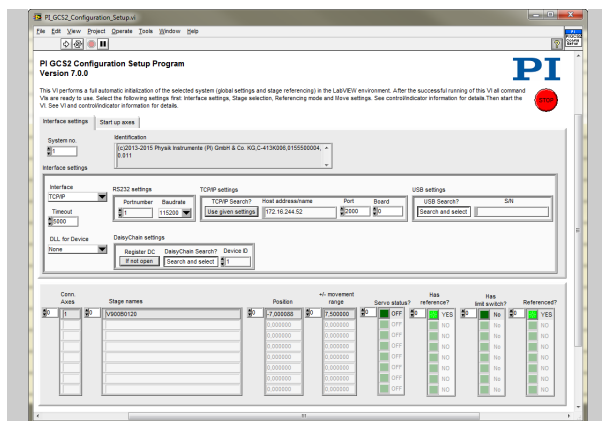


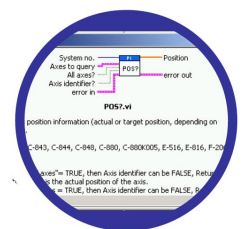
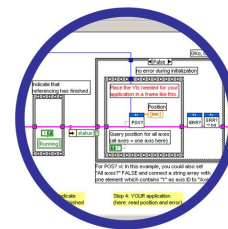
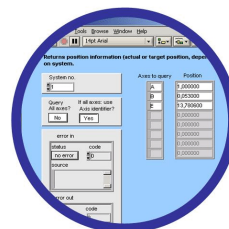
# SM158E Software Manual

## PI GCS2 Driver Library for NI LabVIEW

Release: 7.0.1      Date: 2021-03-03



**This document describes software for use with devices from PI.**



**(c)2002-2021 Physik Instrumente (PI) GmbH & Co. KG**  
 Auf der R  merstr. 1 · 76228 Karlsruhe, Germany  
 Tel. +49 721 4846-0  
 Support-software@pi.ws · www.pi.ws

# Table of Contents

<b>1. DISCLAIMER.....</b>	<b>3</b>
<b>2. INTRODUCTION .....</b>	<b>3</b>
2.1. PI GENERAL COMMAND SET (GCS).....	3
2.2. SCOPE OF THIS MANUAL.....	4
2.3. VI STRUCTURE .....	4
2.4. INSTALLATION .....	7
2.5. UPGRADING FROM OLD PRODUCT-SPECIFIC DRIVERS TO THE PI GCS2 DRIVER.....	7
2.6. USAGE.....	8
2.7. FIRST STEPS FOR GCS2-COMPATIBLE PI CONTROLLERS .....	8
<b>3. HIGH LEVEL VIS .....</b>	<b>10</b>
3.1. PI_GCS2_ALL_Vis.VI.....	10
3.2. PI_GCS2_CONFIGURATION_SETUP.VI .....	10
3.3. PI_GCS2_SLAVEDEVICE_CONFIGURATION_SETUP.VI .....	12
3.4. PI_GCS2_SAMPLE_APPLICATION_1_CONTROLLER.VI.....	14
3.5. PI_GCS2_SAMPLE_APPLICATION_2_CONTROLLERS.VI.....	14
3.6. PI_GCS2_SAMPLE_APPLICATION_WITH_ANALOG.VI.....	15
3.7. PI_GCS2_SIMPLE_TEST.VI.....	15
3.8. PITERMINAL.VI.....	16
3.9. SHOW_SAVE_LOAD_XY_DATA.VI .....	17
3.10. JOYSTICK_OPERATION.VI .....	18
<b>4. LOW LEVEL VIS .....</b>	<b>20</b>
4.1. ANALOG CONTROL.LLB (ANALOG DRIVER DISABLED).....	20
4.2. ANALOG CONTROL.LLB (ANALOG DRIVER ENABLED).....	20
4.3. COMMUNICATION.LLB .....	26
4.4. CONTROLLER ALGORITHMS.LLB .....	41
4.5. CONTROLLER DISPLAY.LLB.....	53
4.6. COORDINATE SYSTEMS.LLB.....	55
4.7. FILE HANDLING.LLB.....	62
4.8. GENERAL COMMAND.LLB.....	65
4.9. JOYSTICK.LLB .....	82
4.10. LIMITS.LLB .....	91
4.11. MACROS.LLB.....	106
4.12. OLD COMMANDS.LLB .....	111
4.13. OPTICAL OR ANALOG INPUT.LLB .....	118
4.14. PZT VOLTAGE.LLB.....	124
4.15. SCAN SUPPORT.LLB.....	134
4.16. SPECIAL COMMAND.LLB.....	137
4.17. SUPPORT.LLB .....	192
4.18. WAVEGENERATOR.LLB .....	210
<b>5. ERROR CODES .....</b>	<b>234</b>

© Copyright 2002 – 2021 Physik Instrumente (PI) GmbH & Co. KG

The following company names and brands are registered trademarks of Physik Instrumente (PI) GmbH & Co. KG: PI®, PICMA®, PILine®, PiezoWalk®, NEXACT®, NEXLINE®, NanoCube®, Picoactuator®, PInano®, PIMag®, Q-Motion®

Notes on third-party brand names and trademarks:

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and / or other countries.

LabVIEW, National Instruments and NI are trademarks of National Instruments. Neither the driver Software, nor any software programs or other goods or services offered by PI, are affiliated with, endorsed by, or sponsored by National Instruments.

The patents owned by PI can be found in our [patent list](#).

Software products that are provided by PI are subject to the General Software License Agreement of Physik Instrumente (PI) GmbH & Co. KG and may incorporate and/or make use of third-party software components. For more information, please read the General Software License Agreement and the Third Party Software Note linked below.

[General Software License Agreement](#)

[Third Party Software Note](#)

File: PI\_GCS2\_LabVIEW\_SM158E.doc

## 1. Disclaimer

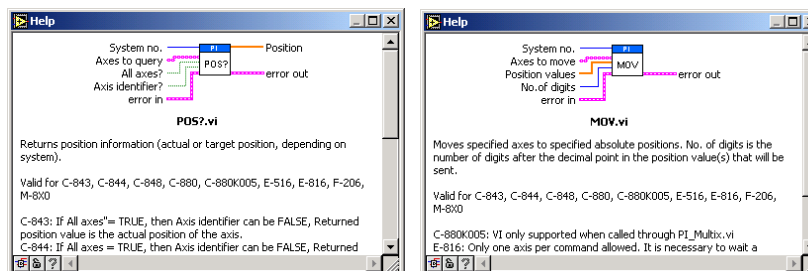
This software is provided “as is”. Physik Instrumente (PI) does not guarantee that this software is free of errors and will not be responsible for any damage arising from the use of this software. The user agrees to use this software on his own responsibility.

## 2. Introduction

The software for NI LabVIEW consists of a collection of virtual instrument (VI) drivers. All functionality involves invoking one or more VIs with the appropriate parameter and global variable settings.

These VIs are provided to ease the task of programming your application. They, and the accompanying documentation, assume a prior knowledge of proper NI LabVIEW programming techniques. The provided “Simple Test” and “Configuration Setup” VIs help to solve the essential initialization steps, but are not intended to provide an out-of-the-box, universal solution to a particular application.

To minimize the need for consulting the manual during programming, each VI comes with a detailed VI description that appears in the *Context Help* window when you move the cursor over the VI icon. Use the *Help*→*Show Context Help* menu sequence in the NI LabVIEW environment to display the *Context Help* window.



NI LabVIEW 8.6 or higher and NI-VISA 3.6 or higher must be installed prior to using this driver set.

To control an analog system, DAQmx 8.3 or higher and a DAQmx-compatible National Instruments DAC card which supports waveform generation must also be installed.

### 2.1. PI General Command Set (GCS)

This VI driver set supports the *PI General Command Set* (GCS), which is based on ASCII communication with well-defined commands and replies. This makes it possible to control different PI systems, such as the *E-518 Interface Module* or the *C-884 Multi-Axis Controller*, with only one driver set simply by “wiring” the correct command parameters to the associated VIs. To achieve this, a unique System no. must be selected in each *PI\_GCS2\_Configuration\_Setup.vi*. This System no. is then used in all sub-VIs to tell NI LabVIEW which connected system to talk to.

## Translation Libraries

To control PI systems with a non-GCS command set that is not compatible with the PI General Command Set, e.g. the *E-710 Digital Piezo Controller* or the *C-843 Motion Control Board*, controller-specific libraries are used. Each such library translates PI General Command Set commands to the controller's native language. There is also a universal library which adds this functionality: GCSTranslator; it must be installed on the computer in the LabVIEW\Low Level folder, no matter whether the system being controlled is PI General Command Set compatible or not.

If you install this driver set using the setup program from the main directory of the installation medium, this installation is done automatically. The installation wizard makes sure that all necessary libraries and their data files are correctly registered in the operating system environment and can be found by the GCS drivers.

To access the GCS2 driver set for NI LabVIEW, all VIs are available at the *Functions* palette in the category *Instrument I/O* → *Instrument Drivers* → *PI*.

Once the libraries for the system to control are installed, this driver set for NI LabVIEW can be used to control any GCS-compatible system. To be compatible with previous versions of the driver set for NI LabVIEW, all former XXXX\_Configuration\_Setup VIs (with XXXX being the name of your PI device) can be found in the \Obsolete subfolder of the driver set.

## Units and GCS

The GCS system uses physical units of measure. Most controllers and GCS software have default conversion factors chosen to convert hardware-dependent units (e.g. encoder counts) into mm or degrees, as appropriate. These defaults are generally taken from a database of stages that can be connected. The direction of motion associated with positive and negative relative moves can also be controlled by parameter settings. In some cases an additional scale factor can be applied, making a second physical unit available without overwriting the conversion factor for the first. It is also sometimes possible to enter a conversion factor as numerator and denominator of a fraction, reducing the number of digits and outside calculations needed for high-precision entry of gearhead system values. See the DFF.vi and SPA.vi command descriptions (if supported by your PI controller), taking special note of the sections referring specifically to your controller.

## 2.2. Scope of This Manual

This manual contains descriptions of all VIs provided by PI. To find out if your controller supports a certain command and hence the corresponding VI, check the HLP? response of your controller, which is available via the HLP?.vi, or use the HasCmd?.vi.

For VIs which are based on GCS commands, see the User manual of the controller or, if present, the special GCS commands manual for further details.

## 2.3. VI Structure

The folder structure of the drivers for NI LabVIEW consists of the main folder "LabVIEW" with the sub-folders "Low Level" and "Obsolete".

The main folder "LabVIEW" contains a terminal VI (for command-based systems), a configuration VI (PI\_GCS2\_Configuration\_Setup.vi), a simple test VI, and several sample programs.

The sub-folder “Low Level” contains VIs for the following functions:

- Establishing communication with different PI systems which support the PI General Command Set via RS-232, USB or TCP/IP interfaces, defining the parameter IDs of the connected axes or sending and receiving ASCII characters to/from the specified system. These VIs are mainly sub-VIs for the PI\_GCS2\_Configuration\_Setup.vi.
- Support functions which are helpful for several common tasks in NI LabVIEW and are used by the command VIs
- Sending system-specific commands (system-specific commands are separated into function-specific LLBs) which are the “construction set” to build your application.

Additionally, the sub-folder “Low Level” contains the GCSTranslator.dll (for 32 Bit NI LabVIEW versions) and the GCSTranslator\_x64.dll (for 64 Bit NI LabVIEW versions).

Following the data flow concept of LabVIEW, all VIs have their wiring inputs on the left side and their wiring outputs on the right side of each connector pane. For quick integration, this *connector pane* in most cases has the following pattern:

1					15
2	7	9	11	13	16
3					17
4					18
5	8	10	12	14	19
6					20

The terminals are assigned as follows (if the mentioned, the control/indicator is present in one of the supplied LabVIEW libraries (\*.llb)):

- 1 System number
- 2 Optical board, Interface, or other main input control
- 3 Axes to query, Affected axes, Number of systems, or other main input control
- 4 All axes?, Invert order?, or other main input control
- 5 Axis identifier?, No. of digits, or other main input control
- 6 Error in
- 7 Parameter number, Without axis ID?, or other input control
- 8 Step size, or other input control
- 9 AA step size, or other input control
- 10 Input control
- 11 Input control or output indicator
- 12 Input control or output indicator
- 13 Input control or output indicator
- 14 Input control or output indicator
- 15 Hidden error, Connected axes, String read, or other main output indicator
- 16 Axes to query out, Bytes read, or other main output indicator
- 17 No. of rows, or other main output indicator

- 18 Output indicator
- 19 Output indicator
- 20 Error out

Also note that this driver set does not use the standard NI LabVIEW error numbers recommended by National Instruments, but rather those used by PI controllers. As a result, the error texts displayed by NI LabVIEW will not describe the error accurately. Use `GCSTranslateError.vi` to get the description of a PI GCS error number. Some VIs use an additional indicator `Controller error` to indicate that the selected system has been queried for a controller error with `ERR?` and reported an error number  $\neq$  zero.

See also chapter 4 for a summary of error numbers produced by this driver set.

In NI LabVIEW, uncheck *Enable automatic error handling dialogs* in *Tools→Options→New and Changed in 7.x* to prevent that NI LabVIEW suspends execution and displays an error dialog box for any error that occurs during the execution of the VIs.

### Important:

Before running any VIs to control a connected system, **PI\_GCS2\_Configuration\_Setup.vi** (located in the main folder) must be run. This initialization VI performs all necessary steps automatically:

1. It opens the communications port,
2. It defines the IDs for the connected axes,
3. It references the connected stages (if appropriate), depending on if the controller requires a referencing before axes can be moved and on your custom settings,
4. It defines the controller name.

After these steps all parameters are saved into global variables, so that other VIs invoked during the same NI LabVIEW session can access this data at runtime.

As the initialization is a complex procedure which uses a large number of sub-VIs, **PI\_GCS2\_Configuration\_Setup.vi** is password-protected, meaning that you cannot see or modify the diagram. In this way, the full initialization is packed into one single and fully tested procedure which you simply insert into your own application program. For security reasons as well as your convenience, we recommend that you not modify this VI.

For testing a PI system using a command-based interface, the easiest method is to call `PITerminal.vi`, which is located in the "LabVIEW" main folder. `PITerminal.vi` will open `PI_GCS2_Configuration_Setup.vi` if you are not already connected to a PI device. `PITerminal.vi` can be operated in parallel to any other application since communication only takes place when sending commands actively to the PI device.

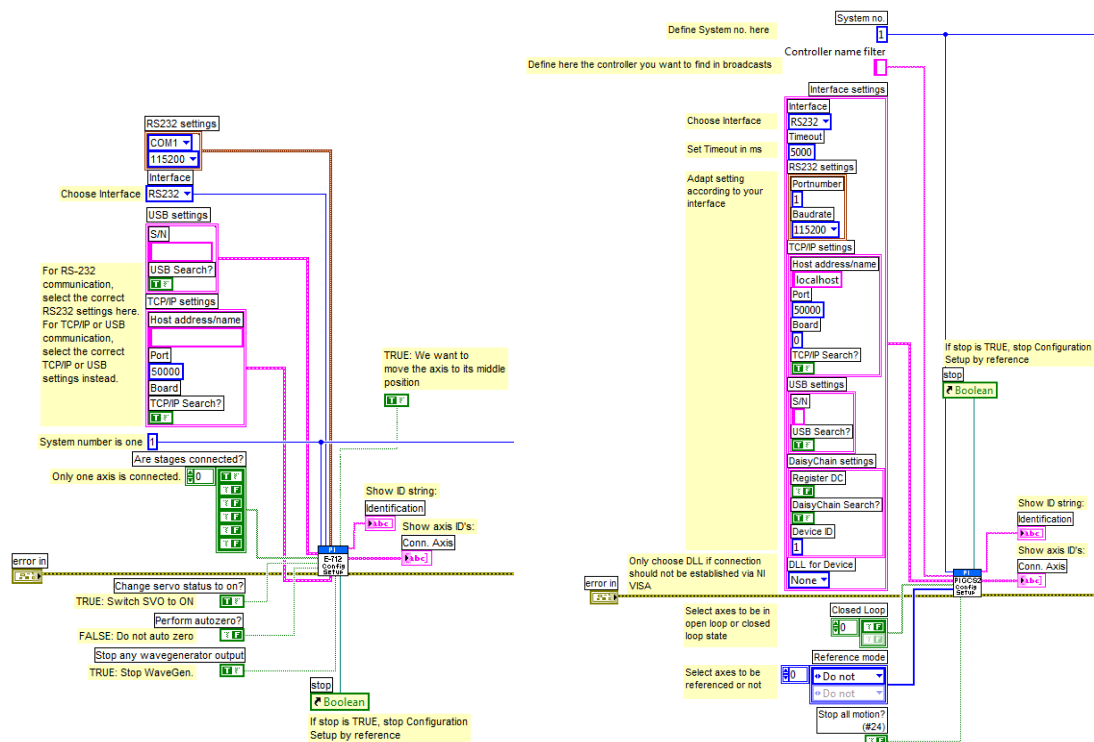
## 2.4. Installation

The PI GCS2 driver for NI LabVIEW is a controller-independent driver and is designed to work directly from the NI LabVIEW environment. The GCS2LabVIEW driver will be installed by the setup PISoftwareSuite.exe on the supplied installation media to the path C:\ProgramData\PI\LabVIEW.

If you copied the driver to the instr.lib folder, NI LabVIEW will automatically find the new driver. If you have not copied the driver there, NI LabVIEW might ask where the new PI GCS2 driver for NI LabVIEW is located.

**Note:** A XXXX \_Configuration\_Setup.vi should no longer be used to configure a certain controller. These configuration setups will still work but are set to obsolete and will be removed in later versions of the PI\_GCS2\_driver for NI LabVIEW. Use the new controller-independent PI\_GCS2\_Configuration\_Setup.vi instead. The following section describes how to replace XXXX \_Configuration\_Setup.vi by the new PI\_GCS2\_Configuration\_Setup.vi in your application.

## 2.5. Upgrading from Old Product-Specific Drivers to the PI GCS2 Driver



The figure above shows the old product-specific XXXXX\_Configuration\_Setup.vi (left) and the new general PI\_GCS2\_Configuration\_Setup.vi (right). The configurations settings are similar but differ in details.

The connection settings (RS232, USB, TCP/IP and DaisyChain settings and the Interface definition) are combined as single interface settings cluster for the new PI\_GCS2 driver. In order to replace the old product-specific XXXX\_Configuration\_Setup.vi, all you have to do for communication settings is to cluster them as interface settings.

The broadcast function of the USB connection or the TCP/IP connection in the new PI\_GCS2\_Configuration\_Setup.vi will find all PI controllers using the same



connection settings. In order to filter the found controllers by a specific controller type, use the “Controller name filter”.

“Change servo status to on?”, “Perform autozero?” and “Move axis to middle position” are product-specific options. They are replaced by the more general “closed loop” and “reference mode” options.

The “Are stages connected” array is no longer required because the PI\_GCS2\_Configuration\_Setup.vi will check automatically whether the stage is connected or not. The number of stages which are available is defined by the number of elements used for the “Closed Loop” and “Reference Mode” arrays.

“Stop any wavegenerator output” is also a product-specific option. This is replaced by the more general “Stop all motion?” option.

## 2.6. Usage

If the GCS2 driver for NI LabVIEW is located in the NI LabVIEW instr.lib folder, NI LabVIEW will find all necessary SubVIs automatically if you open a VI that contains SubVIs from PI.

Please do not modify the GCS2 driver for NI LabVIEW or work in the NI LabVIEW instr.lib folder directly. When updating the driver, all VIs will be replaced by the new VIs.

## 2.7. First Steps for GCS2-Compatible PI Controllers

*Step 1:* To configure positioners connected to your PI GCS2 device, please use PIMikroMove.

*Step 2 (advanced users can skip this step):* To check communication between the PI GCS2 device and the host PC, run PI\_GCS2\_Simple\_Test.vi. This VI will return the ID string of the PI device. See chapter 2 for a description of this VI and use the *Help→Show Context Help* menu sequence in the NI LabVIEW environment to display the *Context Help* window with the VI and control/indicator descriptions.

*Step 3:*

### **WARNING: PI\_GCS2\_Configuration\_Setup.vi May Cause Move**

When you start PI\_GCS2\_Configuration\_Setup.vi with Reference mode ≠ “Do not reference”, the VI will automatically determine which axes have a reference switch and which have limit switches and will move these stages to these sensor switches. It is therefore important to make sure that items connected to or mounted on connected stages cannot be damaged by such a move. If referencing is not possible (because the connected stage has no reference or limit switch) or not desired, Reference Mode should be set to “Do not reference”.

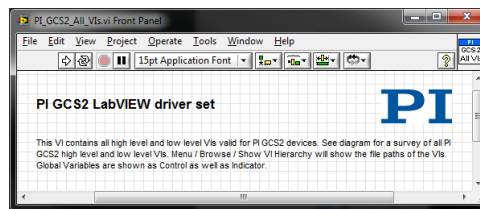
Open PI\_GCS2\_Configuration\_Setup.vi. Select the correct interface and run the VI. This VI performs all steps necessary for a full configuration of the driver VIs in the NI LabVIEW environment: the definition of axis IDs, the initialization of the connected stages and the definition of the controller name. During your testing phase (when you simply run the VIs without wiring them together into a program), do not close PI\_GCS2\_Configuration\_Setup.vi; otherwise all global settings will be lost and the driver VIs will not work.

When programming your application, you can implement `PI_GCS2_Configuration_Setup.vi` as an initialization VI in your software. See chapter 2 for a detailed description of `PI_GCS2_Configuration_Setup.vi` and use the *Help*→*Show Context Help* menu sequence in the NI LabVIEW environment to display the *Context Help* window with the VI and control/indicator descriptions.

Before using a joystick connected to the host PC, install the joystick driver (if required) and calibrate the joystick in the Windows control panel.

### 3. High Level VIs

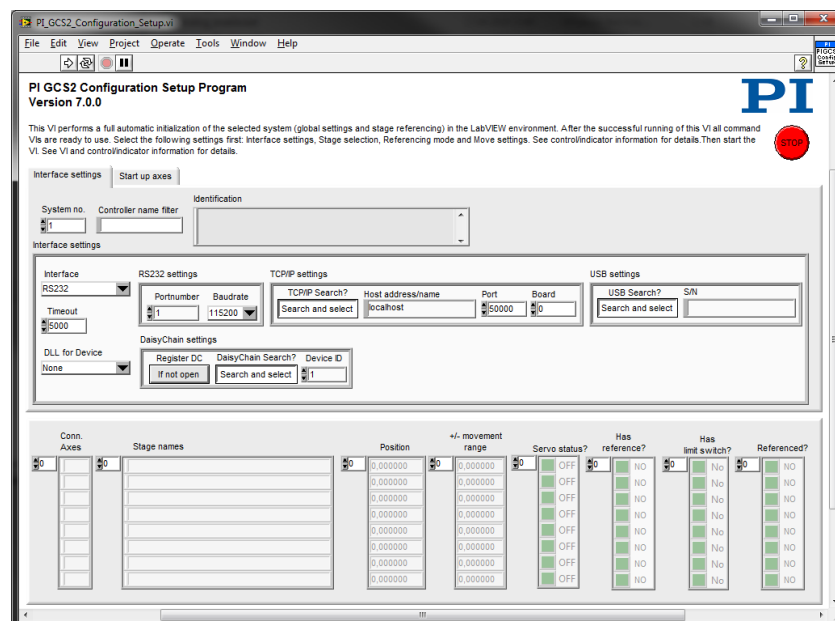
#### 3.1. PI\_GCS2\_All\_VIs.vi



This VI contains all high level and low level VIs valid for PI GCS2 devices. See diagram for a survey of all PI GCS2 high level and low level VIs. Menu / Browse / Show VI Hierarchy will show the file paths of the VIs. Global Variables are shown as Control as well as Indicator.



#### 3.2. PI\_GCS2\_Configuration\_Setup.vi



This VI performs a full automatic initialization of the selected system (global settings and stage referencing) in the LabVIEW environment. After the successful running of this VI all command VIs are ready to use. Select the following settings first: Interface settings, Stage selection, Referencing mode and Move settings. See control/indicator information for details. Then start the VI.

See VI and control/indicator information for details.

- System No.: 1 in a one-system-only configuration
- Interface settings (= RS232, RS232 DaisyChain, USB, USB DaisyChain, TCP/IP, TCP/IP DaisyChain or PCI (currently not supported)). For interface = RS232, select the correct Portnumber and Baudrate. For interface = TCP/IP, select if you want to call a dialog to select the connected PI GCS2 device from a list (TRUE) or if you know the host address/name of the controller to connect (FALSE). For interface = USB, select the appropriate Serial Number of the controller to connect to (USB Search? FALSE) or if you want to search for the controller (USB Search? TRUE). DLL for Device: If the connection should be established via a special DLL file, please select the DLL here (For interface USB or DaisyChain communication, the PI\_GCS2\_DLL will be used automatically). If DLL for Device is None, NI VISA communication will be used.
- Timeout value (in milliseconds)
- Stop all motion? (#24): Select here if motion (e.g. WaveGenerator) should be stopped when connection to the PI GCS2 device
- Reference mode: To perform closed loop motion all connected axes with incremental sensors have to be referenced. If an axis is already referenced it will not be referenced again
- Closed Loop: Select if the connected axes can be operated in closed loop mode after the successful execution of the PI\_GCS2\_Configuration\_Setup.vi  
Then start the VI.

PI\_GCS2\_Configuration\_Setup.vi performs the following initialization tasks:

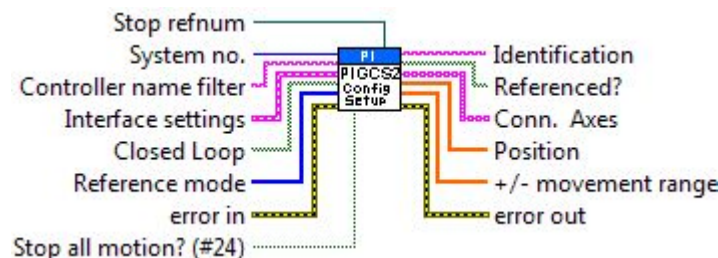
1. Initializes all Global variables.
2. Runs PI Open Interface of one system.vi to open a connection to the controller.
3. Runs \*IDN?.vi to query for the controller identification string.
4. Defines the selected system via the \*IDN? answer
5. Runs Define connected axes.vi with Read from controller = TRUE to query for the axis IDs of the connected axes.
6. Runs CST?.vi to query for the names of the connected stages.
7. If Stop all motion? (#24) is TRUE and #24 is available, sends #24 to the device and resets error 10.
8. Checks if EAX command is available or if system is hexapod type.
9. If EAX is available and axes should be referenced, enables the connected axes via the EAX command
10. Determines which axes have a reference switch (TRS?.vi).
11. Performs the reference motion via FRF, FNL or FPL depending on the connected device. Reference motion only is performed if axes have not been referenced before.
12. Depending on the selection of "Closed Loop" the servo of the connected axes will be set via the SVO command.
13. Reads the position range (TMN?, TMX?) of the axes.
14. Runs POS?.vi to query the position of the axes.

15. Runs FRF?.vi to query the reference status of the connected axes
16. Runs SVO?.vi to display servo status of axes
17. Runs ERR?.vi to query the controller for its error status.
18. Runs GCSTranslateError.vi to append the error message which corresponds with a GCS error number returned by ERR?.vi to Source from Error out.

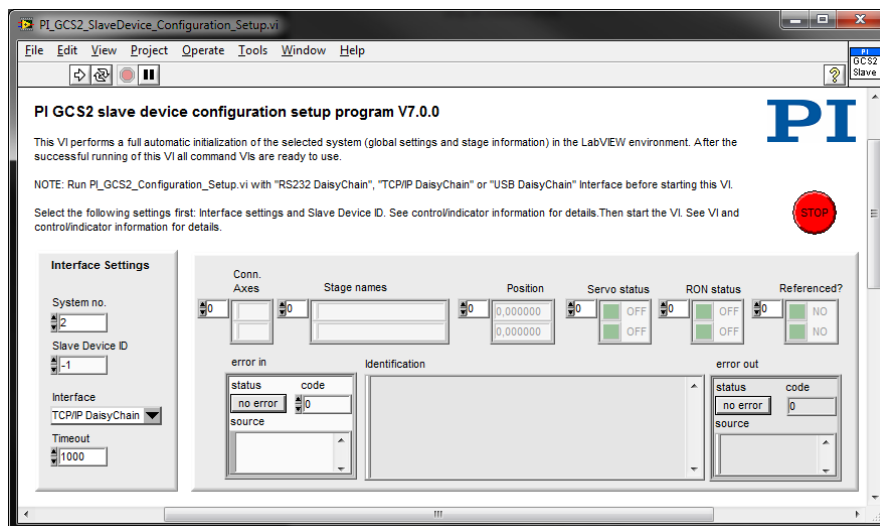
Use this VI as the initialization VI for the PI GCS2 device in your application.

