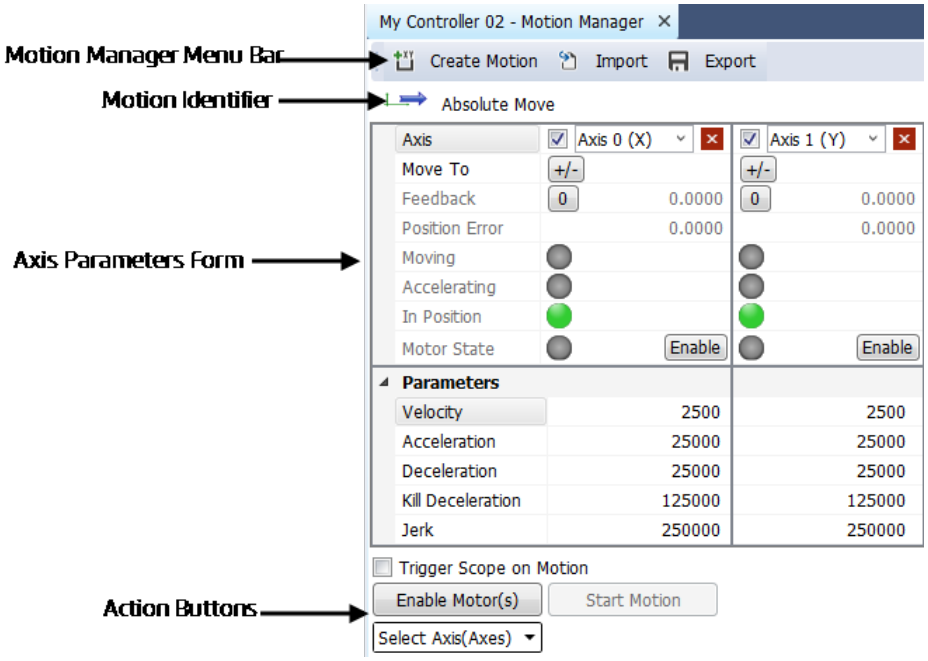



There are five types of motion that can be set:

| | | |
|---|---|---|
|  | Absolute Move | - Used for moving the motor(s) to a specific target. |
|  | Relative Move | - Used for moving the motor(s) a specific distance from current position. |
|  | One Direction Incremental Repeated Move | - Used for repeatedly moving the motor(s) incrementally in one specified direction from current position. |
|  | Back and Forth Move | - Used for moving the motor(s) back and forth between two specified points. |
|  | Jog Motion | - Used for moving the motor(s) with constant velocity and without a defined end point. The motion continues until the next motion command stops it, or the limit switch has been activated, or some other condition. |

### *6.5.2 Motion Manager Window*

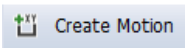
The Motion Manager window components are:



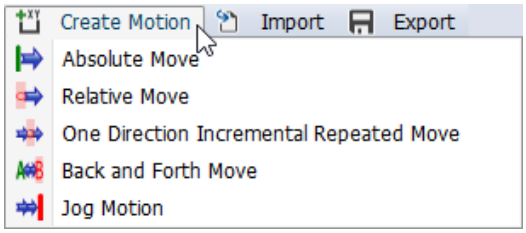

The Motion Manager window enables you to set up and manage the motion of motors, both along individual axes and along all axes.

### *6.5.2.1 Motion Manager Menu Bar*

The Motion Manager Menu bar has four options:

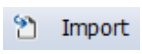
> 


This option enables you to create a new motion type. When you click the down arrow, a dropdown menu appears from which you can select the motion type:

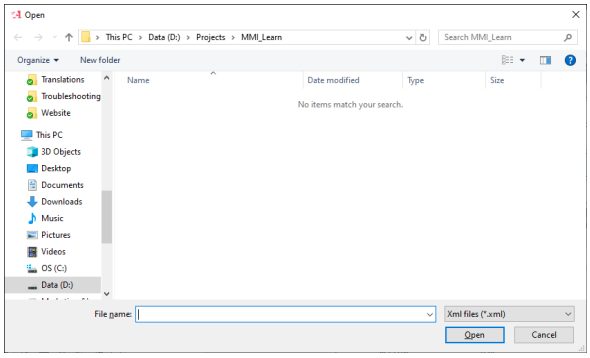



To select: click the required motion type, a new Axis Motion Parameters Form opens.

While the **Axis Motion Parameters Form** allows you to add as many axes as you need, their parameters will all be of the same motion type. You use this option in the event that the motion type of other axes is different.

> 


This option enables you to import the parameter settings from a previously saved file. When you select the option, a browser window opens:

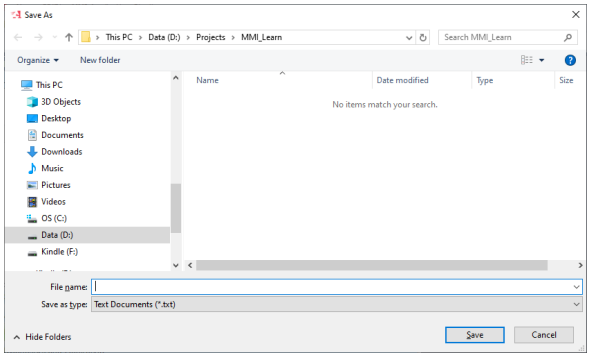



Select a directory and filename and click **OK**.

Motion parameters files have an **mmf** file extension.

> Export

This option enables you to save the parameter settings to a file. When you select the option, a browser window opens:




Select the directory into which you want to save the file and enter a filename. Then click **OK**.

### *6.5.2.2 Motion Manager Window Action Buttons*

The Motion Manager window has the following action buttons:

| | |
|---|---|
| Enable Motor(s) | When clicked, all motors are enabled. The button changes to **Disable Motor(s)**. |
| Disable Motor(s) | When clicked, all motors are disabled. The button changes to **Enable Motor(s)**. |
| Start motion | Initiates motion for all enabled motors. The button changes to **Stop motion**. |

| | |
|---|---|
|   | Stops all motion in progress. The button changes to **Start motion**. |
|   | By default, the form displays only the 0 and 1 axes. Clicking this button enables you to add additional axes. When you click this button, a list of axes is displayed:<br>  <br>Select the required axis or axes by clicking the checkbox(es), and then click **OK**. The axis(es) are displayed in the form. |

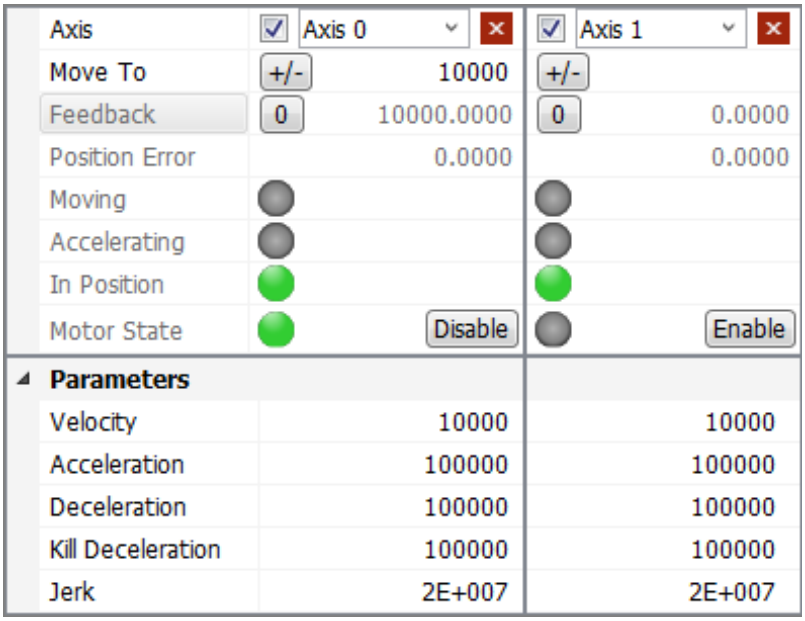
In the upper right corner of the window there are two buttons:

| | |
|---|---|
|  | Collapses all the Axis Parameters forms in the window. This is an aid when you need to reduce the window to view other items.<br>When you click this button, it changes to  , which when clicked, restores all the Axis Parameters forms to the window. |
|  | Closes the window. |

The window contains one checkbox, **Trigger Scope on Motion**. When this is selected, upon the start of motion, the is automatically displayed and set to the Single Trigger mode.

### *6.5.2.3 Axis Motion Parameters Form*

The Axis Motion Parameters form is displayed when any type of motion is selected, for example:




Each form has the same layout, it is divided into two sections:

- **Axis** - motion type specific settings and common parameters (motor states, feedback position and position error). The section contains four motor status indicators:
  - **Moving** - the motor is in motion
  - **Accelerating** - the motor is accelerating
  - **In Position** - the motor is in initial position
  - **Motor State** - the motor is enabled

The indicators are gray (switched off) by default. When the motion is in progress, the indicator of the corresponding status turns green.

- **Motion Parameters** - for viewing or setting values specifically applicable to the motor.

If one of these parameters is changed when a motion is in progress, the change affects the currently executing motions, or motions that were created before the substitution.

The form has one action button:

| | |
|---|---|
| Enable | Enables a single motor. When clicked, just the motor for the given axis is enabled. The button changes to **Disable**. |
| Disable | Disables a single motor. When clicked, just the motor for the given axis is disabled. The button changes to **Enable**. |
| | Closes the form window. |

### 6.5.3 Working with Axis Motion Parameters Form

The following are general instructions applying to all Axis Motion Parameters forms:

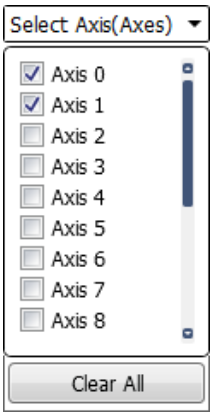
**Field Values**

Values entered into the various fields are, when motion is initiated, inserted in the appropriate ACSPL+ variable. For example, the value in the **Feedback Position** field is supplied to the ACSPL+ **FPOS** axis state variable.

If you let the mouse cursor linger on the name of a field, a Tool Tip appears informing you of the name of the ACSPL+ variable with which the field is associated.

**Changing Axis**

When you first enter the Axis Motion Parameters Form, by default, the form displays fields for Axis 0 and Axis 1. You can change the axis designation by clicking the down arrow of the **Axis** field. A dropdown menu appears:




Select the required axis from the list by clicking it.

**The Move By Field Value**

You can toggle the value of the **Move By** field between positive and negative by clicking  .

**Zeroing the Feedback Position Field**

You can set the value of the **Feedback Position** field to zero (0) by clicking  .

**Expanding/Collapsing the Parameters List**

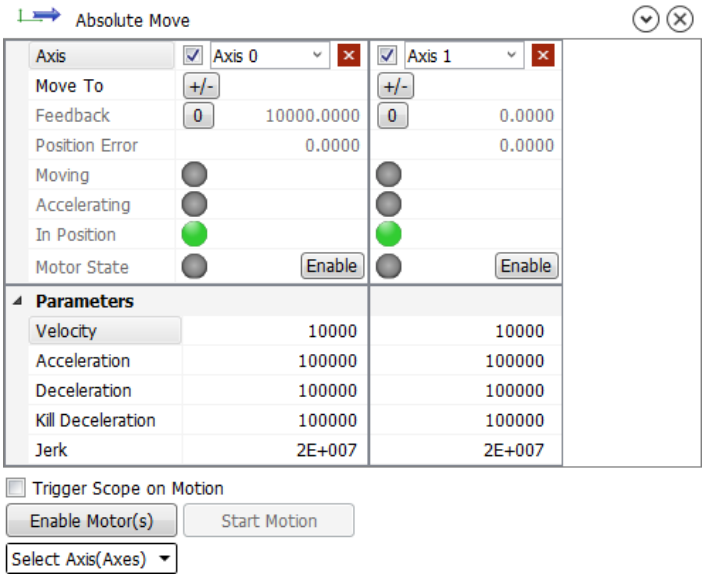
You can collapse the Parameters list display by clicking  . And you can expand its display by clicking  .

### 6.5.4 Absolute Move

Absolute Move represents the motion from the origin of coordinates to the initially given destination point - target position.

To set the parameters for Absolute Move:

1. Activate Motion Manager.
2. Click **Absolute Move** ( ). The Absolute Move form is displayed.



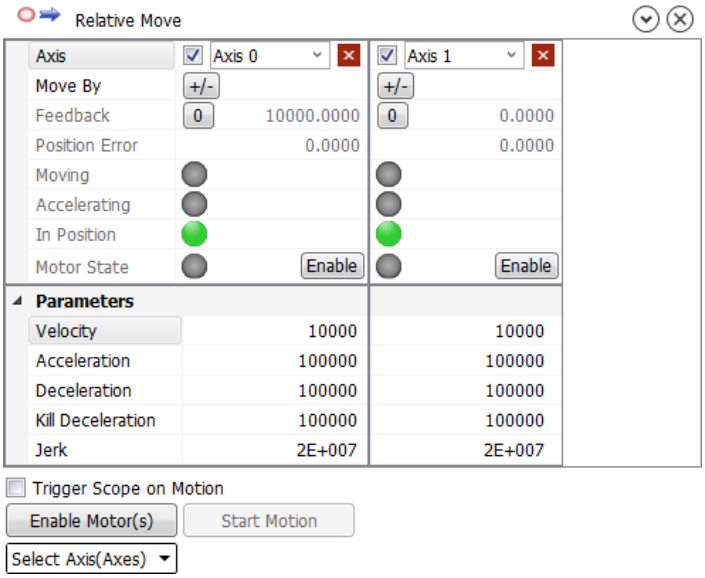

3. Set the motion parameters:
   - **Move To** - the destination point in accordance to the selected axis
   - **Feedback Position** - the current position
   - **Position Error** - the difference between current and required position coordinates
   - **Velocity** - defines the default velocity of the motion profile
   - **Acceleration** - how quickly the speed will reach the required limit
   - **Deceleration** - defines the speed reduction rate
   - **Kill Deceleration** - defines deceleration when a motion is killed or fails due to a fault
   - **Jerk** - stands for the jerk of the motion profile
4. If you need to define the motion parameters for axes other than X and Y, click **Add Axis (Axes)** and select the additional axis or axes.
5. Click **Enable Motor(s)**.
6. To start the motion click **Start motion** .

### *6.5.5 Relative Move*

Relative Move is used to move the axis a specific distance from current position.

To set the parameters for Relative Move:

1. Activate Motion Manager.
2. Click **Relative Move** ( ). The Relative Move form is displayed.



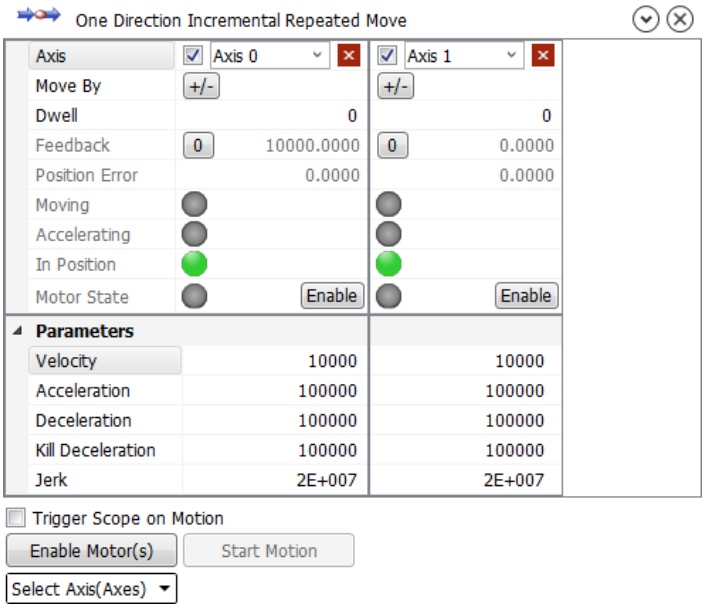

3. Set the motion parameters:
   - **Move By** - the distance for the motor to move
   - **Feedback Position** - the current position
   - **Position Error** - the difference between the current and required position coordinates
   - **Velocity** - defines the default velocity of the motion profile
   - **Acceleration** - how quickly the speed will reach the required limit
   - **Deceleration** - defines the speed reduction rate
   - **Kill Deceleration** - defines deceleration when a motion is killed or fails due to a fault
   - **Jerk** - stands for the jerk of the motion profile
4. If you need to define the motion parameters for axes other than X and Y, click **Add Axis (Axes)** and select the additional axis or axes.
5. Click **Enable Motor(s)**.
6. To start the motion click **Start motion**.

### *6.5.6 One Direction Incremental Repeated Move*

One Direction Incremental Repeated Move is used to move the motor(s) incrementally and repeatedly in one specified direction from the current position.

To set the parameters for One Direction Incremental Repeated Move:

1. Activate Motion Manager.
2. Click **One Direction Incremental Repeated Move** ( ). The One Direction Incremental Repeated Move form is displayed.



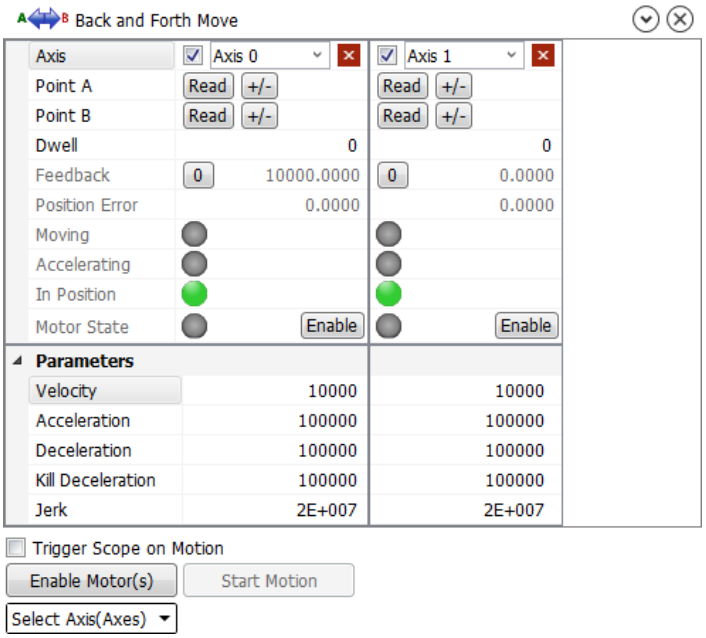

3. Set the motion parameters:
   - **Move By** - the required distance
   - **Dwell** - the amount of time, in ms, to dwell in each point
   - **Feedback Position** - the current position
   - **Position Error** - the difference between current and required position coordinates
   - **Velocity** - defines the default velocity of the motion profile
   - **Acceleration** - how quickly the speed will reach the required limit
   - **Deceleration** - defines the speed reduction rate
   - **Kill Deceleration** - defines deceleration when a motion is killed or fails due to a fault
   - **Jerk** - stands for the jerk of the motion profile
4. If you need to define the motion parameters for axes other than X and Y, click **Add Axis (Axes)** and select the additional axis or axes.
5. Click **Enable Motor(s)**.
6. To start the motion click **Start motion**.

### *6.5.7 Back and Forth Move*

Back and Forth Move is used to move the axis back and forth between two specific points.

To set the parameters for Back and Forth Move:

1. Activate Motion Manager.
2. Click **Back and Forth Move** ( ). The Back and Forth Move form is displayed.



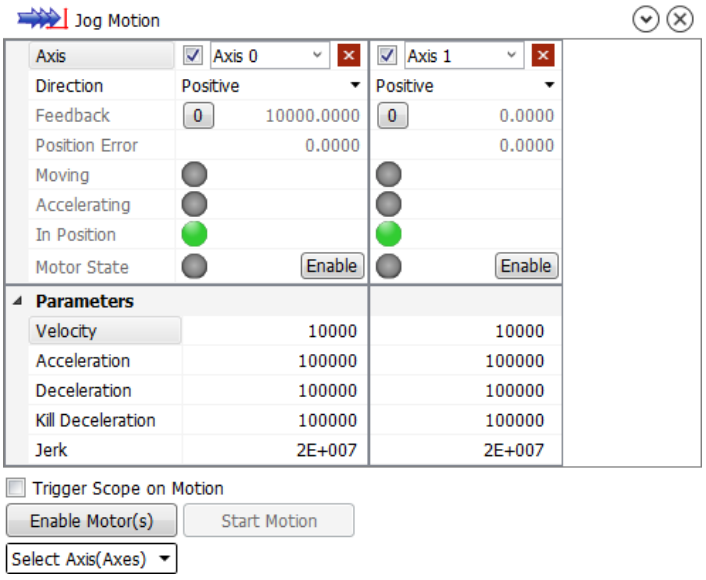

3. Set the motion parameters:
   - **Point A** - the starting point
   - **Point B** - the end point
   - **Dwell** - the amount of time, in ms, to dwell at the end point
   - **Feedback Position** - the current position
   - **Position Error** - the difference between current and required position coordinates
   - **Velocity** - defines the default velocity of the motion profile
   - **Acceleration** - how quickly the speed will reach the required limit
   - **Deceleration** - defines the speed reduction rate
   - **Kill Deceleration** - defines deceleration when a motion is killed or fails due to a fault
   - **Jerk** - stands for the jerk of the motion profile
4. If you need to define the motion parameters for axes other than X and Y, click **Add Axis (Axes)** and select the additional axis or axes.
5. Click **Enable Motor(s)**.
6. To start the motion click **Start motion**.

### *6.5.8 Jog Motion*

Jog Motion is a motion with constant velocity and without a defined end point. The motion continues until the next motion command stops it, or the motion fails because of limit switch activation or other condition.

To set the parameters for Jog Motion:

1. Activate Motion Manager.
2. Click **Jog Motion** ( ). The Jog Motion form is displayed.




3. Set the motion parameters:
   - **Direction** - select between positive and negative directions in accordance to the axis
   - **Feedback Position** - the current position
   - **Position Error** - the difference between current and required position coordinates
   - **Velocity** - defines the default velocity of the motion profile
   - **Acceleration** - how quickly the speed will reach the required limit
   - **Deceleration** - defines the speed reduction rate
   - **Kill Deceleration** - defines deceleration when a motion is killed or fails due to a fault
   - **Jerk** - stands for the jerk of the motion profile
4. If you need to define the motion parameters for axes other than X and Y, click **Add Axis (Axes)** and select the additional axis or axes.
5. Click **Enable Motor(s)**.
6. To start the motion click **Start motion**.

## 6.6 Safety and Faults Monitor

The Safety and Faults Monitor is, as it were, an expanded LED display enabling you to view the status of the variable faults that were set in *Section 3.4, Safety and Faults Configurator*.

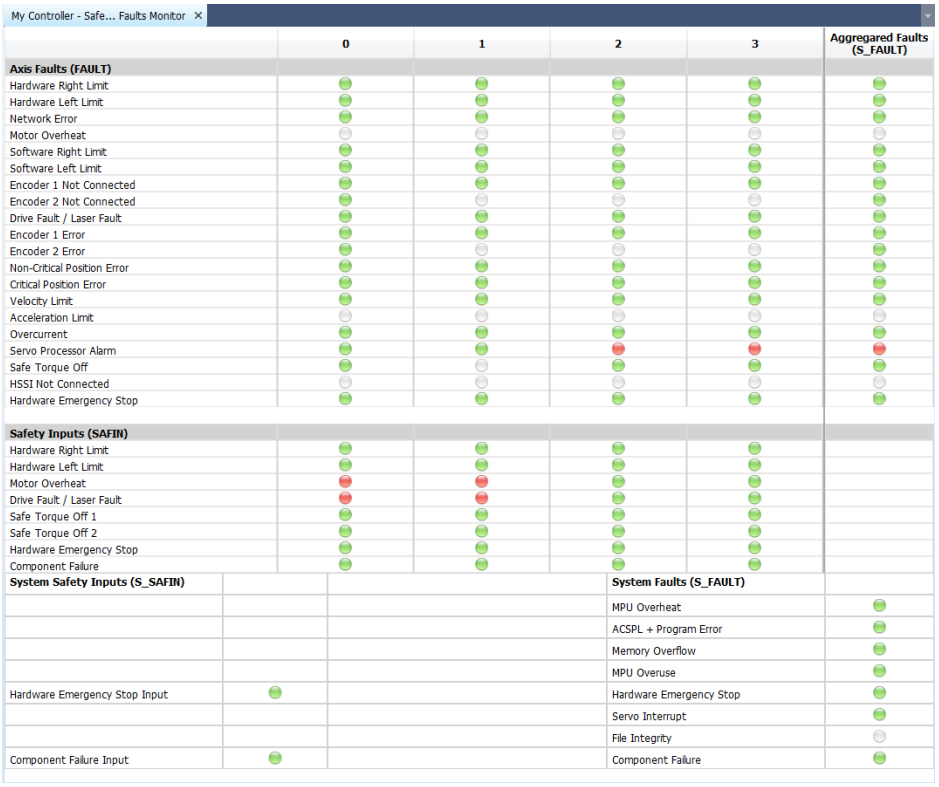
### 6.6.1 Activating Safety and Faults Monitor

To activate the Safety and Faults Monitor:

1. From the Toolbox click **Diagnostics and Monitoring** .
2. Click **Safety and Faults Monitor**.

You may use **Add Component** from the right-click menu of the controller in the Workspace Tree (see Controller Right-Click Options ) to activate the Safety and Faults Monitor.