When using as a sub-VI, connect Stop refnum terminal to stop VI from caller.

As the initialization is a complex procedure which uses a large number of sub-VIs, PI\_GCS2\_Configuration\_Setup.vi is password-protected, meaning that you cannot see or modify the diagram. In this way, the full initialization is packed into one single and fully tested procedure which you simply insert into your own application program. For security reasons as well as your convenience, we recommend that you not modify this VI.



### 3.3. PI\_GCS2\_SlaveDevice\_Configuration\_Setup.vi



This VI performs the initialization of the selected slave device (global settings and stage referencing) in the LabVIEW environment. After the successful run of this VI all command VIs are ready to use.

NOTE: Run PI\_GCS2\_Configuration\_Setup.vi with "RS232 DaisyChain", "TCP/IP DaisyChain" or "USB DaisyChain" Interface before starting this VI.

Specify the correct parameters first:

- System No.: if the System No. of the master device is 1 (selected in PI\_GCS2\_Configuration\_Setup.vi), the first slave device normally has System No. 2.
- Slave Device ID: As the DeviceID of the master device is always 1, the first slave device of the system normally has SlaveDeviceID 2.
- Interface

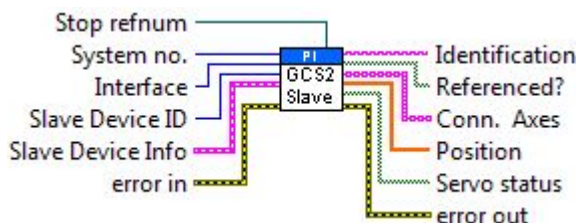
Then start the VI.

PI\_GCS2\_SlaveDevice\_Configuration\_Setup.vi performs the following initialization tasks:

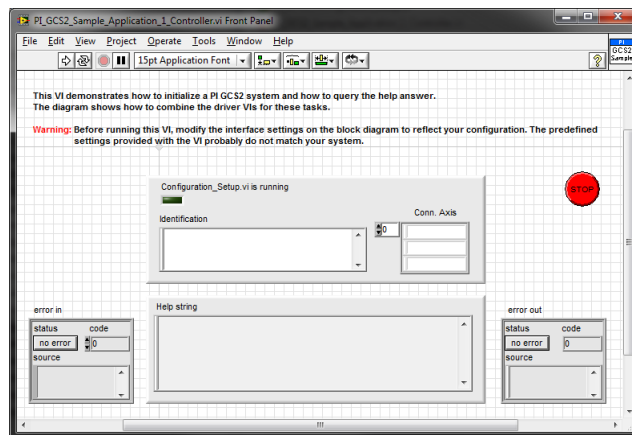
1. Runs PI Open Interface of one system.vi to open a connection to the controller.
2. Runs \*IDN?.vi to query for the controller identification string.
3. Defines the selected system via parsing the \*IDN? answer for known systems.
4. Queries the connected axes and stages using Define connected axes.vi and CST?.vi
5. Determines the servo status of all connected axes (SVO?)
6. Determines the RON status of all connected axes.
7. Determines the FRF status of all connected axes.
8. Runs POS?.vi to query for the position of all axes.
9. Runs ERR?.vi to query the controller for its error status.
10. Runs GCSTranslateError.vi to append the error message which corresponds with a GCS error number returned by ERR?.vi to Source from Error out.

Use this VI as the initialization VI for a slave device in your application.

As the initialization is a complex procedure which uses a large number of sub-VIs, PI\_GCS2\_SlaveDevice\_Configuration\_Setup.vi is password-protected, meaning that you cannot see or modify the diagram. In this way, the full initialization is packed into one single and fully tested procedure which you simply insert into your own application program. For security reasons as well as your convenience, we recommend that you not modify this VI.



### 3.4. PI\_GCS2\_Sample\_Application\_1\_Controller.vi



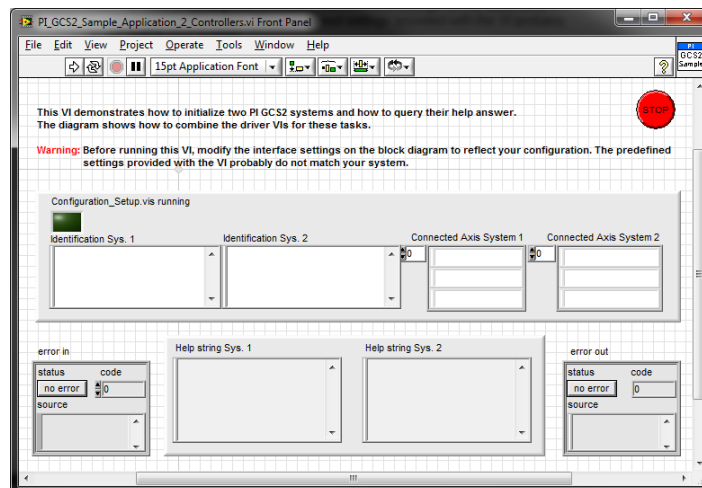
This VI demonstrates how to initialize a PI GCS2 system and how to query the help answer.

The diagram shows how to combine the driver VIs for these tasks.

Warning: Before running this VI, modify the interface settings on the block diagram to reflect your configuration. The predefined settings provided with the VI probably do not match your system.



### 3.5. PI\_GCS2\_Sample\_Application\_2\_Controllers.vi



This VI demonstrates how to initialize two PI GCS2 systems and how to query the help answers.

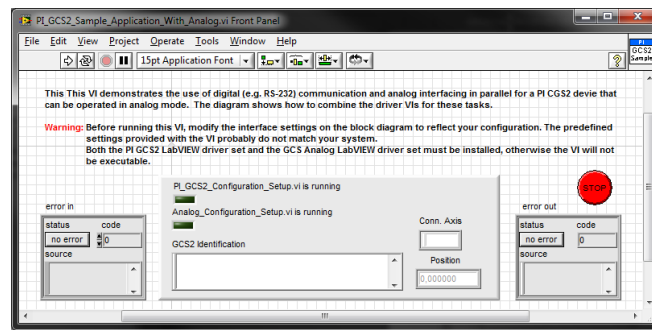
The diagram shows how to combine the driver VIs for these tasks.

Warning: Before running this VI, modify the interface settings on the block diagram to reflect your configuration. The predefined settings provided with the VI probably do not match your system.





### 3.6. PI\_GCS2\_Sample\_Application\_With\_Analog.vi



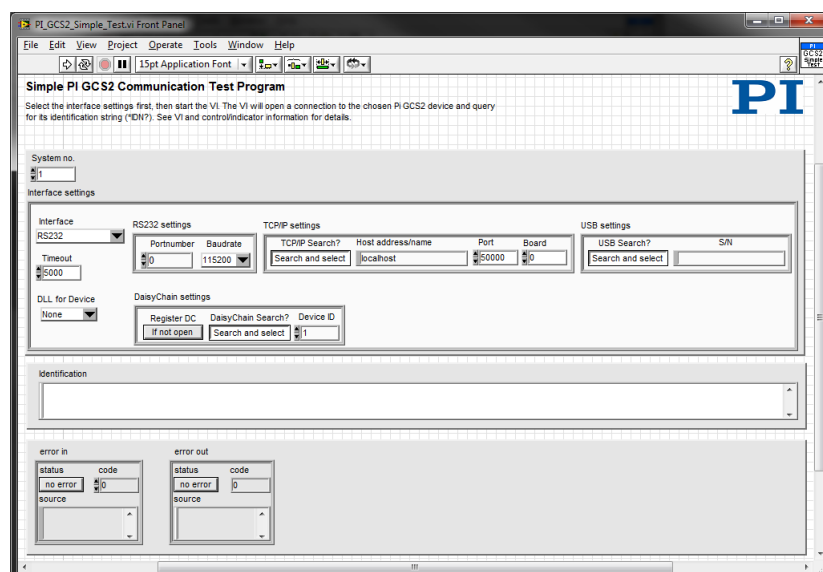
This VI demonstrates the use of digital (e.g. RS-232) communication and analog interfacing in parallel for a PI GCS2 device which can be operated in analog mode. In this example the PI GCS2 device is connected through COM port 1 with baudrate 115200. The VI first initializes the device via the selected digital interface by running PI\_GCS2\_Configuration\_Setup.vi. It then switches to analog control using SwitchToAnalogOrDigital.vi and initializes the analog interface by running Analog\_Configuration\_Setup.vi. After these initialization steps, it moves one axis relative to its current position by outputting a voltage by the DAC card used, waits 10 ms and queries for the position of this axis. Finally, it switches back to digital control and closes both the analog and the digital connection to the device.

The diagram shows how to combine the driver VIs for these tasks.

Warning: Before running this VI, modify the interface settings to reflect your configuration. The predefined settings provided with the VI probably do not match your system.



### 3.7. PI\_GCS2\_Simple\_Test.vi



This simple test VI is a stand-alone sample application. Use the *Help->Show Context Help* menu sequence in the LabVIEW environment to display the *Context Help* window with the VI and control/indicator descriptions.



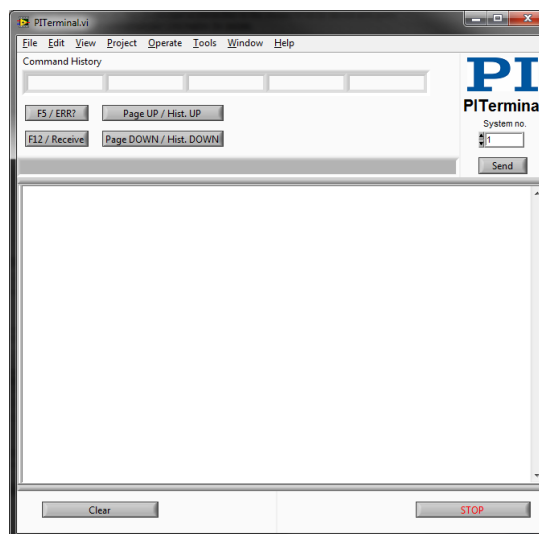
Specify

- System number (= 1 in a one-system configuration),
- Interface settings
- For Interface = RS232: Choose the port number the device is connected to and an appropriate baud rate.
- For Interface = USB: Choose the appropriate serial number of the device to connect to or if you want to search for the controller.
- For Interface = TCP/IP: Host address/name and portnumber (default is 50000) of the device to connect to or if you want to search for the controller.
- Timeout value (in milliseconds)

Then start the VI. The VI will open a connection to the chosen device and query the device for its identification string. The diagram shows how to combine the driver and support VIs for these tasks.



### 3.8. PITerminal.vi



Simple Terminal Application.

This VI can be executed in parallel to any existing PI LabVIEW application.

If not already connected to a PI device, PI\_GCS2\_Configuration\_Setup.vi is opened automatically. Please use PI\_GCS2\_Configuration\_Setup.vi to connect to your PI GCS2 device.

Shortcut keys:

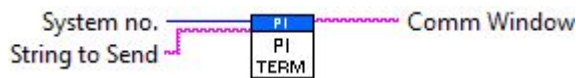
Return: Send or query

F5: ERR? (with error translation)

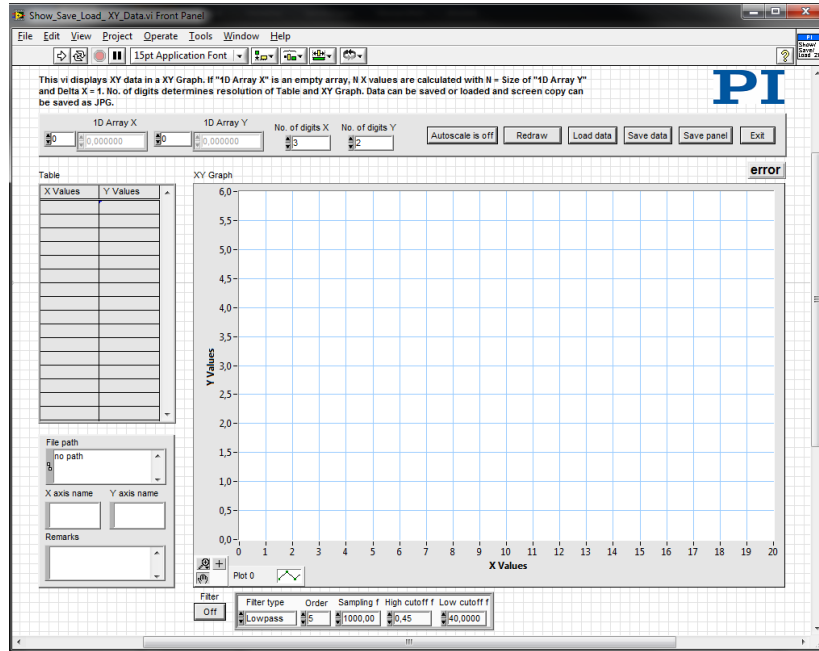
F12: Additional receive function (e.g. to clear output buffer of the connected device)

Page Up: Command history previous entry

Page down: Command history next entry



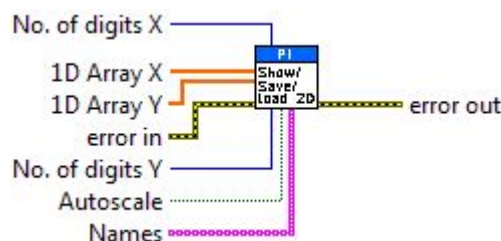
### 3.9. Show\_Save\_Load\_XY\_Data.vi



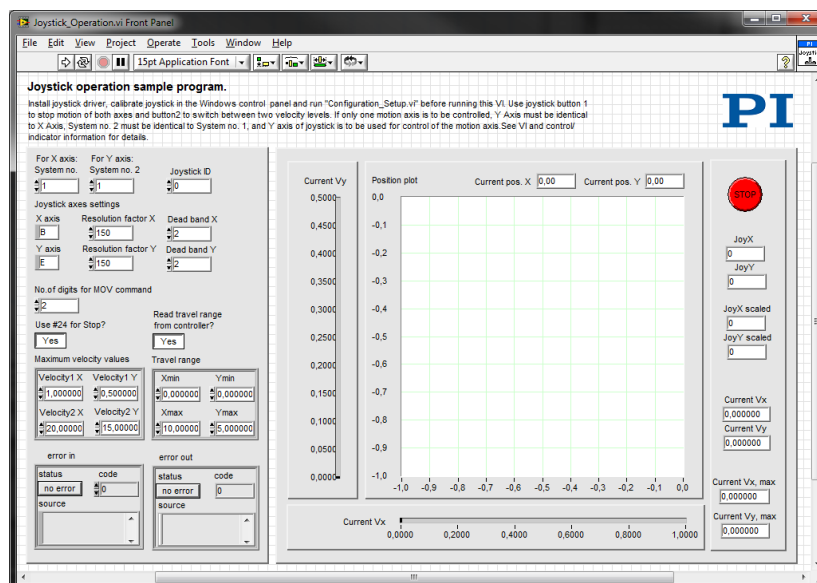
This VI displays XY data in an XY Graph. If 1D Array X is an empty array, N X values are calculated with  $N = \text{Size of 1D Array Y}$  and  $\Delta X = 1$ . No. of digits determines the resolution of Table and XY Graph. Data can be saved or loaded and a screen copy can be saved as JPG.

If data (1D Array X, 1D Array Y) are sent to the VI via the corresponding connectors, the VI will display the corresponding graphics after being called. To load data at runtime, press the Load data button. A dialog will pop up where a data file to open can be selected. The VI can read data in GCSArray, GCSTable and simple ASCII column format. Autoscale can be switched on or off. If Autoscale is off, the Y axis of the graph is scaled from 0-10.

Press Save data to save data (file header and numerical data). Data will be saved in GCS Array format. The file header will contain information given in X axis name, Y axis name and Remarks. With Save panel a screen copy of this VI can be saved as a JPG file. XY Graph will show the Y values over corresponding X values. Table contains the numerical values for X and Y. Press Exit to stop execution of this VI.



### 3.10. Joystick\_Operation.vi



This VI can be used to control two closed-loop axes (which can belong to one or two connected systems) with a 2-button, 2-axis joystick connected to the host computer. The absolute value of the joystick position is converted into velocity values for the two stages connected to the system being commanded. Two velocity levels for each axis can be specified, e.g. one for fast, rough positioning and one for slow, fine positioning. Joystick button 2 switches between these levels. The sign of the joystick position determines whether the move command issued contains the positive or negative travel limit (read automatically if Read travel range from controller? is TRUE) of the corresponding axis. When the joystick is "in the middle position", the velocity of the corresponding axis is set to zero.

Dead band \* is the maximum size of the scaled joystick position value that does not result in any motion.

Resolution factor \* determines the joystick resolution.

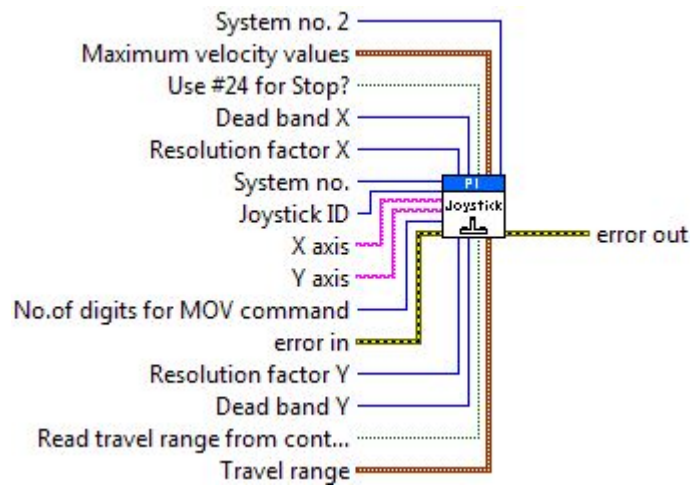
No. of digits is the number of digits after the decimal point in the position values that will be sent by the MOV command.

If joystick button 1 is pressed, a stop command (STP or #24, depending on Use 24 for stop) is sent to the controller. The diagram shows how to combine the driver and support VIs for these tasks.

If only one motion axis is to be controlled, Y Axis must be identical to X Axis, System no. 2 must be identical to System no. 1, and Y axis of joystick is to be used for control of the motion axis.

If required: Install the joystick driver and calibrate the joystick in the Windows control panel before running this VI.

Run PI\_GCS2\_Configuration\_Setup.vi prior to running this VI.

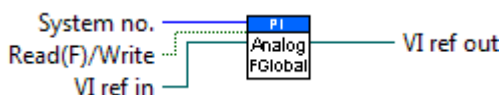


## 4. Low Level VIs

### 4.1. Analog control.Ilb (Analog Driver disabled)

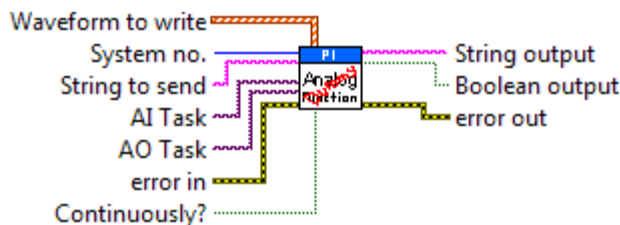
#### 4.1.1. Analog FGlobal.vi

This VI works as a functional global variable for VI references



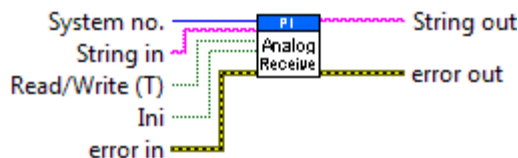
#### 4.1.2. Analog Functions.vi

This VI calls Analog Functions (dyn).vi functions dynamically during runtime, depending on String to send.



#### 4.1.3. Analog Receive String.vi

This VI works as an old style global variable for String out.



#### 4.1.4. Global Analog.vi

A global variable which contains setup information for analog systems.

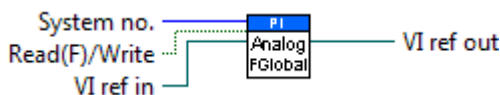


### 4.2. Analog control.Ilb (Analog Driver enabled)

#### 4.2.1. Analog FGlobal.vi

This VI works as a functional global variable for VI references.

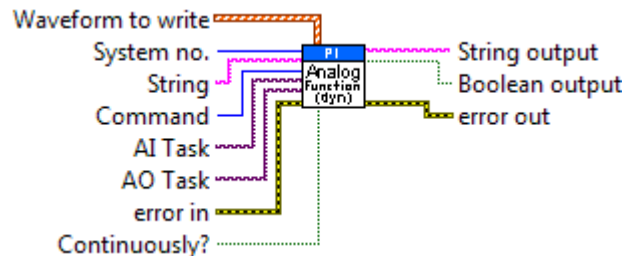
Valid for Analog Systems, but must be present for all other systems, too.



#### 4.2.2. Analog Functions (dyn).vi

This VI executes DAQmx functions and other tasks depending on String to send. This VI is called dynamically from Analog Functions.vi.

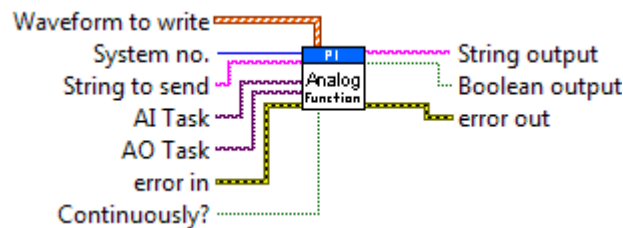
Valid for Analog Systems.



#### 4.2.3. Analog Functions.vi

This VI calls Analog Functions (dyn).vi functions dynamically during runtime, depending on String to send.

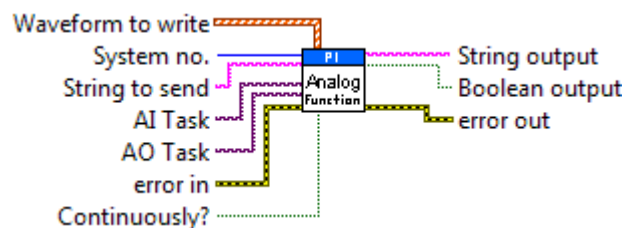
Valid for Analog Systems (but must be present for all other systems, too).



#### 4.2.4. Analog Functions\_old.vi

This VI calls Analog Functions (dyn).vi functions dynamically during runtime, depending on String to send.

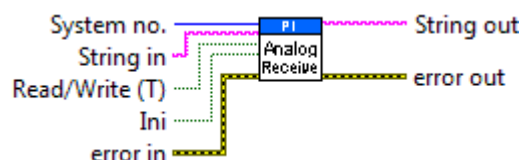
Valid for Analog Systems (but must be present for all other systems, too).



#### 4.2.5. Analog Receive String.vi

This VI works as an old style global variable for String out.

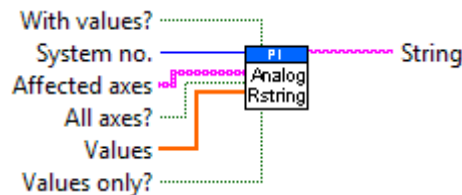
Valid for Analog Systems (but must be present for all other systems, too).



#### 4.2.6. Build analog reply string.vi

This VI builds a query command reply string by combining axes and values into a string following GCS 2.0 conventions. If With values? is FALSE, only axes are returned. For Values only? = TRUE, only values are returned without axes.

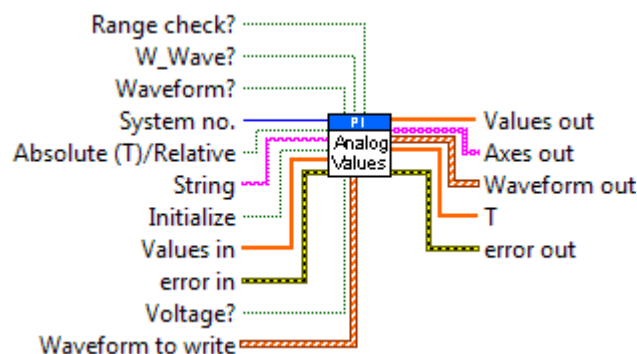
Valid for Analog Systems.



#### 4.2.7. Build analog value array.vi

This VI returns an array of numbers or waveforms and axes parsed from String, which must follow the GCS 2.0 conventions. If Initialize? is TRUE, the VI is initialized with Values in. If Absolute (T)/Relative is TRUE, the values in String are considered to be absolute position values, for FALSE they are considered relative to the current voltage/position. If Voltage? is FALSE, the conversion factor V/ $\mu$ m is read from GlobalAnalog.vi, the values in String are considered to be position values and Values out/Waveform out contains voltage information. If Voltage? is TRUE, the values in String are considered to be voltage values. T is the total time for Waveform out.

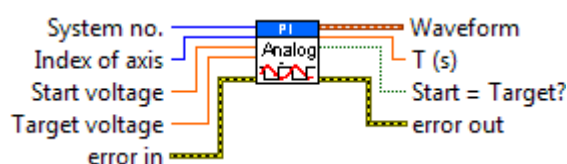
Valid for Analog systems.



#### 4.2.8. Build analog waveform.vi

This VI returns a waveform calculated by start and target voltage for the given axis.

Valid for Analog Systems.



#### 4.2.9. Get Terminal Name with Device Prefix.vi

Sample VI from NI which takes in a DAQmx Task and a terminal and converts this to a terminal which includes the device prefix specified in the DAQmx Physical Channel Control.

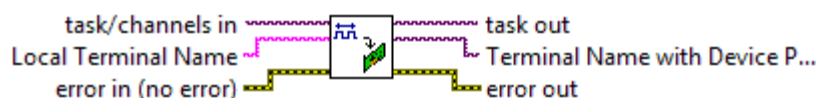
For example:

DAQmx Task/channels in=Dev1/ai1:2

Terminal=ai/StartTrigger

Device+Terminal=/Dev1/ai/StartTrigger

Valid for Analog Systems.



#### 4.2.10. Global Analog.vi

A global variable which contains setup information for analog systems.

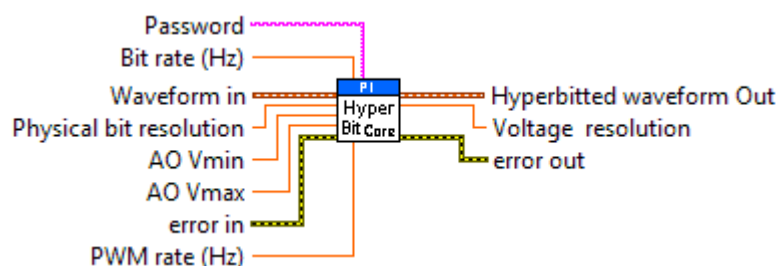
Valid for Analog Systems (but must be present for all other systems, too).



#### 4.2.11. HyperBit Core.vi

PI HyperBit Core VI. Calculates hyperbitted waveform for incoming waveform with given settings.

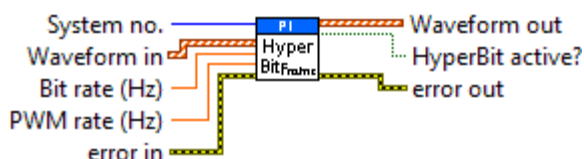
Valid for Analog Systems.



#### 4.2.12. HyperBit Frame.vi

PI HyperBit Frame VI. Calls HyperBit Core.vi with given settings and returns hyperbitted waveform.

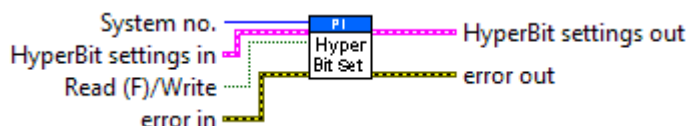
Valid for Analog Systems.





#### 4.2.13. HyperBit settings.vi

This VI sets or returns HyperBit settings, depending on Read (F)/Write.  
Valid for Analog Systems.



#### 4.2.14. Increase waveform by n values.vi

This VI increases a given waveform by N percent of the original length. The additional points are equal to the last point of the incoming waveform.  
Valid for Analog Systems



#### 4.2.15. Initialize Global Analog.vi

This VI initializes Global Analog according to the given System no.  
Valid for Analog Systems.



#### 4.2.16. Multiboard Trigger handling.vi

This VI checks if AI and AO use the same DAC device and if this is the case, it uses the DAQmx Start Trigger (Digital Edge).vi for triggering.

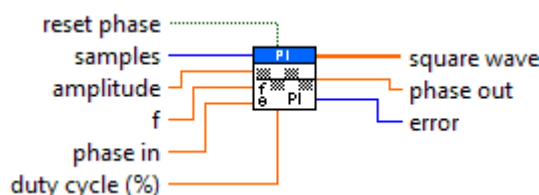
Triggering analog input and output channels by using the "ai/StartTrigger" terminal as the source for Start Trigger (Digital Edge).vi works only if AI and AO belong to the same DAC device. When Equal? returns FALSE, you can implement another trigger function in the FALSE frame of this VI. Please keep in mind that PCMCIA cards do not offer a RTSI line. One solution is to use digital lines for triggering. See DAQmx manual for details about triggering of DAC channels.