The Safety and Faults Monitor window is displayed:




The fault status is distributed as follows:

> **Axis Faults** - These show the status of the axis motion as gathered by the ACSPL+ **FAULT** variable and reflect the following bits of the variable:

| Bit | Fault | Fault Description |
|---|---|---|
| 0 | #RL | Hardware Right Limit.<br>1 = Right limit switch is activated. |
| 1 | #LL | Hardware Left Limit<br>1 = Left limit switch is activated. |
| 2 | #NT | Network Error<br>1 = Network error has occurred. Valid only for SPiiPlus NT Motion Controllers. |
| 4 | #HOT | Motor Overheat |

| Bit | Fault | Fault Description |
|---|---|---|
| | | 1 = Motor's temperature sensor indicates overheat. |
| 5 | #SRL | Software Right Limit<br>1 = Axis reference position (**RPOS**) is greater than the software right limit margin (**SRLIMIT**). |
| 6 | #SLL | Software Left Limit<br>1 = Axis reference position (**RPOS**) is less than the software left limit margin (**SLLIMIT**). |
| 7 | #ENCNC | Encoder Not Connected<br>1 = Primary encoder (for digital encoder type only) is not connected. |
| 8 | #ENC2NC | Encoder 2 Not Connected<br>1 = Secondary encoder (for digital encoder type only) is not connected. |
| 9 | #DRIVE | Drive Fault / Laser Fault<br>1 = Signal from the drive reports a failure. |
| 10 | #ENC | Encoder Error<br>1 = Primary encoder miscounts. |
| 11 | #ENC2 | Encoder 2 Error<br>1 = Secondary encoder miscounts. |
| 12 | #PE | Non-Critical Position Error<br>1 = Non-critical position error has occurred.<br>PE is defined by the following variables:<br>**ERRI -** Maximum position error while the axis is idle<br>**ERRV** - Maximum position error while the axis is moving with constant velocity<br>**ERRA** - Maximum position error while the axis is accelerating or decelerating<br>**DELI** - Delay on transition from **ERRA** to **ERRI**<br>**DELV** - Delay on transition from **ERRA** to **ERRV** |

| Bit | Fault | Fault Description |
|---|---|---|
| 13 | #CPE | Critical Position Error<br>1 = Position error exceeds the value of the critical limit.<br>#CPE errors occur outside normal range of operation and #CPE > #PE.<br>The critical limit depends on the axis state and is defined by the following variables:<br>**CERRI** if the axis is idle (not moving)<br>**CERRV** if the axis is moving with constant velocity<br>**CERRA**if the axis is accelerating or decelerating<br>**DELI** - Delay on transition from **ERRA** to **ERRI**<br>**DELV** - Delay on transition from **ERRA** to **ERRV** |
| 14 | #VL | Velocity Limit<br>1 = Absolute value of the reference velocity (**RVEL**) exceeds the limit defined by the **XVEL** parameter. |
| 15 | #AL | Acceleration Limit<br>1 = Absolute value of the reference acceleration (**RACC**) exceeds the limit defined by the **XACC** parameter. |
| 16 | #CL | Current Limit<br>1 = **RMS** current calculated in the Servo Processor exceeds the limit value defined by the **XRMSD** or **XRMSM** parameter. |
| 17 | #SP | Servo Processor Alarm<br>1 = Axis Servo Processor loses its synchronization with the MPU. The fault indicates a fatal problem in the controller. |
| 18 | #STO | Safe Torque Off<br>1 = STO is activated. |
| 20 | #HSSINC | HSSI Not Connected<br>1 = HSSI module is not connected. |
| 23 | #ES | Hardware Emergency Stop<br>1 = **ES** signal is activated. |

> **Aggregated Faults** - These show the status of the axis motion as gathered by the ACSPL+ S_FAULT variable and reflect the following bits of the variable:

| Bit | Fault | Fault Description |
|---|---|---|
| 0 | #RL | Hardware Right Limit.<br>1 = Right limit switch is activated. |
| 1 | #LL | Hardware Left Limit.<br>1 = Left limit switch is activated. |
| 2 | #NT | Network Error<br>1 = Network error has occurred. Valid only for SPiiPlus NT Motion Controllers. |
| 4 | #HOT | Motor Overheat<br>1 = Motor's temperature sensor indicates overheat. |
| 5 | #SRL | Software Right Limit<br>1 = Axis reference position (**RPOS**) is greater than the software right limit margin (**SRLIMIT**). |
| 6 | #SLL | Software Left Limit<br>1 = Axis reference position (**RPOS**) is less than the software left limit margin (**SLLIMIT**). |
| 7 | #ENCNC | Encoder Not Connected<br>1 = Primary encoder (for digital encoder type only) is not connected. |
| 8 | #ENC2NC | Encoder 2 Not Connected<br>1 = Secondary encoder (for digital encoder type only) is not connected. |
| 9 | #DRIVE | Drive Fault / Laser Fault<br>1 = Signal from the drive reports a failure. |
| 10 | #ENC | Encoder Error<br>1 = Primary encoder miscounts. |
| 11 | #ENC2 | Encoder 2 Error<br>1 = Secondary encoder miscounts. |
| 12 | #PE | Non-Critical Position Error<br>1 = Non-critical position error has occurred.<br>PE is defined by the following variables:<br>**ERRI -** Maximum position error while the axis is idle |

| Bit | Fault | Fault Description |
|---|---|---|
| | | **ERRV** - Maximum position error while the axis is moving with constant velocity<br>**ERRA** - Maximum position error while the axis is accelerating or decelerating<br>**DELI** - Delay on transition from **ERRA** to **ERRI**<br>**DELV** - Delay on transition from **ERRA** to **ERRV** |
| 13 | #CPE | Critical Position Error<br>1 = Position error exceeds the value of the critical limit.<br>#CPE errors occur outside normal range of operation and #CPE > #PE.<br>The critical limit depends on the axis state and is defined by the following variables:<br>**CERRI** if the axis is idle (not moving)<br>**CERRV** if the axis is moving with constant velocity<br>**CERRA**if the axis is accelerating or decelerating<br>**DELI** - Delay on transition from **ERRA** to **ERRI**<br>**DELV** - Delay on transition from **ERRA** to **ERRV** |
| 14 | #VL | Velocity Limit<br>1 = Absolute value of the reference velocity (**RVEL**) exceeds the limit defined by the **XVEL** parameter. |
| 15 | #AL | Acceleration Limit<br>1 = Absolute value of the reference acceleration (**RACC**) exceeds the limit defined by the **XACC** parameter. |
| 16 | #CL | Current Limit<br>1 = **RMS** current calculated in the Servo Processor exceeds the limit value defined by the **XRMSD** or **XRMSM** parameter. |
| 17 | #SP | Servo Processor Alarm<br>1 = Axis Servo Processor loses its synchronization with the MPU. The fault indicates a fatal problem in the controller. |
| 18 | #STO | Safe Torque Off<br>1 = STO is activated. |

| Bit | Fault | Fault Description |
|---|---|---|
| 20 | #HSSINC | HSSI Not Connected<br>1 = HSSI module is not connected. |
| 23 | #ES | Hardware Emergency Stop<br>1 = **ES** signal is activated. |

> **Safety Inputs** - These show the status of the axis motion as gathered by the ACSPL+ SAFIN variable which indicates the raw state, before processing, of the axis safety inputs and reflect the following bits of the variable:

| Bit | Bit Name | Description |
|---|---|---|
| 0 | #RL | Hardware Right Limit |
| 1 | #LL | Hardware Left Limit |
| 4 | #HOT | Motor Overheat |
| 9 | #DRIVE | Drive Fault / Laser Fault |
| 17 | #STO1 | Safe Torque Off 1 |
| 18 | #STO2 | Safe Torque Off 2 |
| 28 | #ES | Hardware Emergency Stop |
| 31 | #COMP | Component Failure |

> **System Safety Inputs** - These show the status of the controller program as gathered by the ACSPL+ S_SAFIN variable which indicates the raw state, before processing, of the inputs and reflect the following bits of the variable:

| Bit | Fault | Fault Description |
|---|---|---|
| 28 | #ES | Hardware Emergency Stop<br>1 = **ES** signal is activated. |
| 31 | #FAILURE | Component Failure<br>1 = An MC4U hardware component other than the drive, such as Power Supply, I/O card, or encoder card, has failed. |

> **System Faults** - These show the status of the controller program as gathered by the ACSPL+ S_FAULT variable and represent the following bits of the variable:

| Bit | Fault | Fault Description |
|---|---|---|
| 25 | #PROG | Program Fault<br>1 = Run time error occurs in one of the executing ACSPL+ programs. |
| 26 | #MEM | Memory Fault<br>1 = User application requires too much memory. |
| 27 | #TIME | MPU Overuse<br>1 = User application consumes too much time in the controller cycle. |
| 28 | #ES | Hardware Emergency Stop<br>1 = ES signal is activated. |
| 29 | #INT | Servo Interrupt<br>1 = The servo interrupt that defines the controller cycle is not generated. The fault indicates a fatal controller problem. |
| 30 | #INTGR | File Integrity<br>1 = The integrity of the user application in controller RAM is checked by the controller at power-up and whenever an **#IR** command is issued. |
| 31 | #FAILURE | Component Failure<br>1 = An MC4U hardware component other than the drive, such as the Power Supply, I/O card, or encoder card, has failed. |

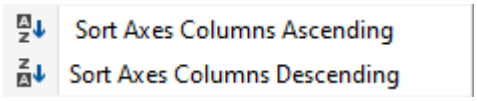
The "LED" indicators have the following meanings:

| | |
|---|---|
|  | No fault. |
|  | Fault detected. |
|  | Not applicable. |

### *6.6.2 Working with Safety and Faults Monitor*

You have the following actions that you can take when viewing the Safety and Faults Monitor:

For Axis Faults to get the action menu for any column, right-click the column number. The following popup menu appears:




Your options are:

> To sort a column by ascending order click Sort Axes Columns Ascending.

> To sort a column by descending order click Sort Axes Columns Descending.

The **System Faults** is fixed and you cannot change it.

## 6.7 Scope

The SPiiPlus MMI Application Studio **Scope** enables you to view a real-time graphic display of the motion. It provides you with a wide diversity of display options that allow you to define the graph's display to meet your needs.

### 6.7.1 Activating the Scope

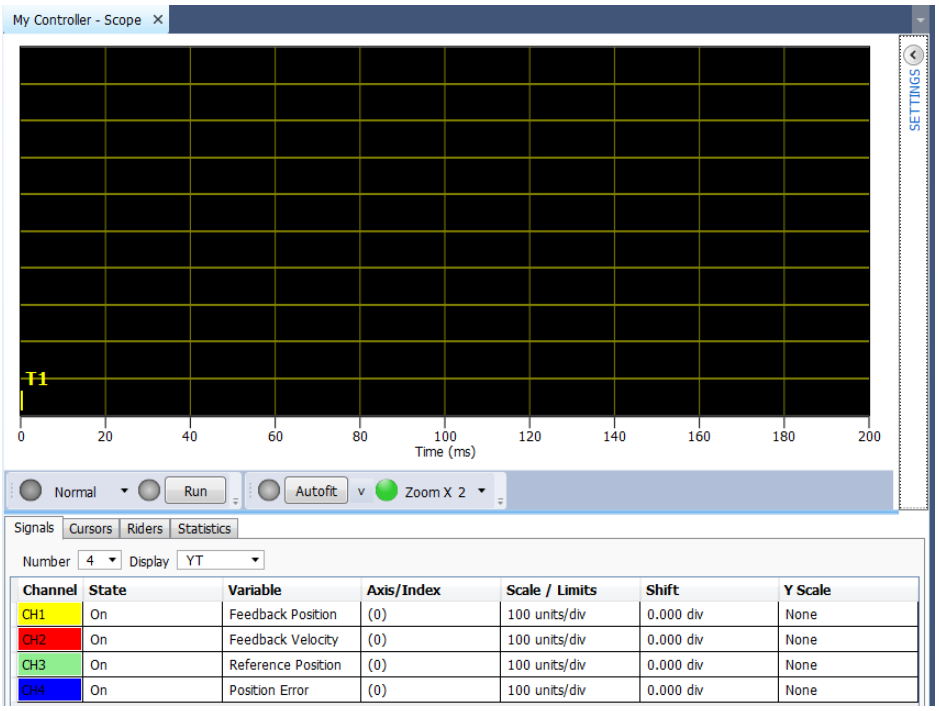
To activate the Scope:

1. From the Toolbox, click **Diagnostics and Monitoring** to display the Diagnostics and Monitoring tools list.
2. Click **Scope**.

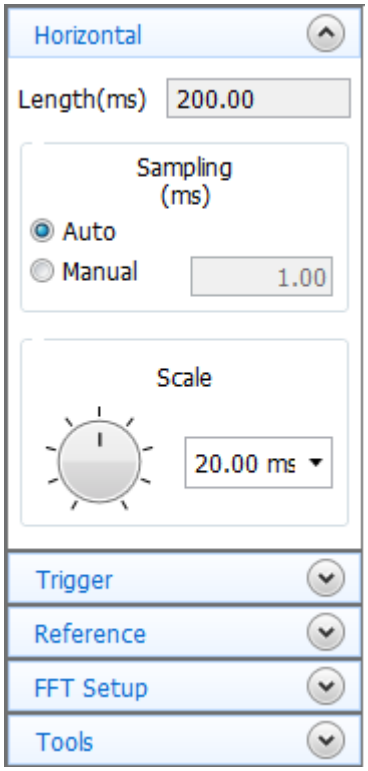
You can use **Add Component** from the right-click menu of the controller in the Workspace Tree (see Controller Right-Click Options) to activate the Scope.

The Scope display can also be triggered from the Communication Terminal by the clicking **Send and Trigger Scope**, and from the Program Manager by clicking  or pressing **Ctrl+Alt+T**, and from Motion Manager by selecting the **Trigger Scope on Motion** checkbox.

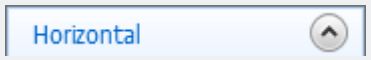
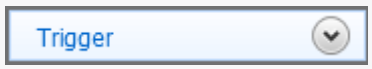
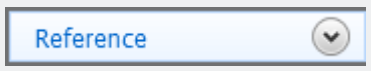
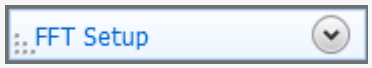
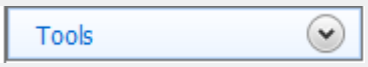
The **Scope** window is displayed:




You can set the graphic display parameters by clicking **[Settings]** on the right of the Scope display.

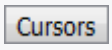
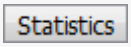


There are four options for setting the graphic display parameters:

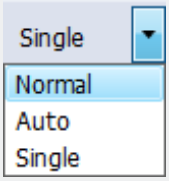
| | |
|---|---|
|  | Used for setting the horizontal scale of the graph. See Setting Horizontal Scale. |
|  | Used for setting the trigger parameters at which points the motion is sampled. See Setting Triggers. |
|  | Used for setting the reference points to assist in analyzing simple graphical signals of motion. See Setting Reference Axis |
|  | Used for selecting an FFT (Fast Fourier Transform) algorithm to be used in displaying the graph. See FFT Setup. |
|  | Save and load signal recordings. |

To expand an option click:  . To reduce, click  .

Underneath the Scope display there is a section for defining what channels are displayed and for viewing the measured values. This section is divided into four tabs:

| | |
|---|---|
|  | This tab enables you to set the number of channels to be sampled, how they are to be sampled, and to set what is to be displayed. See Signals Tab. |
|  | This tab enables you to set cursors on the graph to view the values between selected points on the graph. See Cursors Tab. |
|  | This tab enables you to see the specific values along the graph. You can move the Riders along the graph and view the values at the intersection of the Rider and the graph. See Riders Tab. |
|  | This tab displays all the measured statistics for the channels being sampled. See Statistics Tab. |

There are four display controls directly under the Scope. These are:




You use this to set how the graph is displayed. Your options are:

- **Normal** - The graph is displayed according to the Trigger setting. If a signal is outside of the Trigger setting, it is not displayed - only those signals within the Trigger.
- **Auto** - If the signals are all within the Trigger setting, this setting acts the same as **Normal**; however, if a signal is outside of the Trigger setting, after some delay it is also displayed.
- **Single** - The graph is displayed for a single frame, that is, the data is displayed for the length of the time set in the Horizontal scale and then stop. Each time you click **Run**, another frame is displayed.

These options are connected to the definition of the Trigger.


This appears if the data collection for the graph is not running. To start the data collection and the graph display, you click this button.


When you click 

to start the data collection, it changes to this button. You click this button to stop the data collection.


In some cases the graph may exceed the boundaries of the Scope screen. Click for the Scope to resize the display so that all data is displayed. You have four **Autofit** options:

- **Full Screen** - This option sets the Scope to resize the entire screen, thus each time you click **Autofit**, the data is refreshed and displayed in the Full Screen mode.
- **Between Cursors** - This option reduces the Scope display to the area between the cursors. Each time you click **Autofit**, the data is refreshed and displayed for the area between the cursors.
- **Permanent Full Screen** - This option sets the Scope display permanently to the Autofit Full Screen display mode.
- **Permanent Between Cursors** - This option sets the Scope display permanently to the Autofit Between the Cursors screen display mode.

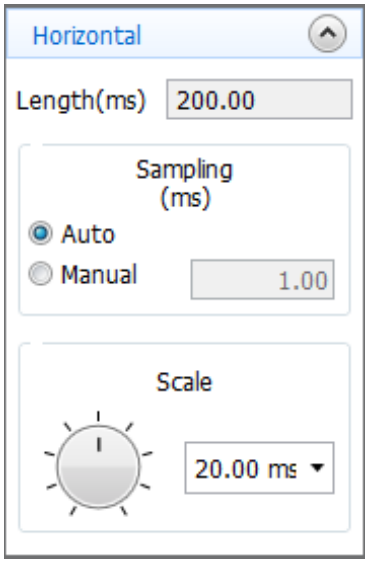
| | |
|---|---|
| Zoom | You have the following Zoom options:<br>> Zoom In: Hold Shift key and place the mouse cursor in the Scope display in area you want to zoom into. Click the Left mouse button to display the Zoom icon. When the icon appears, click the Left mouse button. Without releasing the Shift key, each time you click the Left mouse button, you increase the zoomed display.<br>> Zoom In Area: While holding the Shift key, press the Left mouse button and drag to mark the area you want to zoom, then release the mouse button.<br>> Zoom Out: While holding the Shift key, place the mouse cursor in the zoomed area and click the Right mouse button. Without releasing the Shift key, each time you click the Right mouse button, you zoom out one level.<br>> Move Graph: Hold Ctrl key and Left mouse button - a hand is displayed. Drag the graph to where you want.<br>> Undo - To undo all zooms, press Shift + Backspace keys. (You can also undo the zoom by clicking associated with Zoom). |

Whenever you have activated one of the above display functions, appears to indicate that it is active.

### 6.7.2 Setting Horizontal Scale

You set the horizontal scale (the X axis) of the Scope graph using the 

options. If the options are not visible, click  to expand.




Your options are:

- **Length**
- The length of the Scope X axis is for information only. It reflects the Sampling rate and Scale that you establish.
- **Sampling**
- Sampling reflects the point in time at which Scope measures the values. It can be:
  - **Automatic** - In which case Scope determines, based on the Scale, where the sampling points are.
  - **Manual** - Which enables you to enter your own value.

If you enter a value that not within range of the Scope algorithm, Scope will enter the closest minimum value to the one you have entered.

- **Scale**
- Scale establishes the value of each division of the X axis. It ranges between 0.05 ms/division to 10 sec/division.

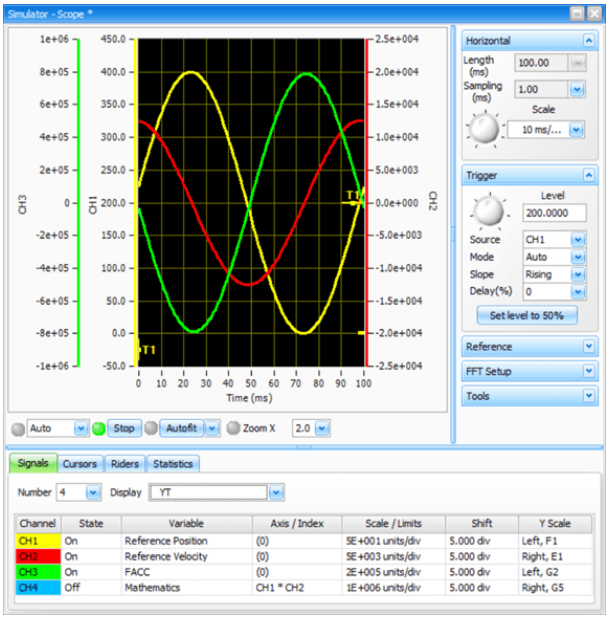
You can either select the value from the dropdown list, or use the dial button:  .

To use the dial button, place the mouse cursor over the button and drag it left or right.