Valid for Analog Systems



#### 4.2.17. PI Square Wave.vi

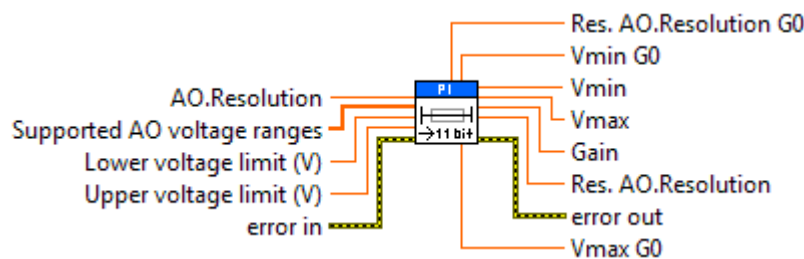
Generates an array specifying a square wave.



#### 4.2.18. Resulting resolution.vi

This VI calculates which analog output voltage range is best for the given Lower and Upper voltage limit and returns the appropriate offset and the resulting resolution value. It does also calculate range and resolution if gain must be zero.

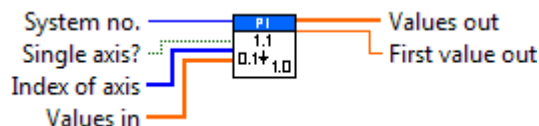
Valid for Analog Systems.



#### 4.2.19. Subtract analog offset.vi

Calculates an offset-corrected array of numerical values. If Single axis? is TRUE, Values in are considered to belong to one axis only and the correction factor of that axis is used. If Single axis? is FALSE, each value in Values in is considered to belong to another axis and must contain as many values as there are axes connected. VI also checks whether Values in are in allowed voltage range and if they are, it coerces Values out to be so also.

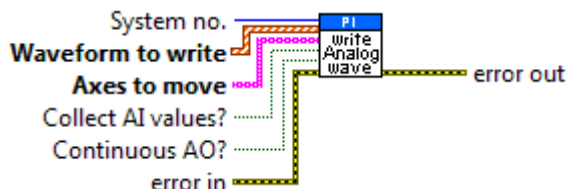
Valid for Analog Systems



#### 4.2.20. Write analog waveform.vi

This VI outputs the given waveform for Axes to move. If Collect AI values? is TRUE, waveform can only be outputted once and voltage values can be read back with DRR?.vi or DRR? and display data.vi. If Collect AI values? is FALSE, waveform can be selected to be outputted continuously.

Valid for Analog Systems.



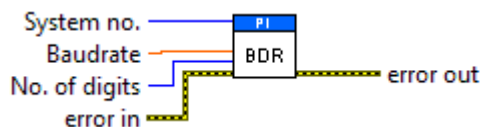
### 4.3. Communication.IIb

#### 4.3.1. BDR.vi

Set baud rate for RS-232 communication. Valid baud rates are 9.6, 19.2, 38.4, 57.6 and 115.2 (GCS 1.0) and 9600, 19200, 38400, 57600, 115200 (GCS 2.0). No. of digits is the number of digits after the decimal point in the baud rate specification that will be sent. VI waits 3 sec after sending BDR.

E-755: No. of digits must be 0. Baudrate is given in baud. Baud rate changes with BDR.vi are done in RAM only. After the baud rate was changed with BDR.vi, Close connection if open.vi with Close Daisychain? = TRUE must be called and communications must be re-established with the new baud rate (see instruction in E-755 user manual). Incorrect entries may have unpredictable results and may not set an error status. Use WPA.vi to write the current baud rate setting to non-volatile memory where it becomes the power-on default. Check RAM setting with BDR?.vi before running WPA.vi.

E-816: No. of digits must be 1. Baudrate is given in kbaud. Baudrate changes must be written to non-volatile memory with WPA.vi and do not take effect before the next power on or RST. Incorrect entries (such as 56) have unpredictable results and may not set an error status. Check RAM setting with BDR?.vi after setting the baudrate before running WPA.vi. This command cannot be issued to a slave E-816.



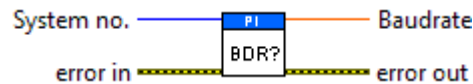
#### 4.3.2. BDR?.vi

Returns current RAM baud rate setting for RS-232 communication.

E-755: The value returned reflects the value that is currently in effect. This value will be saved to non-volatile memory if parameters are saved with WPA.vi. Returned baud rate value is baud.

E-816: The value returned reflects the value that will be saved to non-volatile memory if parameters are saved with WPA.vi. This may differ from the value currently in effect. Returned Baudrate value is in kbaud.

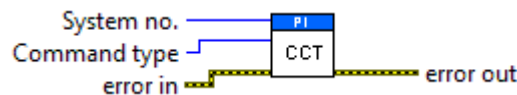
This command cannot be issued to a slave.



#### 4.3.3. CCT.vi

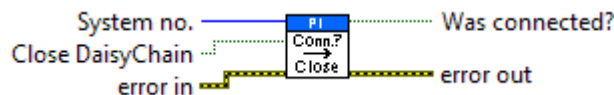
This VI sets the command type (e.g. ASCII or binary) of the controller.

E-761: Command type can be 1 (ASCII) or 2 (binary). Note: After power up the controller communicates via ASCII protocol.



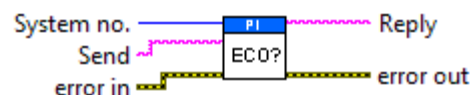
#### 4.3.4. Close connection if open.vi

This VI checks if the connection to the selected system is already open and, if it is, it closes this connection.



#### 4.3.5. ECO?.vi

Returns echo string. Reply should be equal to Send string.



#### 4.3.6. Find baudrate.vi

Opens COM port of given system with valid baud rates until status of Error out is false.

C-413: Input and output HW handshake must be TRUE. All other controls = default.

C-702: Input and output HW handshake must be TRUE. All other controls = default.

C-848: Input and output HW handshake must be TRUE. All other controls = default.

C-867: Input and output HW handshake must be FALSE. All other controls = default. The baud rate is set via the DIP switches on the controller front panel. See the controller user manual for details.

C-877: Input and output HW handshake must be FALSE. All other controls = default. Check controller user manual for supported baud rates.

C-880: Input and output HW handshake must be TRUE. All other controls = default.

C-880K005: All controls = default.

C-884: Input and output HW handshake must be TRUE. All other controls = default. The baud rate is set via IFC/IFS.vi. See the controller user manual for details.

C-891: Input and output HW handshake must be FALSE. All other controls = default.

E-135: Input and output HW handshake must be TRUE. All other controls = default.

E-516: Input and output HW handshake must be TRUE. All other controls = default.

E-517: Input and output HW handshake must be TRUE. Not available for Interface = GPIB, TCP/IP or DLL (USB). All other controls = default.

E-518: Input and output HW handshake must be TRUE. Not available for Interface = TCP/IP or DLL (USB). All other controls = default.

E-709: Input and output HW handshake must be TRUE. Not available for Interface = USB. All other controls = default.

E-712: Input and output HW handshake must be TRUE. Not available for Interface = TCP/IP or DLL. All other controls = default.

E-725, E-727: Input and output HW handshake must be TRUE. Not available for Interface = TCP/IP or DLL (USB). All other controls = default.

E-753: Input and output HW handshake must be TRUE. Not available for Interface = TCP/IP. All other controls = default.

E-754: Input and output HW handshake must be TRUE. Not available for Interface = TCP/IP or DLL. All other controls = default.

E-755: Input and output HW handshake must be TRUE. Not available for Interface = DLL and DLL Interface = RS232DC (DaisyChain). Interface clear = \18 (Use "\"Codes Display" to enter), String to Send = err?. All other controls = default.

E-816: Input and output HW handshake must be TRUE. All other controls = default.

E-861: Input and output HW handshake must be FALSE. All other controls = default. The baud rate is set via the DIP switches on the controller front panel. See the E-861 user manual for details.

E-871: Input and output HW handshake must be FALSE. All other controls = default. The baud rate is set via the DIP switches on the controller front panel. See the controller user manual for details.

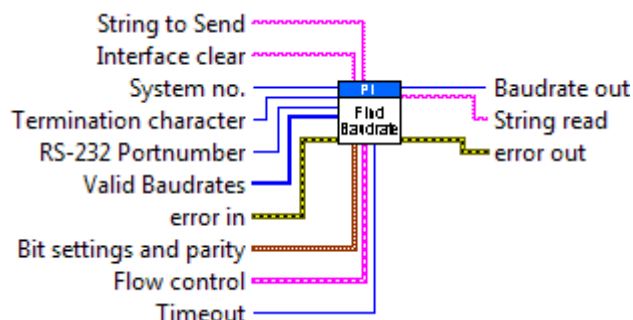
E-872: Input and output HW handshake must be FALSE. All other controls = default.

E-873: Input and output HW handshake must be FALSE. All other controls = default.

E-874: Input and output HW handshake must be FALSE. All other controls = default.

C-887, F-206, M-8X0: All controls = default.

Mercury\_GCS: Input and output HW handshake must be FALSE. All other controls = default. The baud rate is set via the DIP switches on the controller front panel. See the controller user manual for details.



#### 4.3.7. Find host address.vi

Performs an UDP broadcast and returns IP addresses of all controllers matching Controller name. VI will also stop if Stop refnum or Local stop is TRUE.

C-413: Controller names = C-413, Mode = Find XPort by UDP, Port = 30718

C-413: Controller names = C-413, Mode = Find USR-K3 by UDP, Port = 1901

C-702: Controller names = C-702, Mode = Find controller by UDP, Port = 50000

C-867: Controller names = C-867, Mode = Find controller by UDP, Port = 50000

C-884: Controller names = C-884, Mode = Find controller by UDP, Port = 50000

C-885: Controller names = C-885, Mode = Find controller by UDP, Port = 50000

C-886: Controller names = C-886, Mode = Find controller by UDP, Port = 50000

C-891: Controller names = C-891, Mode = Find XPort by UDP, Port = 30718

E-135: Controller names = E-135, Mode = Find USR-K3 by UDP, Port = 1901

E-517: Controller names = E-517, Mode = Find controller by UDP, Port = 50000

E-518: Controller names = E-518, Mode = Find controller by UDP, Port = 50000

E-712: Controller names = E-712, Mode = Find controller by UDP, Port = 50000

E-725: Controller names = E-725, Mode = Find XPort by UDP, Port = 30718

E-727: Controller names = E-727, Mode = Find controller by UDP, Port = 50000

E-753: Controller names = E-753, Mode = Find controller by UDP, Port = 50000

E-754: Controller names = E-754, Mode = Find controller by UDP, Port = 50000

E-871: Controller names = E-871, Mode = Find XPort by UDP, Port = 30718

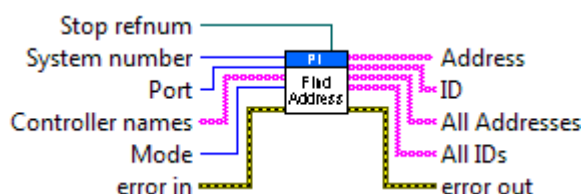
E-872: Controller names = E-872, Mode = Find controller by UDP, Port = 50000

E-873: Controller names = E-873, Mode = Find XPort by UDP, Port = 30718

E-874: Controller names = E-874, Mode = Find USR-K3 by UDP, Port = 1901

F-206: Controller names = F-206, F-HEX, Mode = Find XPort by UDP, Port = 30718

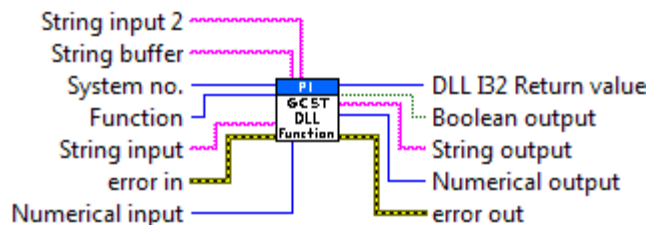
C-887, M-8X0: Controller names = M-8X0, HEXAPOD, Mode = Find XPort by UDP, Port = 30718



#### 4.3.8. GCSTranslator DLL Functions.vi

This VI calls a given function from GCSTranslator.dll. GCSTranslator.dll must be installed. To call a system-specific function, the system-specific GCS DLL must be installed also.

Warning: For XXX\_GcsGetANswer , String buffer must be large enough, otherwise the application may crash. Call XXX\_GcsGetANSwerSize first to determine necessary string length.



#### 4.3.9. Get subnet.vi

Calls system specific operating system function and returns subnet broadcast addresses of all installed network cards. Supports Windows (default case, calls IPCONFIG), Unix (calls IFCONFIG) and Mac.OS (calls IFCONFIG).



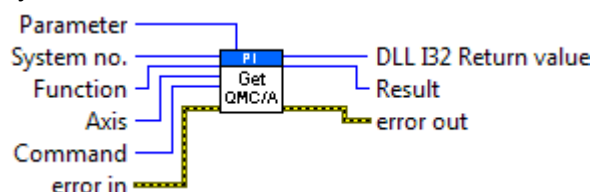
#### 4.3.10. GetGCSTranslatorDLLPath.vi

VI to create and store the path of the dynamic link library GCSTranslator.dll (Windows) for the different bitnesses as function variable.



#### 4.3.11. GetQMC\_GetQMCA.vi

This VI calls the GetQMC or GetQMCA function in the GCSTranslator.dll. GCSTranslator.dll and the corresponding system specific GCS DLL must be installed. The VI reads the system name from Global 2 (Array).vi and calls the corresponding DLL function. Parameter is only valid for GetQMCA. When using as a sub-VI, take special care to use the correct data representation when connecting values to the input terminals. If coercion dots appear at the terminals, the VI may not function correctly.



#### 4.3.12. Global DaisyChain.vi

Global DaisyChain is a global variable which contains setup information for DaisyChain systems.



#### 4.3.13. Global1.vi

A global variable which contains communication setup information.



#### 4.3.14. IFC.vi

Changes the current interface configuration. After IFC is sent, the new interface configuration of the controller becomes active and the host must change the interface configuration too. Settings made with IFC are valid until the controller is powered down.

To save them to EPROM and make them the default use IFS instead.

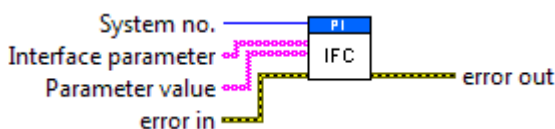
C-884: MACADR cannot be changed with IFC. To change IPADR the use of IFS is recommended instead.

C-885: MACADR cannot be changed with IFC. To change IPADR the use of IFS is recommended instead.

E-517: The current active RS-232 baud rate and GPIB (IEEE 488) address can also be set on the E-517 front panel in the COMMUNICATION display screen. This screen can be accessed by the trackball. The front panel settings interdepend with the interface parameter settings set with IFC. See E-517 user manual for details.

E-709: Use IFC to change baud rate settings only.

E-712, E-725, E-727, E-753, E-754, E-871, E-872, E-873, E-874: Use IFC to change baud rate settings only, for all other settings use IFS instead.

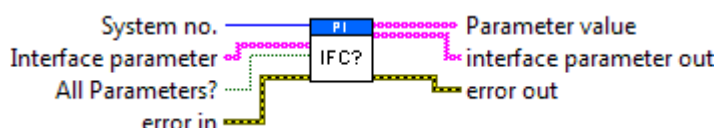


#### 4.3.15. IFC?.vi

Returns the current interface configuration.

E-517, E-518, E-753, E-754, E-871, E-872, E-873, E-874: Note that when the controller is part of a network with DHCP, the static IP address of the controller is returned, not the currently used IP address which was obtained from the DHCP server.

F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.3.16. IFS.vi

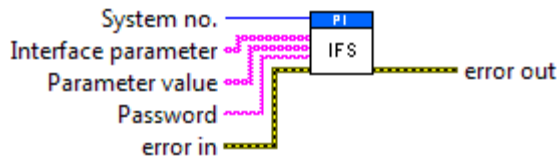
If Password is correct, the default parameter(s) for the interface are changed, but the current active parameters are not changed. Settings made with IFS are saved



to EPROM and become active with the next startup/reboot. To change settings immediately (but temporarily) use IFS instead (if supported by your controller).

E-709: Interface parameter can only be RSBAUD.

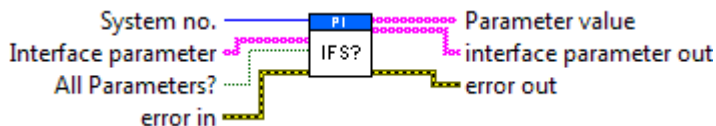
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.3.17. IFS?.vi

Returns the default interface configuration which is stored in EPROM.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.3.18. Initialize Global DaisyChain.vi

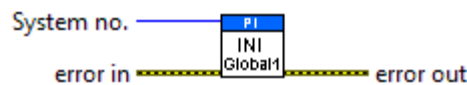
This VI initializes Global DaisyChain according to the given System no.

E-709: Only supported if E-709 is used inside C-867K012/K013.



#### 4.3.19. Initialize Global1.vi

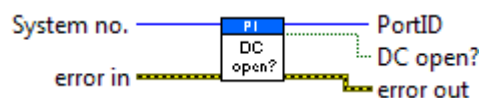
This VI initializes Global1 according to the given system no.



#### 4.3.20. Is DaisyChain open.vi

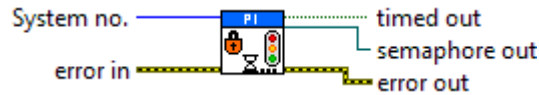
This VI checks if a DaisyChain connection is already open for the communication port defined for the given System no. It does also return the Port ID of the DaisyChain connection if any exists.

E-709: Only supported if E-709 is used inside C-867K012/K013.



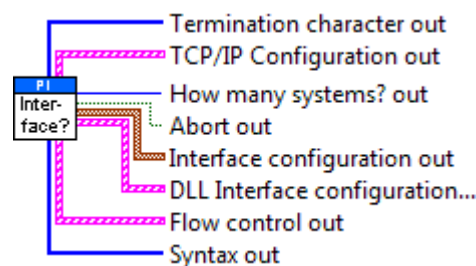
#### 4.3.21. PI acquire semaphore.vi

This VI acquires the semaphore that was defined in Global1.vi for the specified system during initialization.



#### 4.3.22. PI Ask for Communication Parameters.vi

A user-interface VI for setting up communications parameters (RS-232 or GPIB, number of systems, baudrate, timeout etc.) for up to 4 systems. Press F1 for displaying a help window with the appropriate interface configuration of each PI controller.



#### 4.3.23. PI Open Interface of one system.vi

Establishes communication with one connected system. This VI is called automatically by PI\_GCS2\_Configuration\_Setup.vi and must be completed successfully before any other VI can use the interface. The interface and error status of the chosen system are cleared by this VI, which sends XXX (no command), \*IDN? and ERR?.

C-413: Interface = RS232, DLL, or TCP/IP (TCP/IP support available depending on controller version! If unsure whether your controller supports TCP/IP communication, please refer to the controller user manual). RS232: Input and output HW handshake must be TRUE. DLL: DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

C-702: Interface = RS232 or TCP/IP, RS232: Input and output HW handshake must be TRUE, Syntax: GCS 1.0; Term char = LF.

C-843: Interface = DLL, DLL for Device = C-843, DLL Interface = Board, Parameter = Board number (1 for first C-843 board), Syntax: GCS 1.0; Term char = LF.

C-843.PM: Interface = DLL, DLL for Device = C-843.PM, DLL Interface = Board, Parameter = Board number (1 for first C-843 board), Syntax: GCS 1.0; Term char = LF.

C-844: Interface = DLL, DLL for Device = C-844, DLL Interface = RS232 or GPIB, Parameter = empty string, RS232 baud rate = 9600

C-865: Interface = DLL, DLL for Device = C-865, DLL Interface = RS232, Parameter = empty string, RS232 baud rate = set as appropriate, Syntax: GCS 1.0; Term char = LF.

C-866: Interface = DLL, DLL for Device = C-866, DLL Interface = RS232 or USB, RS232: Parameter = empty string, RS232 baud rate = set as appropriate, USB: Parameter = Serial no. of system to connect to,

Syntax: GCS 1.0; Term char = LF.

C-867: Single Device: Interface = RS232, TCP/IP or DLL, RS232: Input and output HW handshake must be FALSE. DLL (USB): DLL for Device = C-867, DLL Interface = USB, Parameter = Serial no. of system to connect to.

DaisyChain: Interface = DLL, DLL for Device = C-867, DLL Interface = RS232\_DC, Parameter = Number of device in chain, Register DC: FALSE. Syntax: GCS 2.0; Term char = LF.

C-877: Interface = RS232 or DLL, RS232: Input and output HW handshake must be FALSE. DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

C-880: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE, Syntax: GCS 1.0; Term char = LF.

C-848: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE, Syntax: GCS 1.0; Term char = LF.

C-880K005: Interface = RS232, Input and output HW handshake must be FALSE, Syntax: GCS 1.0; Term char = LF.

C-884: Interface = RS232, TCP/IP or DLL, RS232: Input and output HW handshake must be TRUE. DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

C-885: Interface = TCP/IP or DLL. DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF. DaisyChain: Interface = DLL, DLL for Device = PI\_GCS2\_DLL, DLL Interface = TCPIP\_DC or USB\_DC, Parameter = Number of device in chain, Register DC: FALSE. Syntax: GCS 2.0; Term char = LF.

C-886: Interface = TCP/IP or DLL. DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

C-891: Interface = RS232, TCP/IP or DLL. TCP/IP support available depending on controller version! If unsure whether your controller supports TCP/IP communication, please refer to your user manual. RS232: Input and output HW handshake must be FALSE. DLL: DLL for Device = C-891, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-135: Interface = RS232, DLL, or TCP/IP. RS232: Input and output HW handshake must be TRUE. DLL: DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-516: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE, Syntax: GCS 1.0; Term char = LF.

E-517: Interface = RS232, GPIB, TCP/IP or DLL, RS232: Input and output HW handshake must be TRUE, DLL (USB): DLL for Device = E-517, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-518: Interface = RS232, TCP/IP or DLL, RS232: Input and output HW handshake must be TRUE, DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = SB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-709: Interface = RS232 or USB, RS232: Input and output HW handshake must be TRUE, DLL: DLL for Device = E-709, DLL Interface = USB, Parameter = Serial no. of system to connect to.

Syntax: GCS 2.0; Term char = LF.

E-710: Interface = DLL, DLL for Device = E-710, DLL Interface = RS232 or GPIB, Parameter = empty string, Syntax: GCS 1.0; Term char = LF.

E-712: Interface = RS232, TCP/IP or DLL, RS232: Input and output HW handshake must be TRUE. DLL: DLL for Device = E-712, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-725: Interface = RS232, TCP/IP or DLL, RS232: Input and output HW handshake must be TRUE. DLL: DLL for Device = E-725, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-727: Interface = RS232, TCP/IP or DLL, RS232: Input and output HW handshake must be TRUE. DLL: DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-753: Interface = RS232 or TCP/IP, RS232: Input and output HW handshake must be TRUE, Syntax: GCS 2.0; Term char = LF.

E-754: Interface = RS232, TCP/IP or DLL, RS232: Input and output HW handshake must be TRUE. DLL: DLL for Device = E-754, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-755: Single Device: Interface = RS232, Input and output HW handshake must be TRUE.

DaisyChain: Interface = DLL, DLL for Device = E-755, DLL Interface = RS232\_DC, Parameter = Number of device in chain (first device: 1), Register DC: FALSE.

Syntax: GCS 2.0; Term char = LF.

E-761: Interface = DLL, DLL for Device = E-761, DLL Interface = Board, Parameter = Board number (1 for first E-761 board), Syntax: GCS 1.0; Term char = LF.

E-816: Interface = RS232 or DLL, RS232: Input and output HW handshake must be TRUE. DLL (USB): DLL for Device = E-816, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 1.0; Term char = LF.

E-861: Single Device: Interface = RS232 or DLL, RS232: Input and output HW handshake must be FALSE. DLL (USB): DLL for Device = E-861, DLL Interface = USB, Parameter = Serial no. of system to connect to.

DaisyChain: Interface = DLL, DLL for Device = E-861, DLL Interface = RS232\_DC or USB\_DC, Parameter = Number of device in chain, Register DC: FALSE.

Syntax: GCS 2.0; Term char = LF.

E-870: Interface = DLL, DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-871: Single Device: Interface = RS232 or DLL, RS232: Input and output HW handshake must be FALSE. DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to.

DaisyChain: Interface = DLL, DLL for Device = PI\_GCS2\_DLL, DLL Interface = RS232\_DC or USB\_DC, Parameter = Number of device in chain, Register DC: FALSE. Syntax: GCS 2.0; Term char = LF.

E-872: Interface = RS232, TCP/IP or DLL, RS232: Input and output HW handshake must be FALSE. DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-873: Interface = RS232, TCP/IP or DLL, RS232: Input and output HW handshake must be FALSE. DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

E-874: Interface = RS232, TCP/IP or DLL, RS232: Input and output HW handshake must be FALSE. DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to. Syntax: GCS 2.0; Term char = LF.

F-206: F-206 (GCS 1.0): Interface = RS232, GPIB or TCP/IP, The error status will not be cleared by this VI. The first ERR? query will report a hidden error with error code 1, which will be cleared during system initialization (INI). RS232: Input and output handshake settings must be FALSE, Syntax: GCS 1.0; Term char = LF.

C-887 + H-206 (GCS 2.0): Interface = RS232 or TCP/IP, RS232: Input and output handshake settings must be FALSE, Syntax: GCS 2.0; Term char = LF.

Hydra: Interface = DLL, DLL (TCP/IP and RS-232): DLL for Device = PI\_HydraPollux\_GCS2\_DLL, DLL Interface = RS232 or TCP/IP, Syntax: GCS 2.0; Term char = LF.

M-8X0: M-810/11/24/40/50 (GCS 1.0): Interface = RS232, GPIB or TCP/IP, RS232: Input and output handshake settings must be FALSE, Syntax: GCS 1.0; Term char = LF.

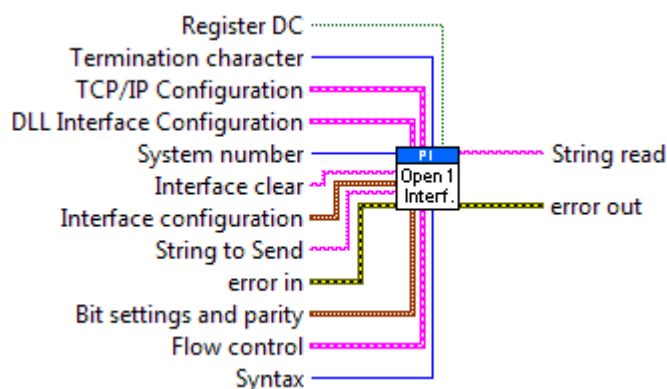
C-887 + H-810/11/24/40/50 (GCS 2.0): Interface = RS232 or TCP/IP, RS232: Input and output handshake settings must be FALSE, Syntax: GCS 2.0; Term char = LF.

Mercury: Interface = DLL, DLL for Device = Mercury, DLL Interface = RS232 (even if using USB), Parameter = empty string, RS232 baud rate = same as controller hardware setting (even if using USB), Syntax: GCS 1.0; Term char = LF.

Mercury\_GCS: Single Device: Interface = RS232 or DLL, RS232: Input and output HW handshake must be FALSE. DLL (USB): DLL for Device = PI\_GCS2\_DLL, DLL Interface = USB, Parameter = Serial no. of system to connect to.

DaisyChain: Interface = DLL, DLL for Device = PI\_GCS2\_DLL, DLL Interface = RS232\_DC or USB\_DC, Parameter = Number of device in chain, Register DC: FALSE. Syntax: GCS 2.0; Term char = LF.

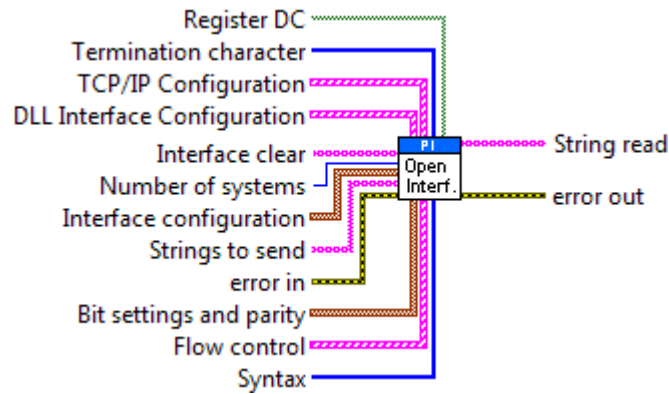
Pollux: Interface = DLL, DLL (RS-232): DLL for Device = PI\_HydraPollux\_GCS2\_DLL, DLL Interface = RS232, Baudrate must be 19200. Syntax: GCS 2.0; Term char = LF.



#### 4.3.24. PI Open Interface.vi

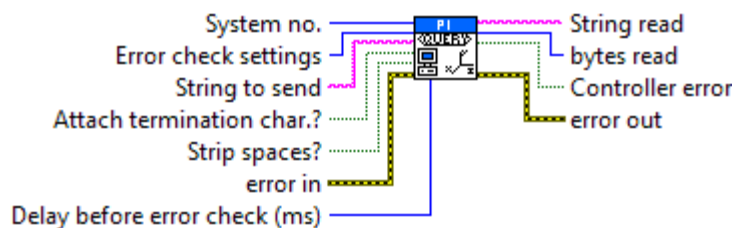
Establishes communication with the connected systems. The interface and error statuses of all connected systems are cleared by this VI, which sends XXX (no command), \*IDN? and ERR?.

See PI Open Interface of one system.vi for control settings.



#### 4.3.25. PI Query.vi

This VI acquires the semaphore for the corresponding system via PI acquire semaphore.vi, sends a string w/o termination character via the selected interface, receives the answer and releases the semaphore via PI release semaphore.vi



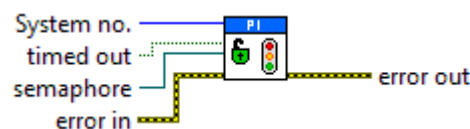
#### 4.3.26. PI Receive String.vi

Read string from selected system.



#### 4.3.27. PI release semaphore.vi

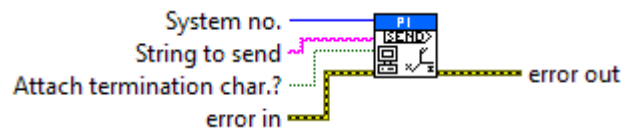
This VI releases the semaphore that was defined in Global1.vi for the specified system.





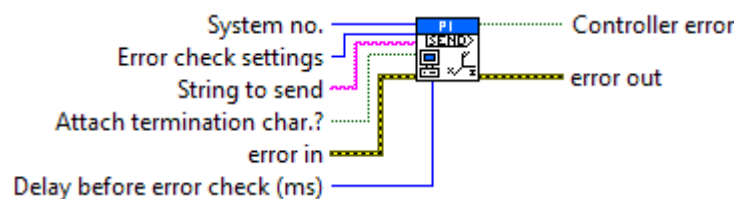
#### 4.3.28. PI Send String.vi

Sends command with or without trailing termination character to selected system.



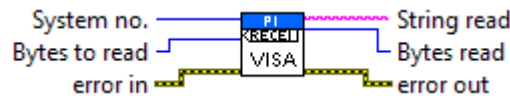
#### 4.3.29. PI Send.vi

This VI acquires the semaphore for the corresponding system via PI acquire semaphore.vi, sends a string w/o termination character via the selected interface and releases the semaphore via PI release semaphore.vi



#### 4.3.30. PI VISA Receive Characters.vi

This VI reads n bytes (characters) via the chosen VISA interface. Sub-vi for PI Receive String.vi.



#### 4.3.31. Select DaisyChain device.vi

Performs a broadcast, returns names of all controllers matching Controller name and lets the user select the appropriate controller from a ring control. VI will also stop if Cancel is TRUE.

C-867: Controller name = "C-867", DLL Interface = RS232\_DC

C-885: Controller name = "C-885", DLL Interface = USB\_DC or TCPIP\_DC

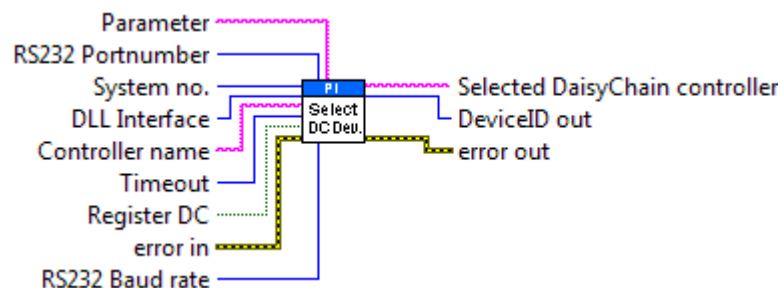
E-709: Controller name = "E-709", DLL Interface = RS232\_DC. Only supported if E-709 is used inside C-867K012/K013.

E-755: Controller name = "E-755", DLL Interface = RS232\_DC

E-861: Controller name = "E-861", DLL Interface = RS232\_DC or USB\_DC

E-871: Controller name = "E-871", DLL Interface = RS232\_DC or USB\_DC

Mercury\_GCS: Controller name = "Mercury\_GCS", DLL Interface = RS232\_DC or USB\_DC



#### 4.3.32. Select host address.vi

Performs an UDP broadcast, returns IP addresses and names of all controllers matching Controller names and lets the user select the appropriate controller from a ring control. VI will also stop if Cancel is TRUE.

C-413: Controller names = C-413, Mode = Find XPort by UDP, Port = 30718

C-413: Controller names = C-413, Mode = Find USR-K3 by UDP, Port = 1901

C-702: Controller names = C-702, Mode = Find controller by UDP, Port = 50000

C-867: Controller names = C-867, Mode = Find controller by UDP, Port = 50000

C-884: Controller names = C-884, Mode = Find controller by UDP, Port = 50000

C-885: Controller names = C-885, Mode = Find controller by UDP, Port = 50000

C-886: Controller names = C-886, Mode = Find controller by UDP, Port = 50000

C-891: Controller names = C-891, Mode = Find XPort by UDP, Port = 30718

E-135: Controller names = E-135, Mode = Find USR-K3 by UDP, Port = 1901

E-517: Controller names = E-517, Mode = Find controller by UDP, Port = 50000

E-518: Controller names = E-518, Mode = Find controller by UDP, Port = 50000

E-712: Controller names = E-712, Mode = Find controller by UDP, Port = 50000

E-725: Controller names = E-725, Mode = Find XPort by UDP, Port = 30718

E-725: Controller names = E-725, Mode = Find controller by UDP, Port = 50000

E-753: Controller names = E-753, Mode = Find controller by UDP, Port = 50000

E-754: Controller names = E-754, Mode = Find controller by UDP, Port = 50000

E-871: Controller names = E-871, Mode = Find XPort by UDP, Port = 30718

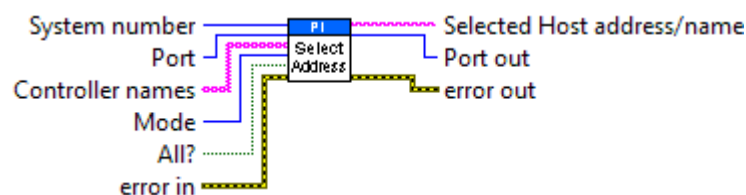
E-872: Controller names = E-872, Mode = Find controller by UDP, Port = 50000

E-873: Controller names = E-873, Mode = Find XPort by UDP, Port = 30718

E-874: Controller names = E-874, Mode = Find USR-K3 by UDP, Port = 1901

F-206: Controller names = F-206, F-HEX, Mode = Find XPort by UDP, Port = 30718

C-887, M-8X0: Controller names = M-8X0, HEXAPOD, Mode = Find XPort by UDP, Port = 30718



#### 4.3.33. Select USB device.vi

Performs a broadcast, returns names of all controllers matching Controller name and lets the user select the appropriate controller from a ring control. VI will also stop if Cancel is TRUE.

C-413: Controller name = "C-413", DLL Interface = USB

C-866: Controller name = "C-866", DLL Interface = USB

C-867: Controller name = "C-867", DLL Interface = USB

C-877: Controller name = "C-877", DLL Interface = USB

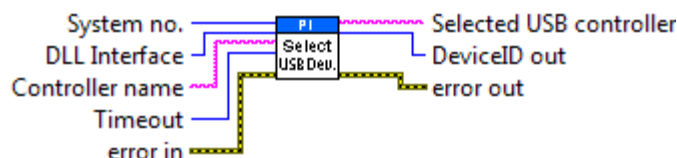
C-884: Controller name = "C-884", DLL Interface = USB

C-885: Controller name = "C-885", DLL Interface = USB

C-886: Controller name = "C-886", DLL Interface = USB

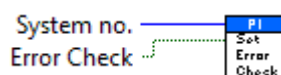


C-891: Controller name = "C-891", DLL Interface = USB  
 E-135: Controller name = "E-135", DLL Interface = USB  
 E-517: Controller name = "E-517", DLL Interface = USB  
 E-518: Controller name = E-518 , DLL Interface = USB  
 E-709: Controller name = E-709 , DLL Interface = USB  
 E-712: Controller name = E-712 , DLL Interface = USB  
 E-725: Controller name = E-725 , DLL Interface = USB  
 E-727: Controller name = E-727 , DLL Interface = USB  
 E-754: Controller name = E-754 , DLL Interface = USB  
 E-816: Controller name = "E-816", DLL Interface = USB  
 E-861: Controller name = "E-861", DLL Interface = USB  
 E-870: Controller name = "E-870", DLL Interface = USB  
 E-871: Controller name = "E-871", DLL Interface = USB  
 E-872: Controller name = "E-872", DLL Interface = USB  
 E-873: Controller name = "E-873", DLL Interface = USB  
 E-874: Controller name = "E-874", DLL Interface = USB  
 Mercury\_GCS: Controller name = "Mercury\_GCS", DLL Interface = USB



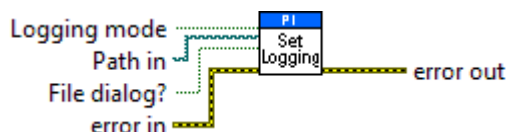
#### 4.3.34. Set Error Check.vi

This VI enables the error check in Global1.vi for the selected System no.



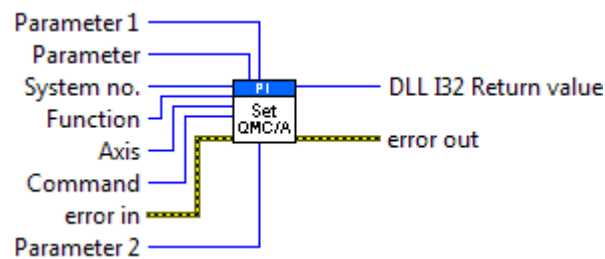
#### 4.3.35. Set logging mode.vi

Sets logging mode for all communication interfaces. When Logging mode is ON, each string sent to or received from the controller is written to a .txt file for debugging. When File dialog is TRUE, a dialog box will pop up where the file to write can be selected, otherwise Path in must contain a valid path to a .txt file. Depending on the call chain of Set logging mode.vi, the VI will either stop (correct behavior when called from another VI) or it will remain idle (correct behavior when command VIs from this driver set are to be run manually, i.e. non-programmatically). In the latter case do not forget to press the STOP button when you have finished working with the command VIs.



#### 4.3.36. SetQMC\_SetQMCA.vi

This VI calls the SetQMC or SetQMCA function in the GCSTranslator.dll. GCSTranslator.dll and the corresponding system specific GCS DLL must be installed. The VI reads the system name from Global 2 and calls the corresponding DLL function. Parameter is only valid for SetQMC, Parameter 1 and Parameter 2 are only valid for SetQMCA. When using as a sub-VI, take special care to use the correct data representation when connecting values to the input terminals. If coercion dots appear at the terminals, the VI may not function correctly. After changing values, the update function must be called to activate the new values. See user manual for details.

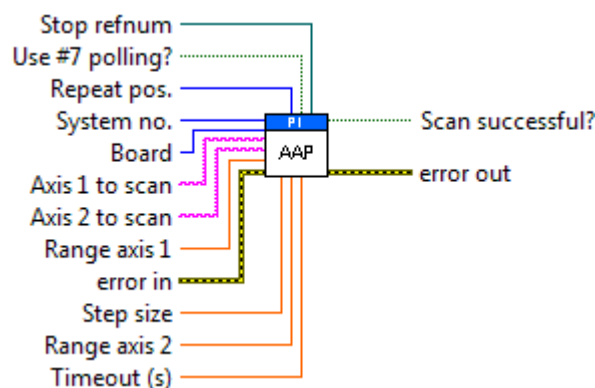


### 4.4. Controller Algorithms.llb

#### 4.4.1. AAP.vi

Performs a fast automated alignment in specified axes, waits until scan is finished using #7 polling (if checked) and indicates whether scan was successful or not. Timeout is only valid if Use #7 polling is FALSE. If Refnum stop or Local stop is TRUE, VI sends #24 and stops.

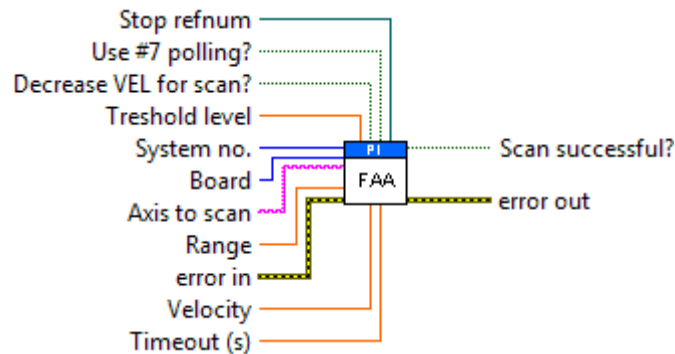
C-887, F-206, M-8X0: Use #7 polling = FALSE, Range axis 2 must be identical with Range axis 1 (no rectangular scan implemented). For GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), VI calls FSS? automatically to determine if scan was successful or not.



#### 4.4.2. FAA.vi

Performs a fast angular line scan to maximum in specified axis, waits until scan is ready (using #7 polling if checked) and indicates whether scan was successful or not. Scan starts at (Current position - ½ Range) and stops at (Current position + ½ Range). "Velocity" is only valid if "Decrease VEL for scan?" is TRUE. If "Decrease VEL for scan?" is TRUE, velocity is decreased before and reset after the scan.

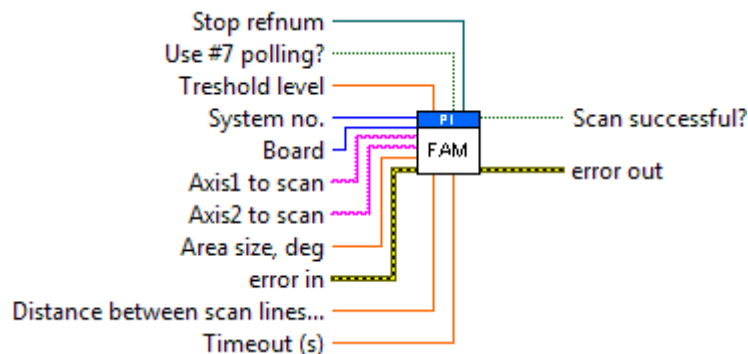
"Timeout" is only valid if "Use #7 polling" is FALSE. If "Refnum stop" or "Local stop" is TRUE, VI resets velocity if it was decreased before, sends #24 and stops.  
C-887, F-206: Only for GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For compatibility reasons, VI calls FLM automatically for GCS 2.0.



#### 4.4.3. FAM.vi

Performs a fast angular scan to maximum in specified axes, waits until scan is finished (using #7 polling if checked) and indicates whether scan was successful or not. "Timeout" is only valid if "Use #7 polling" is FALSE. If Refnum stop or Local stop is TRUE, VI sends #24 and stops.

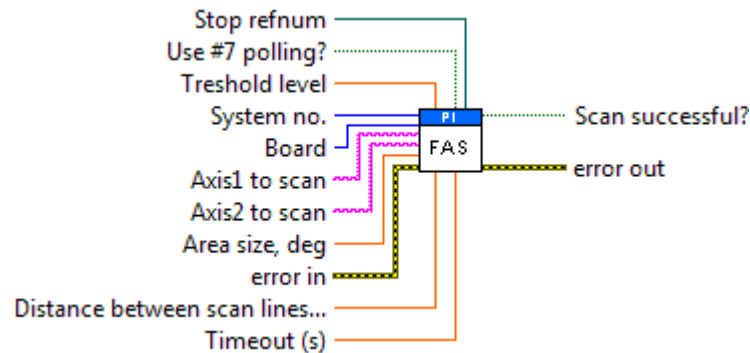
C-887, F-206: Use #7 polling = FALSE. Only for GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For compatibility reasons, VI calls FSM automatically for GCS 2.0.



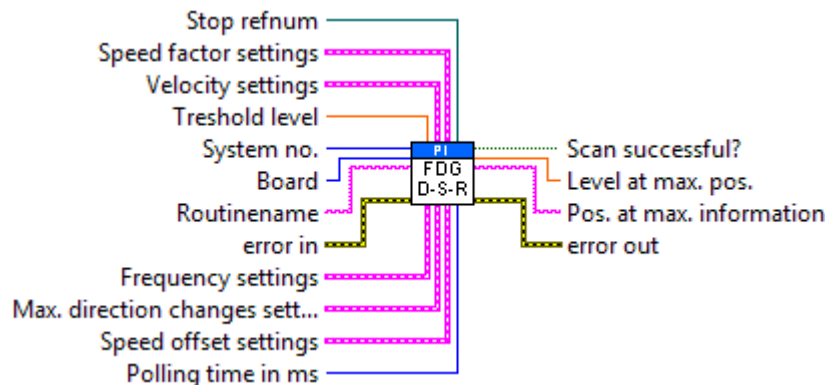
**4.4.4. FAS.vi**

Performs a fast angular scan in specified axes, waits until scan is finished (using #7 polling if checked) and indicates whether scan was successful or not. "Timeout" is only valid if "Use #7 polling" is FALSE. If Refnum stop or Local stop is TRUE, VI sends #24 and stops.

C-887, F-206: Use #7 polling = FALSE. Only for GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For compatibility reasons, VI calls FSC automatically for GCS 2.0.

**4.4.5. FDG Def\_Start\_Results.vi**

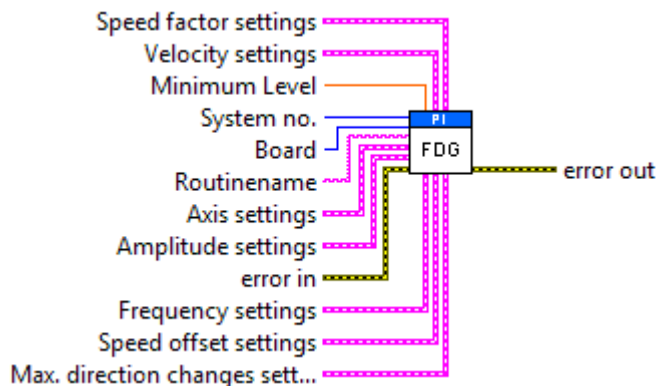
Defines a fast alignment gradient search routine, starts search, waits until search has finished and queries results. If Stop refnum or Local stop is TRUE, VI sends #24 and stops.



#### 4.4.6. FDG.vi

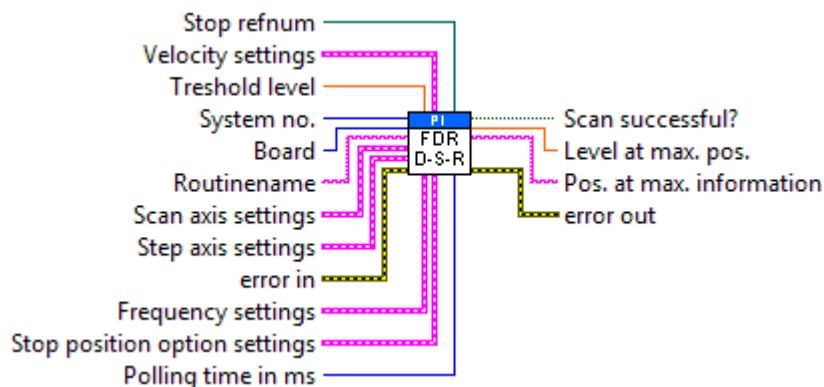
Defines a fast alignment gradient search routine. Use FRS to start the routine and FRR? to read out the results of the routine.

C-887, E-712:



#### 4.4.7. FDR Def\_Start\_Results.vi

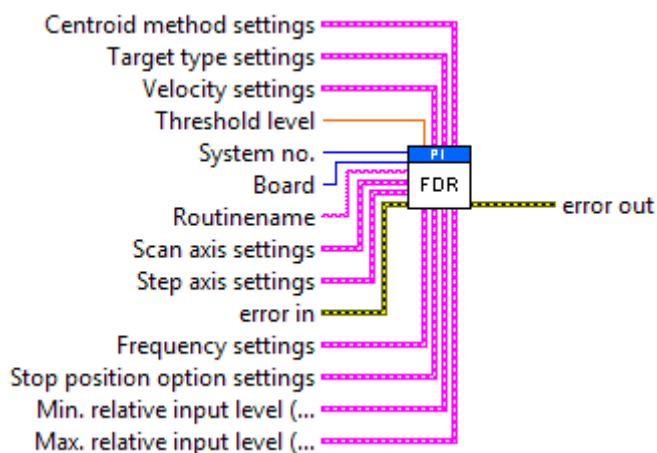
Defines a fast alignment area scan routine, starts scan, waits until scan has finished and queries results. If Stop refnum or Local stop is TRUE, VI sends #24 and stops.



#### 4.4.8. FDR.vi

Defines a fast alignment area scan routine. Use FRS to start the routine and FRR? to read out the results of the routine.

C-887, E-712, E-727.AS:

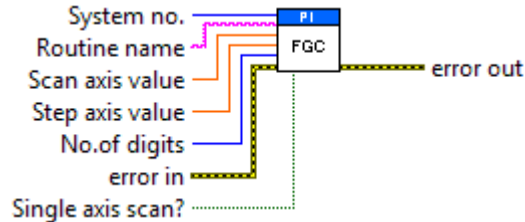


#### 4.4.9. FGC.vi

Sets fast alignment gradient search dither center:

If only one axis is used for the routine, "Step Axis Value" will be set identical to "Scan Axis Value".

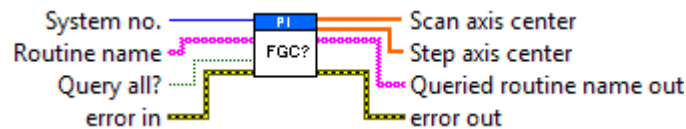
C-887, E-712:



#### 4.4.10. FGC?.vi

Query fast alignment gradient search dither center values

C-887, E-712:

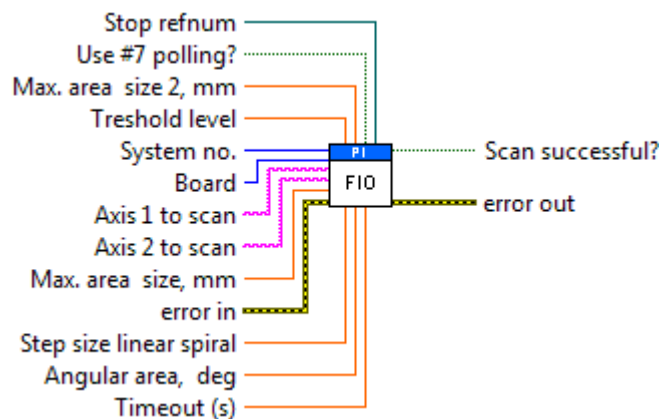


#### 4.4.11. FIO.vi

Performs a fast input/output automated alignment procedure in specified axes, waits until scan is ready (using #7 polling if checked) and indicates whether scan was successful or not. If Refnum stop or Local stop is TRUE, VI sends #24 and stops.

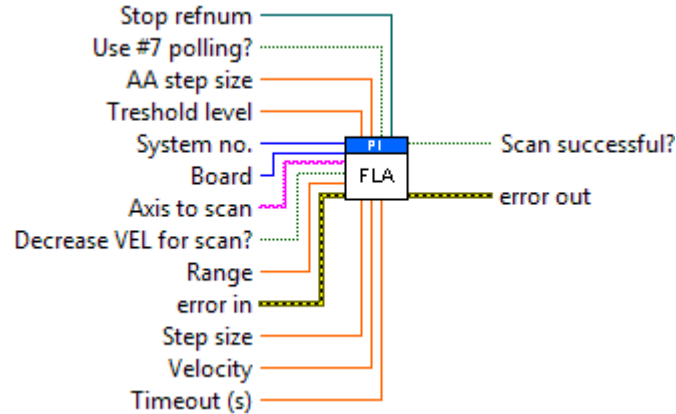
"Timeout" is only valid if "Use #7 polling" is FALSE.

C-887, F-206, M-8X0: : For GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), VI calls FSS? automatically to determine if scan was successful or not. Use #7 polling = FALSE. Max. area size 2 must be identical to Max. area size.



#### 4.4.12. FLA.vi

Performs a line scan and align in specified axis, waits until scan is ready using #7 polling and indicates whether scan was successful or not. If Refnum stop or Local stop is TRUE, VI sends #24 and stops.

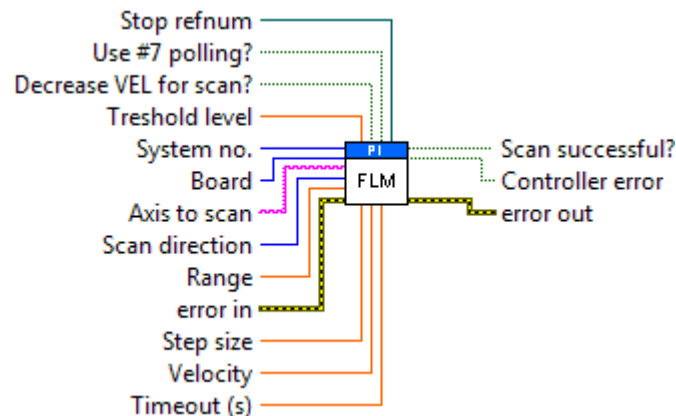


#### 4.4.13. FLM.vi

Performs a line scan to maximum in specified axis, waits until scan is ready (using #7 polling if checked) and indicates whether scan was successful or not. Velocity is only valid if Decrease VEL for scan? Is TRUE. If Decrease VEL for scan? is TRUE, velocity is decreased before and reset after the scan. Timeout is only valid if Use #7 polling is FALSE. If Refnum stop or Local stop is TRUE, VI sends #24 and stops. When using as a sub-VI, use Refnum stop to stop VI from caller.

C-880: Use #7 polling? = TRUE

C-887, F-206, M-8X0: Use #7 polling? = FALSE. Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). Step size is not valid.

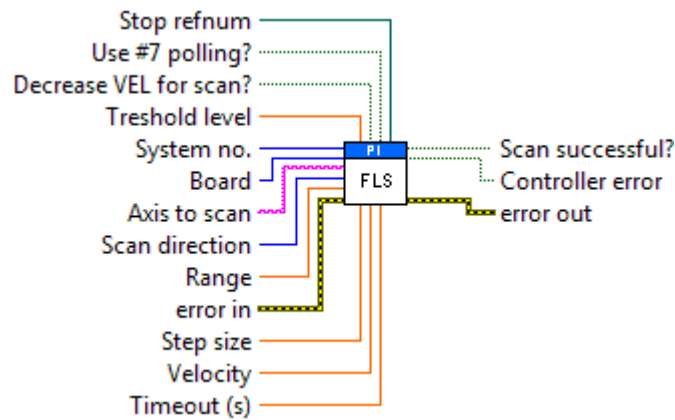


#### 4.4.14. FLS.vi

Performs a line scan specified axis, waits until scan is ready (using #7 polling if checked) and indicates whether scan was successful or not. Velocity is only valid if Decrease VEL for scan? Is TRUE. If Decrease VEL for scan? is TRUE, velocity is decreased before and reset after the scan. Timeout is only valid if Use #7 polling is FALSE. If Refnum stop or Local stop is TRUE, VI sends #24 and stops. When using as a sub-VI, use Refnum stop to stop VI from caller.

C-880: Use #7 polling? = TRUE

C-887, F-206, M-8X0: Use #7 polling? = FALSE. Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). Step size is not valid.

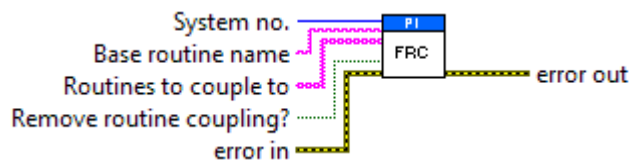


#### 4.4.15. FRC.vi

Set fast alignment routine coupling:

If coupling should be removed, "Routines to couple to" has to be 0 or "Remove routine coupling" has to be TRUE.

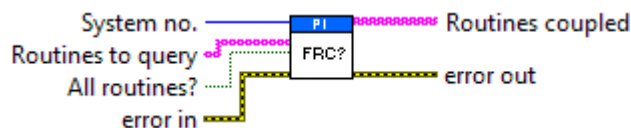
C-887, E-712:



#### 4.4.16. FRC?.vi

Returns fast alignment routine coupling.