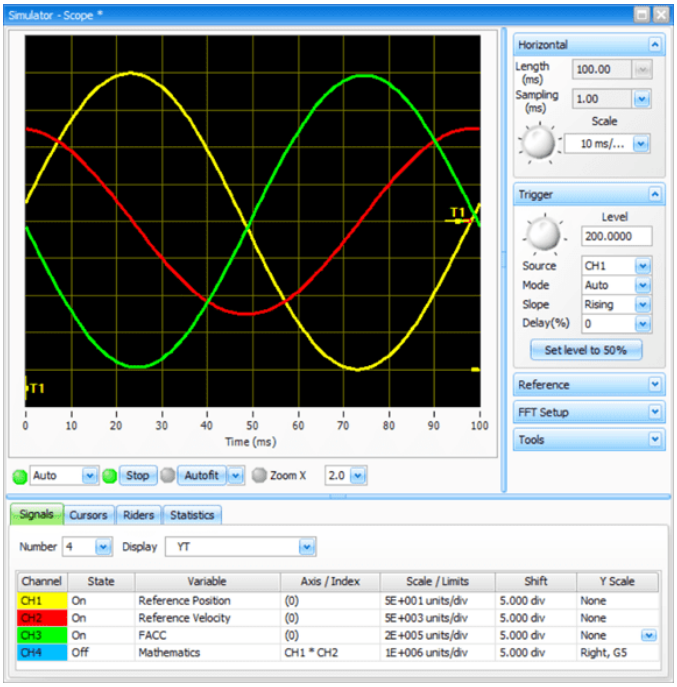
### *6.7.3 Ability to display Y-axis units*

This feature includes three new properties for each Channel:

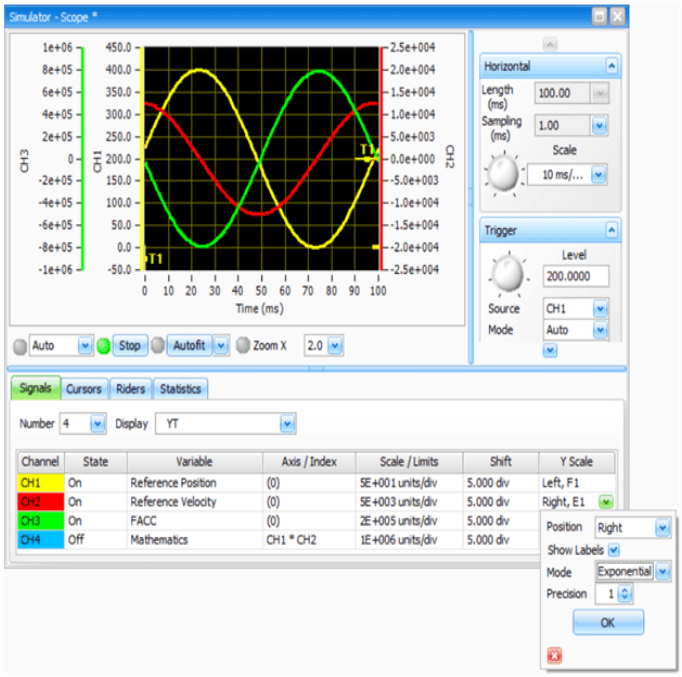
- Scale Label Mode (General, Fixed Point, Exponential)
- Scale Label Precision (digital number from 0 up to 99 )
- Scale Label Visible (true, false)
- Scale Position (None, Left, Right)



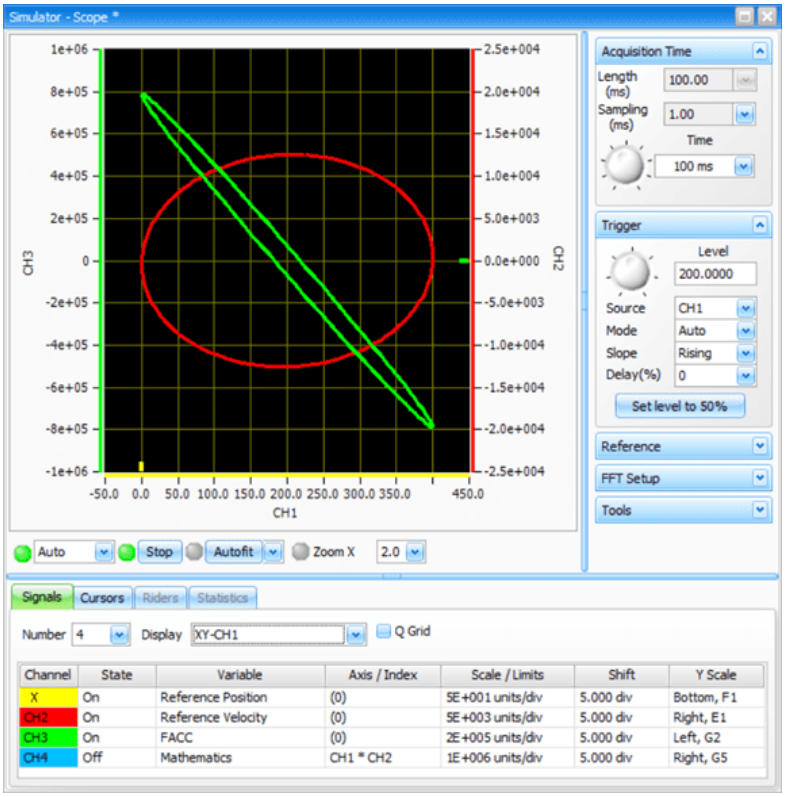

For YT or FFT modes the scales display as shown below.




If the scale position for all scales select as "None", the Scope display is:



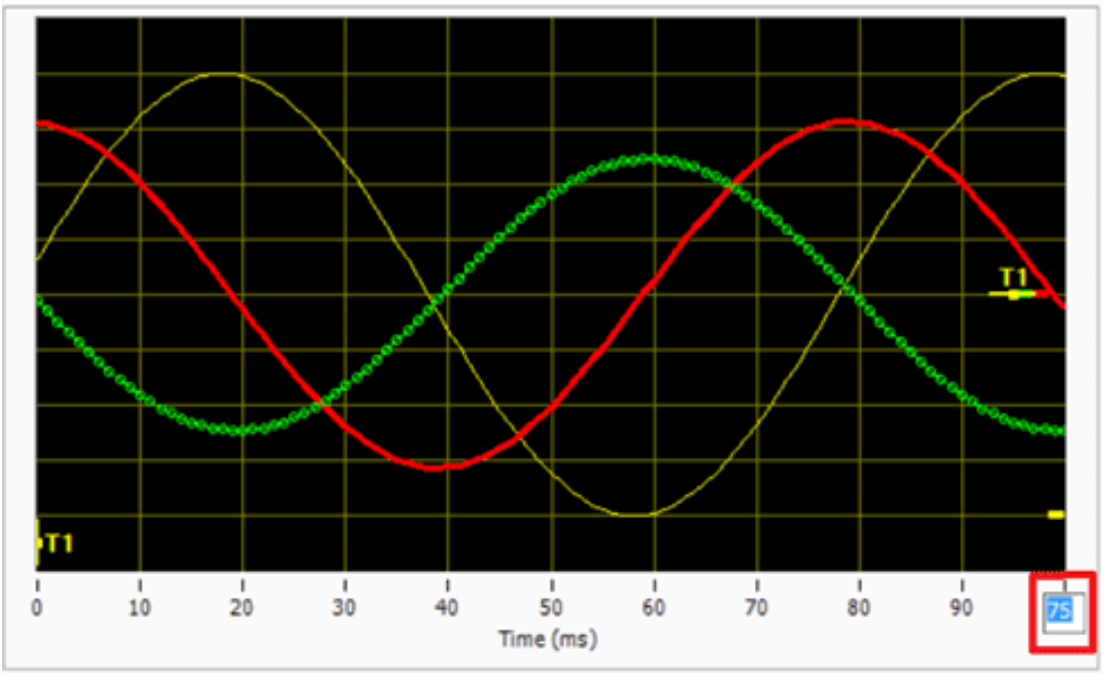

If the Scope display is set to the XY mode, then the Left/Right positions display as bottom position as shown.



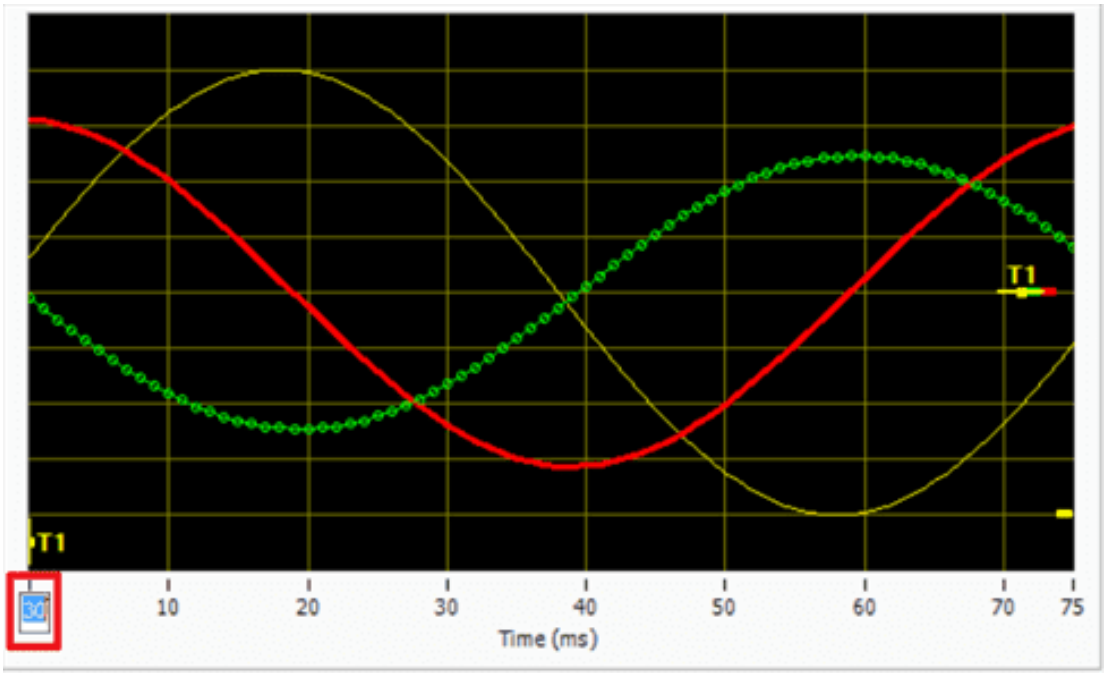

### *6.7.4 Setting X-axis zoom interval bounds*

To select the X- Axis zoom interval bounds, click on Left/Right labels of the Time scale and enter the desired the value.

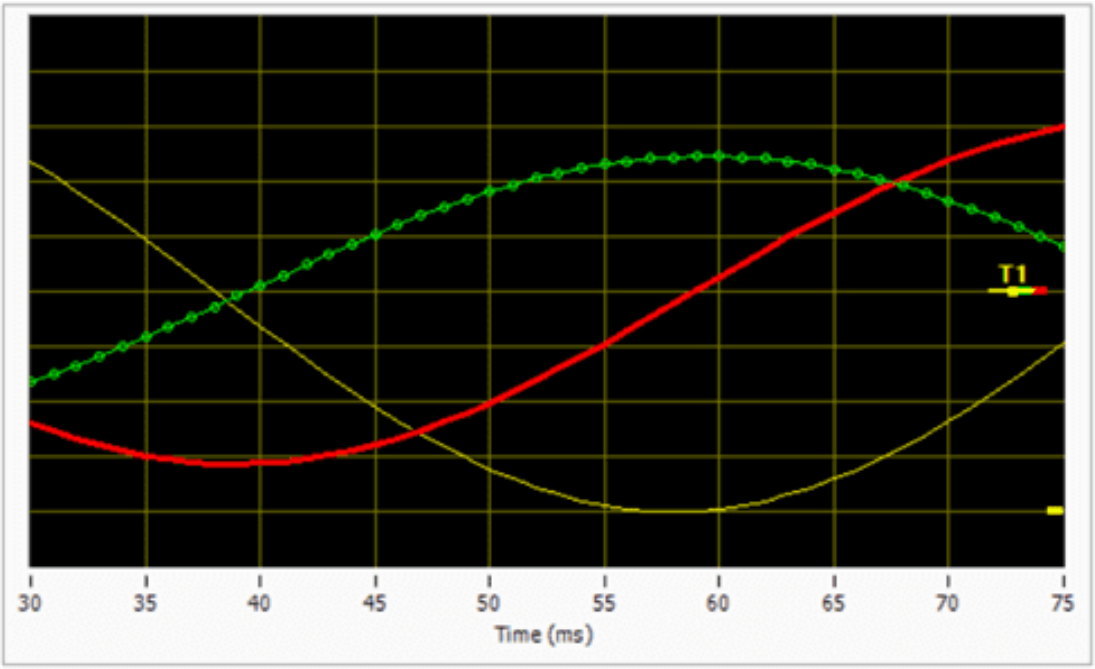
**Example to change the maximum (upper) bound.**


T1
T1
0
10
20
30
40
50
60
70
80
90
75
Time (ms)

**Example to change the minimum (lower) bound.**




**Result of after resetting the lower and upper bounds.**

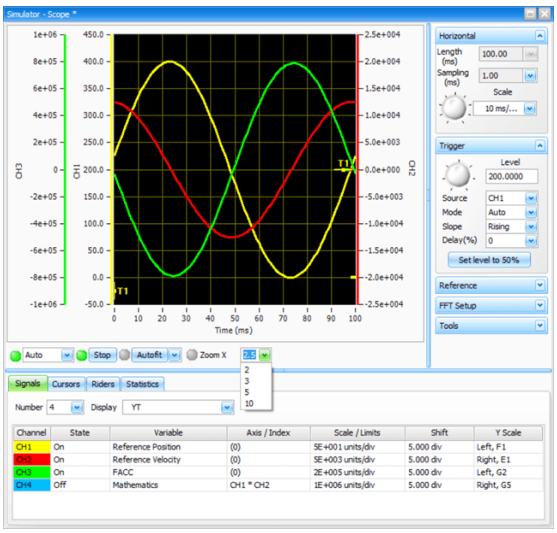



### *6.7.5 Zoom factor*

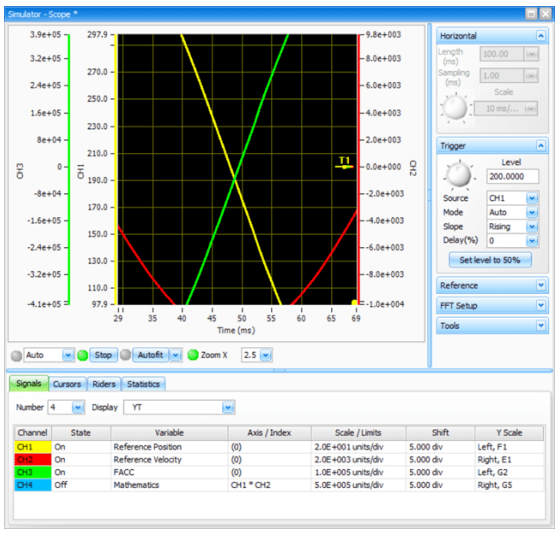
The zoom factor provides the ability to specify a desired zoom level. The selectable range for the Zoom Factor is from 1 up to 10.

**Example of Zoom Factor 2.5.**

Scope display before setting the zoom factor



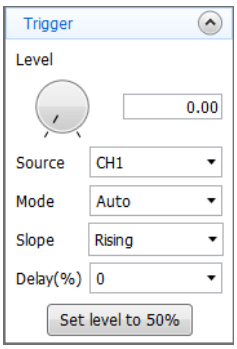
Scope display after setting the zoom facor to 2.5.



## *6.7.6 Setting Triggers*

You set the Triggers, that is, the points at which you want data collection to begin, of the Scope graph using the 

options. If the options are not visible, click  to expand.




Your options are:

> **Level**
This is the threshold value of the signal for the start of data collection. When the signal reaches this value, data collection begins and the signal's graph is displayed.

It is recommended that you set the Trigger value broad enough to obtain a usable graph, then slowly fine tune the results.

You can either enter the value directly into the field, or use the dial button:  .

To use the dial button, place the mouse cursor over the dial and drag it left or right.

- **Source**
  Select the triggering channel from the dropdown list.
- **Mode**

  There are three possible modes (selected from the dropdown list):

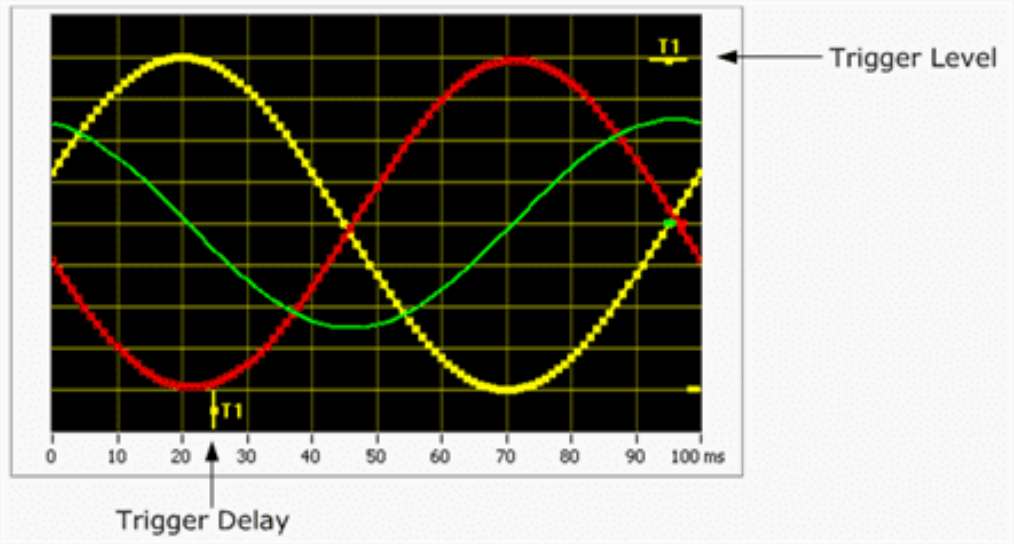
  - **Normal** - Samples when the triggering signal is within the settings in the Trigger dialog box.
  - **Auto** - If the triggering signal is within the Trigger setting, this mode is the same as **Normal**. However, even if the triggering signal is outside the Trigger setting, after a short delay, it will be sampled. The sampling period is proportional to the Horizontal setting.
  - **Single** - Samples the triggering signal only once over the length of the Horizontal setting and then stops. It repeats this for the next frame each time you click **Run**.
- **Slope**

  You can set the triggering to:

  - **Rising** – Trigger on signal rising past trigger level.
  - **Falling** – Trigger on signal falling past trigger level.
- **Delay**

  Delay polling for trigger until a percentage of the X axis, selected from the dropdown list, has passed.

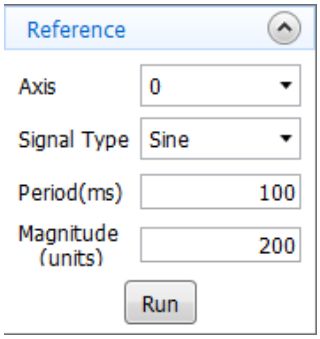
The Triggers are displayed as follows:




Clicking Set level to 50% sets the Y axis trigger point to 50% of the height of the Y axis.

### 6.7.7 Setting Reference Axis

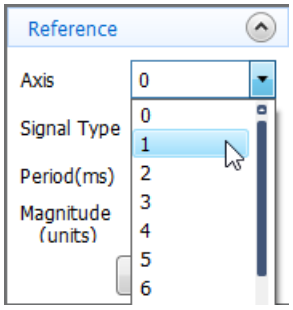
You can use a References axis to assist you in analyzing simple graphical signals of motion. You set the Reference axis using the Reference options. If the options are not visible, click to expand.




Your options are:

- **Axis**

  Select the required axis from the dropdown list.




- **Signal Type**

  Select the signal type from the dropdown list. The available signal types are:

  - PULSE
  - RAMP
  - SINE

- **Period**

  Enter the required period (in ms).

- **Magnitude**

  Enter the required signal magnitude.

Once you have entered the your settings, click Run and the graph will be displayed reflecting your choices.

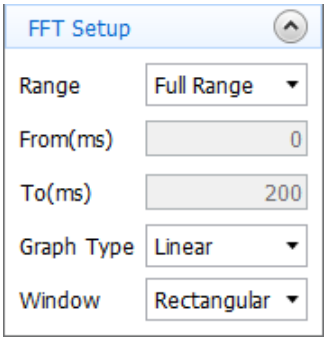
You may use the Riders (see Riders Tab) to view the exact values at points along the graph, like the maximum and minimum values.

To stop the display and return to the regular Scope display click Stop.

### *6.7.8 FFT Setup*

The FFT option enables you to view the signals on the Scope in an FFT (Fast Fourier Transform) format. The MMI Scope has a wide variety of FFT algorithms that are available to you. The FFT format displays the Spectral Density against the Frequency

You set the FFT display using the FFT Setup options. If the options are not visible, click to expand.




Your options are:

- **Range**

  The range can be:

  - Full Signal - Covers the continuous run of the signal
  - Window - Covers a portion of the signal, which you set by entering values in **From** and **To**.

- **Graph Type**

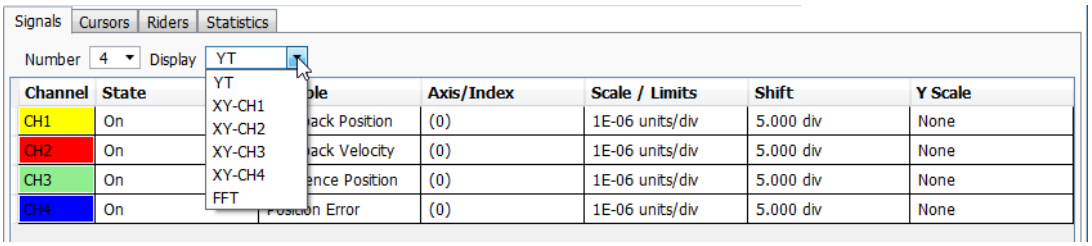
  The graph type can be:

  - Linear
  - dB (logarithmic)

- **Window**

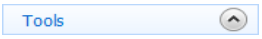
You select the specific FFT format from this dropdown list.

Once you have set the parameters, in the **Signals** tab (see Signals Tab) select FFT from the Display dropdown to view the FFT display in the Scope.

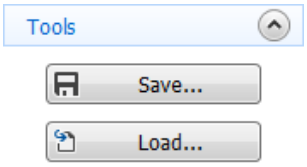



## *6.7.9 Tools*

The Scope Tools option enables you to save your signal data to a file for later retrieval.

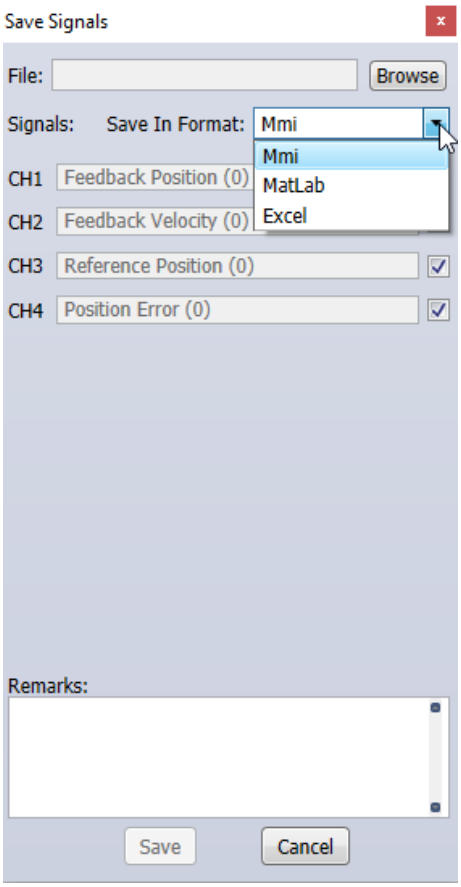
To save or load your data use the 

options. If the options are not visible, click  to expand.




**To Save Signal Data**

1. Click **Save**. The Save Signals dialog is displayed:




2. Select the file type desired. You have the options of:

> MMI signal file with **.sgn** file extension.

> Matlab data file with **.m** file extension.

> Xcel CSV file with **.csv** file extension.

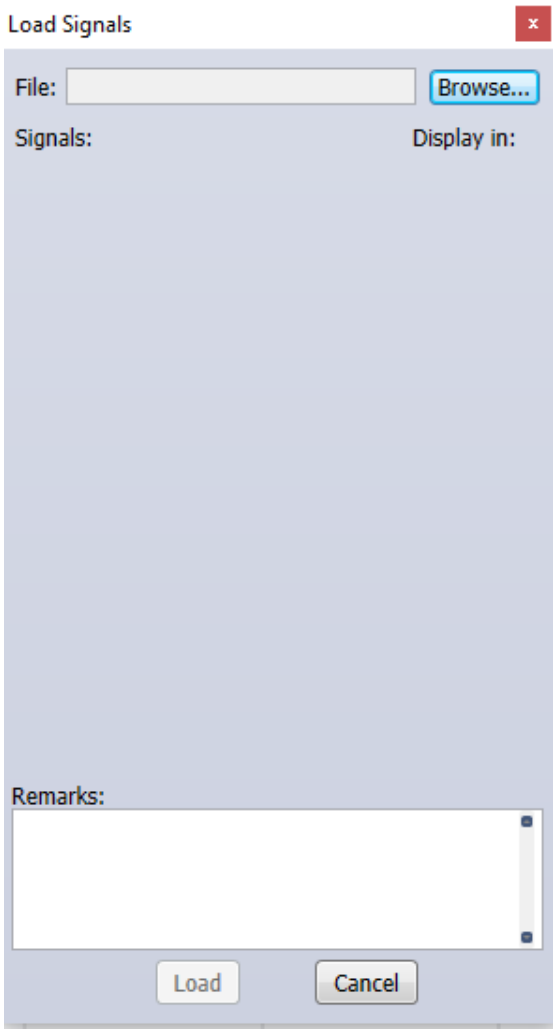
3. Either enter a path and filename in the **File** field, or click **Browse** to find the path (and filename, if needed).

4. Click  to unselect those signals whose data you do not want to save.

5. Click **Save**.

**To Load Signal Data**

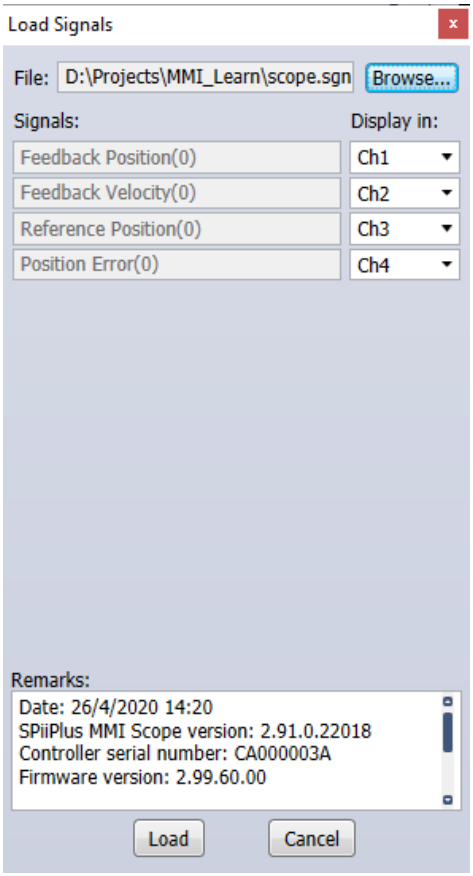
1. Click **Load**.




2. Either enter a path and filename in the **File** field, or click **Browse** to find the file.

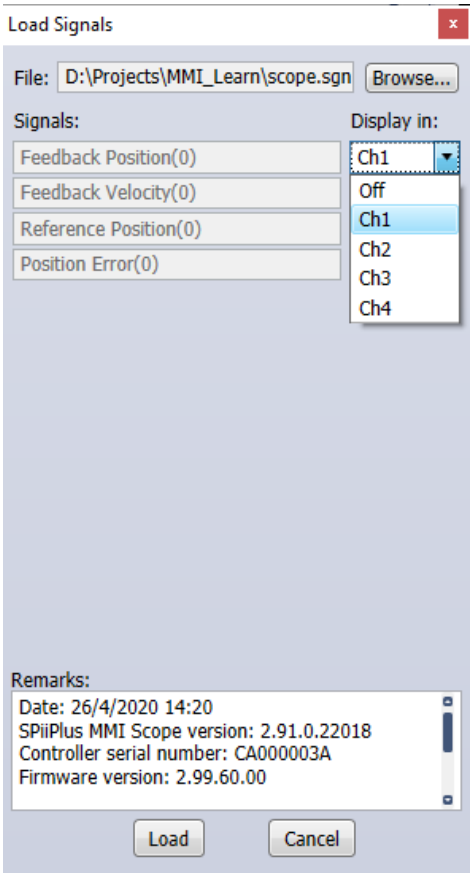
Scope signal data is stored in files having **.sgn** file extensions. Files saved on older versions of the MMI can be loaded.

The contents of the file are displayed, for example,




3. The Scope file contains the signal data that was collected in a previous session, arranged according the channels that were sampled. You have the option to change the assignment of these signals if you desire.

Click  to view the channels to which the signal may be applied.




You may select:

> **Off** - does not retrieve this particular data

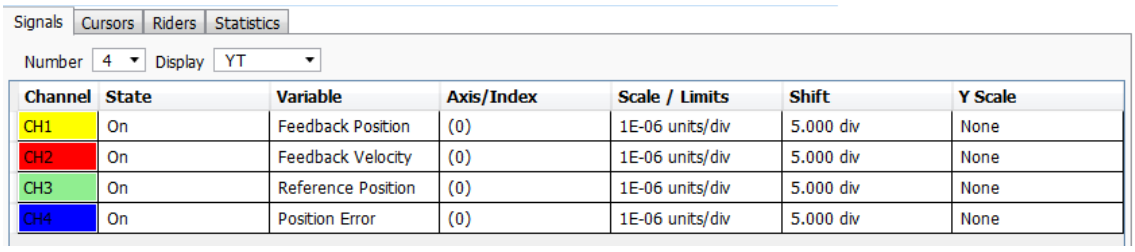
> **CH#** - puts this data into the selected channel

4. Click **Load**

When the signals are loaded, they are loaded in the **Freeze** state. Once loaded, you can change the state to whatever you want. See State.

### 6.7.10 Signals Tab

The Signals tab is used for setting the channels that are to be graphed in the Scope.




| Channel | State | Variable | Axis/Index | Scale / Limits | Shift | Y Scale |
|---|---|---|---|---|---|---|
| CH1 | On | Feedback Position | (0) | 1E-06 units/div | 5.000 div | None |
| CH2 | On | Feedback Velocity | (0) | 1E-06 units/div | 5.000 div | None |
| CH3 | On | Reference Position | (0) | 1E-06 units/div | 5.000 div | None |
| CH4 | On | Position Error | (0) | 1E-06 units/div | 5.000 div | None |

The following describes how you work with each field in the tab.

> Number

- Display
- State
- Variable
- Axis/Index
- Scale/Limits
- Shift

### Number

You use the **Number** field to set the number of channels to be tracked.

1. Click  to display the number of channels available. A list of numbers from 1 to 8 is displayed.
2. Click the last number to be displayed, for example, if you are measuring four channels, you would click **4**, if you are measuring eight channels (the maximum), you would click **8**.

The channels are displayed in the **Channel** column.

### Display

You use the **Display** field to set the type of display you want on the Scope.

1. Click  to display the options available.

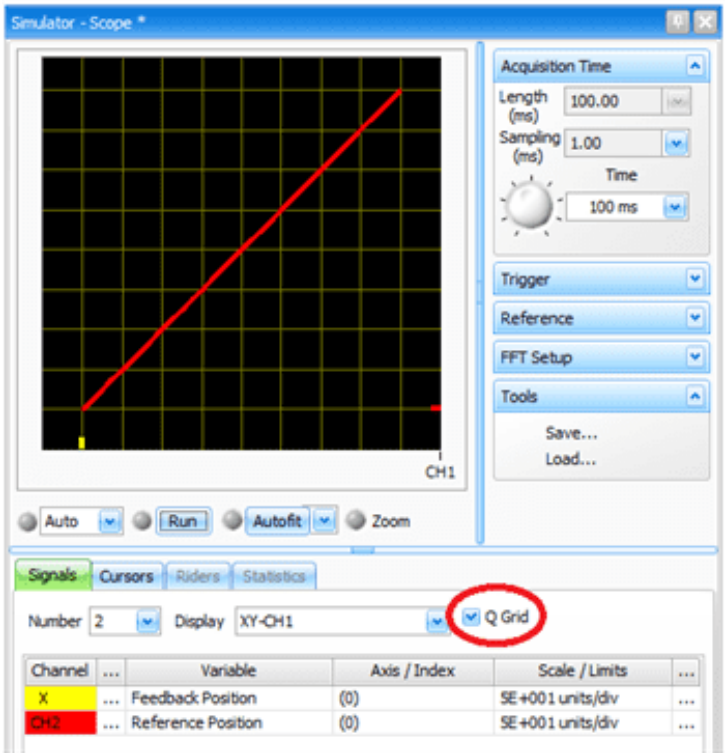
   Your display options are:

   - YT - Channel(s) vs. Time. This is the most common display.
   - XY-CH# - All signals measured against the selected channel.

If the XY-CH# display option is selected, then an option for a quadrature grid is available.

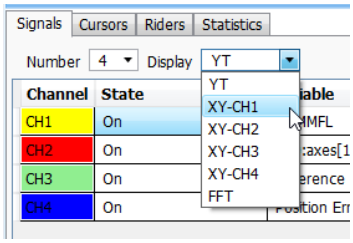
- To enable the quadrature grid, select the **Q grid** check box.

  When **Q grid** is enabled, the ratio of the X axis scaling to the Y axis scaling remains constant as the display window is resized.

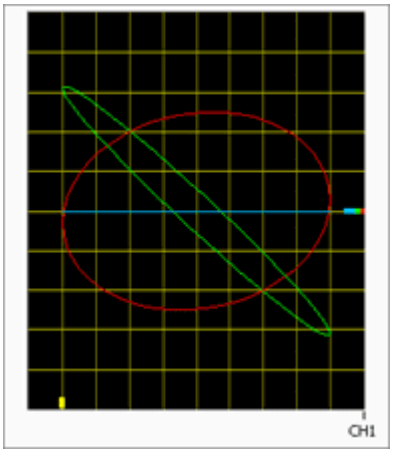



> FFT - Fast Fourier Transform display (also see FFT Setup)

2. Click the display option to activate it. For example:

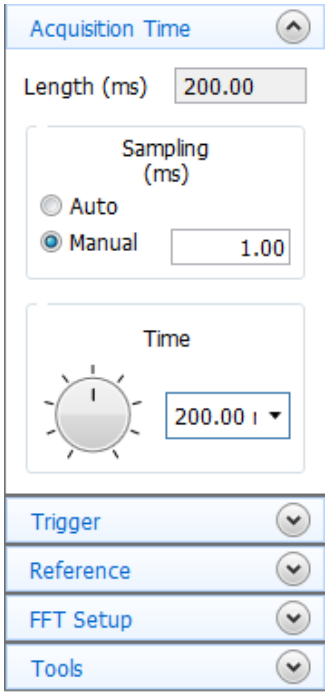



The Scope display changes to:




mapping the signals against the selected channel.

The **Horizontal** option changes to **Acquisition Time**:




This enables you to set the data collection time for measuring the value (in the same way as Setting Horizontal Scale).

**State**

A signal can have one of three states:

- On - the signal is active
- Off - the signal is inactive
- Freeze - the signal is frozen at its current value

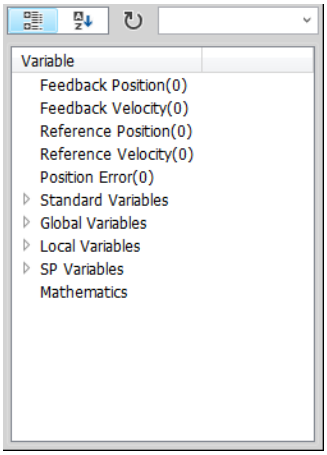
To set the signal's state click the **State** field and select the state from the dropdown menu.

**Variable**

Signal values are tracked through various program variables, such as ACSPL+ variables, SP variables, user-defined variables, and the like. You use this field to track the signal via a specific variable.

To select the variable you want to track:

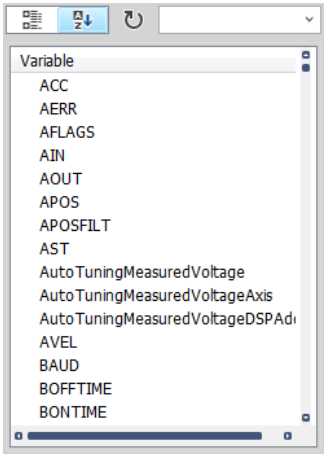
1. Click the **Variable** field, the Variables tree appears:



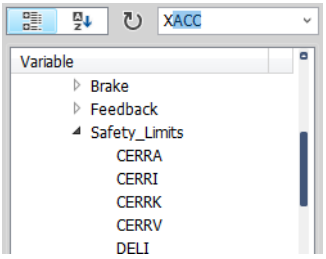

The list is accumulative, it maintains a history of all variables you have selected in the past.

2. Drill down to the variable you want by expanding the tree, where needed, and click the variable.

You can view the variables list sorted alphabetically by clicking **Sorted**.




You can type the variable name in the search bar:




Click to ... (refresh???)

The signal variable list items are:

| Item | Description |
|---|---|
| Feedback Position | Motor position based on the **FPOS** variable. |
| Feedback Velocity | Motor velocity based on the **FVEL** variable. |
| Feedback Acceleration | Motor acceleration based on the **FACC** variable. |
| Reference Position | Reference position based on the **RPOS** variable. |
| Reference Velocity | Reference velocity based on the **RVEL** variable. |
| Standard Variables | This branch lists the built-in ACSPL+ variables. These variables are broken down into:<br>> Axis_State<br>> Monitoring<br>> Motion<br>> Safety_Control<br>> Inputs_Outputs<br>> Program_Execution_Control<br>> System_Configuration<br>> Axis_Configuration<br>> Communication<br>> Commutation<br>> Data_Collection<br>> Servo_Loop<br>> Miscellaneous<br>> Obsolete |
| Global Variables | Lists all global variables declared in any compiled buffer. |
| Local Variables | Lists variables broken down by the buffers within which they are declared. |
| SP Variables | Lists variables stored in the SPiiPlus controller RAM as part of the controller's firmware. The variables are broken down by controller. |
| Mathematics | Signal is derived from an arithmetic operation performed on two channels. The operation is defined in the Axis/Index field. |

If a variable is an array or matrix, select an array element in the Axis/Index field.

Refer to the *SPiiPlus Command & Variable Reference Guide* and the *SPiiPlus ACSPL+ Programmer's Guide* for details on all the variables.

### Axis/Index

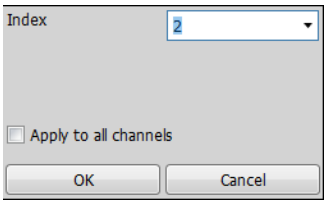
The **Axis/Index** field is used in three cases:

- Selecting an index for an array variable
- Selecting a bit for a variable that can be bit-accessed (including elements of an array variable)
- Defining a mathematical operation for two channels

### Selecting a Real Array Index

To select an index for a real array variable:

1. Click the **Axis/Index** field for the array variable. A Select Index dialog box appears, for example:



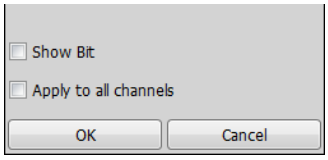

If the variable is a two-dimensional matrix, two **Index** fields appear.

2. Enter the index number in the **Index** field
3. If you want it to apply the index to all of the channels being measured, select the **Apply to all channels** checkbox.
4. Click **OK**.

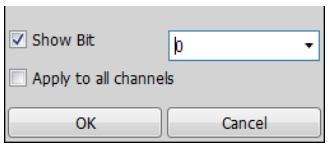
### Selecting a Scalar Bit Number

To select a bit number for a bit-accessible scalar variable:

1. Click the **Axis/Index** field for the variable, the following dialog box appears:

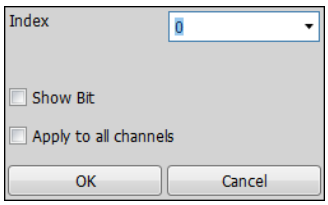



2. Select the **Show Bit** checkbox. The dialog box expands to:




3. Enter the bit number and click **OK**.

If array is a two-dimensional scalar matrix, the following appears:

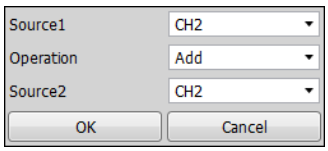



In this case you have to select both the array index, and the bit number.

### Defining a Mathematical Operation

To define a mathematical operation between two channels:

1. Click the **Axis/Index** field for a variable defined as a Mathematics array variable. A dialog box appears:




2. Use the down arrow to select **Source1**
3. Use the down arrow to select the **Operation**. Your options are:
   - > Add
   - > Subtract
   - > Multiply
   - > Divide
4. Use the down arrow to select **Source2**. Click **OK**.

### Scale/Limits

The **Scale/Limits** field is used in two cases:

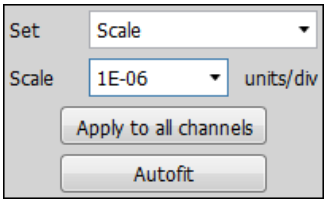
- > To adjust the Y axis (vertical) units used for displaying of a selected signal in the graph

> To set the Y axis (vertical) limits used for displaying of a selected signal in the graph

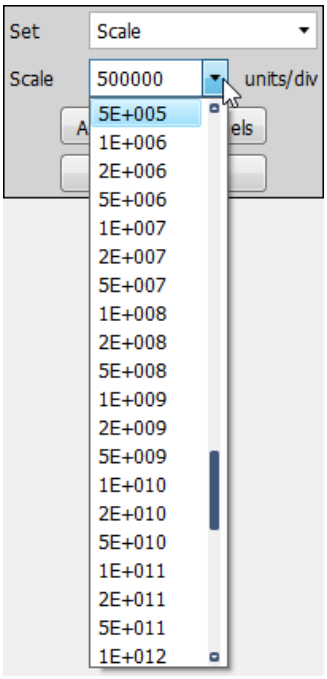
This is often useful for when the graph exceeds the borders of the Scope and you want to see the complete graph.

### Adjusting the Y axis units

1. Click the **Scale/Limit** field for the desired signal. A dialog box appears:




2. Select **Scale** (default).
3. Use the dropdown list to adjust the scale:

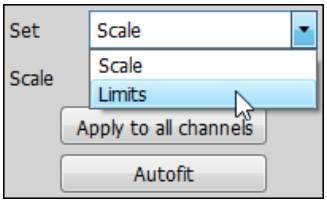



or
Click **Autofit** to center the signal by altering the scale so that the entire graph is visible.

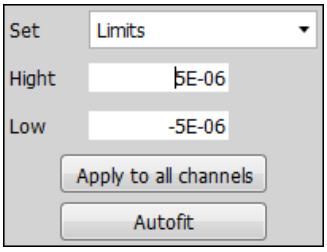
4. You can apply it to all channels by selecting the **Apply to all channels** checkbox.

### Setting the Y axis limits

1. Click the **Scale/Limit** field for the desired signal, and in the **Set** dropdown field select **Limits**.




2. Enter the maximum limit in the **Height** field,
and the minimum limit in the **Low** field.




You can click **Autofit** to center the signal by altering the limits so that the entire graph is visible.

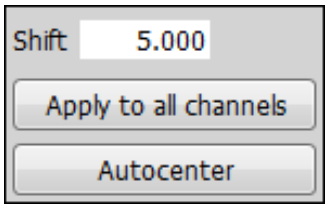
3. You can apply it to all channels by selecting the **Apply to all channels** checkbox.

**Shift**

You use the **Shift** field to move the graph of a selected signal along the Y axis (vertical).

This is often useful for when the graph is cluttered and you want to see a particular signal on the graph separated from the others.

1. Click the **Shift** field of the channel the display of which you want to move. A dialog box appears:



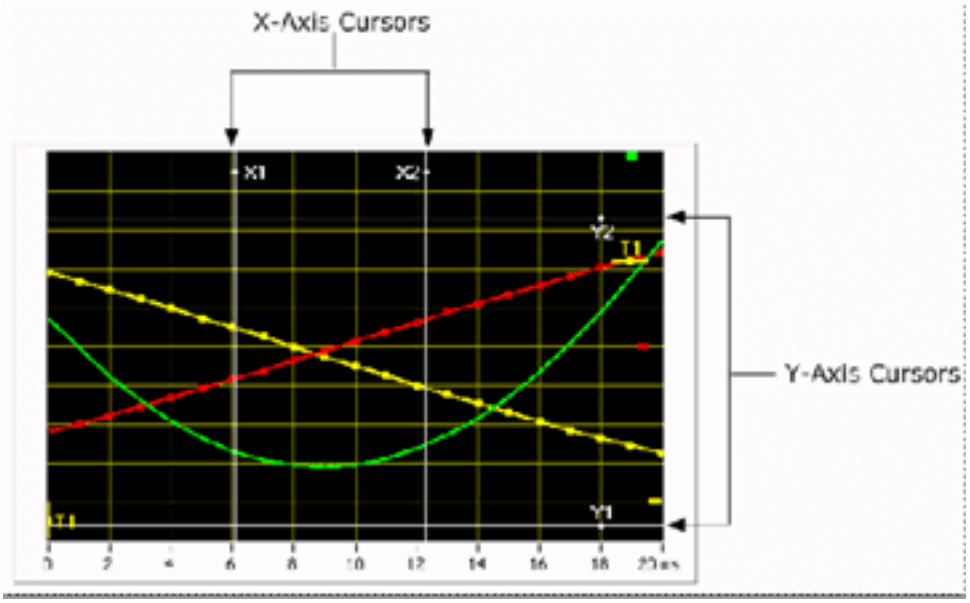

2. In the **Shift** field, enter the wanted adjusting of the scalel, or
Click **Autocenter** to center the signal in the Scope display.
3. You can apply it to all channels by selecting the **Apply to all channels** checkbox.

### *6.7.11 Cursors Tab*

The Cursors tab enables you to mark points on the Scope display where you are interested in seeing exactly what the values at the points are.

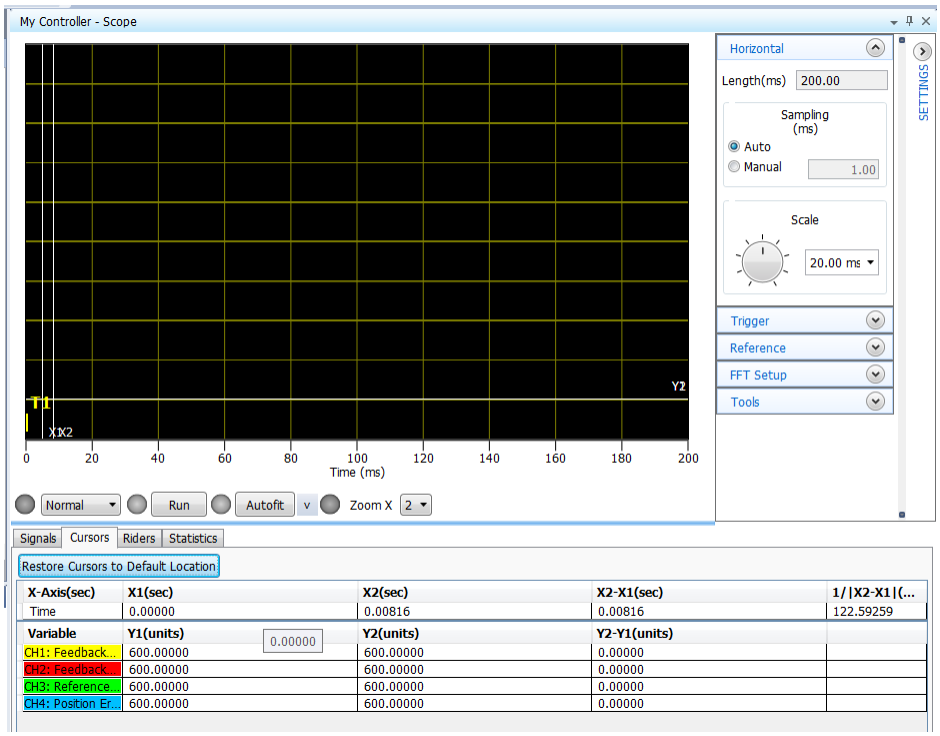
In addition, the cursors enable you to reduce the Scope display to a particular area of interest where you want to see the signal reactions within a given range.

There are four Cursors:




To set the Cursors:

1. Click Cursors. The Cursors tab is displayed and the Cursors are displayed on the Scope.




2. Set the X Cursors by placing the mouse cursor on them and dragging them left or right (depending on the cursor you are setting).
   As you move the X Cursors the values in the following change:

   > **X1(sec)** - the time on the X axis where the X1 Cursor is sitting.

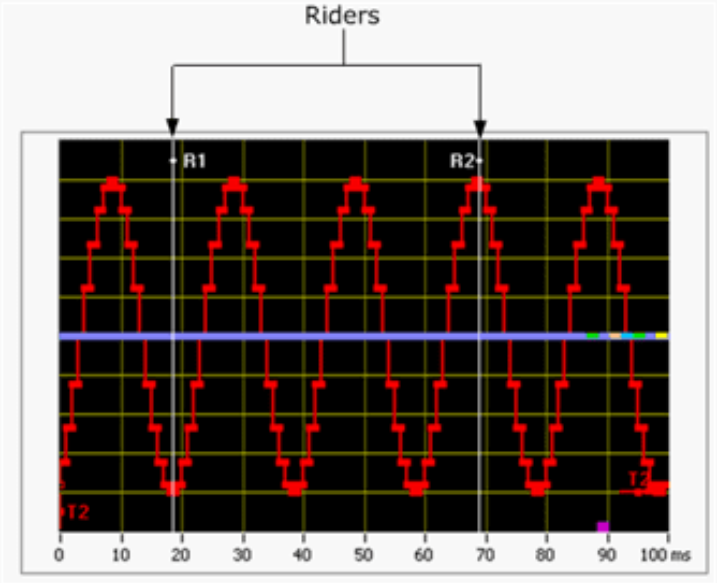
- **X2(sec)** - the time on the X axis where the X2 Cursor is sitting.
- **X2-X1(sec)** - the time between the X2 Cursor and the X1 Cursor.
- **1/(X2_X1)(Hz)** - the frequency of signal between the X2 Cursor and the X1 Cursor.