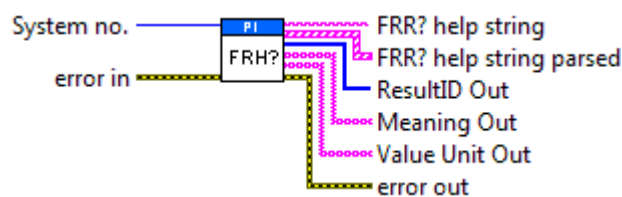
C-887, E-712:



#### 4.4.17. FRH?.vi

The command returns an ASCII help string with available query options ("result IDs") for the FRR? command.

C-887, E-712, E-727.AS:





#### 4.4.18. FRP.vi

Stops, pauses or resumes the given fast alignment routine(s).

C-887, E-727.AS: Option can be

0 - stop given routine

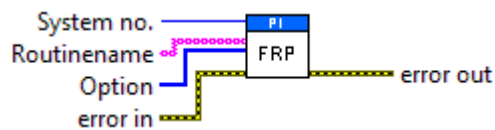
E-712: Option can be

0 - stop given routine

1 - pause given routine

2 - resume given routine

C-887, E-712, E-727.AS:



#### 4.4.19. FRP?.vi

Returns current status of given fast alignment routine(s).

C-887, E-727.AS: Status can be

0 - given routinehas been stopped / is not running

2 - given routineis running

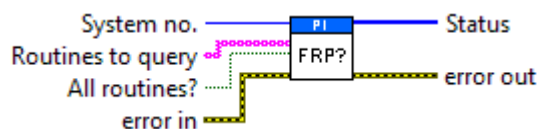
E-712: Status can be

0 - given routinehas been stopped / is not running

1 - given routinehas been paused

2 - given routineis running

C-887, E-712, E-727.AS:

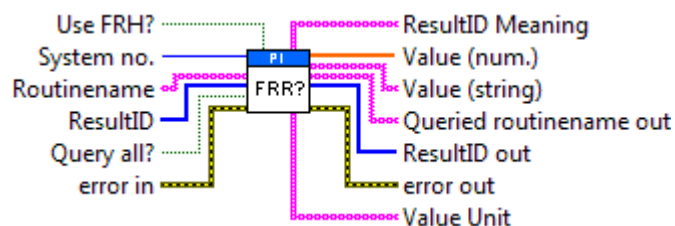


#### 4.4.20. FRR?.vi

Get results of given fast alignment routine. If Use FRH? is TRUE, VI queries FRH? and returns meaning of ResultIDs and units of Values. Requires FRH? and Parse\_FRH\_Type\_String.vi.

E-712, E-727.AS: Query all? must be FALSE, Use FRH? can be TRUE.

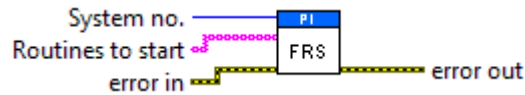
C-887, E-712, E-727.AS:



#### 4.4.21. FRS.vi

Starts the given fast alignment routine (one or more). Routine must be predefined, e.g. with FDR or FDG.

C-887, E-712, E-727.AS:

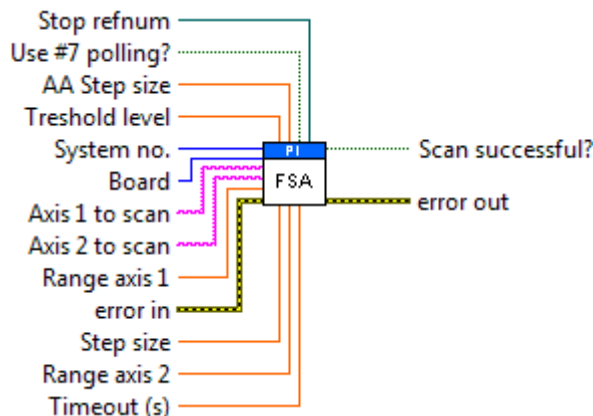


#### 4.4.22. FSA.vi

Performs a 2D scan and align in specified axes, waits until scan is ready using #7 polling (if checked) and indicates whether scan was successful or not. If Stop refnum or Local stop is TRUE, VI sends #24 and stops. For a square scan, Range axis 2 must be identical to Range axis 1, otherwise a rectangular scan is performed. Timeout is only valid if Use #7 polling is FALSE.

C-880: Use #7 polling = TRUE

C-887, F-206, M-8X0: Use #7 polling = FALSE, Range axis 2 must be identical with Range axis 1 (no rectangular scan implemented). For GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), VI calls FSS? automatically to determine if scan was successful or not.

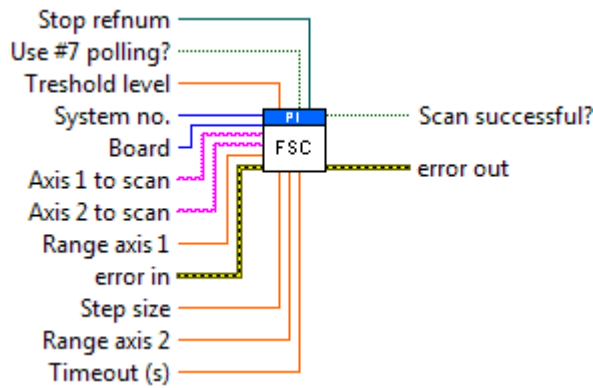


#### 4.4.23. FSC.vi

Performs a 2D scan in specified axes, waits until scan is ready using #7 polling (if checked) and indicates whether scan was successful or not. If Stop refnum or Local stop is TRUE, VI sends #24 and stops. For a square scan, Range axis 2 must be identical to Range axis 1, otherwise a rectangular scan is performed. Timeout is only valid if Use #7 polling is FALSE.

C-880: Use #7 polling = TRUE

C-887, F-206, M-8X0: Use #7 polling = FALSE, Range axis 2 must be identical with Range axis 1 (no rectangular scan implemented). For GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), VI calls FSS? automatically to determine if scan was successful or not.

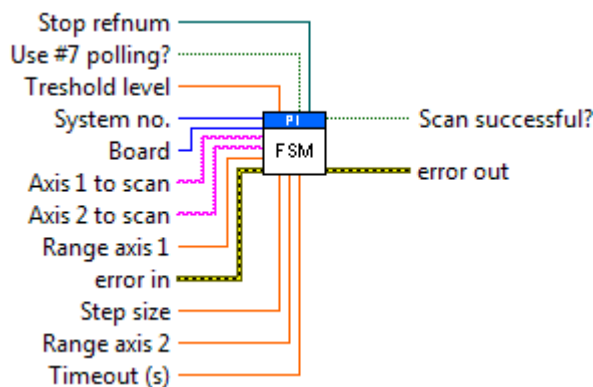


#### 4.4.24. FSM.vi

Performs a 2D scan to maximum in specified axes, waits until scan is ready using #7 polling (if checked) and indicates whether scan was successful or not. If Stop refnum or Local stop is TRUE, VI sends #24 and stops. For a square scan, Range axis 2 must be identical to Range axis 1, otherwise a rectangular scan is performed. Timeout is only valid if Use #7 polling is FALSE.

C-880: Use #7 polling = TRUE

C-887, F-206, M-8X0: Use #7 polling = FALSE, Range axis 2 must be identical with Range axis 1 (no rectangular scan implemented). For GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), VI calls FSS? automatically to determine if scan was successful or not.

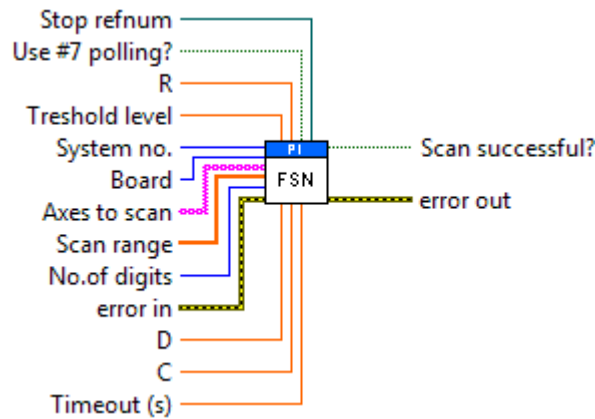


#### 4.4.25. FSN.vi

Performs a 1D scan following a trajectory described by the given parameters, waits until scan is ready (using #7 polling if checked) and indicates whether scan was successful or not. Timeout is only valid if Use #7 polling is FALSE. If Refnum stop or Local stop is TRUE, VI sends #24 and stops. See system manual for further details.

F-206: Use #7 polling = FALSE, Axes to scan can be a subset of (X,Y,Z,U,V,W). Scan cannot be stopped. Only for GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

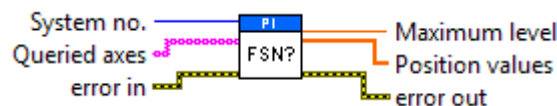
C-887, M-8X0: Use #7 polling = FALSE, Axes to scan can be a subset of (X,Y,Z,U,V,W). Scan cannot be stopped. Only for GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.4.26. FSN?.vi

Returns the maximum level found during the last FSN scan and the coordinates where it was found.

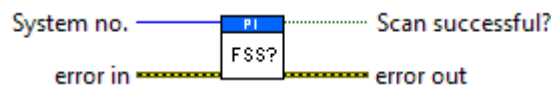
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.4.27. FSS?.vi

Returns the result of the last scan algorithm.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.4.28. MVS.vi

Move given axes in Steps relative to their current position. The new target position is calculated by adding "Position values" to the last commanded target values. The movement is performed in steps with the size of "Step size". After each step a delay of "Delay time (ms)" (in milliseconds) takes place.

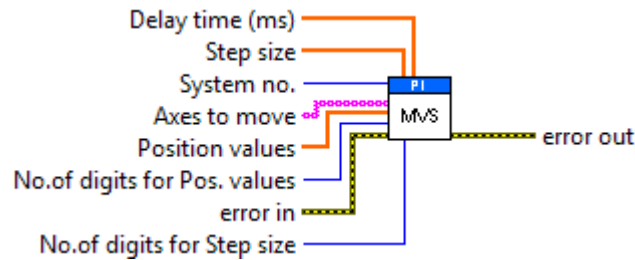
During the execution of the command, the controller is in "not ready" state, which can be checked by the #7 command.

This command can be interrupted by #24, HLT and STP.

Servo must be enabled prior to using this command.

Note: The Step size is automatically adapted so that the resulting number of steps is an integer value.

No. of digits is the number of digits after the decimal point in the position and step size value(s) that will be sent.

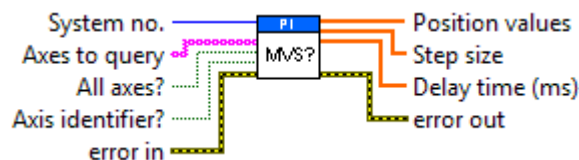


#### 4.4.29. MVS?.vi

Returns last valid configuration for relative step move (caused by MVS) for the given axis.

Note: The reported Step size may not be the last commanded value because during MVS the step size was automatically adapted so that the resulting number of steps is an integer value.

C-702: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.4.30. SIC.vi

Set fast alignment input calculation.

Calculation type:

0 = Calculation OFF

1 = Exponential Calculation

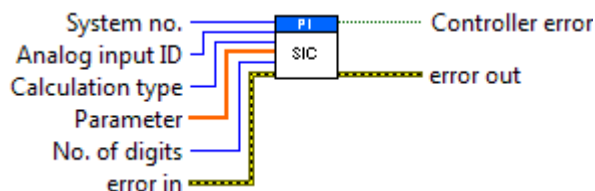
Calculated Input Signal = Parameter0 + Parameter1 \*  
Parameter2^(Parameter3\*Voltage)

2 = Polynomial Calculation

Calculated Input Signal = Parameter0 + Parameter1\*Voltage^1 +  
Parameter2\*Voltage^2 + Parameter3\*Voltage^3 + Parameter4\*Voltage^4)

C-887, E-712:

E-727: Only calculation type 0 supported.



#### 4.4.31. SIC?.vi

Query fast alignment input calculation

Signal input calculation type:

0 = Calculation OFF

1 = Exponential Calculation:

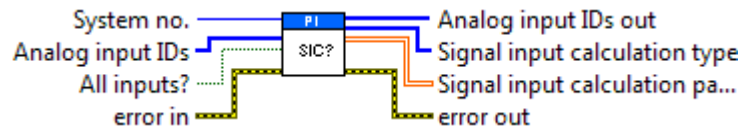
Calculated Input Signal = Parameter0 + Parameter1 \*  
Parameter2^(Parameter3\*Voltage)

2 = Polynomial Calculation:

Calculated Input Signal = Parameter0 + Parameter1\*Voltage^1 +  
Parameter2\*Voltage^2 + Parameter3\*Voltage^3 + Parameter4\*Voltage^4)

C-887, E-712:

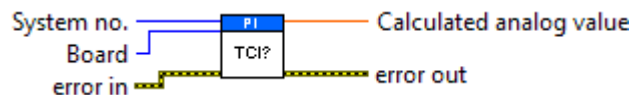
E-727: Only calculation type 0 supported.



#### 4.4.32. TCI?.vi

Queries calculated fast alignment input value. Answer is TAV? value calculated with method set by SIC.

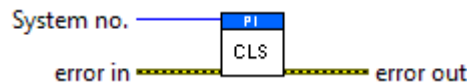
C-887, E-712, E-727.AS:



### 4.5. Controller display.llb

#### 4.5.1. CLS.vi

Clears the controller display.



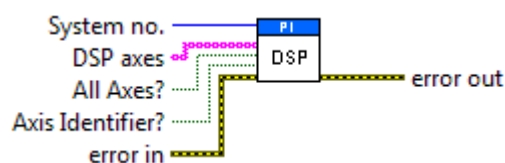
#### 4.5.2. DSP.vi

Displays selected axes on the controller screen.

C-702: If All axes = TRUE, then Axis identifier must be TRUE

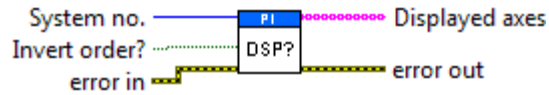
C-848: If All axes = TRUE, then Axis identifier must be TRUE

C-880: If All axes = TRUE, then Axis identifier must be TRUE



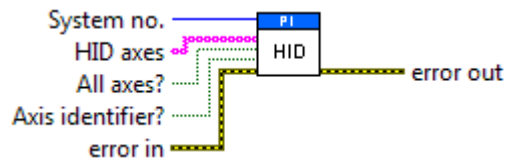
#### 4.5.3. DSP?.vi

Replies which axes are displayed on the controller screen.  
 C-702: Invert order should be TRUE  
 C-848: Invert order should be TRUE  
 C-880: Invert order should be TRUE



#### 4.5.4. HID.vi

Hides selected axes on the controller screen.  
 C-702: If All axes = TRUE, then Axis identifier must be TRUE  
 C-848: If All axes = TRUE, then Axis identifier must be TRUE  
 C-880: If All axes = TRUE, then Axis identifier must be TRUE



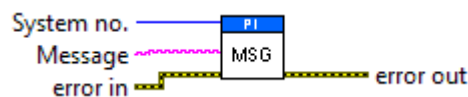
#### 4.5.5. HID?.vi

Replies which axes are hidden on the controller screen.  
 C-702: Invert order should be TRUE  
 C-848: Invert order should be TRUE  
 C-880: Invert order should be TRUE



#### 4.5.6. MSG.vi

Displays a message on the controller display.

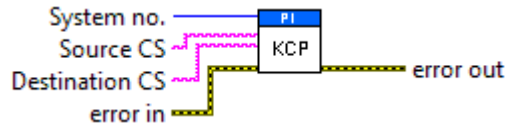


## 4.6. Coordinate Systems.llb

### 4.6.1. KCP.vi

Copies a coordinate system (e. g. in order to create a backup copy), and queries ERR?. Controller error is TRUE if selected system reports error code not equal to 0.

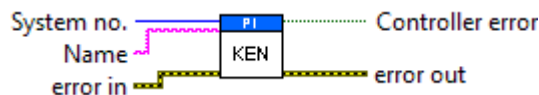
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



### 4.6.2. KEN.vi

Activates an already defined coordinate system; i.e. assigns "active" state and queries ERR?. KEN sets the pivot point coordinates (SPI.vi) to zero when a KSD, KSW or KST coordinate system is active. Activating KLF, KLD, KSB or KSF does not change the pivot point settings. If the hexapod is moving, the command cannot be applied. Activating coordinate systems of type KLD, KLF and KSB requires command level 1 (CCL.vi). KEN settings are volatile but can be saved as power-on default with WPA.vi using the password "SKS". Controller error is TRUE if selected system reports error code not equal to 0.

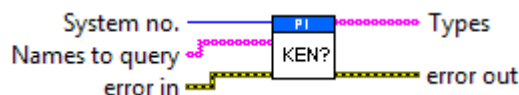
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



### 4.6.3. KEN?.vi

Returns active coordinate systems. The returned information depends on the arguments used. If no argument is set, all active coordinate systems are returned. The KEN? command sets an error code if a coordinate system with the specified name is not defined.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).





#### 4.6.4. KET?.vi

Returns active coordinate system types. The returned information depends on the arguments used. If no argument is set, all names of active coordinate systems are returned sorted by type.

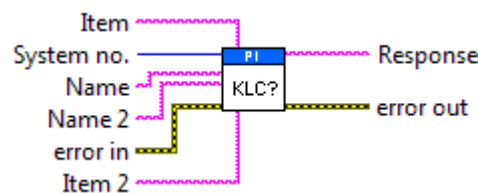
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.6.5. KLC?.vi

Returns parameters of available Work/Tool combinations. The returned information depends on the used arguments. Applicable parameters are: name of Work coordinate system, name of Tool coordinate system, NLM, PLM, SSL, and SST. Name 2 is only valid in combination with Name; Item is only valid in combination with Name and Name 2; Item 2 is only valid in combination with Name, Name 2 and Item.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

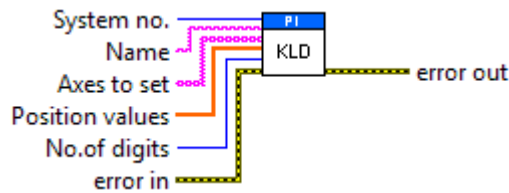


#### 4.6.6. KLD.vi

Defines a levelling coordinate system (KLD-type) and queries ERR?. A coordinate system defined with this command is intended to eliminate hexapod misalignment. Use KLD in case misalignment is known via an external measurable deviation.

If there is already a coordinate system with the same name, KLD will replace it if it is not used. Activating a KLD type coordinate system ("KEN.vi") does not cause any motion, and "POS?.vi" will report new position values. If the hexapod is moving, the command cannot be applied. All pivot point coordinates (SPI.vi) will remain unchanged. SPI.vi is allowed when a KLD type coordinate system is active. For new or replaced KLD type coordinate systems, the parent coordinate system will be set to "PI\_LEVELLING". When replacing an existing coordinate system of this type, the chain relations will be changed and the parent will be set to "PI\_LEVELLING". For axes which are not given the position values will be set to 0. Coordinate system settings are volatile but can be saved as power-on default with WPA.vi using the password "SKS". This command requires command level 1 (CCL.vi).

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

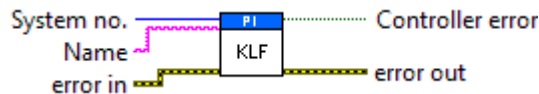


#### 4.6.7. KLF.vi

Defines a levelling coordinate system (KLF-type) and queries ERR?. A coordinate system defined with this command is intended to eliminate hexapod misalignment. Use KLF in case the hexapod is already in the aligned position. Controller error is TRUE if selected system reports error code not equal to 0.

If there is already a coordinate system with the same name, KLF will replace it if it is not used. Activating a KLF type coordinate system (KEN.vi) does not cause any motion, and POS?.vi will report new position values. If the hexapod is moving, the command cannot be applied. All pivot point coordinates (SPI.vi) will remain. SPI.vi is allowed when a KLF type coordinate system is active. For new or replaced KLF type coordinate systems, the parent coordinate system will be set to "PI\_LEVELLING". Coordinate system settings are volatile but can be saved as power-on default with WPA.vi using the password "SKS". This command requires command level 1 (CCL.vi).

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

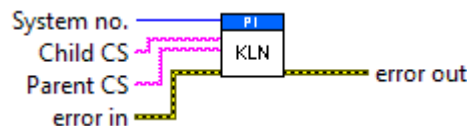


#### 4.6.8. KLN.vi

Links two coordinate systems together by defining a parent-child relation; thus forming a chain, and queries ERR?.

KLN does not allow linking a coordinate system to itself. KLN does not check for cyclic coordinate system relations. Check is done with KEN. Linking a KSB-type coordinate system as child requires command level 1 (CCL.vi). Coordinate system settings are volatile but can be saved as power-on default with WPA.vi using the password "SKS". Controller error is TRUE if selected system reports error code not equal to 0.

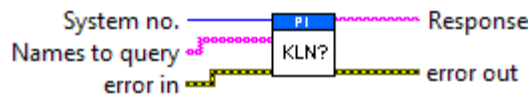
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



**4.6.9. KLN?.vi**

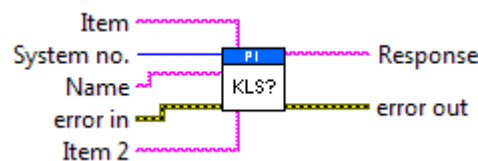
Returns coordinate system chains. A chain consists of minimum two linked coordinate systems. In case of a cyclic definition the reported chain ends with the cyclic coordinate system.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

**4.6.10. KLS?.vi**

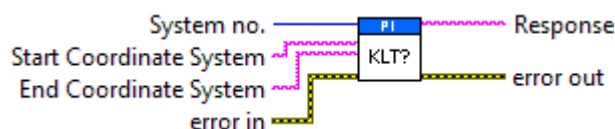
Returns parameters of all available coordinate systems. The returned information depends on the used arguments. To query parameters of Work/Tool combinations use the KLC? command. The x, y, z, u, v, w values of all levelling coordinate systems are displayed according to current coordinate system settings. Item is only valid in combination with Name; Item 2 is only valid in combination with Name and Item.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

**4.6.11. KLT?.vi**

Returns the resulting coordinate system of a chain. A chain consists of minimum two linked coordinate systems. The resulting coordinate system is calculated beginning with the first specified coordinate system. The calculation ends with the second coordinate system if specified. If no second coordinate system is specified, "ZERO" is used for calculation. The position of the resulting coordinate system is canonical (SPI swivel (x, y, z) = (0, 0, 0)). If no parameters are used, the positions of all coordinate systems are returned with second coordinate system "ZERO". End Coordinate System is only valid in combination with Start Coordinate System.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

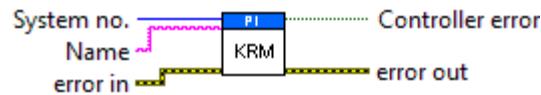
**4.6.12. KRM.vi**

Removes a coordinate system and queries ERR?.

Coordinate system settings are volatile but can be saved as power-on default with WPA.vi using the password "SKS". If the coordinate system to remove is in use or does not exist, an error is set. If the coordinate system is used (e.g. part of an active chain), it cannot be removed. If a coordinate system which is part of a chain

is removed, its parent and child are joined, so that the chain is still valid but KLT? results may differ. Coordinate systems of type KLD, KLF and KSB require command level 1 to remove them (CCL.vi). Controller error is TRUE if selected system reports error code not equal to 0.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

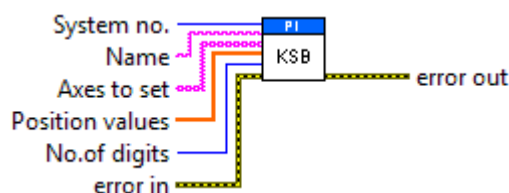


#### 4.6.13. KSB.vi

Defines a base coordinate system (KSB-Type) and queries ERR?. Use KSB in order to change the orientation of the base coordinate system.

KSB only can rotate the coordinate system by 0, 90, 180, 270, -90, -180, -270 degrees. If there is already a coordinate system with the same name, KSB will replace it if it is not used. Activating a KSB type coordinate system (KEN.vi) does not cause any motion, and POS?.vi will report new position values. If the hexapod is moving, the command can be applied. All pivot point coordinates (SPI.vi) will remain. SPI.vi is allowed when a KSB type coordinate system is active. For new or replaced KSB type coordinate systems, the parent coordinate system will be set to the currently active KSB coordinate system. When replacing a coordinate system of this type, the chain relations will not be changed if the coordinate system to replace is of the same type. For axes which are not given, the position values will be set to 0. Coordinate system settings are volatile but can be saved as power-on default with WPA.vi using the password "SKS". This command requires command level 1 (CCL.vi).

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



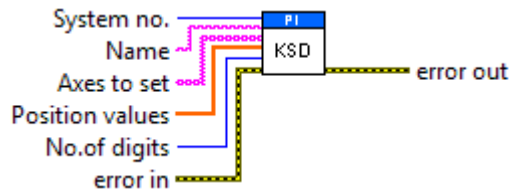
#### 4.6.14. KSD.vi

Defines a new KSD-type coordinate system in order to set a "directed" swivel with the parameters X, Y, Z (relative to the hexapod platform) and queries ERR?. The coordinate system is rotated with the parameters U, V, W.

If there is already a coordinate system with the same name, KSD will replace it if it is not used. Activating a KSD type coordinate system (KEN.vi) does not cause any motion, and POS?.vi will report new position values. If the hexapod is moving, the command can be applied. All pivot point coordinates (SPI.vi) will be reset to the previous values after disabling. SPI.vi is not allowed when a KSD type coordinate system is active. For new or replaced KSD type coordinate systems, the parent coordinate system will be set to "ZERO". When replacing a coordinate system of this type, the chain relations will not be changed if the coordinate system to replace is of the same type. For axes which are not given, the position values will be set to 0. Coordinate system settings are volatile but can be saved as power-on

default with WPA.vi using the password "SKS". This command requires command level 0 or 1.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

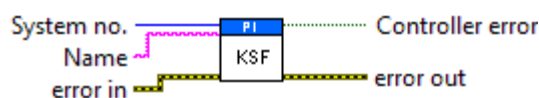


#### 4.6.15. KSF.vi

Defines a coordinate system based on the current position of the hexapod platform and queries ERR?.

If there is already a coordinate system with the same name, KSF will replace it if it is not used. Activating a KSF type coordinate system (KEN.vi) does not cause any motion, and POS?.vi will report position (0, 0, 0, 0, 0, 0). If the hexapod is moving, the command cannot be applied. All pivot point coordinates (SPI.vi) will remain. SPI.vi is allowed when a KSF type coordinate system is active. For new or replaced KSF type coordinate systems, the parent coordinate system will be set to "ZERO". When replacing a coordinate system of this type, the chain relations will be changed and the parent will be set to "ZERO". Coordinate systems that are currently in use cannot be changed. Coordinate system settings are volatile but can be saved as power-on default with WPA.vi using the password "SKS". This command requires command level 0 or 1. Controller error is TRUE if selected system reports error code not equal to 0.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



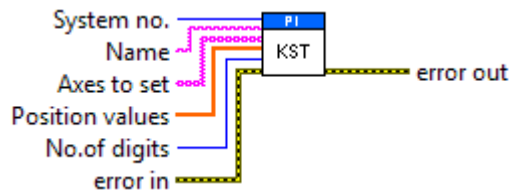
#### 4.6.16. KST.vi

Defines a new tool coordinate system (KST-type) and queries ERR?.

If there is already a coordinate system with the same name, KST will replace it if it is not used. Activating a KST type coordinate system (KEN.vi) does not cause any motion, and POS?.vi will report new position values. If the hexapod is moving, the command can be performed. All pivot point coordinates (SPI.vi) will be reset to the previous values after disabling. SPI.vi is not allowed when a KST type coordinate system is active. For new or replaced KST type coordinate systems, the parent coordinate system will be set to "ZERO". When replacing a coordinate system of this type, the chain relations will not be changed if the coordinate system to replace is of the same type. Coordinate systems that are currently in use cannot be changed. For axes which are not given, the position values will be set to 0.

Coordinate system settings are volatile but can be saved as power-on default with WPA.vi using the password "SKS". This command requires command level 0 or 1.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



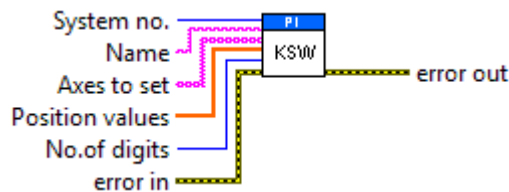
#### 4.6.17. KSW.vi

Defines a new work coordinate system (KSW-type) and queries ERR?.