3. Set the Y Cursors by placing the mouse cursor on them and dragging them up or down (depending on the cursor you are setting).
   As you move the Y Cursors the values in the following change:>
   - **Y1(units)** - the amplitude of the signal at the time where the Y1 Cursor is sitting.
   - **Y2(units)** - the amplitude of the signal at the time where the Y2 Cursor is sitting.
   - **Y2-Y1(units)** - the difference in signal amplitude between the Y2 Cursor and the Y1 Cursor.
4. To view just the area of the Scope graph bounded by the Cursors select **Between Cursors** or **Permanent Between Cursors** from the **Autofit** dropdown list.
5. To reset the cursors to their original position, click Restore Cursors to Default Location.

### *6.7.12 Riders Tab*

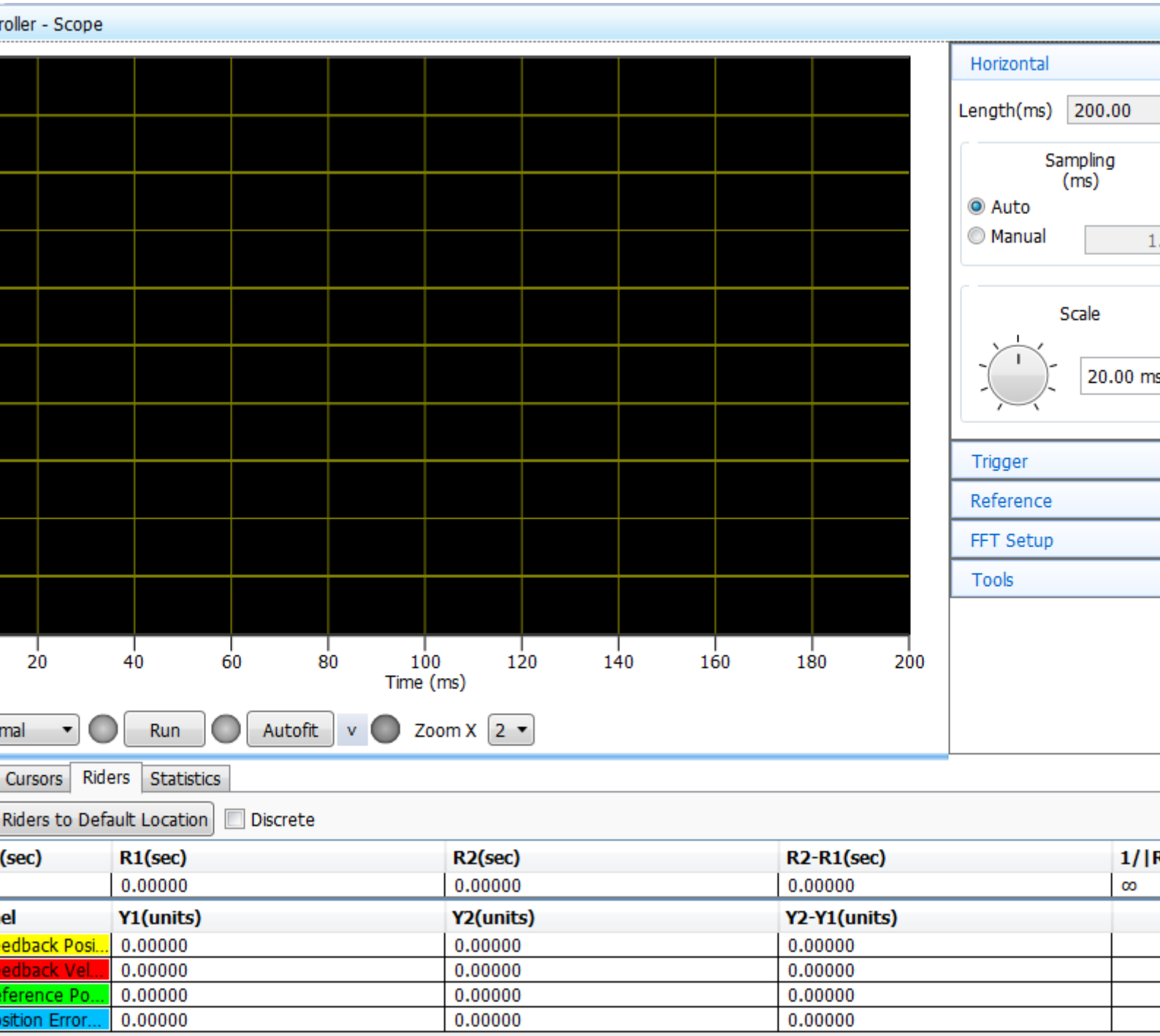
Riders are like Cursors (see Cursors Tab), but unlike Cursors they provide both the X and Y values at the same time.

There are two Riders:




To set the Riders:

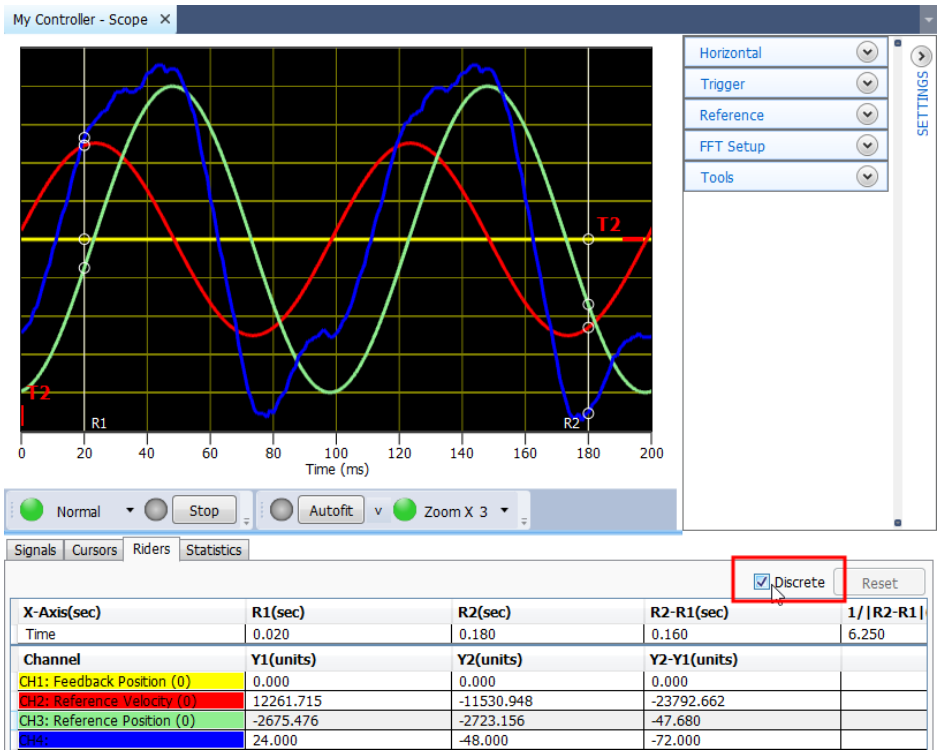
1. Click Riders. The Riders tab is displayed and the Riders are displayed on the Scope.




2. Set the Riders by placing the mouse cursor on them and dragging them left or right. As you move the Riders the values in the following change:
   - **R1(sec)** - the time on the X axis where the R1 Rider is sitting.
   - **R2(sec)** - the time on the X axis where the R2 Rider is sitting.
   - **R2-R1(sec)** - the time between the R2 Rider and the R1 Rider.

- **1/(R2_R1)(Hz)** - the frequency of signal between the R2 Rider and the R1 Rider.
- **Y1(units)** - the amplitude of the signal at the time where the R1 Rider is sitting.
- **Y2(units)** - the amplitude of the signal at the time where the R2 Rider is sitting.
- **Y2-Y1(units)** - the difference in signal amplitude between the R2 Rider and the R1 Rider.

3. To reset the cursors to their original position, click Restore Riders to Default Location.
4. Checking the Discrete checkbox causes TBD.



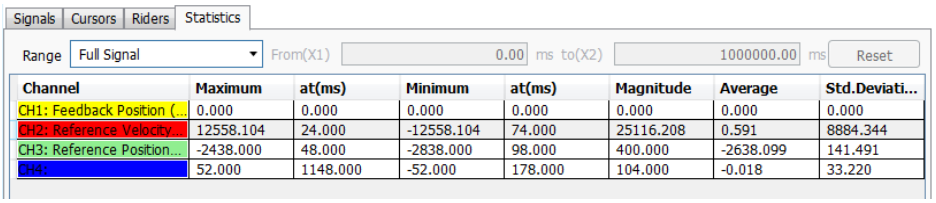

5. You can reset the values by clicking Reset **Reset**.

**Reset** is avaliable only when the Discrete mode is off.

### *6.7.13 Statistics Tab*

The Statistics tab provides you with various data connected with measurements of the signals displayed in the Scope.

Click Statistics to display the Statistics tab.




| Channel | Maximum | at(ms) | Minimum | at(ms) | Magnitude | Average | Std.Deviati... |
|---|---|---|---|---|---|---|---|
| CH1: Feedback Position (... | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| CH2: Reference Velocity... | 12558.104 | 24.000 | -12558.104 | 74.000 | 25116.208 | 0.591 | 8884.344 |
| CH3: Reference Position... | -2438.000 | 48.000 | -2838.000 | 98.000 | 400.000 | -2638.099 | 141.491 |
| CH4: | 52.000 | 1148.000 | -52.000 | 178.000 | 104.000 | -0.018 | 33.220 |

Using the **Range** dropdown list you can obtain statistics for:

- Full Screen - Displays the statistics for all signals that appear within the Scope boundaries.
- Full Signal - Displays the statistics for all signals whether they appear within the Scope or are beyond the Scope boundaries.
- Between Cursors - Displays the statistics for the signals within the boundaries of the X1 and X2 Cursors.

If you select **Between Cursors**, the X1 and X2 Cursors set using the appear. You can change the locations of the X1 and X2 Cursors using the **From(X1)** and **To(X2)** fields, or use the mouse to drag them to different positions.
To reset values to default click**Reset** .

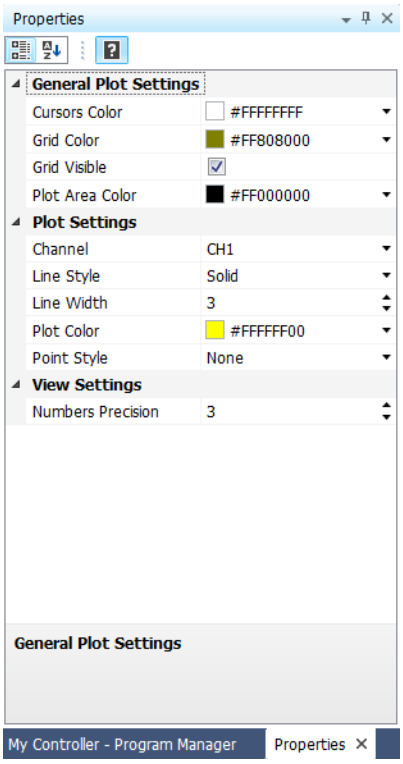
The signal statistics that are given are:

- Maximum - the maximum value a signal obtained, and the point on the X axis where it was obtained displayed in the at(ms) column.
- Minimum - the minimum value a signal obtained, and the point on the X axis where it was obtained displayed in the at(ms) column.
- Magnitude - the calculated maximum minus minimum signal values.
- Average - the average value, maximum plus minimum divided by 2.
- Std. Deviation - the standard deviation of the signal.

### *6.7.14 Setting Scope Plot Properties*

The Scope's plot properties, such as color, line width, line style, and the like, of each signal is set in the Scope **Properties** window.

If the **Properties** window is not displayed, press **Ctrl+Alt+P**, or select **Properties Window** from **View** in the Main Menu (or use the Workspace right-click option - see Working with Workspace Tree).




The Scope properties are divided into three groups:

### General Plot Settings

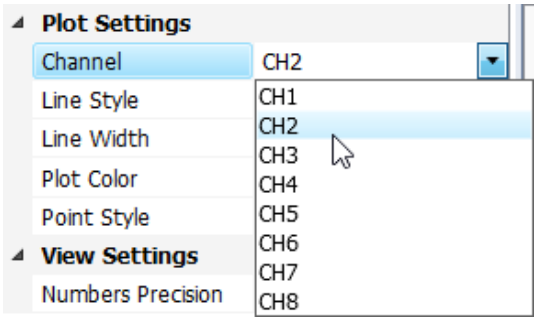
These properties relate to the Scope graphic display:

- Cursors Color - this sets the color of both the cursors and riders.
- Grid Color - this sets the color of the grid.
- Grid Visible - this checkbox toggles between displaying and not displaying the grid.
- Plot Area Color - this sets the color of the background of the graph.

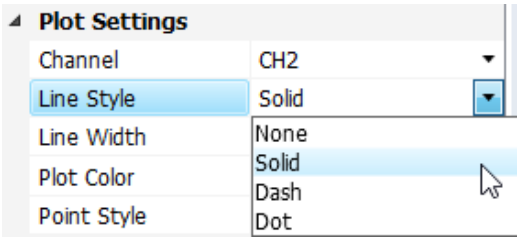
### Plot Settings

These properties relate to the plots of the signals:

- Channel - you use this dropdown list to select the signals to which the Plot Settings are to be applied:



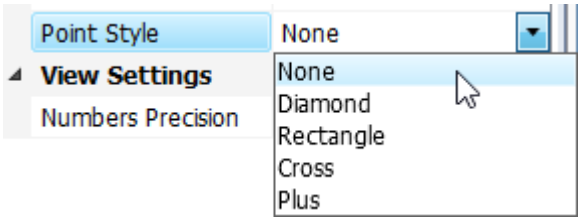

> Line Style - this sets the line style of the plot of the selected channel, selected from a dropdown list:




> Line Width - this sets the width (in points) of the plot of the selected channel.

> Plot Color - this sets the color of the plot of the selected channel which you select from a dropdown list (see Setting Color Properties).

> Point Style - you have the option of displaying the plot as a smooth curve, in which case there is no Point Style, or you can select the format that individual points on the plot are to be displayed from the dropdown list:




### View Settings

This property enables you to set the precision that the plot values are to be displayed.

## 6.8 Sin Cos Encoder Compensation and Monitoring

The **Sin Cos Encoder Compensation** tool enables you to:

> Select axis for encoder measurements.

> Display a Sin-Cos Lissajous curve graph of the wave form.

> Read a controller's gain, phase and offsets compensation parameters (**SCGAIN**, **SCPHASE**, **SCSOFFS**, and **SCCOFFS**).

> Write new compensation parameters to the controller.

> Examine the compensation effect of each parameter.

The **Sin Cos Encoder Compensation** can do the following:

> Perform step-by-step calculations of the compensation parameters for Sin-Cos encoders based on user input.

> Perform automatic calculations of the optimum compensation parameters set.

> Perform monitoring for the calculated compensation parameters set.

## 6.8.1 Activating Sin Cos Encoder Compensation and Monitoring

To activate the Sin Cos Encoder Compensation tool:

1. From the Toolbox click **Diagnostics and Monitoring** to display the Diagnostics and Monitoring tools list.
2. Click **Sin Cos Encoder Compensation**.

You may use **Add Component** from the right-click menu of the controller in the Workspace Tree (see Controller Right-Click Options ) to activate Sin Cos Encoder Compensation.

The **Sin Cos Encoder Compensation** window is displayed.

## 6.8.2 Sin Cos Encoder Compensation Window

The **Sin Cos Encoder Compensation** window consists of the following sections:

> Measurement Curve Display

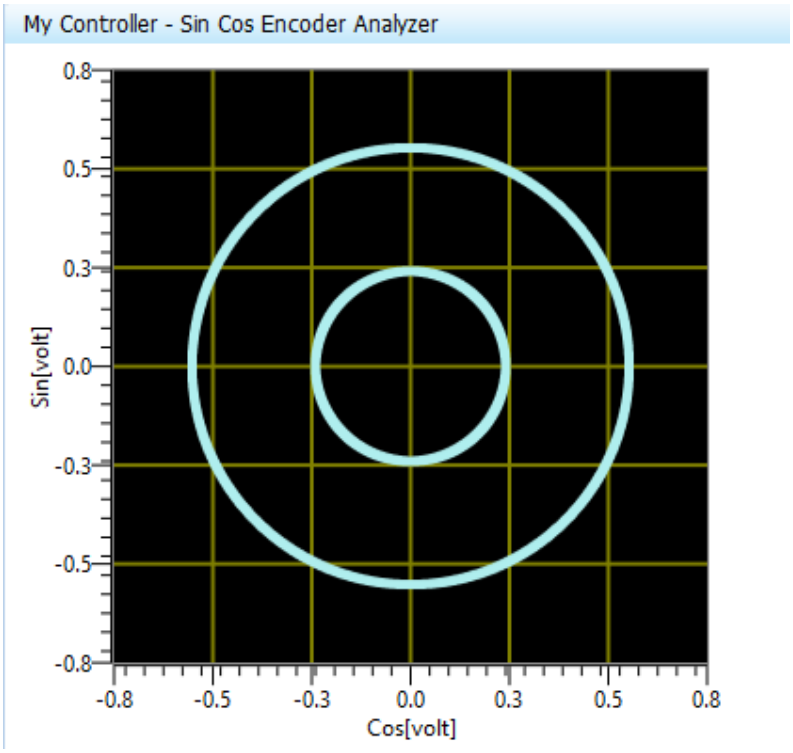
> Compensation Execution Panel

> Motion Operation Panel

> Parameters Panel

### 6.8.2.1 Measurement Curve Display

The Measurement Curve is a Lissajous plot, an example of which is shown below, which displays the curve of the results of the measurements performed by **Sin Cos Encoder Compensation**.

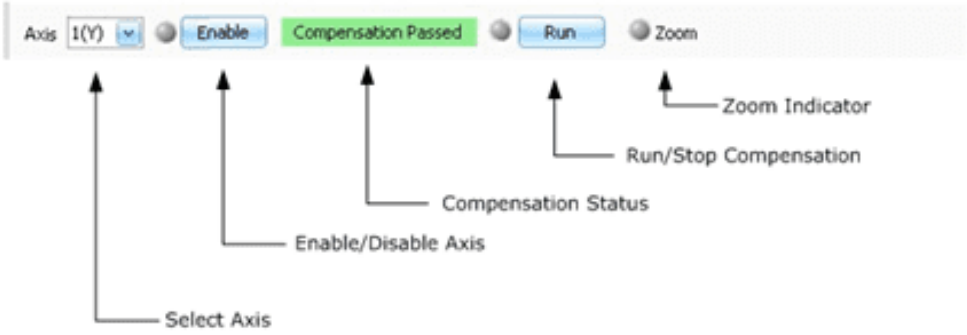



The Lissajous plot displays the aspect ratio of the phase shift between the input and output, with an aspect ratio of 1 (perfect circle) corresponding to a phase shift of ±90° as well as amplitude differences (ellipse).

As with the SPiiPlus MMI Application Studio , you can set cursors on the display through the Cursors tab of the Parameter Selection panel (see Setting Cursors).

### *6.8.2.2 Compensation Execution Panel*

The Sin Cos Encoder Compensation Execution panel serves for executing the **Sin Cos Encoder Compensation** calculation routines:




The elements in this panel are:

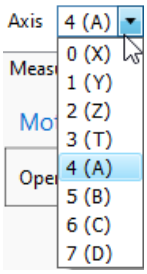
- **Select Axis** - used for selecting the axis from a dropdown list, the contents of which depend on the type of controller. There is a LED associated with the selected axis that can be:

- Grey - the axis is not enabled
- Green - the axis is enabled.

If the axis is not enabled, you enable it by clicking the **Enable** button.

If you select an axis that is not connected to a Sin-Cos Encoder, it cannot be enabled.

Select the required axis from the dropdown list, for example:




- **Compensation Status** - this field displays messages associated with the **Sin Cos Encoder Compensation** status. The messages that can be displayed are:
  - **Compensation Off** - the Sin Cos Encoder Compensation has not been activated. (You activate it by clicking the **Run** button).
  - **Compensation in Progress** - the Sin Cos Encoder Compensation is performing its measurements.
  - **Compensation Passed** - the Sin Cos Encoder Compensation has completed its measurements and the compensation is correct.
  - **Compensation Failed** - the Sin Cos Encoder Compensation has completed its measurements and the compensation is not correct.
- **Compensation Cancelled** - the Sin Cos Encoder Compensation has been stopped (when you click the **Stop** button) before the measurements have been completed.
  There is a LED associated with the status which can be:
  - Grey - the Sin Cos Encoder Compensation has not been activated, or has completed its measurements.
  - Green/blinking - the Sin Cos Encoder Compensation has been activated and is working.
- **Run/Stop** button - activates or halts the Sin Cos Encoder Compensation measurements, when clicked, the **Run** button changes to the **Stop** button for manually halting the measurements in progress.
- **ZoomIndicator** - you have the following Zoom options:

When the display has been zoomed, the Zoom LED ( ) **turns green.**

- Zoom In - Hold Shift key and place the mouse cursor in the Scope display in area you want to zoom into. Click the Left mouse button to display the Zoom icon. When the icon appears, click the Left mouse button. Without releasing the Shift key, each time you click the Left mouse button, you increase the zoomed display.
- Zoom In Area - While holding the Shift key, press the Left mouse button and drag to mark the area you want to zoom, then release the mouse button.
- Zoom Out - While holding the Shift key, place the mouse cursor in the zoomed area and click the Right mouse button. Without releasing the Shift key, each time you click the Right mouse button, you zoom out one level.
- Move Graph - Hold Ctrl key and Left mouse button - a hand is displayed. Drag the graph to where you want.
- Undo - To undo all zooms, press Shift + Backspace keys. (You can also undo the zoom by clicking the Zoom LED  ).

#### *6.8.2.3 Motion Operation Panel*

You use the Motion Operation panel to select the operation type for the Sin Cos Encoder: compensation or monitoring.

#### *6.8.2.4 Parameters Panel*

You use the Parameter panel to set up parameters and display options you want. The panel has three tabs:

- Measurement Tab
- Cursors Tab
- Statistics Tab

##### *6.8.2.4.1 Measurement Tab*

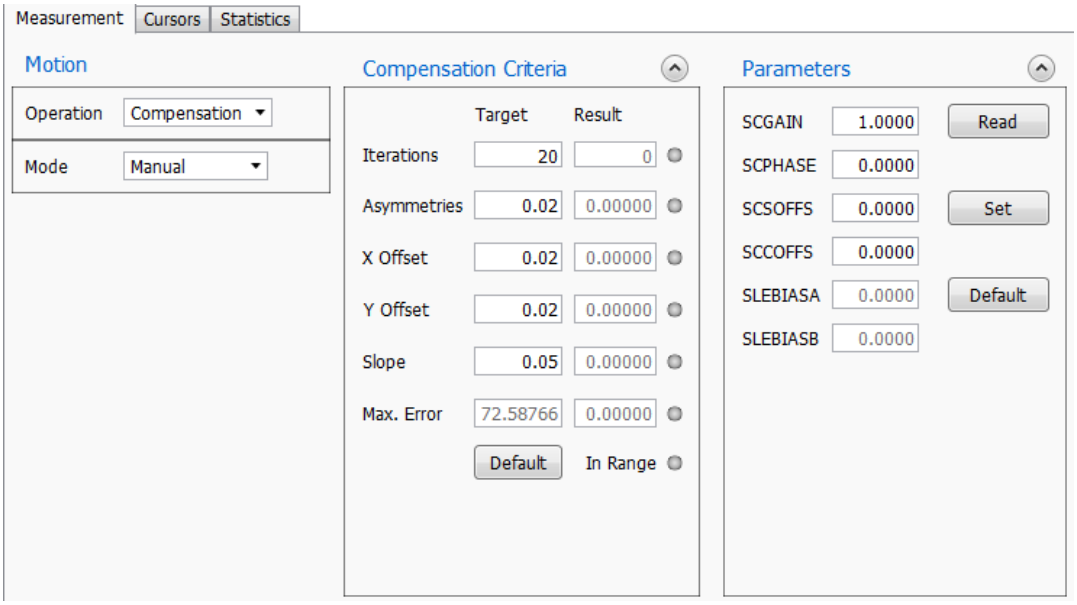
The Measurement tab is used for defining the required input measurement parameters and to control the output results.

You can change the Target values in Monitoring and Compensation modes.