If there is already a coordinate system with the same name, KSW will replace it if it is not used. Activating a KSW type coordinate system (KEN.vi) does not cause any motion, and POS?.vi will report new position values. If the hexapod is moving, the command can be performed. All pivot point coordinates (SPI.vi) will be reset to the previous values after disabling. SPI.vi is not allowed when a KSW type coordinate system is active. For new or replaced KSW type coordinate systems, the parent coordinate system will be set to "ZERO". When replacing a coordinate system of this type, the chain relations will not be changed if the coordinate system to replace is of the same type. Coordinate systems that are currently in use cannot be changed. For axes which are not given, the position values will be set to 0.

Coordinate system settings are volatile but can be saved as power-on default with WPA.vi using the password "SKS". This command requires command level 0 or 1.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

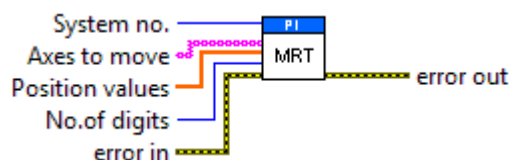


#### 4.6.18. MRT.vi

Executes a relative move in the tool coordinate system. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

Target position results from calculating the translation first and then the rotation. Only hexapod axes can be moved. If no tool coordinate system is explicitly defined, the default tool coordinate system (0, 0, 0, 0, 0, 0) is used.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

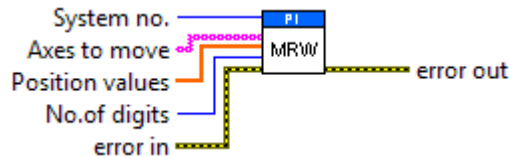


#### 4.6.19. MRW.vi

Executes a relative move in the work coordinate system. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

The target position results from calculating the translation first and then the rotation. Only hexapod axes can be moved. If no work coordinate system is explicitly defined, the default work coordinate system (0, 0, 0, 0, 0, 0) is used.

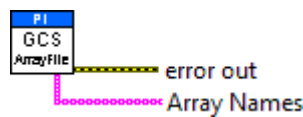
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



## 4.7. File handling.llb

### 4.7.1. ArrayFile.vi

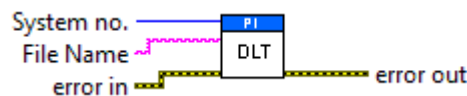
This VI checks the names of all arrays in a data file or deletes a given array from a data file.



### 4.7.2. DLT.vi

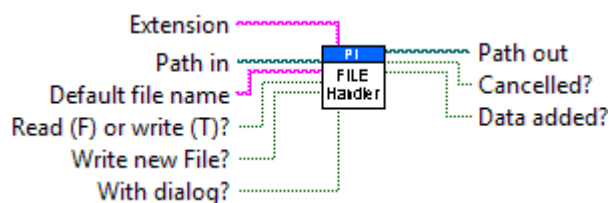
Deletes a data file that was saved in nonvolatile memory using the STD command. Deleting a data file with DLT requires switching to command level 1 with the CCL command

E-754: Check HLP? answer to find out if DLT is supported



### 4.7.3. File handler.vi

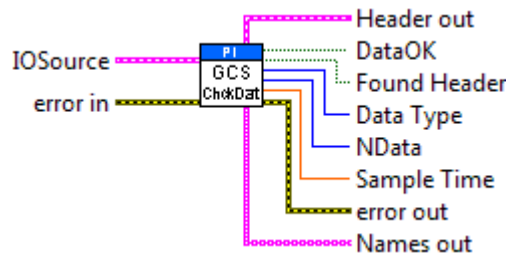
This VI handles file name selections with or without a user interface. Files can be read or written. Path in is the path to the file to read or write. Extension is the file extension for the file to write (e.g. txt, jpg). If Read (F) or write (T) is TRUE, Extension must be given and entry must not have a dot. If With dialog? is TRUE, in every case a dialog box will pop up where the file to read or write can be selected. Default file name is used for naming suggestions if a dialog pops up. If Read (F) or write (T)? is TRUE and Write new file? is TRUE, a dialog box will pop up if the selected file name already exists. If Write new file? is FALSE and the selected file name already exists, a dialog box will pop up to ask if data should be added. Data added? indicates if data was added to an existing file. Cancelled? indicates if the user has cancelled the operation. Path out is NotAPath if operation was cancelled or not successful and contains the selected path for the file which was read or written if the operation was successful.





#### 4.7.4. GetDataFormat.vi

This vi checks the format of a data file. Sub-VI for Show\_Save\_Load\_XY\_Data.vi. See separate manual GCSData User SM146E.pdf and control descriptions in the diagram for more information.



#### 4.7.5. LST?.vi

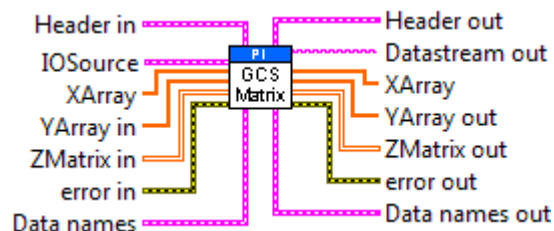
Lists names of data files in nonvolatile memory.

E-754: Check HLP? answer to find out if LST? is supported



#### 4.7.6. MatrixIO.vi

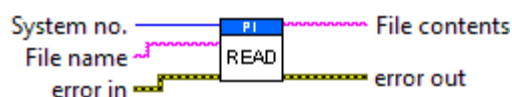
This VI reads or writes data files in matrix format. Sub-VI for 1D\_Scan.vi, 2D\_Scan.vi, Show\_Save\_Load\_XY\_Data.vi and Show\_Save\_Load\_XYZ\_Data.vi. See separate manual GCSData\_User\_SM146E.pdf and control descriptions in the diagram for more information.



#### 4.7.7. READ.vi

Reads the contents of the given file.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). File name can be any file stored on the controller with file extension .dat, .txt or .mac. For files outside the HEXAPOD directory the full path must be given.

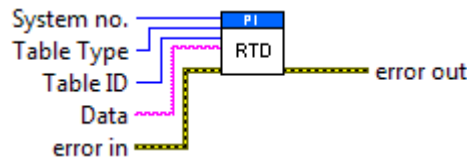


#### 4.7.8. RTD.vi

Reads the content of a data file in nonvolatile memory and writes it to a given table in volatile memory.



E-754: Check HLP? answer to find out if RTD is supported

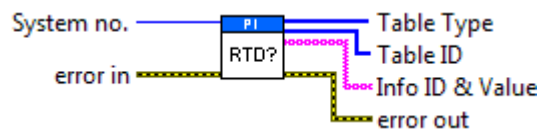


#### 4.7.9. RTD?.vi

Reads the content of a data table in volatile memory.

Before reading, the content must be loaded into the volatile memory from a data file with RTD 100 100 <Name>. For details, see RTD.vi.

E-754: Check HLP? answer to find out if RTD? is supported

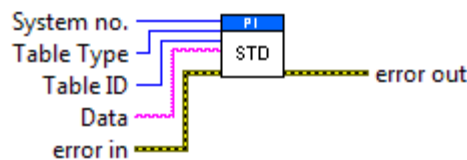


#### 4.7.10. STD.vi

Saves the content of a given table from volatile memory to a data file in nonvolatile memory.

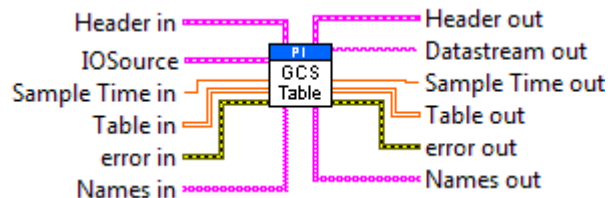
With header tables, STD is also used to write the information in the table before it is saved to data file in nonvolatile memory.

E-754: Check HLP? answer to find out if STD is supported



#### 4.7.11. TableIO.vi

This VI reads or writes data files in table format. Sub-VI for DDL.vi, DRR?.vi, HIT?.vi, GWD?.vi, JLT?.vi, TWS.vi etc. See separate manual GCSDData User SM146E.pdf and control descriptions in the diagram for more information.



## 4.8. General command.lib

### 4.8.1. \*IDN?.vi

Returns system identification string.

E-816: This command cannot be issued to a slave.

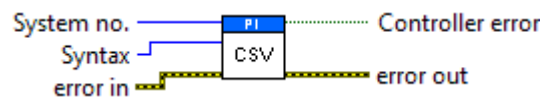


### 4.8.2. CSV.vi

Sets GCS syntax version. Controller error is TRUE if selected system reports an error code which is not 0.

E-517: GCS syntax version must be GCS 2.0. GCS 1.0 switches to E-516 compatibility mode (use E-516 GCS LabVIEW driver set instead when working in E-516 compatibility mode). Requires command level 1, see CCL.vi.

C-887, F-206, M-8X0: Only for combinations of mechanics and C-887 controller.



### 4.8.3. CSV?.vi

Returns current GCS syntax version.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



### 4.8.4. Define connected axes.vi

Writes connected axes into Global2 (Array).vi. This VI is called automatically by PI\_GCS2\_Configuration\_Setup.vi and must be completed successfully before any other axis-specific command VI is called. Requires SAI?.vi to be present.

Analog: Only supported when called by Analog\_Configuration\_Setup.vi

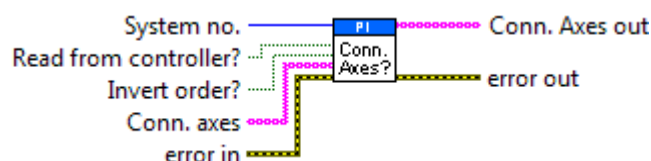
C-702: Read from controller = TRUE, Invert order = TRUE

C-848: Read from controller = TRUE, Invert order = TRUE

C-880: Read from controller = TRUE, Invert order = TRUE

C-887, F-206, M-8X0: Read from controller = FALSE, Invert order = FALSE, Connected axes = X,Y,Z,U,V,W, (A,B,K,L,M optional)

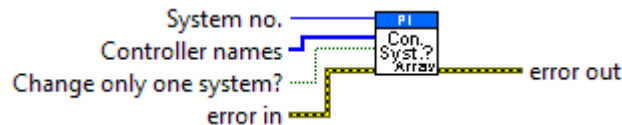
All other systems: Read from controller = TRUE, Invert order = FALSE



#### 4.8.5. Define connected systems (Array).vi

Defines connected systems and writes controller names into Global2 (Array).vi. This VI is called automatically by PI\_GCS2\_Configuration\_Setup.vi and must be completed successfully before General wait for movement to stop.vi is called. If "Change only one system?" is FALSE, all entries from "Controller names" are written into Global2 (Array).vi. If "Change only one system?" is TRUE, only the first field of "Controller names" is valid and only the entry for the given system number is overwritten in Global2 (Array).vi.

Analog system: Only supported when called by Analog\_Configuration\_Setup.vi

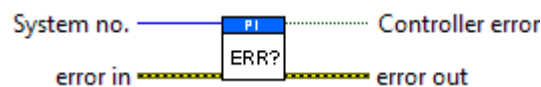


#### 4.8.6. ERR?.vi

Returns error information. Controller error is TRUE if selected system reports an error code which is not 0. See appendix A of the GCS LabVIEW manual for a list of PI error codes and use GCSTranslateError.vi to translate error codes into error descriptions programmatically.

Analog system: VI does not report any error.

E-816: This command cannot be issued to a slave



#### 4.8.7. Global2 (Array).vi

A global variable which contains identifiers for all connected axes of all connected systems and the names of all connected systems.



#### 4.8.8. HLP?.vi

Returns help string.

C-887, F-206, M-8X0: For GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), check HELP answer to determine if HLP? is supported. HLP? and HELP are equivalent.

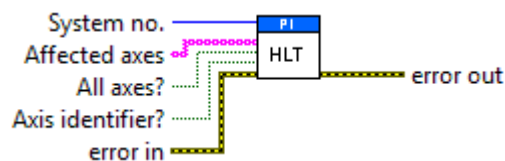


#### 4.8.9. HLT.vi

Stops motion of specified axes. HLT sets error code 10, call ERR?.vi to reset error after HLT has been called.

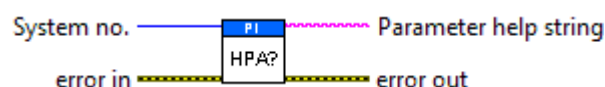
C-413: If All axes? = TRUE, then Axis identifier? can be FALSE

C-702: If All axes = TRUE, then Axis identifier must be TRUE  
 C-843: If All axes = TRUE, then Axis identifier must be TRUE  
 C-843.PM: If All axes = TRUE, then Axis identifier must be TRUE  
 C-844: If All axes = TRUE, then Axis identifier must be TRUE  
 C-848: If All axes = TRUE, then Axis identifier must be TRUE  
 C-865: If All axes = TRUE, then Axis identifier can be FALSE  
 C-866: If All axes? = TRUE, then Axis identifier? can be FALSE  
 C-867: If All axes? = TRUE, then Axis identifier? can be FALSE  
 C-877: If All axes? = TRUE, then Axis identifier? can be FALSE  
 C-880: If All axes = TRUE, then Axis identifier must be TRUE  
 C-884: If All axes? = TRUE, then Axis identifier? can be FALSE  
 C-885: If All axes? = TRUE, then Axis identifier? can be FALSE  
 C-886: If All axes? = TRUE, then Axis identifier? can be FALSE  
 C-891: If All axes? = TRUE, then Axis identifier? must be TRUE  
 E-517: If All axes? = TRUE, then Axis identifier? can be FALSE  
 E-518: If All axes? = TRUE, then Axis identifier? can be FALSE  
 E-710: If All axes? = TRUE, then Axis identifier? must be TRUE  
 E-727: If All axes? = TRUE, then Axis identifier? must be TRUE  
 E-754: If All axes? = TRUE, then Axis identifier? can be FALSE  
 E-755: If All axes? = TRUE, then Axis identifier? must be TRUE  
 E-761: If All axes = TRUE, then Axis identifier can be FALSE  
 E-861: If All axes? = TRUE, then Axis identifier? can be FALSE  
 E-871, E-872, E-873, E-874: If All axes? = TRUE, then Axis identifier? can be FALSE  
 C-887, F-206, M-8X0: If All axes? = TRUE, then Axis identifier? can be FALSE.  
 Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).  
 Hydra, Pollux: If All axes? = TRUE, then Axis identifier? can be FALSE.  
 Mercury: If All axes = TRUE, then Axis identifier can be FALSE  
 Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.8.10. HPA?.vi

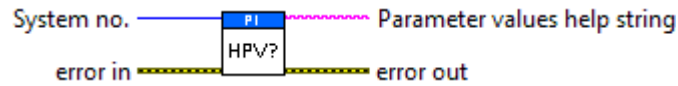
Returns a help string containing information about valid parameter IDs.  
 C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.8.11. HPV?.vi

Returns a help string containing information about valid parameter values.

C-867, C-884, E-709, E-712, E-725, E-753:



#### 4.8.12. Initialize Global2.vi

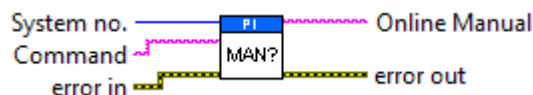
This VI initializes Global2 (Array) according to the given system no.



#### 4.8.13. MAN?.vi

Returns online manual entry for given command.

C-867, Mercury\_GCS:



#### 4.8.14. MOV.vi

Moves specified axes to specified absolute positions. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

C-413: Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

C-867: This command works only in closed-loop operation. Motion commands like MOV are not allowed when a joystick is active on the axis.

C-877: This command works only in closed-loop operation.

C-880K005: VI only supported when called through PI\_Multix.vi

C-884: This command works only in closed-loop operation. Motion commands like MOV are not allowed when control via a Human Interface Device (HID) is active on the axis.

C-885: This command works only in closed-loop operation.

C-886: Motion commands are not allowed when a wave generator is active.

C-891: This command works only in closed-loop operation. Motion commands are not allowed when the wave generator is active.

E-517, E-518: Motion commands like MOV are not allowed when the controller is in OFFLINE mode or when the wave generator output is active. When a macro is running on the controller, MOV will be executed not until the macro is finished or stopped. See "Control Value Generation" and "Control Modes" in the controller user manual for details.

E-709: Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

E-712: Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-725, E-727: Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-753, E-754: Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

E-755: Command not available for E-755.101.

E-816: Only one axis per command allowed. It is necessary to wait a certain time before sending the next command to prevent it from being lost.

E-861: This command works only in closed-loop operation. With open-loop systems, use OAD, OSM, OMA or OMR instead to command motion.

Motion commands like MOV are not allowed when a joystick is active on the axis.

E-871, E-873, E-874: This command works only in closed-loop operation. Motion commands like MOV are not allowed when a joystick is active on the axis.

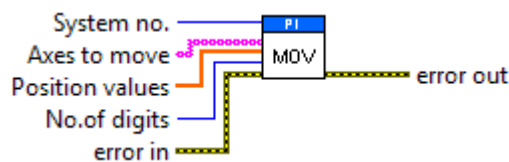
C-887, F-206, M-8X0: For GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), no mix between F-206 axes X,Y,Z,U,V,W, separate axes A, B and NanoCube axes K, L, M in one call allowed. Subsequent MOV commands are handled in a buffer on controller, so please use General Wait for motion to stop.vi or #5.vi to check for motion to stop before sending the next MOV.

For GCS syntax version = GCS 2.0 or higher, subsequent MOV commands override the target position of the previous MOV command. Behavior can be changed by setting parameter 0x19001900 (SPA.vi), see controller user manual for details.

Hydra, Pollux: This command works only in closed-loop operation.

Mercury\_GCS: This command works only in closed-loop operation.

Motion commands like MOV are not allowed when a joystick is active on the axis.



#### 4.8.15. MOV?.vi

Returns commanded target position.

E-516: If All axes? = TRUE, then Axis identifier? must be TRUE

E-710: If All axes? = TRUE, then Axis identifier? must be TRUE

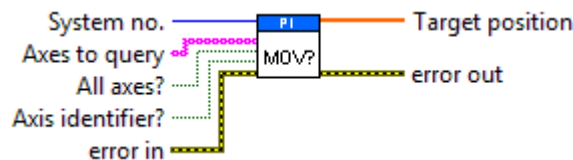
E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101.

E-816: All axes? = FALSE, only one axis per command allowed.

F-206: For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), cCommand has different implementation, please use MOV?\_old.vi

C-887, M-8X0: For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), Ccommand has different implementation, please use MOV?\_old.vi

All other systems: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.8.16. MVR.vi

Moves specified axes relative to current positions. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

C-413: Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

C-867: This command works only in closed-loop operation. Motion commands like MVR are not allowed when a joystick is active on the axis.

C-877: This command works only in closed-loop operation.

C-880K005: VI only supported when called through PI\_Multix.vi

C-884: This command works only in closed-loop operation. Motion commands like MOV are not allowed when control via a Human Interface Device (HID) is active on the axis.

C-891: This command works only in closed-loop operation. Motion commands are not allowed when the wave generator is active.

E-517, E-518: Motion commands like MVR are not allowed when the controller is in OFFLINE mode or when the wave generator output is active. When a macro is running on the controller, MVR will be executed not until the macro is finished or stopped. See "Control Value Generation" and "Control Modes" in the controller user manual for details.

E-709: Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

E-710: See also NMVR.vi in Old commands.llb .

E-712: Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-725, E-727: Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-753, E-754: Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

E-755: Command not available for E-755.101.

E-816: Only one axis per command allowed. It is necessary to wait a certain time before sending the next command to prevent it from being lost.

E-861: This command works only in closed-loop operation. With open-loop systems, use OAD, OSM, OMA or OMR instead to command motion. Motion commands like MVR are not allowed when a joystick is active on the axis.

E-871, E-873, E-874: This command works only in closed-loop operation. Motion commands like MOV are not allowed when a joystick is active on the axis.

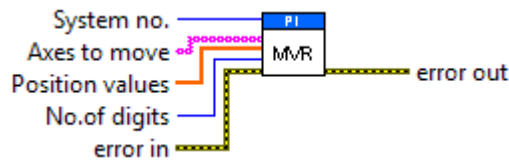
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

Hydra, Pollux: This command works only in closed-loop operation.

Mercury\_GCS: This command works only in closed-loop operation.

Motion commands like MVR are not allowed when a joystick is active on the axis.



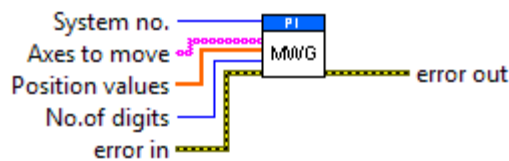


#### 4.8.17. MWG.vi

Moves specified axes to absolute position without updating graphics on the controller screen (fast move). No. of digits is the number of digits after the decimal point in the position value(s) that will be sent. Required by 1D Scan.vi and 2D Scan.vi.

F-206: Only for GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). No mix between F-206 axes X,Y,Z,U,V,W ,separate axes A,B and NanoCube axes K,L,M allowed. For compatibility reasons, VI calls MOV.vi if GCS syntax version is > GCS 1.0.

C-887, M-8X0: Only for GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0) and only for controllers based on C-842.80. If not supported, must be present anyway. For compatibility reasons, VI calls MOV.vi if GCS syntax version is > GCS 1.0.



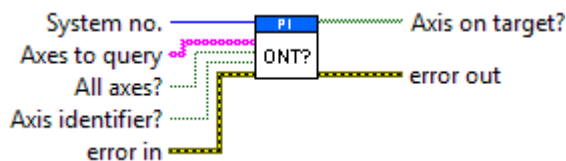
#### 4.8.18. ONT?.vi

Indicates whether or not queried axis is at target position.

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101

E-816: All axes? = FALSE, only one axis per command allowed.

All other systems: If All axes? = TRUE, then Axis identifier? can be FALSE.



#### 4.8.19. POS?.vi

Returns position information (actual or target position, depending on system).

C-880K005: VI only supported when called through PI\_Multix.vi

E-516: If All axes? = TRUE, then Axis identifier? must be TRUE

E-710: If All axes? = TRUE, then Axis identifier? must be TRUE .

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101.

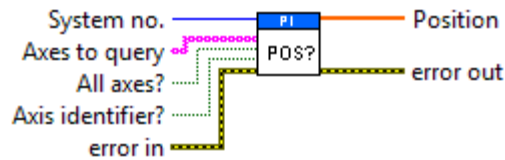
E-816: All axes? = FALSE, only one axis per command allowed.

C-887, F-206: If All axes? = TRUE, then Axis identifier? can be FALSE. For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), returned position value is the commanded target position for the axis.



C-887, M-8X0: If All axes? = TRUE, then Axis identifier? can be FALSE. For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), returned position value is the commanded target position for the axis.

All other systems: If All axes? = TRUE, then Axis identifier? can be FALSE.



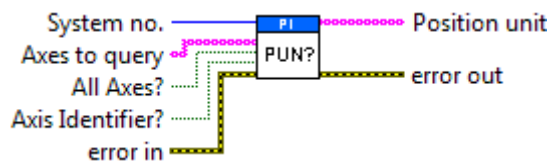
#### 4.8.20. PUN?.vi

Returns the name of the connected stage for queried axes.

C-413, E-712, E-725, E-727, E-753: If All axes? = TRUE, then Axis identifier? can be FALSE.

C-886, E-754: If All axes? = TRUE, then Axis identifier? can be FALSE.

C-887, F-206, M-8X0: If All axes? = TRUE, then Axis identifier? can be FALSE. Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

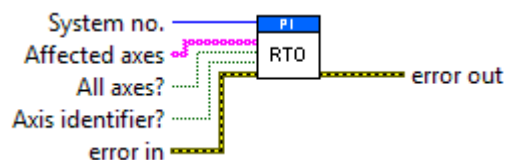


#### 4.8.21. RTO.vi

Ready to turn off. The current position of the given axes is written to non-volatile memory, the servo is switched off automatically and RNP is performed to reduce the applied voltages to 0. When the controller is switched on next time, the saved position is read from non-volatile memory and set as current position. RTO can only be executed if the axis was referenced (e.g. with FNL, FPL, FRF or POS, etc.).

RTO must be repeated when it was already performed in the current session but another motion occurred afterwards or the servo was switched on again.

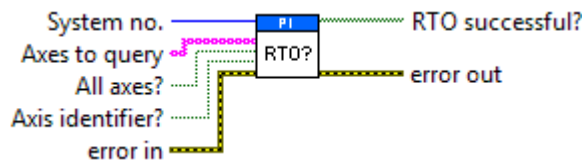
E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101



#### 4.8.22. RTO?.vi

Indicates whether or not RTO was successfully used.

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101



#### 4.8.23. SAI?.vi

Returns axis identifiers of all connected axes and writes them into Global2 (Array).vi. Required by Define connected axes.vi. If SAI? ALL is TRUE, all physically available axes are returned, no matter if configured or not. If SAI? ALL is TRUE, returned identifiers normally may not be written to Global2 (Array).vi. To write them to Global2 (Array).vi nevertheless, set "Write to Global2?" to TRUE.

Analog: Invert order should be FALSE, SAI? ALL must be FALSE

C-702: Invert order should be TRUE, SAI? ALL must be FALSE

C-848: Invert order should be TRUE, SAI? ALL must be FALSE

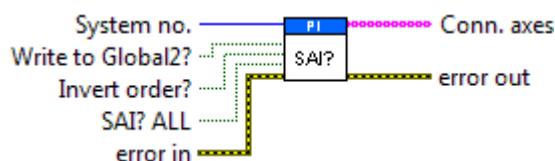
C-880: Invert order should be TRUE, SAI? ALL must be FALSE to read all configured axis IDs and must be TRUE to get all physically defined axis IDs

C-880K005: VI only supported when called through PI\_Multix.vi, SAI? ALL must be FALSE

E-516: Invert order should be FALSE, SAI? ALL must be FALSE

E-816: Invert order should be FALSE, SAI? ALL must be FALSE

All other systems: Invert order should be FALSE, SAI? ALL is supported



#### 4.8.24. SPA.vi

Sets parameters, waits 100 ms and queries ERR?. For axis-related parameters, Axis to set is the axis name, for piezo-, sensor, PIShift or demux-related parameters, the channel number, otherwise a parameter-related code. If parameter no. is in decimal format, use "Parameter number" input, for hexadecimal parameter numbers use "Parameter number (hex)" input and switch "Parameter no. format" to TRUE. For numeric parameter values use "Parameter value" input, for parameter strings use "Parameter string" input and switch "Parameter format" to TRUE. Do not mix decimal and hex. parameter numbers or numeric and string parameter values in one call. Parameter numbers which can be set depend on current CCL level. See GCS DLL manual for available parameter numbers and values. No. of digits is the number of digits after the decimal point in the numeric parameter value(s) that will be sent. Controller error is TRUE if selected system reports an error code which is not 0.

C-413, C-867, C-877, C-884, C-886, E-135, E-516, E-517, E-518, C-891, E-710, E-712, E-725, E-727, E-753, E-754, E-755, E-861, E-870, E-871, E-872, E-873, E-874, Hydra, Pollux, Mercury\_GCS: The SPACommand saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the controller is powered off.

C-413: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the C-413 user manual. Do not set more than 4 parameters at once.

C-702: Parameter no. format is FALSE (decimal).

WARNING: This command is for setting hardware-specific parameters. Wrong values may lead to improper operation or damage of your hardware! Change settings only after consultation with PI.

C-843: Parameter no. format is FALSE (decimal). WARNING

This command is primarily for setting hardware-specific parameters of non-PI stages connected to the controller. Please refer to the stage manual for valid parameter settings. If you have a PI stage connected, please do not change any parameters except P (1), I (2), D (3), I-limit (4) and VFF (5).

For precision and convenience with gearbox systems, the counts per physical unit factor can be entered as numerator and denominator of a fraction (parameters 14 and 15).

C-843.PM: Parameter no. format is FALSE (decimal). See C-843 for warnings.

C-865: Parameter no. format is FALSE (decimal). See C-843 for warnings.

C-866: Parameter no. format is FALSE (decimal). See C-843 for warnings.

C-867, C-877: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the controller user manual. Only one parameter value for only one axis per command allowed.

C-880: Parameter no. format is FALSE (decimal). The most important parameter numbers are:

- 1: P-term (0 to 32767)
- 2: I-term (0 to 32767)
- 3: D-term (0 to 32767)
- 4: I-Limit (integration limit) (0 to 32767)
- 5: VFF (velocity feed forward) (0 to 32767)
- 10: maximum velocity (0 to 1.79769313486231E308)
- 11: maximum allowed acceleration (0 to 1.79769313486231E308)
- 13: maximum allowed Jerk (- 1.79E308 to 1.79E308)

C-880K005: VI only supported when called through PI\_Multix.vi. See C-880 for a description of parameter numbers.

C-884: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the C-884 user manual. Do not set more than 4 parameters at once.