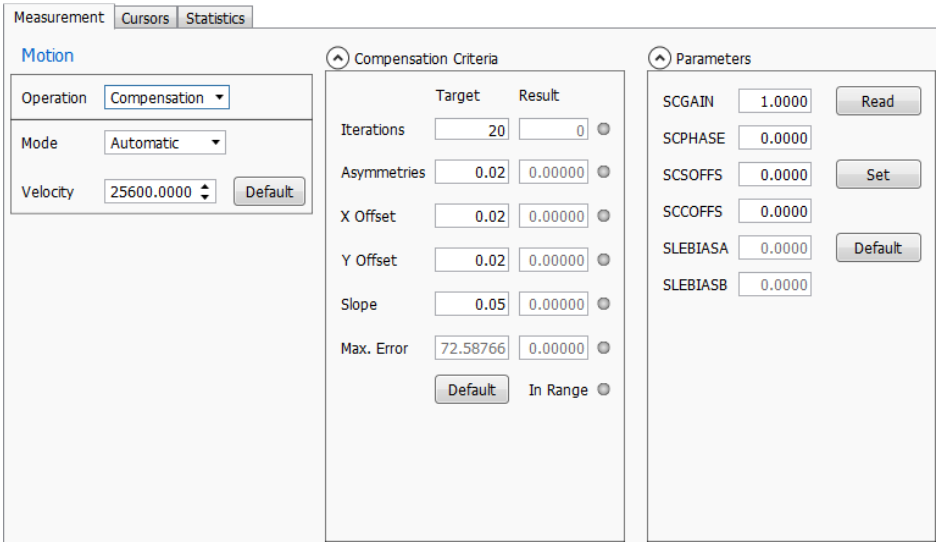
Once you are in compensation mode, pressing the run button will start the compensation algorithm. Based on the value in Mode (Automatic or Manual) - motion will be performed either by MMI commands or the user phsically moving the motor.

If Automatic mode is selected then the user may change the velocity of the stage commanded by the MMI from its default value, as determined by the algorithm.

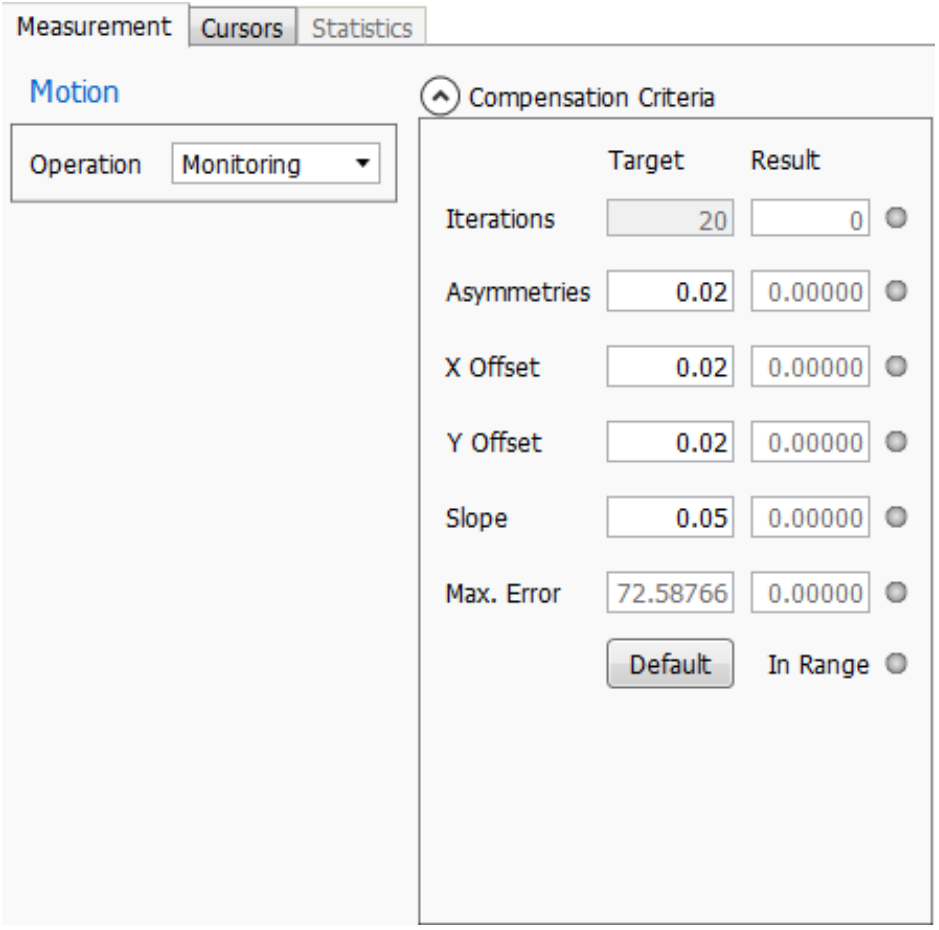
If Manual mode is selected the user may either move the stage by hand or write a program that will execute a motion.




Alternatively, the user can choose automatic mode, and set the velocity, as illustrated below.

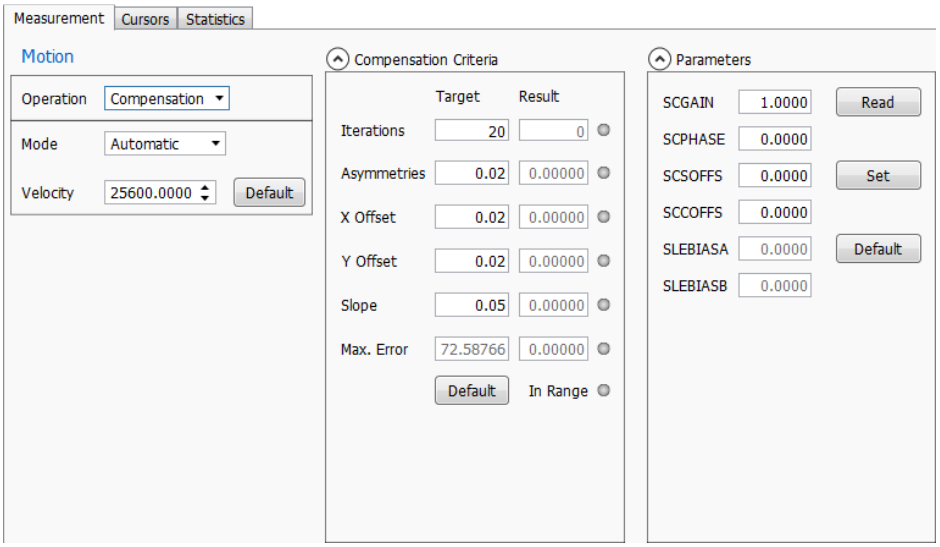


Under monitoring operation a limited set of criteria can be set.




### Motion

When Automatic mode is selected the velocity field appears and can be set.



### Compensation Criteria

This section displays the target criteria used by the **Sin Cos Encoder Compensation** when making the calculations and current values of calculated parameters along with a set of LEDs. If the current value of any parameter is equal to or less than the target value, the associated LED is Green (otherwise it is Red if the value fails, or Grey if the parameter is not relevant).

The criteria are:

- Iterations - the number of iterations that the **Sin Cos Encoder Compensation** is to perform during the measurements.
- Asymmetries - Max(dx,dy) - Min(dx,dy), where dx is $X_{\text{final point}} - X_{\text{start point}}$, and dy is $Y_{\text{final point}} - Y_{\text{start point}}$. If this is zero, the graph is a perfect circle.
- Shift X - the offset on the X axis
- Shift Y - the offset on the Y axis
- Slope - dy/dx, where dx is $X_{\text{final point}} - X_{\text{start point}}$, and dy is $Y_{\text{final point}} - Y_{\text{start point}}$.

You can make changes to the values of the **Target** fields when fine-tuning the compensation.

Clicking **Default** sets default values of the target calculated parameters.

### Parameters

This section lists the ACSPL+ variables that are connected with managing Sin-Cos encoders. These are:

- **SCGAIN** - Modify the amplitude of the COS signal, to compensate amplitude difference between the SIN/COS signals
- **SCPHASE** - Modify the phase of the COS signal, to compensate phase difference between the SIN/COS signals (phase should be 90 degrees)
- **SCSOFFS** - Compensating SIN signal offset
- **SCCOFFS** - Compensating COS signal offset
- **SLEBIASA**
- **SLEBIASB**

**SCGAIN, SCPHASE**, **SCSOFFS**, and **SCCOFFS** are all software-based variables.

The **SLEBIASA** and **SLEBIASB** variables, on the other hand, are hardware-based. Hardware compensation has some advantages over software compensation: possibility to get analog signals out of saturation, and making hardware based features like PEG more accurate.
In principle these variables have similar functionality as **SCSOFFS** and **SCCOFFS**; however, they are only supported by the following controller models:

- SPiiPlusNT-HP/LD (compensation diapason ±350 mV)
- SPiiPlusDC-HP/LD (compensation diapason ±350 mV)
- SPiiPlus-UDMpc (compensation diapason ±625 mV)
- PiiPlus-CMnt-2 (compensation diapason ±750 mV)

If hardware compensation is available then after the calculation of hardware compensation another round of compensation called verification will calculate more accurate software compensation as well.

You have the option of changing the values by inputting new values in the appropriate fields. Details of these variables are provided in the *SPiiPlus Command & Variable Reference Guide*.

There are three action buttons associated with this section:

| | |
|---|---|
| Read | **Sin Cos Encoder Compensation** reads the current variable values from the controller. |
| Set | **Sin Cos Encoder Compensation** writes the variable values into the controller. |
| Default | **Sin Cos Encoder Compensation** sets the default values into the controller. |

*6.8.2.4.2 Cursors Tab*

The Cursors tab enables you to mark points on the display where you are interested in seeing exactly what the values at the points are.

There are two cursors, **X1** and **X2**, for delimiting the Cos voltage values, and two cursors, **Y1** and **Y2**, for delimiting the Sin voltage. As the **Sin Cos Encoder Compensation** performs its routines, the values at the cursor settings are displayed in the field below the graph. For setting the cursors see Setting Cursors.

*6.8.2.4.3 Statistics Tab*

For each run, the **Sin Cos Encoder Compensation** builds a statistics table consisting of compensation calculation measurements in the Statistics tab:

The rows are color-coded, Green for those values that passed the compensation measurements, and Red for those that failed.

At the bottom of the table the **Sin Cos Encoder Compensation** adds a line containing the average values of the variables. There is a separate line for each axis that was checked.

Right-clicking on any row displays a menu with the following options:

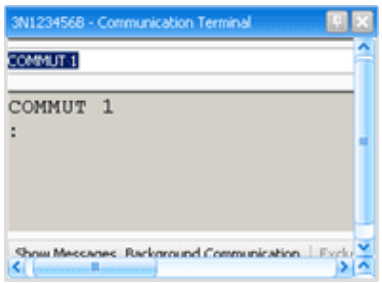
- **Set Parameters** - selecting this option causes the **Sin Cos Encoder Compensation** to use these variable values for purposes of making the measurement.
- **Save to Flash** - selecting this option causes **Sin Cos Encoder Compensation** to save the values to the controller's flash memory.
- **Delete** - selecting this option deletes the row from the Statistics.
- **Clear All** - selecting this option removes all the values from the Statistics.

### 6.8.3 Running Sin Cos Encoder Compensation

To perform automatic encoder compensation:

1. From the Motion Operation panel, select **Compensation**.
2. From the Execution pane select axis from the **Axis** dropdown list.
3. Click **Enable**.
4. Click **Run**.

If the motor associated with the encoder is a brushless motor, it has to be commutated before encoder compensation. In the event that it has not been commutated, the following is displayed:
The easiest way to commutate the motor at this point is to enter the **COMMUT** command via the **Communication Terminal**, for example:




Then click **Run** again.

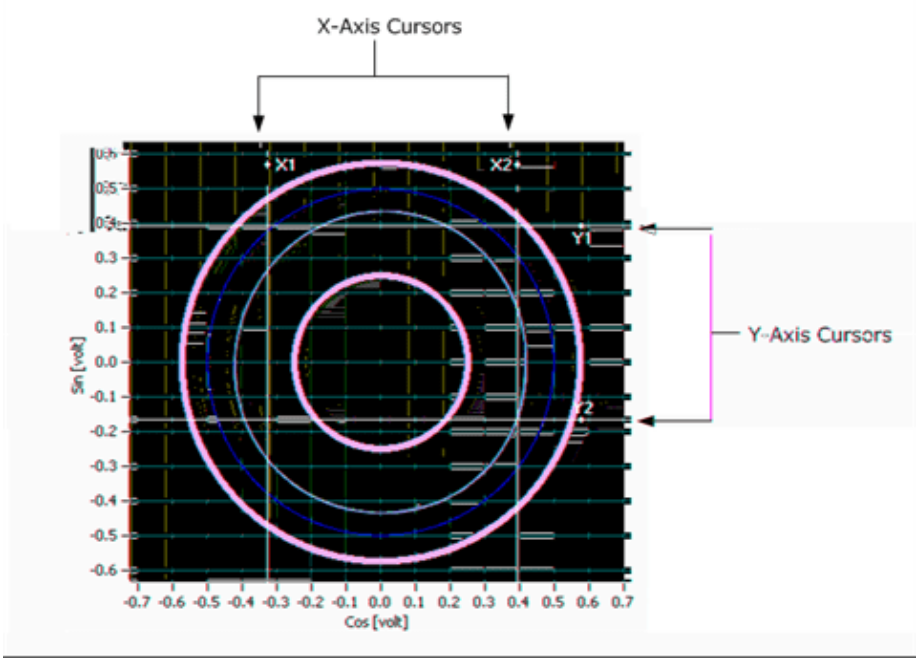
If the compensation is successful, go to the Measurement Tab of the Parameter Selection panel and save the values to the controller's flash memory by clicking **Set** in Measurement Tab.

### 6.8.4 Setting Cursors

You can use Cursors to mark points on the display where you are interested in seeing exactly what the values at the points are.

In addition, the cursors enable you to reduce the display to a particular area of interest where you want to see the signal reactions within a given range.

There are four Cursors:



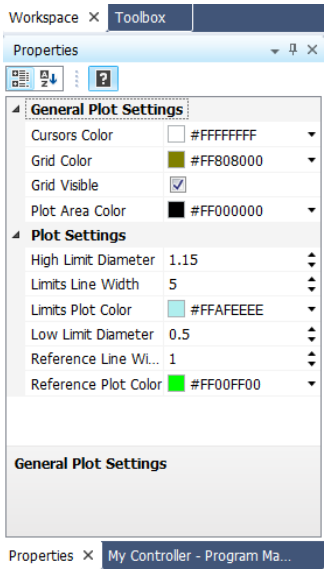

To set the Cursors:

1. Click the Cursors tab of the Parameter Selection panel. The Cursors are displayed on the graph.
2. Set the X Cursors by placing the mouse cursor on them and dragging them left or right (depending on the cursor you are setting).
3. As you move the X Cursors the values in the following change:
   - **X1(volt)** - the cosine voltage on the X axis where the X1 Cursor is sitting.
   - **X2(volt)** - the cosine voltage on the X axis where the X2 Cursor is sitting.
   - **X2-X1(volt)** - the difference of cosine voltages between the X2 Cursor and the X1 Cursor.
4. Set the Y Cursors by placing the mouse cursor on them and dragging them up or down (depending on the cursor you are setting).
5. As you move the Y Cursors the values in the following change:
   - **Y1(volt)** - the sine voltage on the Y axis where the Y1 Cursor is sitting.
   - **Y2(volt)** - the sine voltage on the Y axis where the Y2 Cursor is sitting.
   - **Y2-Y1(volt)** - the difference of sine voltages between the Y2 Cursor and the Y1 Cursor.

### *6.8.5 Setting Sin Cos Encoder Compensation Properties*

The Lissajous Curve properties, such as line color, width, style, and the like, of each graph is set in the Sin Cos Encoder Compensation **Properties** window.

If the **Properties** window is not displayed, press **Ctrl+Alt+P**, or select **Properties Window** from **View** in the Main Menu (or use the Workspace right-click option - see Working with Workspace Tree).




The Sin Cos Encoder Compensation properties are divided into two sections:

### General_Plot_Settings

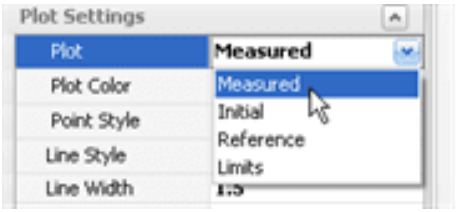
This section enables you to set the background properties of the graph:

- Cursors Color - for setting the color of the cursors.
- Grid Color - for setting the color of the background grid.
- Grid Visible - a checkbox for toggling between the display (default)/non-display of the grid.
- Plot Area Color - for setting the color of the background of the graph.

### Plot Settings

This section enables you to set the properties of the graph:

- Plot - there are four possible plots: Measure, Initial, Reference, Limits, each with its own set of properties. You select the plot for which you want to set the properties from the dropdown list:




- Once you have select the plot, you can set:
  - Plot Color - for setting the color of the selected plot.
  - Point Style - for setting the style of the points used in displaying the graph of the selected plot.
  - Line Style - for setting the style of the line used in displaying the graph of the selected plot.

- Line Width - for setting the thickness of line used in displaying the graph of the selected plot.
- High Limit Diameter - for setting the diameter of the outside limit (maximum) that is displayed.
- Low Limit Diameter - for setting the diameter of the inside limit (minimum) that is displayed.

### 6.8.6 Running Sin Cos Encoder Monitoring

To perform encoder compensation monitoring:

1. From the Motion Operation panel, select **Monitoring**.
2. From the Execution pane select axis from the **Axis** dropdown list.
3. Click **Enable**.
4. Click **Run**.

## 6.9 Variables Manager and Watch

**Variables Manager and Watch** is used for viewing the values of ACSPL+ standard variables, user global and local variables, and SP variables.

It is useful during debugging sessions or when the value of a variable needs to be observed in real-time.

**Variables Manager and Watch** provides two types of display:

- Quick Viewer for viewing the current values of selected variables.
- Watch for viewing the values of selected variables as they change in real-time.

### 6.9.1 Activating Variables Manager and Watch

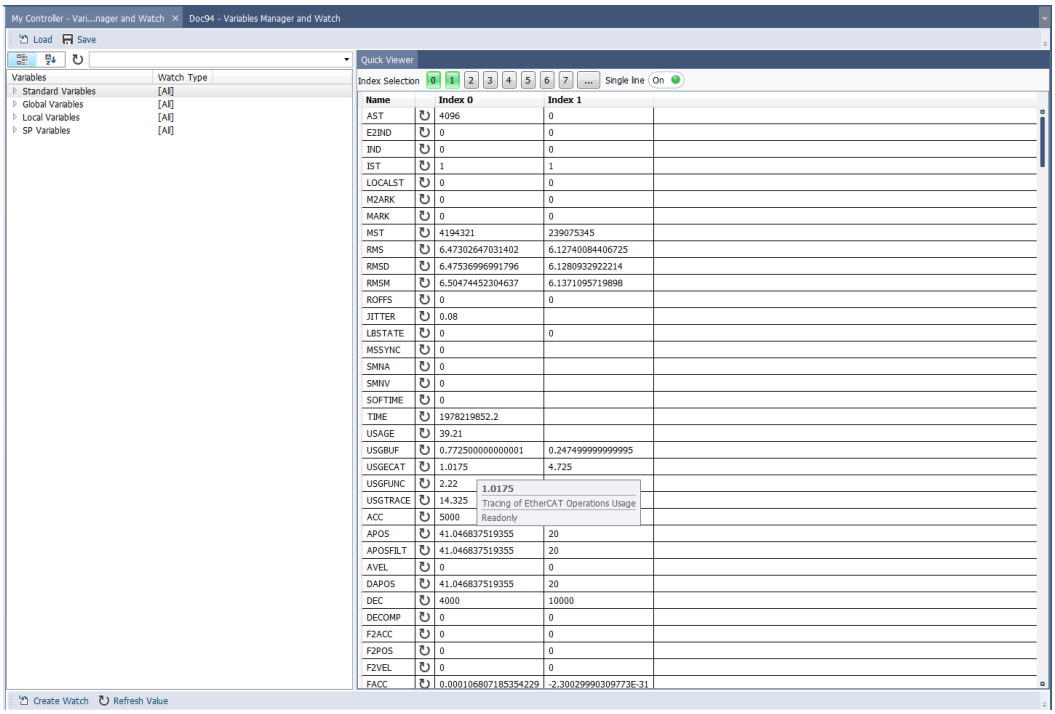
To activate **Variables Manager and Watch**:

1. From the Toolbox click **Diagnostics and Monitoring** ( 

) to display the Diagnostics and Monitoring tools list.
2. Click **Variables Manager and Watch**.

You may use **Add Component** from the right-click menu of the controller in the Workspace Tree (see Controller Right-Click Options ) to activate the **Variables Manager and Watch**.

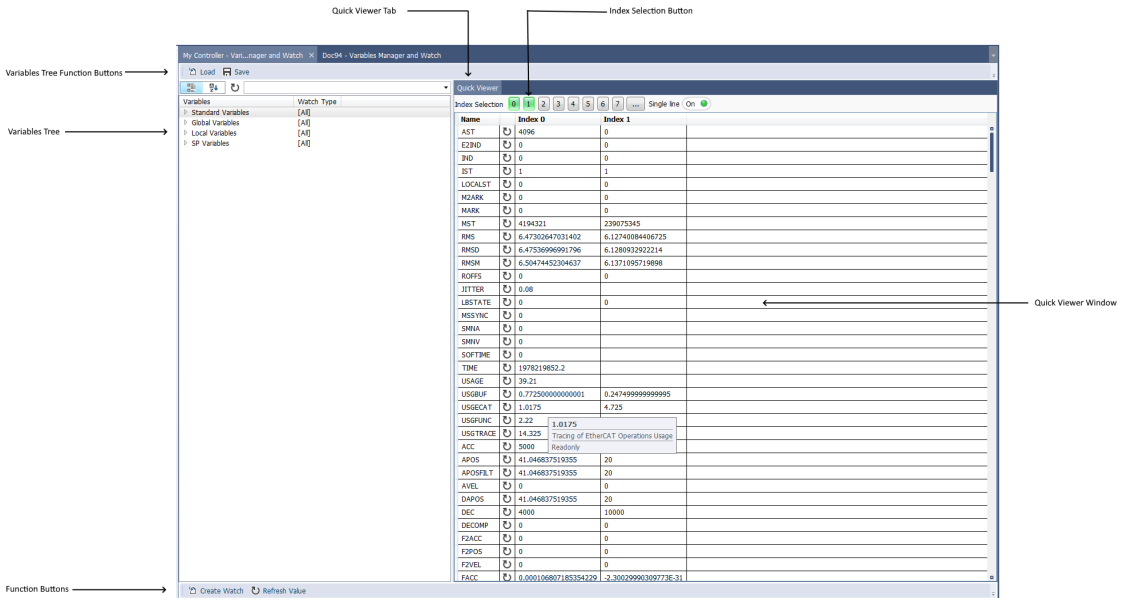
The **Variables Manager and Watch** main window is displayed.




When you first activate **Variables Manager and Watch**, it opens in the Quick View mode. As you add watches, these are displayed in their own tabs.

## *6.9.2 Quick Viewer*

The **Quick Viewer** is used to view the current values of variables selected from the Variables Tree.



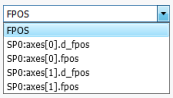

Quick Viewer consists of:

- The Variables Tree for selecting the variable for display
- Index Selector for selecting the index of the variable
- Quick Viewer window displaying the values of the selected variable

### *6.9.2.1 The Variables Tree*

The Variables Tree, which appears in both the **Quick Viewer Mode** and the **Watch Mode**, is used to select the variable for viewing. You can select a group of variables, a single variable, an array element, or a bit of an integer.

The following function buttons are associated with the Variables Tree:

| | |
|---|---|
|  | **Group** - displays the variables according to their groups. |
|  | **Sort** - displays the variables in alphabetic order. |
|  | **Refresh** - refreshes the Variable Selector list. |
|  | **Search** - Entering the name of the required variable in the field (or a partial name) displays the variables starting with the one you entered, for example:   |

### *6.9.2.2 Index Selector*

You use the Index Selector to display the value of the variable for the selected Index. Each Index is associated with an axis:

| | |
|---|---|
|  | 0 |
|  | 1 |
|  | 2 |
|  | 3 |
|  | 4 |
|  | 5 |

| | |
|---|---|
| 6 | 6 |
| 7 | 7 |

If no Index Selection button has been selected, then all of the variable values are displayed in one column, labeled **Value**.