C-891: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the controller user manual. Do not set more than 1 parameter at once.

E-516: Parameter no. format is FALSE (decimal). The following parameter numbers are valid:

- 7: Ksen (Coefficient of Sensor K<sub>s</sub>). When sensor output change is 1V, the position change of stage is K<sub>s</sub> (μm). (- 3.402823466e+38F to 3.402823466e+38F)
- 8: Osen (Offset of Sensor Os). When sensor output is 0V, the actual position of stage is Os (μm). (- 3.402823466e+38F to 3.402823466e+38F)
- 9: Kpzt (Coefficient of PZT voltage amplifier Kpzt). When DAC output change is 1V, the PZT Voltage change is Kpzt (V) (- 3.402823466e+38F to 3.402823466e+38F)
- 10: Opzt (Offset of PZT voltage amplifier Opzt ) When DAC output is 0V, the PZT Voltage is Opzt (V) (- 3.402823466e+38F to 3.402823466e+38F)
- 117442816: Tolerance for ONT software emulation (μm) (0 < value < 1000)

E-135: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the controller user manual.

E-517, E-518: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the controller user manual.

E-709: Parameter no. format is TRUE (hex.). Only one parameter value for only one axis per command allowed. Use HPA?.vi to get valid parameter numbers or see the E-709 user manual.

E-710: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the E7XX\_GCS\_DLL Manual.

E-712: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the E-712 user manual. Do not set more than 10 parameters at once.

E-725: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the E-725 user manual. Do not set more than 10 parameters at once.

E-753: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the E-753 user manual.

E-754: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the E-753 user manual.

E-755: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the E-755 user manual.

E-761: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the user manual. See E-710 for warnings. Do not set more than 10 parameters at once.

The SPA command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the PC is powered off or the E-761 is rebooted.

E-816: Parameter no. format is FALSE (decimal). See E-516 for a description of parameter numbers. This command cannot be issued to a slave. Each command limited to setting one parameter for only one axis.

E-861: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the E-861 user manual. Only one parameter value for only one axis per command allowed. The SPA command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the controller is powered off.

E-870: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the E-870 user manual. Each command is limited to set one parameter for only one channel.

E-871, E-872, E-873, E-874: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the controller user manual. Only one parameter value for only one axis per command allowed.

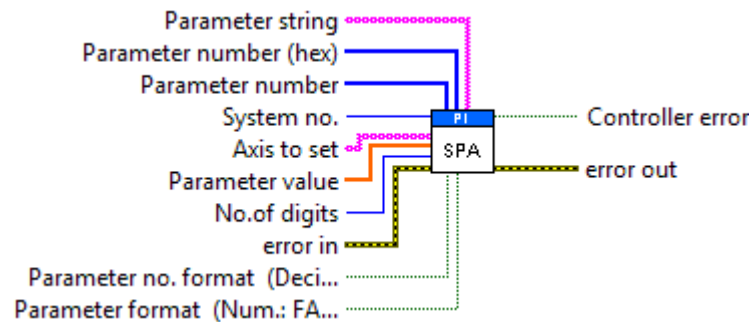
C-887, F-206, M-8X0: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the controller user manual. Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

Hydra, Pollux: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the GCS DLL manual. Do not set more than 10 parameters at once.

Mercury: Parameter no. format is FALSE (decimal). See C-843 for warnings. The SPA command saves the parameters in RAM only. Use PIStageEditor to change parameters or add new stages to the data base permanently.

Mercury\_GCS: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the Mercury user manual. Only one parameter value for only one axis per command allowed.

Warning: This command is for setting hardware-specific parameters. Wrong values may lead to improper operation or damage of your hardware!



#### 4.8.25. SPA?.vi

Returns parameter values for queried items and parameter numbers. For axis-related parameters, Axis to query is the axis name; for piezo-, sensor-, PIShift or demux-related parameters, the channel number, otherwise a parameter-related code. If parameter no. is in decimal format, use "Parameter no." input, for hexadecimal parameter numbers use "Parameter no. (hex)" input and switch "Parameter no. format" to TRUE. If "Without axes" is TRUE, all available parameters for all axes/designators are returned. For parameter numbers which output a string use "Parameter string" output. See GCS DLL manual for available parameter numbers.

C-413: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Do not query more than 6 parameter no. at once (except with Without axes? = TRUE).

C-702: Parameter no. format is FALSE (decimal).

C-843: Parameter no. format is FALSE (decimal). The following parameter number outputs a string:

60: stage name (maximum 14 characters)

C-843.PM: Parameter no. format is FALSE (decimal). The following parameter number outputs a string:

60: stage name (maximum 14 characters)

C-865: Parameter no. format is FALSE (decimal). The following parameter number outputs a string: 60: stage name (maximum 14 characters). Parameter number 25 is read-only.

C-866: Parameter no. format is FALSE (decimal). The following parameter number outputs a string: 60: stage name (maximum 14 characters). Parameter number 25 is read-only.

C-867, C-877: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Only one parameter value for only one axis per command allowed. Use "Without axes?" = TRUE for a query of all parameters.

C-880: Parameter no. format is FALSE (decimal). Additional read-only parameter numbers are:

- 14: Numerator of the counts per physical unit factor (1 to 2147483647)

(factor = num./denom.)

- 15: Denominator of the counts per physical unit factor (1 to 2147483647)

(factor = num./denom.)

- 16: Drive mode: 0=Analog 1=PWM

- 19: Axis type: 0=Linear 1=Rotary

- 20: Reference switch: 0=no present, 1=present

- 28: Reference status: 0=axis not referenced; 1=axis is referenced

C-880K005: VI only supported when called through PI\_Multix.vi

C-884: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Do not query more than 4 parameter no. at once (except with Without axes? = TRUE) Use Without axes? = TRUE for a query of all parameters.

C-891: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Do not query more than 1 parameter no. at once (except with Without axes? = TRUE).

E-135: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers.

E-516: Parameter no. format is FALSE (decimal).

E-517, E-518: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers.

E-709: Parameter no. format is TRUE (hex.). Only one parameter value for only one axis per command allowed. Use Without axes? = TRUE for a query of all parameters. Use HPA?.vi to get valid parameter numbers.

E-710: Use HPA?.vi to get valid parameter numbers and see GCS DLL manual for a description of valid parameter numbers. Parameter no. format is TRUE (Hex).

E-712: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Do not query more than 10 parameter no. at once (except with "Without axes? = TRUE).

E-725, E-727: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Do not query more than 10 parameter no. at once (except with Without axes? = TRUE).

E-753: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers.

E-754: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers.

E-755: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers.

E-761: Use HPA?.vi to get valid parameter numbers and see GCS DLL manual for a description of valid parameter numbers. Parameter no. format is TRUE (Hex). Do not query more than 10 parameter no. at once (except with Without axes? = TRUE).

E-816: Parameter no. format is FALSE (decimal). Only one parameter value for only one axis per command allowed. This command cannot be issued to a slave

E-861: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Only one parameter value for only one axis per command allowed. Use "Without axes?" = TRUE for a query of all parameters. The following parameter number outputs a string:

60 (0x3C): stage name (maximum 16 characters)

E-870: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Only one parameter value for only one channel per command allowed. Use Without axes? = TRUE for a query of all parameters.



E-871, E-872, E-873, E-874: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Only one parameter value for only one axis per command allowed. Use Without axes? = TRUE for a query of all parameters.

C-887, F-206: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Length of command is limited by firmware (appr. 10 parameters per call), see controller user manual for details. Use Without axes? = TRUE for a query of all parameters.

Hydra, Pollux: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Do not query more than 10 parameter no. at once (except with Without axes? = TRUE) Use Without axes? = TRUE for a query of all parameters.

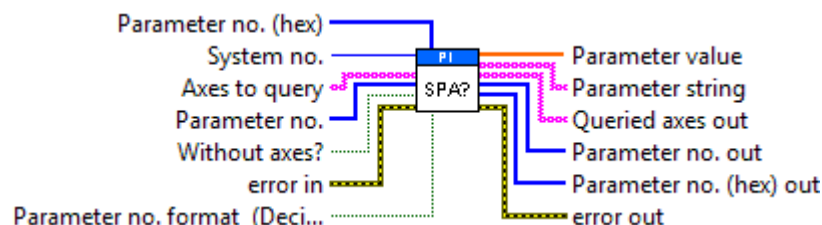
C-887, M-8X0:

For GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Length of command is limited by firmware (appr. 10 parameters per call), see controller user manual for details. Use Without axes? = TRUE for a query of all parameters.

For GCS syntax version = GCS 1.0, Parameter no. format is FALSE (decimal), Axes to query can be 1 to 6 (corresponds to strut no.) and Parameter no. can be 512 (reports if strut is extended or retracted) or 513 (reports commanded strut length). Only one parameter value for only one axis per command allowed.

Mercury: Parameter no. format is FALSE (decimal). The following parameter number outputs a string: 60: stage name (maximum 14 characters)

Mercury\_GCS: Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers. Only one parameter value for only one axis per command allowed. Use Without axes? = TRUE for a query of all parameters.



#### 4.8.26. STP.vi

Stops motion of specified axes. To stop a referencing routine (REF, MNL, MPL) or fast scan routine (FSC, FSA etc.), or AutoZero procedure (ATZ), or wave generator run (WGO), use #24.vi. STP sets error code 10, call ERR?.vi to reset error after STP has been called.

Analog: All axes? = TRUE, Axis identifier = FALSE. STP does not set any error code.

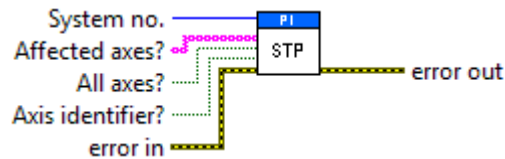
All other systems: All axes? = TRUE, Axis identifier? = FALSE

E-135: STP.vi stops motion of all axes caused by move commands (OSM, POL, SVA).

E-517, E-518: STP.vi stops motion of all axes caused by move commands (MOV, MVR, GOH, SVA, SVR). Furthermore, it stops macros (MAC) and wave generator output (WGO).

C-413, E-709, E-712, E-725, E-727, E-753, E-754: STP.vi stops motion of all axes caused by move commands (MOV, MVR, SVA, SVR), by the wave generator (WGO, if supported), by analog control input and autozero motion (ATZ).

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For GCS 1.0, use STOP.vi instead.



#### 4.8.27. SVO.vi

Sets servo-control mode for given axes. If Without axis ID is TRUE, then Axes to command is ignored and first field of Servo mode array is used.

E-516, E-517, E-518: Make sure that all servo switches on the piezo control electronics are set to "Off" to give the interface/display module complete control over the servo state.

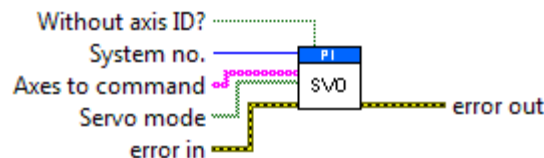
E-755: Without axis ID = FALSE. When the servo mode is switched off, RNP is automatically performed for the corresponding Nexline channel, which could take a few seconds. Command not available for E-755.101.

E-816: Without axis ID = FALSE. Only one axis per command allowed.

E-861: If you have enabled servo controller is busy for 100 ms, only after this period a next command can be performed.

C-887, F-206, M-8X0: For GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), Without axis ID = FALSE. For GCS 1.0, Without axis ID = TRUE and only first field of Servo mode array is valid.

All other systems: Without axis ID = FALSE



#### 4.8.28. SVO?.vi

Returns servo status of queried axes.

C-843, C-843.PM, C-844: If All axes? = TRUE, then Axis identifier? must be TRUE

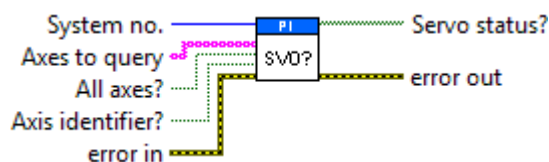
C-865, C-866: If All axes? = TRUE, then Axis identifier? must be TRUE

E-516, E-710: If All axes? = TRUE, then Axis identifier? must be TRUE

E-816: All axes? = FALSE, only one axis per command allowed.

C-887, F-206, M-8X0: For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), only first field of servo status array is valid. All axes? = TRUE, Axis identifier? = FALSE.

All other systems: If All axes? = TRUE, then Axis identifier? can be FALSE





**4.8.29. VEL.vi**

Sets velocity and checks for error. If Without axis ID? is TRUE, then Axes to set is ignored and first field of Velocity values array is used for velocity command. The velocity should not be set to 0. Number of digits is the number of digits after the decimal point in the velocity value(s) that will be sent. Controller error is TRUE if selected system reports an error code which is not 0.

Analog: Without axis ID? = FALSE; Velocity unit is  $\mu\text{m}/\text{sec}$

C-413: Without axis ID? = FALSE. Velocity unit is mm/s. Velocity settings made with VEL are present in RAM only and will be reset to default (parameter 0x06010400) when the controller is powered down or rebooted.

C-867, C-877, C-884, C-891, Hydra, Pollux, Mercury\_GCS: Without axis ID? = FALSE. Velocity unit is mm/s. The VEL command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the controller is powered off.

C-880: Without axis ID? = FALSE, for NanoCube axes command is not valid

C-880K005: VI only supported when called through PI\_Multix.vi

C-885: Without axis ID? = FALSE. The VEL command saves the parameters in RAM only. For how to save the currently valid parameters to flash ROM, where they become the power-on defaults, please refer to the controller's user manual.

E-516: Without axis ID? = FALSE. Velocity unit is  $\mu\text{m}/\text{s}$ . The VEL command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the E-516 is powered off.

E-517, E-518: Without axis ID? = FALSE. Velocity unit is  $\mu\text{m}/\text{s}$  in closed-loop operation and V/s in open-loop operation. Velocity settings made with VEL are present in RAM only and will be reset to default ("Servo Loop Slew Rate" value) when the controller is powered down or rebooted.

E-709, E-712, E-725, E-727, E-753, E-754, E-861: Without axis ID? = FALSE. Velocity unit is  $\mu\text{m}/\text{s}$ . Velocity settings made with VEL are present in RAM only and will be reset to default ("Servo Loop Slew Rate" value) when the controller is powered down or rebooted.

E-710: Without axis ID? = FALSE. Velocity unit is  $\mu\text{m}/\text{ms}$ .

E-755: Without axis ID? = FALSE. Velocity unit is  $\mu\text{m}/\text{s}$ . Command not available for E-755.101.

E-761: Without axis ID? = FALSE. Velocity unit is  $\mu\text{m}/\text{ms}$ . The VEL command saves the "Servo Loop Slew Rate" parameter in RAM only. To save the currently valid parameter to flash ROM, where it becomes the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the PC is powered off or the E-761 is rebooted.

E-861: Without axis ID? = FALSE. The VEL setting only takes effect when the given axis is in closed-loop operation (servo on). For open-loop operation, use OVL instead. The maximum value which can be set with the VEL command is given by the Closed-loop velocity parameter, ID 0xA (can be changed with SPA and SEP). On power-on, the current closed-loop velocity is half the maximum.

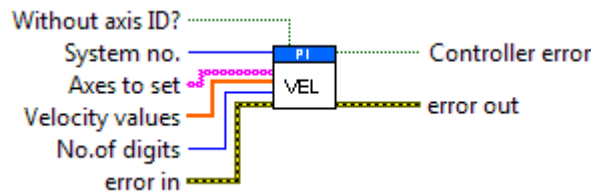
E-873, E-874: Without axis ID? = FALSE. The VEL command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the controller is powered off.

C-886: VEL command is only valid for velocity of additional axes. To set velocity of the parallel kinematics, please use the VLS command instead.

C-887, F-206, M-8X0: For GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), VEL command is only valid for velocity of axes A and/or B, Without axis ID? = False. For platform velocity, VI sends command VLS instead, in this case Without axis ID? = TRUE. For axes K, L, M command is not valid.

For GCS syntax version = GCS 1.0, to set platform velocity: Without axis ID? = TRUE; to set velocity of axes A and/or B: Without axis ID? = False; axes K, L, M: command not valid.

All other systems: Without axis ID? = FALSE



#### 4.8.30. VEL?.vi

Returns velocity setting for specified axes.

Analog: If All axes? = TRUE, then Axis identifier? can be FALSE; Velocity unit is  $\mu\text{m/s}$

C-702, C-848, Mercury: If All axes? = TRUE, then Axis identifier? can be FALSE

C-843, C-843.PM, C-844, E-516: If All axes? = TRUE, then Axis identifier? must be TRUE

C-413, C-865, C-866, C-867, C-877, C-884, C-891, Hydra, Pollux, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE. Velocity unit is mm/s.

C-880: If All axes? = TRUE, then Axis identifier? can be FALSE. NanoCube axes will report velocity = 0

C-880K005: VI only supported when called through PI\_Multix.vi

C-885: If All axes? = TRUE, then Axis identifier? can be FALSE.

C-886: If All axes? = TRUE, then Axis identifier? can be FALSE. VEL? only reports the velocity for additional axes. To query the velocity of the parallel kinematics (system velocity), please use the VLS? command instead.

E-517, E-518: If All axes? = TRUE, then Axis identifier? can be FALSE. Velocity unit is  $\mu\text{m/s}$  in closed-loop operation and V/s in open-loop operation.

E-710: If All axes? = TRUE, then Axis identifier? must be TRUE. Velocity unit is  $\mu\text{m/ms}$ .

E-709, E-712, E-725, E-727, E-753, E-754, E-861: If All axes? = TRUE, then Axis identifier? can be FALSE. Velocity unit is  $\mu\text{m/s}$ .

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Velocity unit is  $\mu\text{m/s}$ . Command not available for E-755.101.

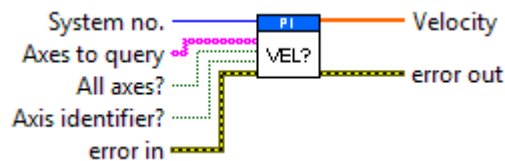
E-761: If All axes? = TRUE, then Axis identifier? can be FALSE. Velocity unit is  $\mu\text{m/ms}$ .

C-887, F-206, M-8X0: For GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), only valid for velocity of axes A and/or B. All axes? must be FALSE. For platform velocity, use VLS? instead. For axes K,L,M command is not valid.

For GCS syntax version = GCS 1.0, Velocity of platform: All axes? = TRUE AND Axis identifier? = FALSE; velocity of axes A,B: All axes? must be FALSE; axes

K,L,M: command not valid. For platform velocity: only first field of velocity array is valid

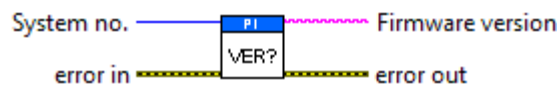
E-873, E-874: If All axes? = TRUE, then Axis identifier? can be FALSE.



#### 4.8.31. VER?.vi

Returns firmware string.

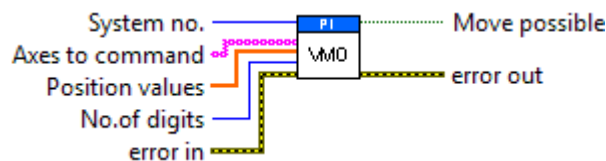
C-867:



#### 4.8.32. VMO.vi

Virtual movement. Indicates whether a move to the specified position is possible or not. Stage will not be moved. is the number of digits after the decimal point in the position value(s) that will be sent.

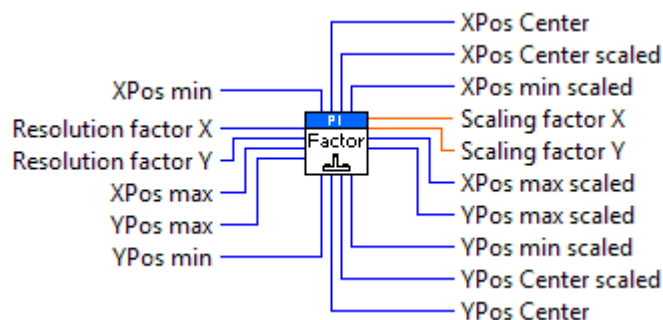
C-887, F-206, M-8X0: For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), VI sends command VMO. For GCS 2.0 and higher, VI sends command VMO? instead.



### 4.9. Joystick.IIb

#### 4.9.1. Calculate joystick scaling.vi

Sub-VI for operation with a joystick connected to the game port of the host computer. Calculates joystick position scaling. If Resolution factor \* = \*Pos max, maximum resolution is achieved. \*Pos min and \*Pos max depend on the Windows joystick calibration.

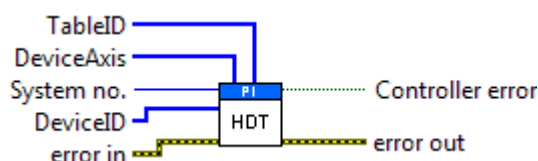


#### 4.9.2. HDT.vi

This VI assigns a lookup table to the given axis of the given Human Interface Device (HID), waits 100 ms and queries ERR?. See controller user manual for details. Controller error is TRUE if selected system reports error code unequal to 0.

C-867: Lookup tables are assigned in volatile memory only (RAM). Use WPA to save the current assignment to non-volatile memory. See controller user manual for details.

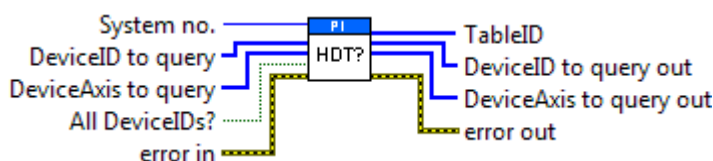
C-884, E-871, E-872, E-873: Lookup tables are assigned in volatile memory only (RAM). Use WPA to save the current assignment to non-volatile memory. See controller user manual for details.



#### 4.9.3. HDT?.vi

Returns the lookup table which is currently assigned to the axis of the Human Interface Device (HID). See controller user manual for details.

C-867:



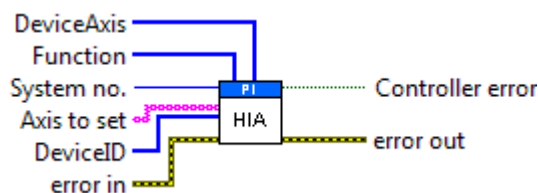
#### 4.9.4. HIA.vi

This VI assigns an axis of a Human Interface Device (HID) to a function of a controller axis. Human interface device control must be disabled (HIN command) when you change HIA. If DeviceID and DeviceAxis is 0, the function is deactivated. See controller user manual for details.

The VI waits 100 ms and queries ERR?. Controller error is TRUE if selected system reports error code which is not equal to 0.

C-867: The configuration with HIA is done in volatile memory only (RAM). Use WPA to save the current configuration to non-volatile memory. See controller user manual for details.

C-884, E-871, E-872, E-873: The configuration with HIA is done in volatile memory only (RAM). Use WPA to save the current configuration to non-volatile memory. See controller user manual for details.



#### 4.9.5. HIA?.vi

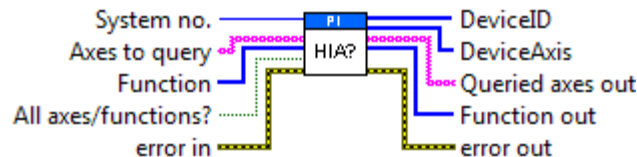
This VI returns the current assignment of axes of Human Interface Devices (HID) to functions of controller axes and queries error. See controller user manual for

details. Requires Build SPA query command substring.vi and Assign SPA values from string to axes.vi to be present.

C-867: If All axes/functions? = FALSE, query is limited to up to one (.1\*) / two (.2\*) axes/functions per call.

C-884: If All axes/functions? = FALSE, query is limited to up to 4 axes/functions per call.

E-871, E-872, E-873: If All axes/functions? = FALSE, query is limited to one axis/function per call.

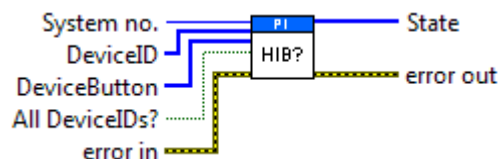


#### 4.9.6. HIB?.vi

Returns the state of the buttons of a Human Interface Device (HID). See controller user manual for details.

C-867:

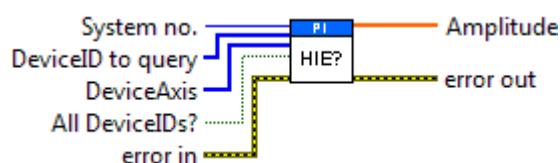
F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.9.7. HIE?.vi

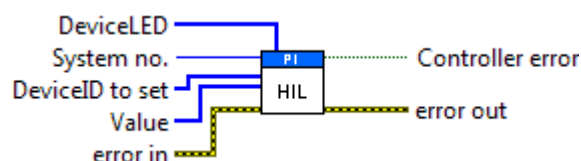
Returns the deflection (current amplitude) of the axes of a Human Interface Device (HID). See controller user manual for details.

C-867:



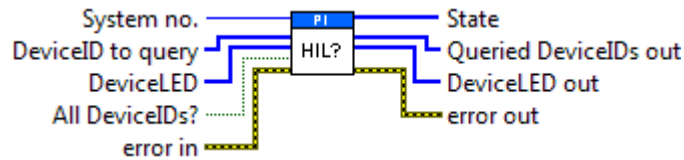
#### 4.9.8. HIL.vi

Sets the state of the LEDs of a Human Interface Device (HID) and queries ERR?. "LED" stands for hardware-dependent output, which can be an LED or a force-feedback actuator. See controller user manual for details. Controller error is TRUE if selected system reports error code which is not 0.



#### 4.9.9. HIL?.vi

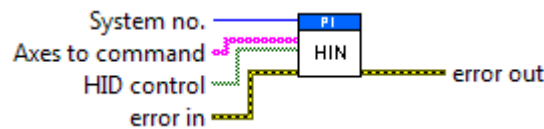
Returns the state of the LEDs of a Human Interface Device (HID). "LED" stands for hardware-dependent output, which can be an LED or a force-feedback actuator. See controller user manual for details.



#### 4.9.10. HIN.vi

Activates or deactivates HID control for given axes of the controller. See controller user manual for details.

C-867:

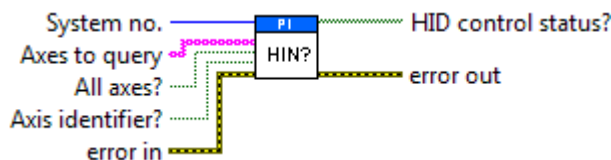


#### 4.9.11. HIN?.vi

Returns current HID control status of queried axes of the controller. See controller user manual for details.

All systems: If All axes? = TRUE, then Axis identifier? can be FALSE

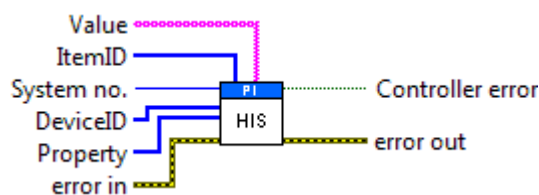
C-867:



#### 4.9.12. HIS.vi

This VI sets values for items of human interface devices (HID). The VI waits 100 ms and queries ERR?. Properties can be colors, the blink time of a LED, the frequency of a force-feed back function, display text etc. See controller user manual for details. Controller error is TRUE if selected system reports error code not equal to 0.

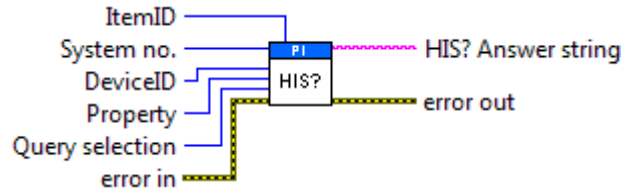
C-884: Command is limited to set up to four DeviceIDs, ItemIDs, Properties and Values per call. ItemID must be of type "LED" which stands for hardware dependent output items or their properties.



#### 4.9.13. HIS?.vi

This VI returns a list of Human Interface Devices (HID) connected to the controller, their items, properties and values and queries error. See controller user manual for details.

C-867:

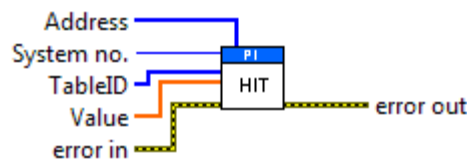


#### 4.9.14. HIT.vi

Sets values of custom lookup tables for Human Interface Devices (HID). See controller user manual for details. Requires Build SPA command substring.vi to be present.

C-867: HIT writes the lookup tables in volatile memory only (RAM). Use WPA to save the current lookup table content to non-volatile memory. See controller user manual for details.