When you click an Index button, the value for that Index replaces whatever was previously displayed.

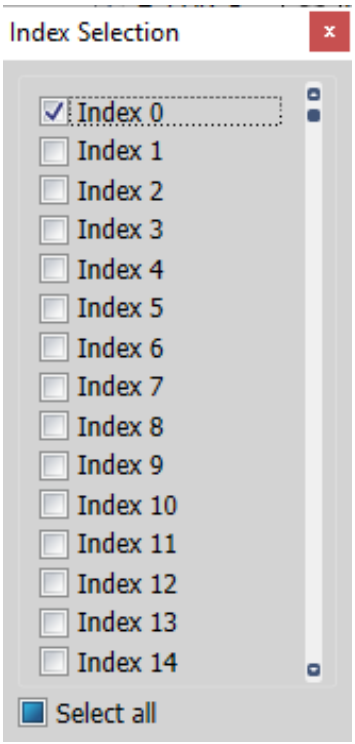
To cancel the display click the button again.

The Index Selector has an additional button: , which enables you to select multiple axes for display.

### *6.9.2.3 Selecting Multiple Axes*

To select multiple axes:

1. Click , the Index Selection dialog is displayed:

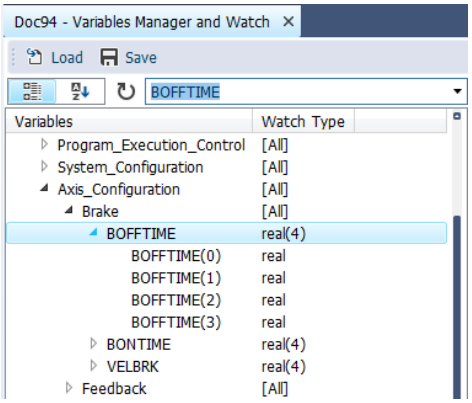



2. Select the required axes by clicking the checkbox associated with the Index (or select **Select All** for all of the axes) and close the dialog window.

To cancel the display of a particular Index click and deselect the Index checkbox.

#### *6.9.2.4 Selecting Variables*

You use the Variables Tree, which appears in both the **Quick Viewer Mode** and the **Watch Mode**, to select the variable. You can select a group of variables, a single variable, an array element, or a bit of an integer.




The Variables Tree consists of the following columns:

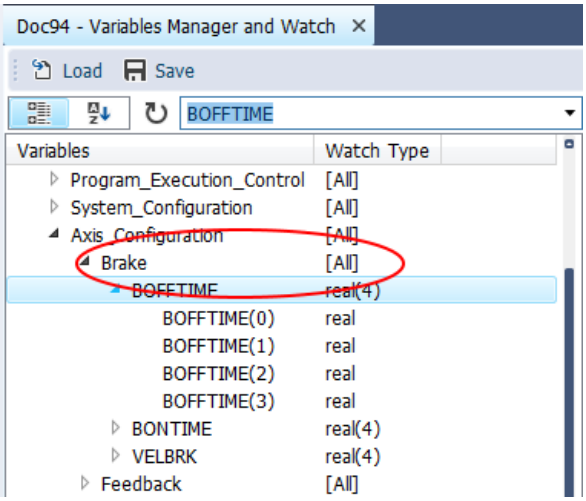
- **Variable** - containing a tree of groups, variables, array element and bits (for integers)
- **Watch Type** - containing the variable type

Using the Variables Selector you can:

- Select a Group
- Select a Single Variable
- Select an Array Element
- Select a Bit of an Integer

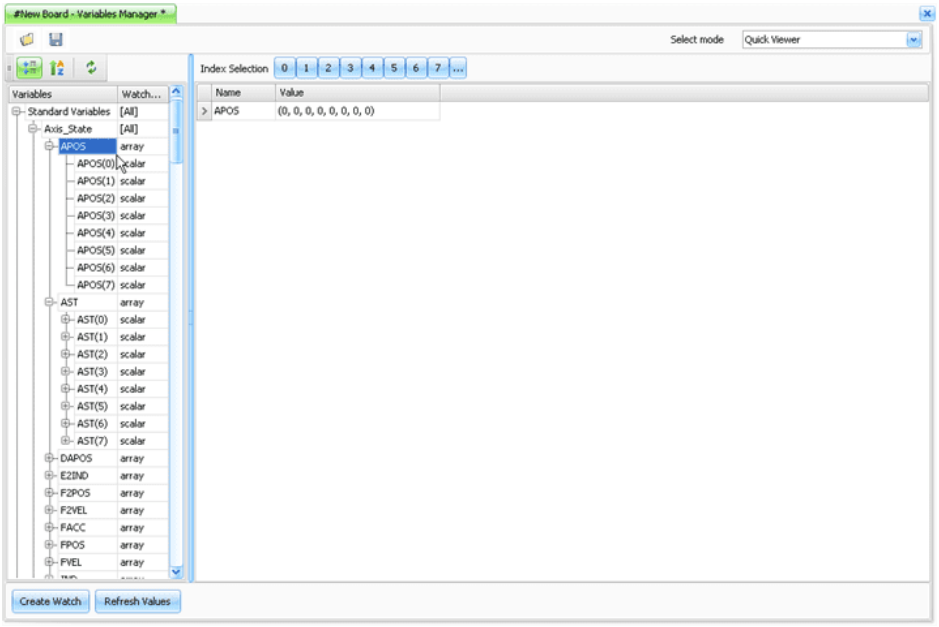
##### *6.9.2.4.1 Select a Group*

To select a group click the variable name with **[All]** in the Variables Selector Watch Type column:



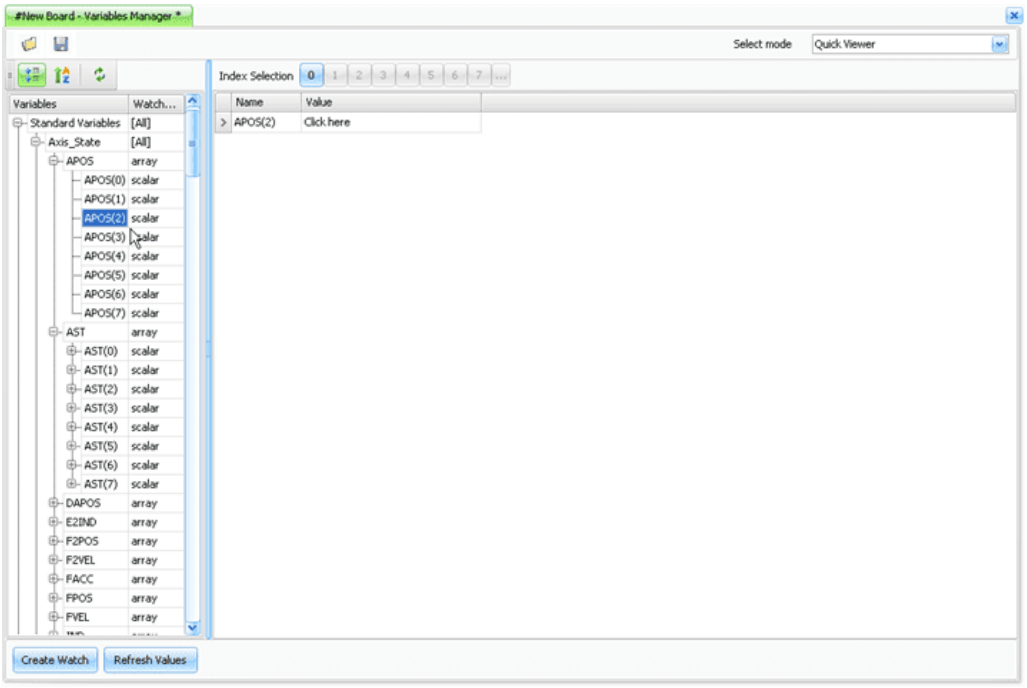

*6.9.2.4.2 Select a Single Variable*

To select a single variable click its corresponding cell in Variables Selector.




*6.9.2.4.3 Select an Array Element*

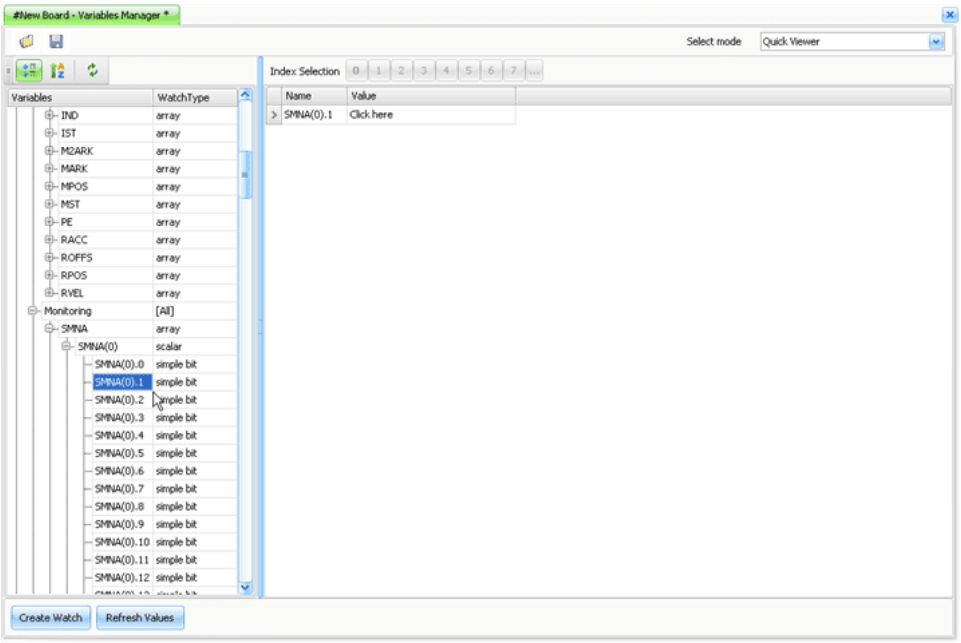
All arrays have Watch Type = array. To expand the list of elements, click  . After you have expanded the list, click any of its elements.




### 6.9.2.4.4 Select a Bit of an Integer

If a single variable or an array element is an integer, then to the right of its name you see  . Clicking this expands the integer to reveal the bits.

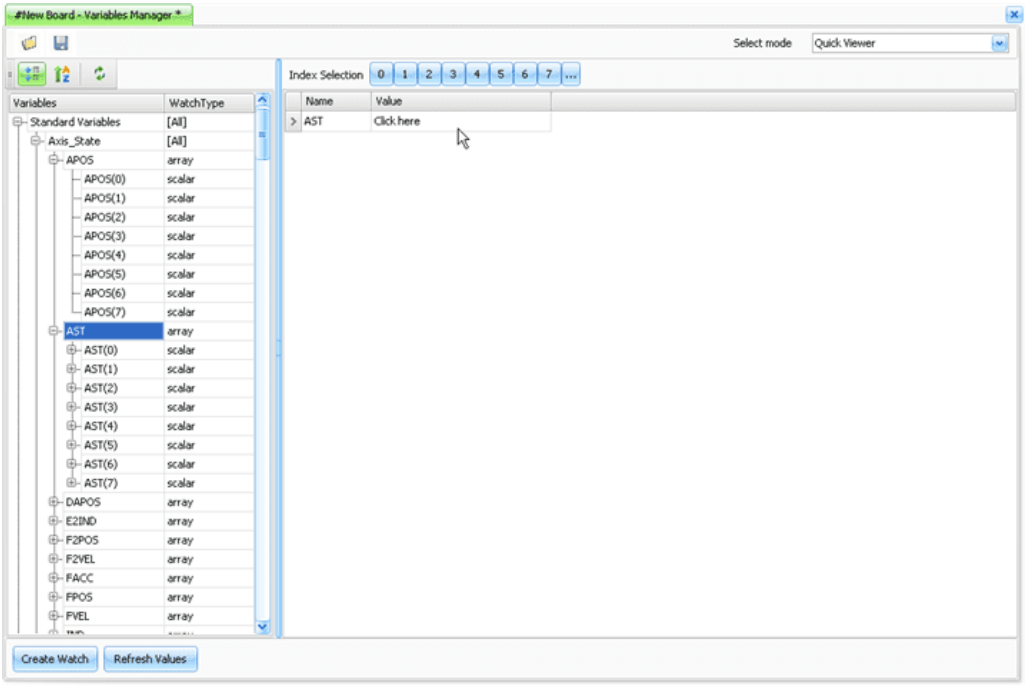
After you have expanded the list of bits, click of them.




### 6.9.2.5 Displaying a Variable's Value

In order to display the current value of a variable

1. Select the variable in the Variables Selector panel.
2. In the Quick Viewer window click its cell.

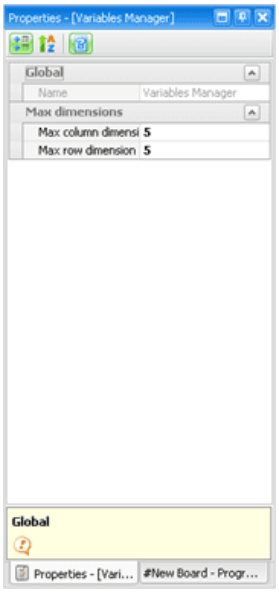



### *6.9.2.6 Update Variable Values*

You should be aware that the Quick Viewer does not maintain a running display of the current values of the variables. From time to time you have to update the values. To do this click **Refresh Values**

### *6.9.2.7 Quick Viewer Properties*

The Quick Viewer properties are:




- **Max column dimension** - width of column.
- **Max row dimension** - height of row.

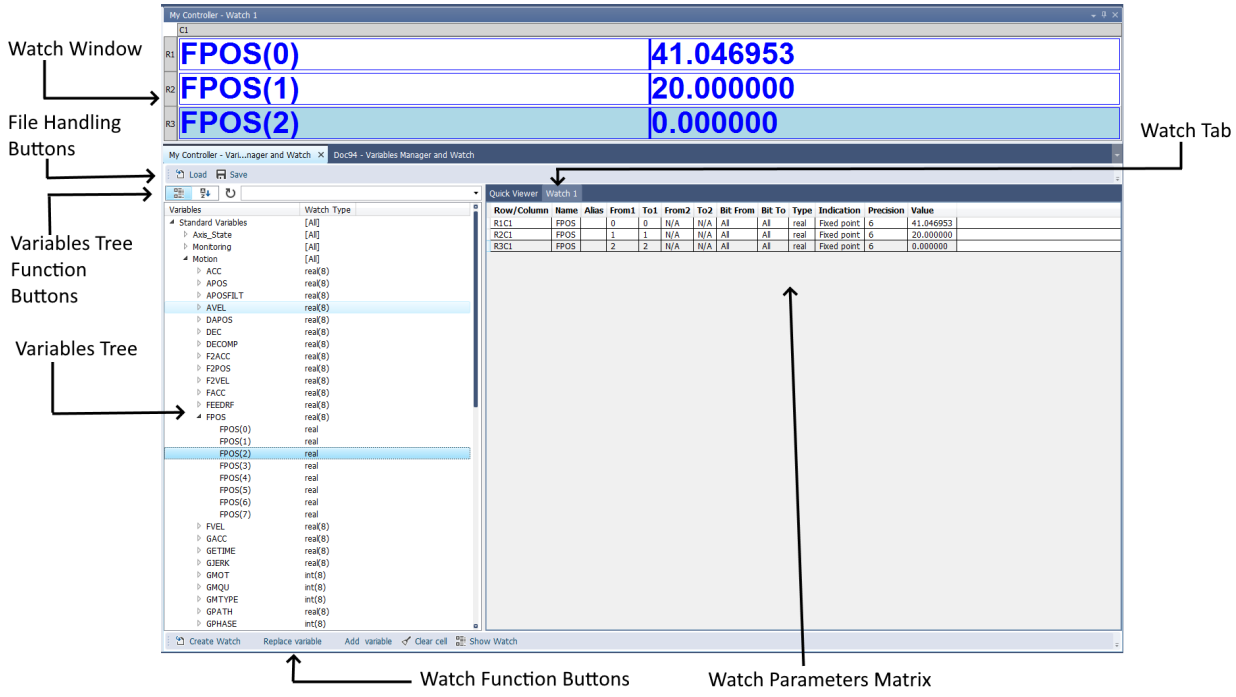
## 6.9.3 Watch

The Watch, unlike the Quick Viewer, is used for real-time monitoring of selected variables.

### 6.9.3.1 Creating a Watch

To create a Watch:

1. Select the variable from the Variables Tree (see Components).
2. Click **Create Watch** ( Create Watch ).
   The Watch is displayed in a new tab.




### *6.9.3.2 Watch Display*

The Watch display has the following elements:

- File Handling Functions
- Watch Parameters Matrix
- Modifying Watch Parameter Matrix Fields

#### *6.9.3.2.1 File Handling Functions*

The file handling functions enable you to save and retrieve Watch data.

| | |
|---|---|
|  | **Save Configuration** - saves the Watch data to a file. |
|  | **Load Configuration** - loads a previously saved Watch data file. |

#### *6.9.3.2.2 Watch Parameters Matrix*

The Watch Parameters Matrix shows the parameters of each cell displayed in the Watch window. The Matrix has the following columns:

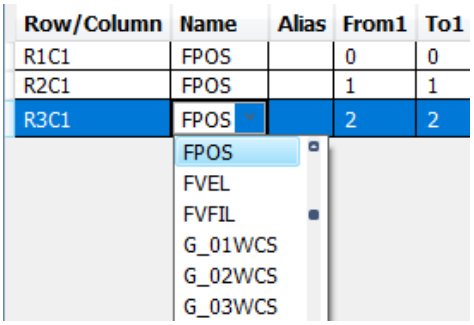
- Name - The name of the variable
- Alias - Enables you to enter a more meaningful name of the variable
- From1 - Starting index of an array

- To1 - Ending index of an array
- From2 - Starting index of a 2-dimensional array
- To2 - Ending index of a 2-dimensional array
- Bit - Bit number of the variable being monitored (or **All** if all the bits are being monitored)
- Type - The type of the variable (int, real, etc.)
- Indication - Numerical representation of the value (**Decimal**, **Hexadecimal**, **Binary**, **Fixed Point**, etc.)
- Precision - For Real numbers this indicates the number of decimal points displayed
- Value - The detected value of the variable being monitored

*6.9.3.2.3 Modifying Watch Parameter Matrix Fields*

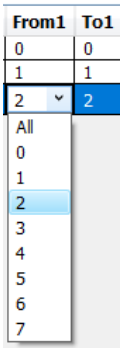
The Watch Parameter Matrix enables you to modify what is displayed in the Watch window. Depending on the variable, you can select what is displayed by clicking the field in the matrix and click the down arrow ( ) to select from the dropdown list.

For example, you may want to see a different variable momentarily. You click the **Name** field and select the required variable:

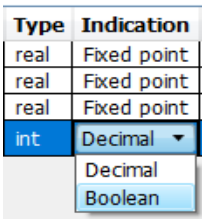



This variable and its values are now displayed in the Watch window. To return to the original variable, you repeat this process and select the original variable.

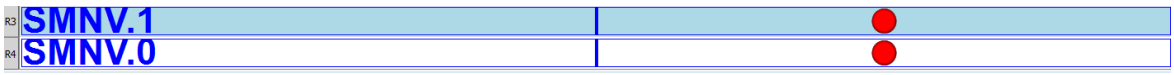
Another modification you can make is change the **To** or **From** indices of the variable by:




If the variable that's being watched is a bit, you can select whether to show its numeric value, or as a Boolean display (True/False).



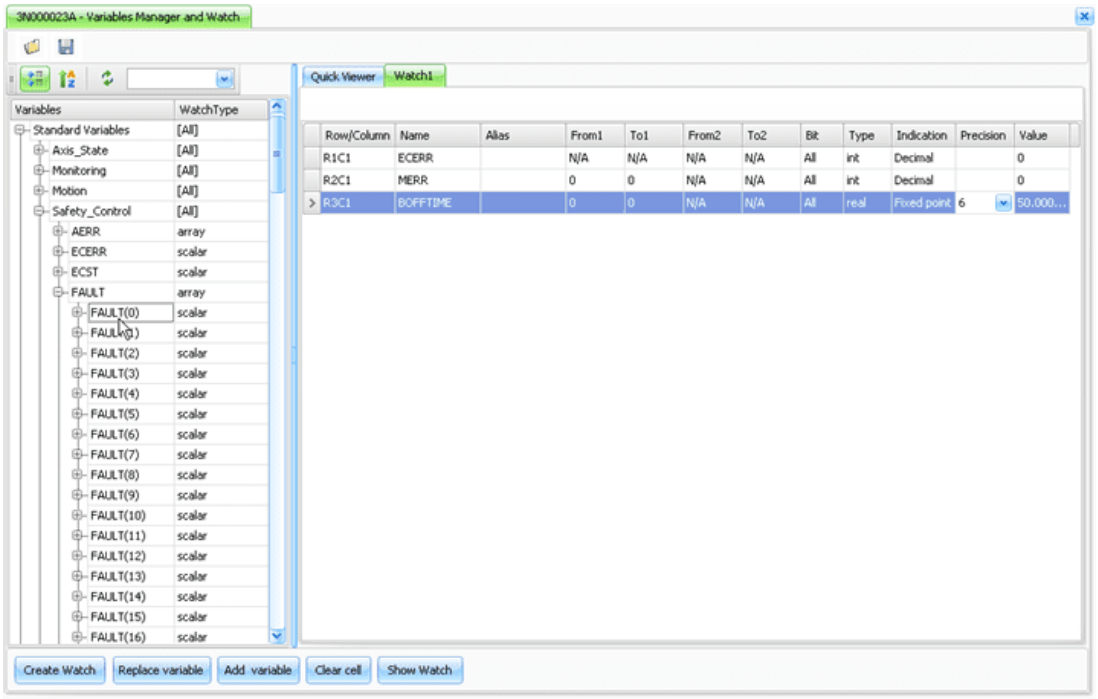

When you select **Boolean**, instead of 0 or 1, you get a graphic display in the Watch window, for example:




### *6.9.3.3 Replace a Variable*

To replace a variable in the Watch:

1. In the Watch Parameter Matrix select the row of the variable you want to replace.
2. In the Variables Tree select the variable you want it to be replace by.




3. Click **Replace variable** ( Replace variable ).

### *6.9.3.4 Add a Variable*

To add a variable to the Watch:

1. In the Variables Tree select the variable you want to add.

2. Click **Add variable** ( Add variable ).

#### *6.9.3.5 Remove a Variable*

To remove a variable from the Watch:

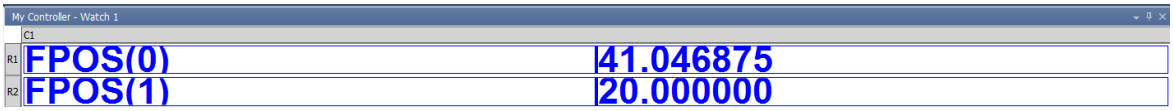
1. In the Watch Parameter Grid select the variable you want to remove.
2. Click **Clear cell** ( Clear cell ).

### *6.9.4 Watch Window*

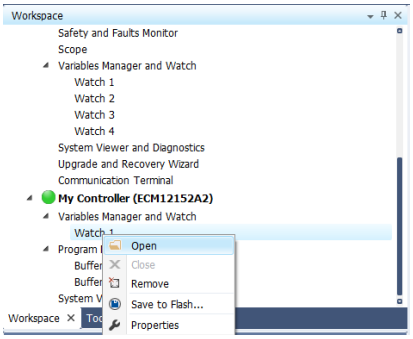
A Watch window consists of rows which can contain several columns. The display of each variable consists of the variable's name and its value. The value is constantly updated in the display.

#### *6.9.4.1 View Watch*

In the event that you have closed the Watch window, you can recall it by clicking **Show Watch** ( Show Watch ) in the Watch Parameters Matrix.



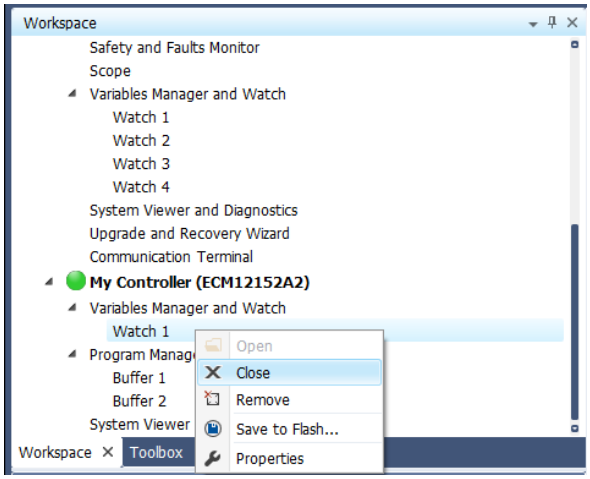

You can also display it from the Workspace Tree by right-clicking its symbol and selecting **Open** from the dropdown menu:




#### *6.9.4.2 Removing a Watch*

To remove a Watch:

1. If it is not displayed, switch to the **Workspace** panel.
2. Right-click the Watch you want to remove.
3. Select **Remove** from the dropdown menu.

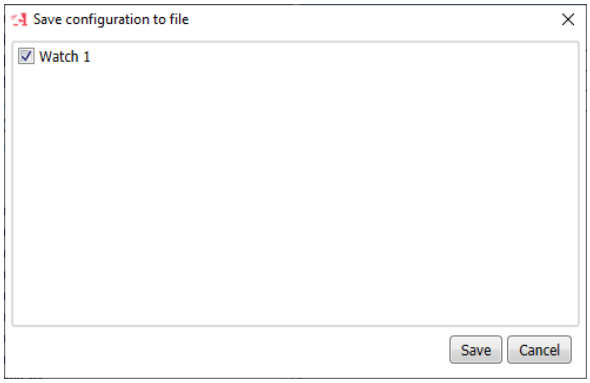



### *6.9.4.3 Save/Load Watch*

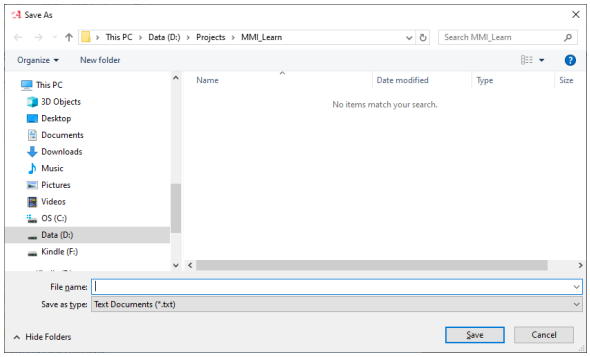
#### Save Watch Data

To save Watch data:

1. From the toolbar click **Save Configuration** ( Save ), the following prompt appears:




2. Click **Save**, a browser opens enabling you to save the data:



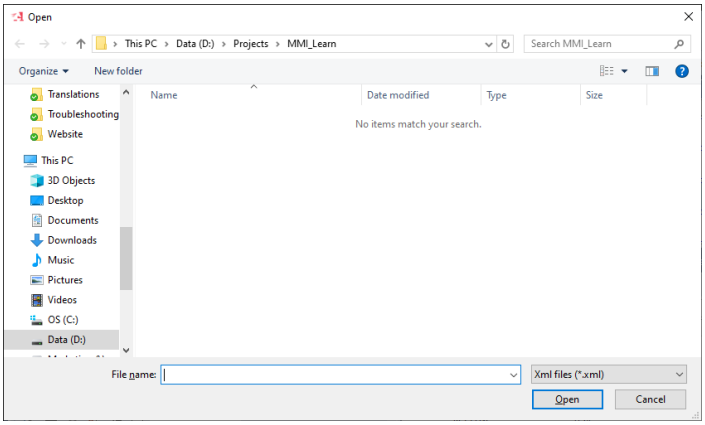

3. Enter a filename and click **Save**.

The data is saved with an **.ini** extension.

### Load Watch Data

To load previously saved Watch data:

1. From the toolbar click **Load Configuration** ( ), a browser opens enabling you to select the Watch you want to load:




2. Select the required Watch file and click **Open**.

#### *6.9.4.4 Watch Right-Click Options*

There are three right-click options associated with the Watch:

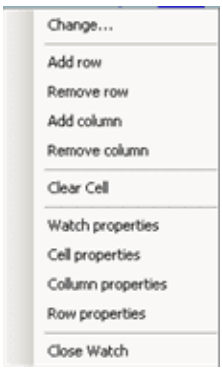
> Watch Window Right-Click Options

> Watch Row Right-Click Options

> Watch Column Right-Click Options

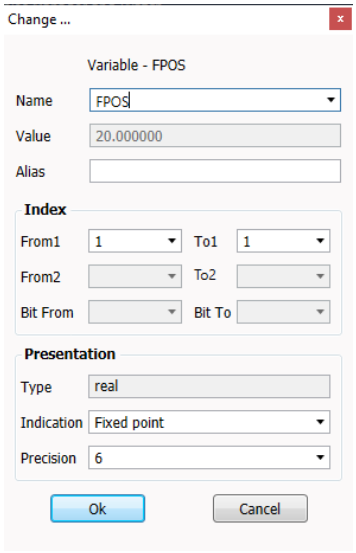
##### *6.9.4.4.1 Watch Window Right-Click Options*

When you right-click the Watch window, you are presented with the following menu:




This menu enables you to:

> **Change** - which enables you to make changes to the selected variable. When you select this option, the following prompt is displayed:

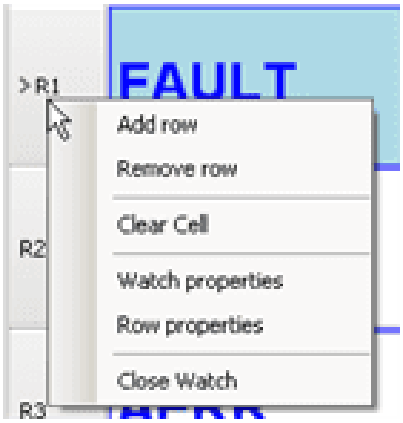



You can make any change and click **OK**.

- **Add row** - which adds a blank row to the Watch window, then you use Add a Variable in the matrix to select the variable for the row.
- **Remove row** - which removes the selected row in the Watch window.
- **Add column** - which adds a blank column to the Watch window, then you use Add a Variable in the matrix to select the variable or variables for the column.
- **Remove column** - which removes the selected column in the Watch window.
- **Clear Cell** - which removes the variable display from the selected cell.
- **Watch properties** - which displays the Watch window properties panel.
- **Cell properties** - which displays the Cell properties panel.
- **Column properties** - which displays the Column properties panel.
- **Row properties** - which displays the Row properties panel.
- **Close Watch** - which closes the Watch window.

#### *6.9.4.4.2 Watch Row Right-Click Options*

When you right-click a Row, you are presented with a number of options:

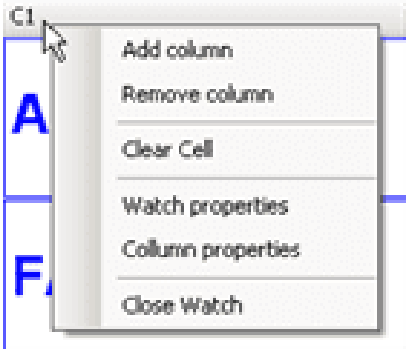



This menu enables you to:

- **Add row** - which adds a blank row to the Watch window, then you use Add a Variable in the matrix to select the variable for the row.
- **Remove row** - which removes the selected row in the Watch window.
- **Clear Cell** - which removes the variable display from the selected cell.
- **Row properties** - which displays the Row properties panel.
- **Close Watch** - which closes the Watch window.

*6.9.4.4.3 Watch Column Right-Click Options*

When you right-click a Column, you are presented with a number of options:




This menu enables you to:

- **Add column** - which adds a blank column to the Watch window, then you use Add a Variable in the matrix to select the variable or variables for the column.
- **Remove column** - which removes the selected column in the Watch window.
- **Clear Cell** - which removes the variable display from the selected cell.
- **Watch properties** - which displays the Watch window properties panel.
- **Column properties** - which displays the Column properties panel.
- **Close Watch** - which closes the Watch window.

#### *6.9.4.5 Watch Properties*

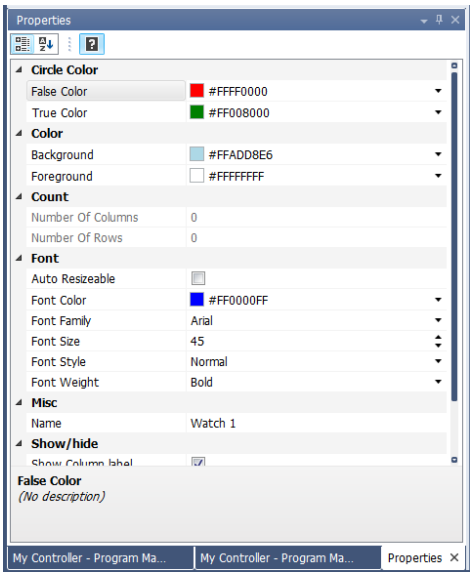
The following properties panels are associated with the Watch window:

- Watch Window Properties - enable you to set certain general properties of the Watch window as a whole.
- Watch Row Properties - enable you to set properties for the Watch window rows.
- Watch Column Properties - enable you to set properties for the Watch window columns.
- Watch Cell Properties - enable you to set the properties of the selected cell in the Watch window.

##### *6.9.4.5.1 Watch Window Properties*

When you select a Watch window, you activate the Watch's Properties.

It can also be displayed by selecting **Watch properties** from any of the right-click menus.




The options are:

- **Circle Color**

  This applies only to the graphic display of bit values in the Watch window and enables you to select the colors in which the values are to be displayed.

  - FalseColor - the color for a False value
  - TrueColor - the color for a True value

- **Color**

  This applies to the Watch window cells and enables you to select the colors in which the cells are to be displayed.

  - Background - indicates selected cells
  - Foreground - indicates non-selected cells

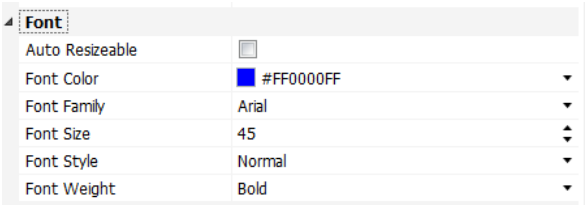
- **Count**

This applies to the number of columns and rows to be displayed:

- NumberOfColumns - the number of columns to be displayed
- NumberOfRows - the number of rows to be displayed

If you enter a number greater than the number of variables being watched, empty rows/columns are inserted.

- **Font**
  - Font - for selecting the display font
    The font section allows the user to select the the required font, size, and attributes:




    from which you can select the required font, size and attributes.

- **Misc**

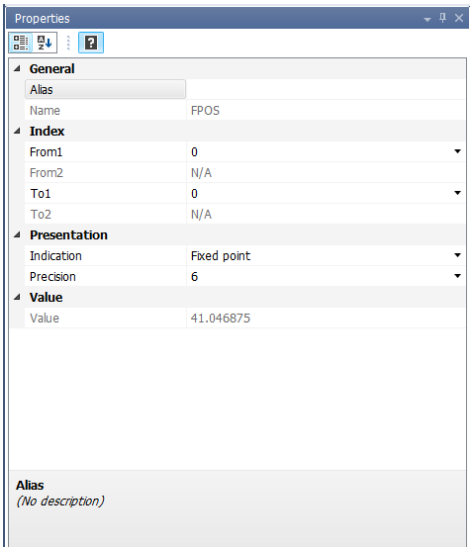
  By default, when you create a Watch, it is given the name **Watch#** (where # is the number of the Watch window). if you have several Watch windows running, you can use this field to give them each a unique identifying name.

- **Show/Hide**

  You can use these buttons to toggle between displaying or hiding the Row, Column and Variable labels.

*6.9.4.5.2 Watch Row Properties*

You can display the Row Properties panel either by selecting a Row, or from the right-click menu.




The options are:

- **Names**

  This enables you to enter a unique identifying name for the Row.

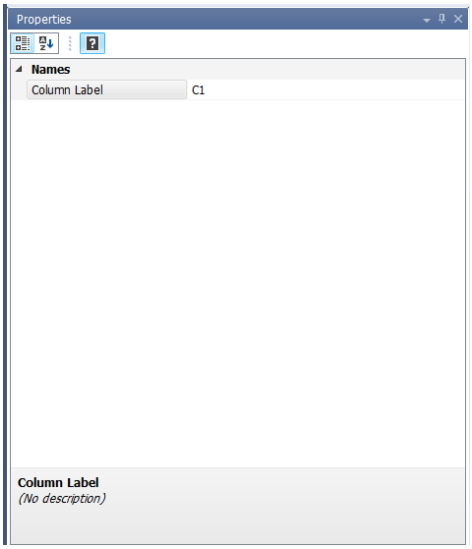
- **Type**

  This enables you to set display characteristics of the values.

**Type** can also be set in the Watch Parameter Matrix - see Modifying Watch Parameter Matrix Fields.

*6.9.4.5.3 Watch Column Properties*

You can display the Column Properties panel either by selecting a Column, or from the right-click menu.




The options are:

- **Names**

  This enables you to enter a unique identifying name for the Column.

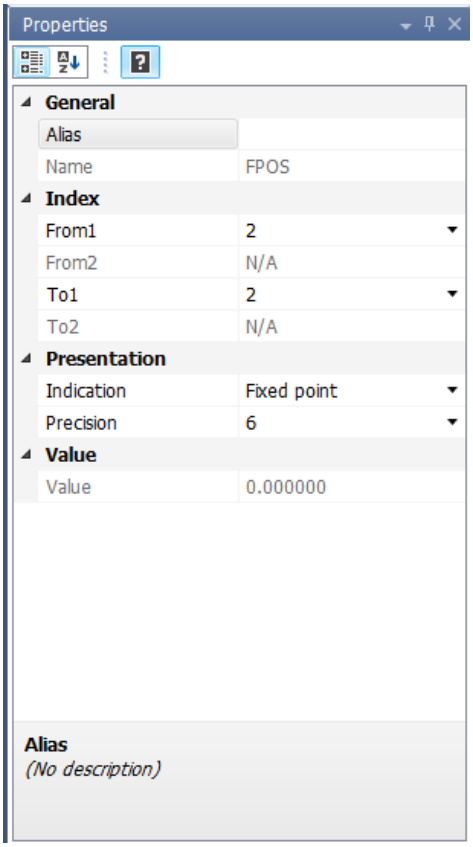
- **Type**

  This enables you to set display characteristics of the values.

**Type** can also be set in the Watch Parameter Matrix - see Modifying Watch Parameter Matrix Fields.

#### *6.9.4.5.4 Watch Cell Properties*

You can display the Cell Properties panel either by selecting a Cell, or from the right-click menu.




The options are:

- **Index**

  This enables you to change the Index of the variable displayed in the Cell.

- **Names**

  This is for information only.

- **Type**

  This enables you to set display characteristics of the values.

- **Value**

  This is for information only.

- **Visible elements**

  The number of elements - this appears for array and bit variables and indicates the total number of elements that can be visible.

## 6.10 System Viewer and Diagnostics

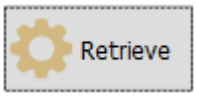
**System Viewer and Diagnostics** displays the system configuration stored on a controller. It also displays any inconsistency between the system configuration stored on the controller and the connected system.

To open **System Viewer and Diagnostics**:

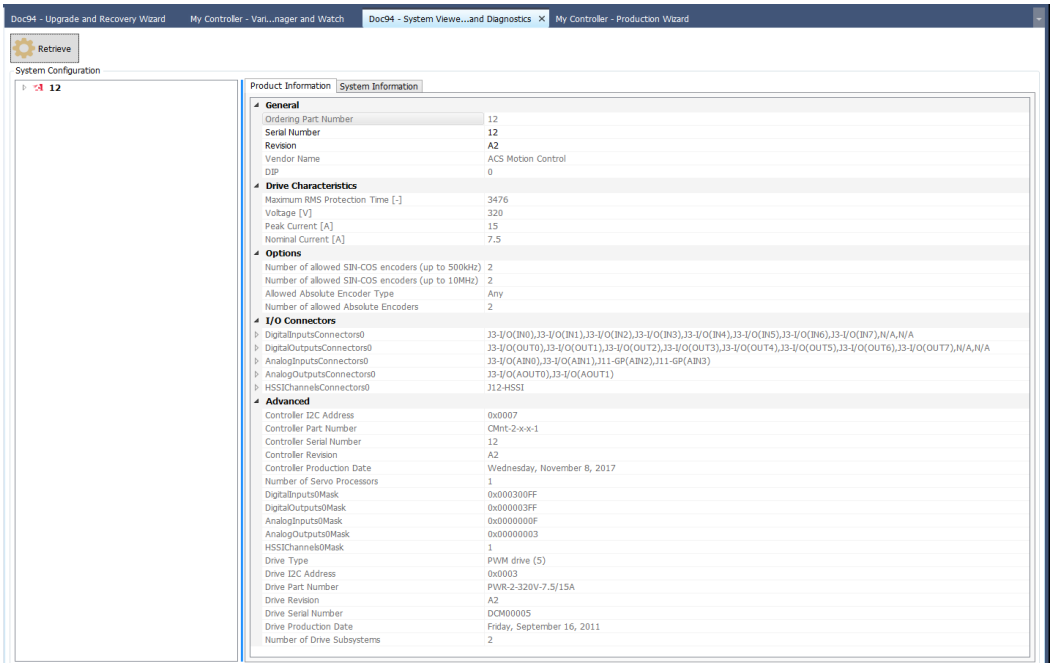
1. From the Toolbox, click **Diagnostics and Monitoring** to display the Diagnostics and Monitoring tools list.
2. Click **System Viewer and Diagnostics**.

You can use **Add Component** from the right-click menu of the controller in the Workspace Tree (see Controller Right-Click Options) to open **System Viewer and Diagnostics**.

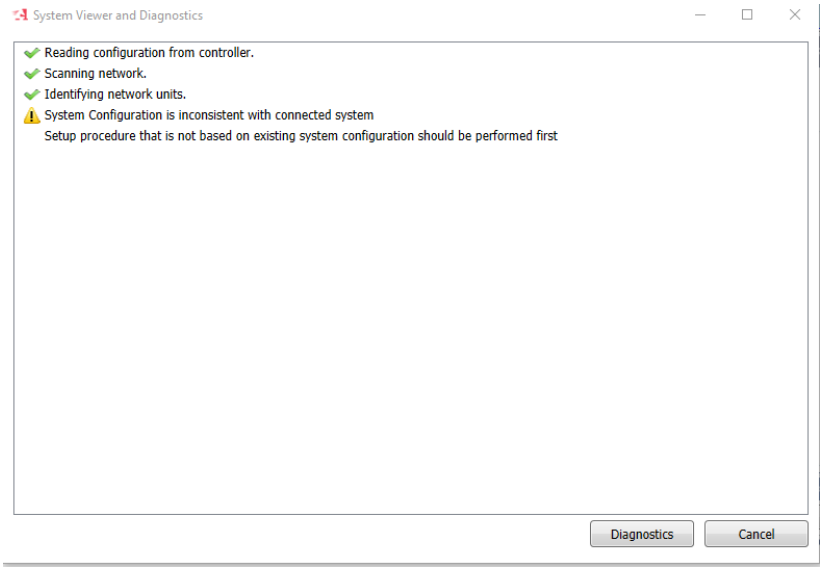
3. Click **Retrieve** :



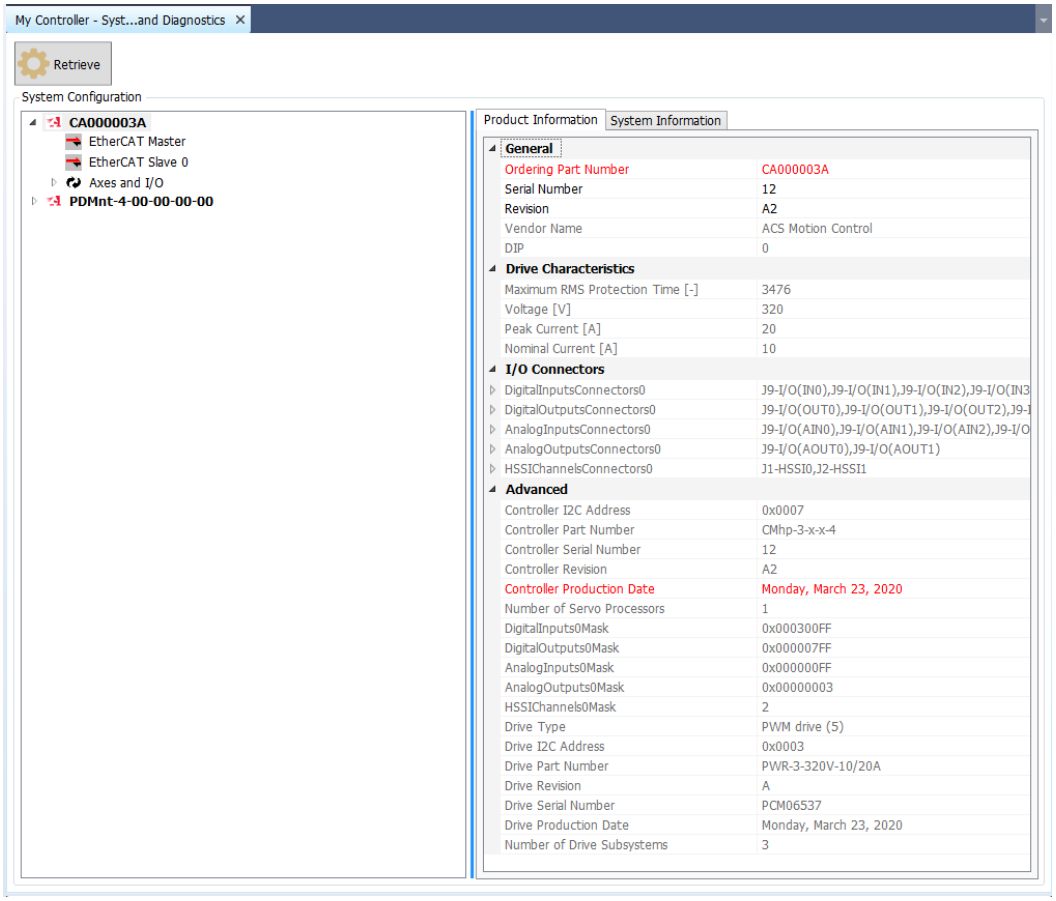

4. The System configuration stored in the controller is displayed:



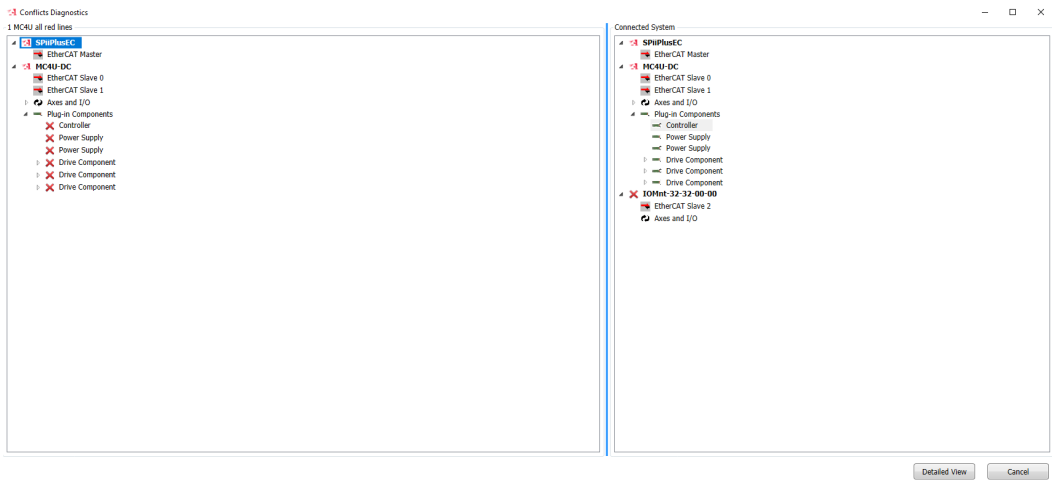

5. If the system contains inconsistent information, the following dialog box will be displayed:




6. Click **Diagnostics** to see additional information in the following window :




7. If the user clicks **Cancel** rather than **Diagnostics**, a window such as the following will be displayed:



Smarter Motion

*5 HaTnufa St.*
*Yokne'am Illit, 2066717*
*Israel*
*Tel: (+972) (4) 654 6440 Fax: (+972) (4) 654 6443*