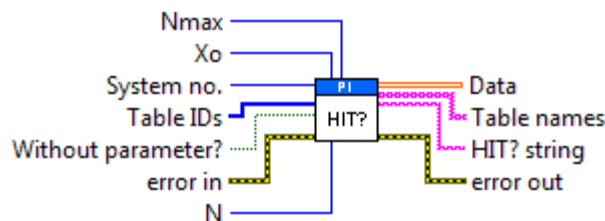
C-884, E-871, E-872, E-873: HIT writes the lookup tables in volatile memory only (RAM). Use WPA to save the current lookup table content to non-volatile memory. See controller user manual for details.



#### 4.9.15. HIT?.vi

Returns N points of the current lookup table content. N must be less than or equal to Nmax. For large N values, communication timeout must be set long enough, otherwise a comm.error may occur.

C-867:



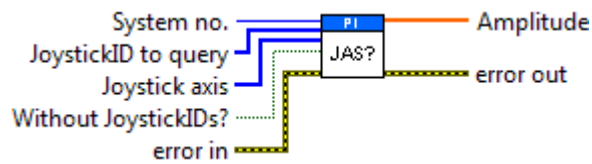
#### 4.9.16. JAS?.vi

Returns joystick axes status values (current amplitude) of joysticks connected to the controller.

C-867, E-709, E-861, E-870, Mercury, Mercury\_GCS: The current amplitude gives a factor which is applied to the velocity set with VEL (closed-loop operation) or OVL (E-861 open-loop operation, E-870), the range is -1.0 to 1.0. Only one Joystick ID to query for one Joystick axis per command allowed. Use Without JoystickIDs? = TRUE to query all Joystick IDs.

E-709: Only supported if E-709 is used inside C-867K012/K013.

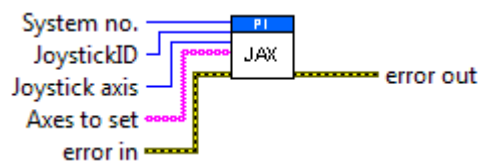




#### 4.9.17. JAX.vi

Defines which axes to control with an axis of a joystick connected to the controller. If Axes to set is an empty array, the current settings are cleared and no axes are controlled. Joystick control is enabled or disabled with JON.vi. Settings will not influence Joystick\_Operation\_Sample\_Program.vi which works with a joystick connected to the host computer.

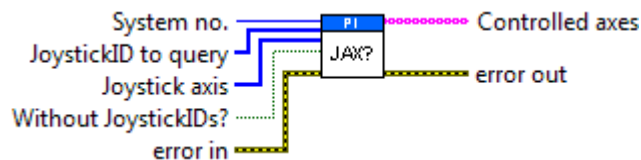
E-709: Only supported if E-709 is used inside C-867K012/K013.



#### 4.9.18. JAX?.vi

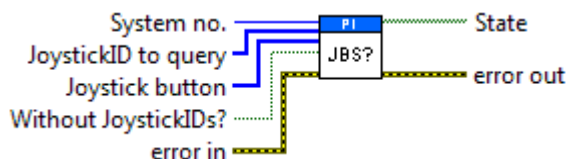
Returns which axes are assigned to be controlled by joystick axes of joysticks connected to the controller.

E-709: Only supported if E-709 is used inside C-867K012/K013.



#### 4.9.19. JBS?.vi

Returns state (TRUE/FALSE) of queried buttons of joysticks connected to the controller.



#### 4.9.20. JDT.vi

Sets default lookup table for joysticks connected to the controller, waits 100 ms and queries ERR?.

Controller error is TRUE if selected system reports error code ? 0. Settings will not influence Joystick\_Operation\_Sample\_Program.vi which works with a joystick connected to the host computer.

C-867, E-709, E-861, Mercury\_GCS: The following default lookup tables are provided:

- 1 = linear
- 2 = parabolic



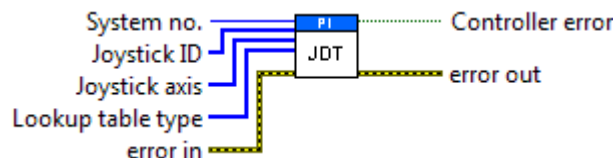
E-709: Only supported if E-709 is used inside C-867K012/K013.

E-870: The following default lookup tables are provided:

- 1 = linear
- 2 = parabolic
- 5 = linear invert
- 6 = parabolic invert

Mercury: The following default lookup tables are provided:

- 1 = linear
- 3 = cubic



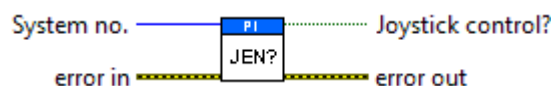
#### 4.9.21. JEN.vi

Switches controller joystick control on (TRUE) or off (FALSE). Before using a joystick connected to the C-848 controller, calibrate joystick by running PI Terminal.vi.. Type "JEN CALIB" and follow the instructions on the screen. Settings will not influence Joystick\_Operation\_Sample\_Program.vi which works with a joystick connected to the host computer.



#### 4.9.22. JEN?.vi

Indicates whether controller joystick control is on (TRUE) or off (FALSE).



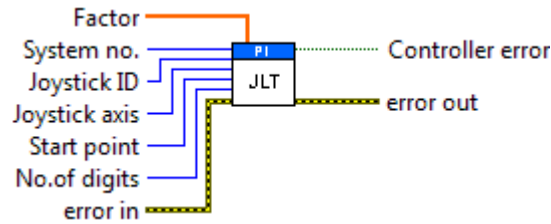
#### 4.9.23. JLT.vi

Sets joystick lookup table for a joystick connected to the controller. The amplitudes of the joystick axes are mapped to velocities of the controller axes. StartPoint is the start point in the lookup table, starting with 1. Valid values for Factor are -1.0 to 1.0 and define the factor which is applied to the currently set velocity (VEL in closed-loop operation, OVL in open-loop operation). No. of digits is the number of digits after the decimal point in the Factor value(s) that will be sent. Controller error is TRUE if selected system reports error code ? 0. Settings will not influence Joystick\_Operation\_Sample\_Program.vi which works with a joystick connected to the host computer.

E-709: Only supported if E-709 is used inside C-867K012/K013. Only one lookup table point for one axis per command allowed. To set more than one point you must create a for-loop around this VI. The joystick table consists of 256 points. The first point corresponds to the maximum joystick axis displacement in negative direction, the 256th point to the maximum displacement in positive direction.

C-867, E-861, E-870, Mercury\_GCS: Only one lookup table point for one joystick axis per command allowed. To set more than one point you must create a for-loop around this VI. The joystick table consists of 256 points. The first point

corresponds to the maximum joystick axis displacement in negative direction, the 256th point to the maximum displacement in positive direction.

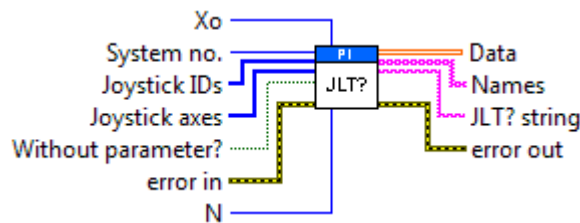


#### 4.9.24. JLT?.vi

Returns joystick lookup table for joysticks connected to the controller. For large N values, communication timeout must be set long enough, otherwise a comm.error may occur.

C-867, E-709, E-861, E-870, Mercury\_GCS: JoystickIDs and Joystick axes must be omitted in the JLT? command, while X0 and N are always required.

E-709: Only supported if E-709 is used inside C-867K012/K013.

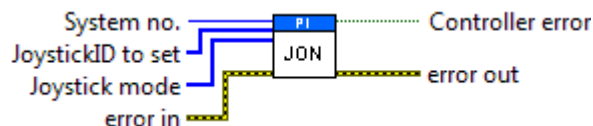


#### 4.9.25. JON.vi

Enables or disables joysticks connected to the controller. Controller error is TRUE if selected system reports an error code which is not 0. Settings will not influence Joystick\_Operation\_Sample\_Program.vi which works with a joystick connected to the host computer.

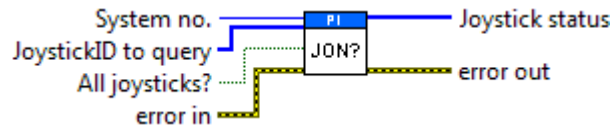
C-843, C-867, E-861, Mercury, Mercury\_GCS: Joystick mode can be 0 (OFF) or 1 (ON). Motion commands like MOV or OSM (if supported) are not allowed when a joystick is active on the axis.

E-709: Only supported if E-709 is used inside C-867K012/K013. Motion commands like MOV are not allowed when a joystick is active on the axis. Joystick mode can be 0 (OFF) or 1 (ON).



#### 4.9.26. JON?.vi

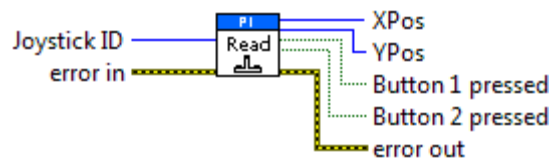
Returns activation status of queried joysticks connected to the controller.  
E-709: Only supported if E-709 is used inside C-867K012/K013.



#### 4.9.27. Read joystick.vi

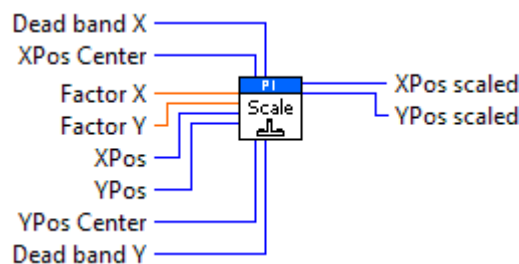
Sub-VI for operation with a joystick connected to the game port of the host computer. Reads joystick position and button status for a standard 2-button 2-axis joystick.

Install joystick driver and calibrate joystick in the Windows control panel before running this VI.



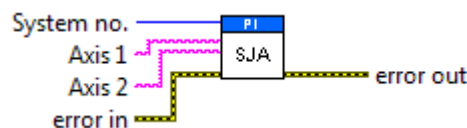
#### 4.9.28. Scale joystick data.vi

Sub-VI for operation with a joystick connected to the game port of the host computer. Scales joystick position. Use output value from Calculate joystick scaling.vi for Factor \*. Dead band \* is the maximum scaled position value that does not result in any motion.



#### 4.9.29. SJA.vi

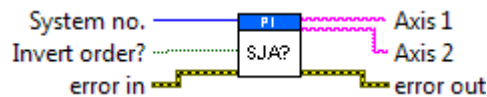
Enables control of the two specified axes with a joystick connected to the controller. Settings will not influence axis settings in Joystick\_Operation\_Sample\_Program.vi which works with a joystick connected to the host computer.



#### 4.9.30. SJA?.vi

Indicates which axes can be controlled with a joystick connected to the controller.  
C-848: Invert order? must be TRUE

C-880: Invert order? must be TRUE



#### 4.9.31. TNJ?.vi

Returns the number of joysticks connected to the controller.



### 4.10. Limits.IIb

#### 4.10.1. ATZ.vi

Performs an automatic zero-point calibration for the specified linear axes (see user manual for details), waits until this procedure has finished and returns whether ATZ was successful or not. Select NaN as Low voltage parameter if you want the controller to use the stored values it has. VI will also stop if Stop refnum or Local stop is TRUE. The home position is reset to default by ATZ. ATZ works independent of servo mode.

C-413 : The result of the AutoZero procedure (new offset value) is present in RAM only. To save the result to EEPROM, where it becomes the power-on default, you must run WPA.vi.

E-709: The result of the AutoZero procedure (new offset value) is present in RAM only. To save the result to EEPROM, where it becomes the power-on default, you must run WPA.vi. For stages with ID-chip the option "Power Up Read ID-Chip" must be disabled by default to make the AutoZero result available in the future. See E-709 user manual for details.

E-712: The result of the AutoZero procedure (new offset value) is present in RAM only. To save the result to EEPROM, where it becomes the power-on default, you must run WPA.vi. For stages with ID-chip the option "Power Up Read ID-Chip" must be disabled by default to make the AutoZero result available in the future. See E-712 user manual for details.

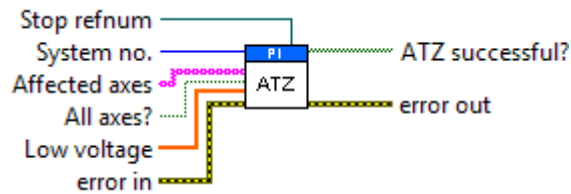
E-725, E-727: The result of the AutoZero procedure (new offset value) is present in RAM only. To save the result to EEPROM, where it becomes the power-on default, you must run WPA.vi. For stages with ID-chip the option "Power Up Read ID-Chip" must be disabled by default to make the AutoZero result available in the future. See controlleruser manual for details.

E-753: The result of the AutoZero procedure (new offset value) is present in RAM only. To save the result to EEPROM, where it becomes the power-on default, you must run WPA.vi. For stages with ID-chip the option "Power Up Read ID-Chip" must be disabled by default to make the AutoZero result available in the future. See E-753 user manual for details.

E-754: The result of the AutoZero procedure (new offset value) is present in RAM only. To save the result to EEPROM, where it becomes the power-on default, you must run WPA.vi. For stages with ID-chip the option "Power Up Read ID-Chip" must be disabled by default to make the AutoZero result available in the future. See E-754 user manual for details.

E-761: Be aware that the result of the AutoZero procedure (new offset value) is automatically written to non-volatile memory (EPROM). For stages with ID-chip the

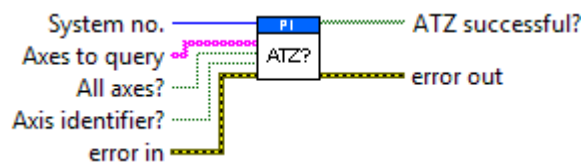
option "Read ID-Chip always" must be disabled by default to make the AutoZero result (new offset value) available in the future. See E-761 user manual for details.



#### 4.10.2. ATZ?.vi

Returns if ATZ was successful or not for queried axes.

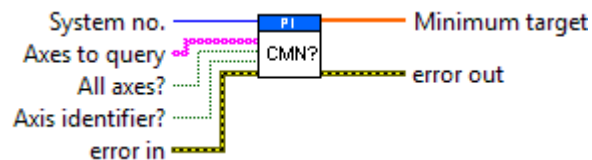
C-413, E-709, E-712, E-725, E-727, E-753, E-754: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.10.3. CMN?.vi

Returns the minimum commandable closed-loop target for the currently selected control mode.

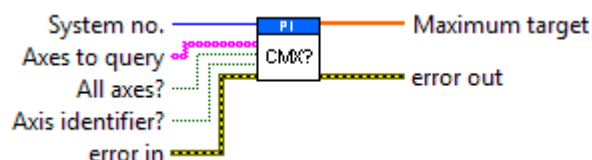
C-413: If All axes? = TRUE, then Axis identifier? can be FALSE.



#### 4.10.4. CMX?.vi

Returns the maximum commandable closed-loop target for the currently selected control mode.

C-413: If All axes? = TRUE, then Axis identifier? can be FALSE.



#### 4.10.5. DFH.vi

Defines current position of the "DFH axes" as home position. Their position value is set to 0. Due to the change of the home position the values for the travel range limits (TMN?, TMX?) are also changed accordingly. The home position is reset to default by ATZ.vi, REF.vi, MNL.vi, MPL.vi or similar command VIs.

C-867, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? must be TRUE.

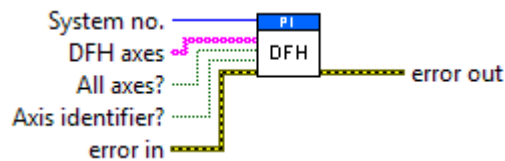
C-880K005: VI only supported when called through PI\_Multix.vi

E-517: If All axes? = TRUE, then Axis identifier? must be TRUE. The values are saved with WPA 100. The position shown in the E-517 display is not affected by DFH because it is that of the sensor channel (can also be queried with TSP?). DFH affects only the axis position which can be queried with POS?.

E-755: If All axes? = TRUE, then Axis identifier? must be TRUE. Command not available for E-755.101.

E-518, E-761: If All axes? = TRUE, then Axis identifier? must be TRUE. The values are saved with WPA 100.

All other systems: If All axes? = TRUE, then Axis identifier? must be TRUE



#### 4.10.6. DFH?.vi

Returns the difference between the current home position and the absolute or initial zero point (default home position) for each of the queried axes.

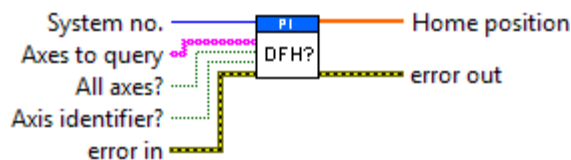
C-702, C-848, C-877, C-880, C-884, E-517, E-518, E-871, E-873, E-874, Hydra, Pollux, Mercury, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE

C-843, C-843.PM, C-844, C-865, C-866, E-710, E-761: If All axes? = TRUE, then Axis identifier? must be TRUE

C-867, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE.

C-880K005: VI only supported when called through PI\_Multix.vi

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101.



#### 4.10.7. FED.vi

Moves given axis to a given signal edge. If multiple axes are to command, they are moved synchronously. This command does not change the reference state of the axis and does not set a certain position value. Valid Parameter values depend on EdgeID:

EdgeID 1: negative limit switch, Parameter value is 0 when the default setting should be used (e.g. from pistages.dat), 1 when active high, -1 when active low

EdgeID 2: positive limit switch, Parameter value is 0 when the default setting should be used (e.g. from pistages.dat), 1 when active high, -1 when active low

EdgeID 3: reference switch, Parameter value is 0 when the default setting should be used (e.g. from pistages.dat), 1 when active high, -1 when active low

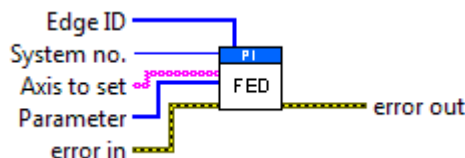
EdgeID 4: autofind AxisIn (is one input line of the motion chip carrying an external sensor signal which changes its state at a certain position), Parameter value gives the signal state to the left of the edge (high or low)

EdgeID 5: find arbitrary edge (i.e. change of the state of the AxisIn signal),  
Parameter value gives the direction of motion: 1=positive, -1= negative

C-843: EdgeID can be 1 to 3. Parameter value can be -1, 0 or 1, depending on EdgeID. The firmware detects the presence or absence of reference switch and limit switches using controller parameters (ID 0x14 for reference switch; ID 0x32 for limit switches). According to the values of those parameters, the controller enables or disables FED motions to the appropriate signal edges. Adapt the parameter values to your hardware using SPA.

C-867,C-877,C-884, E-861, E-871, E-873, E-874, Mercury\_GCS: EdgeID can be 1 to 3. Parameter value must be 0. The firmware detects the presence or absence of reference switch and limit switches using controller parameters (ID 0x14 for reference switch; ID 0x32 for limit switches). See user manual for details and specialities. According to the values of those parameters, the controller enables or disables FED motions to the appropriate signal edges. Adapt the parameter values to your hardware using SPA or SEP.

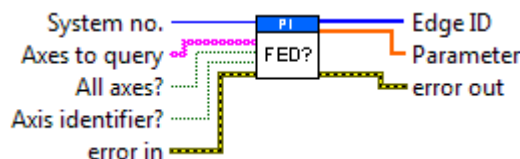
Hydra, Pollux: EdgeID can be 2 or 3. Parameter value must be 0. The firmware detects the presence or absence of reference switch and limit switches using controller parameters (ID 0x14 for reference switch; ID 0x32 for limit switches). See user manual for details and specialities. According to the values of those parameters, the controller enables or disables FED motions to the appropriate signal edges. Adapt the parameter values to your hardware using SPA (or SEP, if supported).



#### 4.10.8. FED?.vi

Get the parameters of the last Find Edge motion performed with FED.

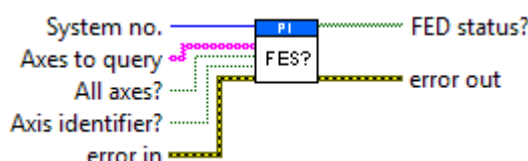
C-843: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.10.9. FES?.vi

Returns "finde edge" status. Indicates whether the last FED command was successful or not.

C-843: If All axes? = TRUE, then Axis identifier? can be FALSE





#### 4.10.10. FNL.vi

This VI starts a fast move to negative limit of the specified axes. Use #7 polling to determine end of this referencing procedure.

C-843, C-843.PM: If All axes? = TRUE, then Axis identifier? must be TRUE

C-867, C-884: If not, you can use FRF instead and set parameter no. 0x70 to select which sensor is to use for referencing. See controller user manual for details.

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101

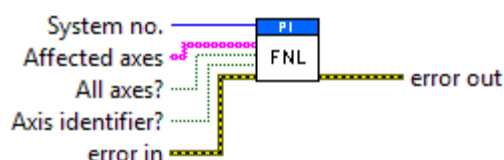
C-867, C-877, C-884, E-871, E-873, E-874, Hydra, Pollux, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? must be TRUE. Servo must be enabled with SVO for the commanded axis prior to using this command (closed-loop operation).

The reference mode must be set to "1" (factory default) with the RON command if referencing is to be done by performing a reference move.

The negative limit switch of the mechanics is used to determine the negative physical limit of the travel range. The difference of VALUE\_AT\_REF\_POS (parameter ID 0x16) and DISTANCE\_REF\_TO\_N\_LIM (parameter ID 0x17) is set as the current position when the axis is at the negative limit switch (value can be negative). If the soft limits (MAX\_TRAVEL\_RANGE\_POS, parameter ID 0x15, and MAX\_TRAVEL\_RANGE\_NEG, parameter ID 0x30) are used to reduce the travel range, the limit switches can not be used for reference moves. FNL and FPL commands will provoke an error message, and only the reference switch can be used for a reference move (FRF command). See user manual for details and specifics.

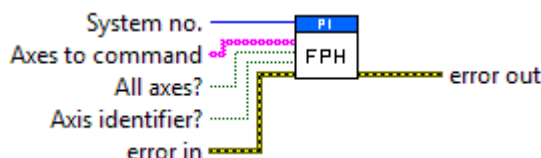
E-861: If All axes? = TRUE, then Axis identifier? must be TRUE. The reference mode must be set to "1" (factory default) with the RON command if referencing is to be done by performing a reference move.

The negative limit switch of the mechanics is used to determine the negative physical limit of the travel range. The difference of VALUE\_AT\_REF\_POS (parameter ID 0x16) and DISTANCE\_REF\_TO\_N\_LIM (parameter ID 0x17) is set as the current position when the axis is at the negative limit switch (value can be negative). If the soft limits (MAX\_TRAVEL\_RANGE\_POS, parameter ID 0x15, and MAX\_TRAVEL\_RANGE\_NEG, parameter ID 0x30) are used to reduce the travel range, the limit switches can not be used for reference moves. FNL and FPL commands will provoke an error message, and only the reference switch can be used for a reference move (FRF command).



#### 4.10.11. FPH.vi

Find phase offset. See controller manual for details.

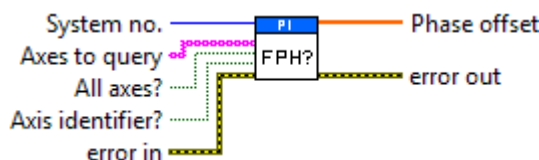




#### 4.10.12. FPH?.vi

Returns phase offset for queried axes.

All systems: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.10.13. FPL.vi

This VI starts a fast move to positive limit of the specified axes. Use #7 polling to determine end of this referencing procedure.

C-843, C-843.PM: If All axes? = TRUE, then Axis identifier? must be TRUE

C-867, C-884: If not, you can use FRF instead and set parameter no. 0x70 to select which sensor is to use for referencing. See controller user manual for details.

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101

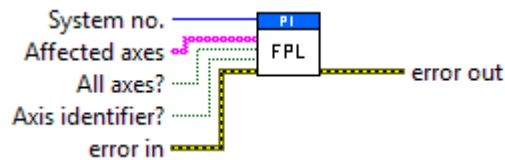
C-867, C-877, C-884, E-871, E-873, E-874, Hydra, Pollux, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? must be TRUE. Servo must be enabled with SVO for the commanded axis prior to using this command (closed-loop operation).

The reference mode must be set to "1" (factory default) with the RON command if referencing is to be done by performing a reference move.

The positive limit switch of the mechanics is used to determine the positive physical limit of the travel range. The sum of VALUE\_AT\_REF\_POS (parameter ID 0x16) and DISTANCE\_REF\_TO\_P\_LIM (parameter ID 0x2F) is set as the current position when the axis is at the positive limit switch. If the soft limits (MAX\_TRAVEL\_RANGE\_POS, parameter ID 0x15, and MAX\_TRAVEL\_RANGE\_NEG, parameter ID 0x30) are used to reduce the travel range, the limit switches can not be used for reference moves. FNL and FPL commands will provoke an error message, and only the reference switch can be used for a reference move (FRF command). See user manual for details and specifics.

E-861: If All axes? = TRUE, then Axis identifier? must be TRUE. The reference mode must be set to "1" (factory default) with the RON command if referencing is to be done by performing a reference move.

The positive limit switch of the mechanics is used to determine the positive physical limit of the travel range. The sum of VALUE\_AT\_REF\_POS (parameter ID 0x16) and DISTANCE\_REF\_TO\_P\_LIM (parameter ID 0x2F) is set as the current position when the axis is at the positive limit switch. If the soft limits (MAX\_TRAVEL\_RANGE\_POS, parameter ID 0x15, and MAX\_TRAVEL\_RANGE\_NEG, parameter ID 0x30) are used to reduce the travel range, the limit switches can not be used for reference moves. FNL and FPL commands will provoke an error message, and only the reference switch can be used for a reference move (FRF command).



#### 4.10.14. FRF.vi

This VI starts a fast referencing of the specified axes. Use #7 polling to determine end of this referencing procedure.

C-843, C-843.PM, C-880: If All axes? = TRUE, then Axis identifier? must be TRUE

C-880K005: VI only supported when called through PI\_Multix.vi

C-413, C-867, C-877, C-884, E-871, E-873, E-874, Hydra, Pollux, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? must be TRUE. Servo must be enabled with SVO for the commanded axis prior to using this command (closed-loop operation).

The reference mode must be set to "1" (factory default) with the RON command if referencing is to be done by performing a reference move.

The value of the VALUE\_AT\_REF\_POS parameter (ID 0x16) is set as the current position when the axis is at the reference switch. Use FNL or FPL (if supported) instead of FRF to perform a reference move for an axis which has no reference sensor but limit switches. See user manual for details and specifics.

C-885: If All axes? = TRUE, then Axis identifier? can be FALSE. Servo must be enabled with SVO for the commanded axis prior to using this command (closed-loop operation).

The reference mode must be set to "1" (factory default) with the RON command if referencing is to be done by performing a reference move.

C-886: If All axes? = TRUE, then Axis identifier? can be FALSE. Servo must be enabled with SVO for additional axes prior to using this command (closed-loop operation).

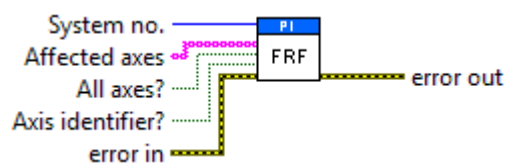
C-891: If All axes? = TRUE, then Axis identifier? must be TRUE. The system mechanics Reference type can be set by changing the parameter ID 0x70. For details please refer to the Controller user manual.

E-712: If All axes? = TRUE, then Axis identifier? must be TRUE.

E-861: If All axes? = TRUE, then Axis identifier? must be TRUE. The reference mode must be set to "1" (factory default) with the RON command if referencing is to be done by performing a reference move.

The value of the VALUE\_AT\_REF\_POS parameter (ID 0x16) is set as the current position when the axis is at the reference switch. Use FNL or FPL (if supported) instead of FRF to perform a reference move for an axis which has no reference sensor but limit switches.

C-887, F-206, M-8X0: If All axes? = TRUE, then Axis identifier? must be TRUE. Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For GCS syntax version = GCS 1.0 use INI.vi for initialization.



#### 4.10.15. FRF?.vi

Indicates whether queried axes have been referenced (using REF, FNL, FPL, FRF, MPL, MNL, or - if reference mode is OFF - using POS) successfully or not.

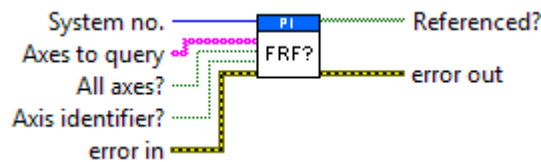
C-413, C-843, C-843.PM, C-866, C-867, C-877, C-880, C-884, C-885, C-886, C-891, E-754, E-861, E-871, E-873, E-874, Hydra, Pollux, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE

C-880K005: VI only supported when called through PI\_Multix.vi

E-712: If All axes? = TRUE, then Axis identifier? can be FALSE.

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101

C-887, F-206, M-8X0: If All axes? = TRUE, then Axis identifier? can be FALSE. Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

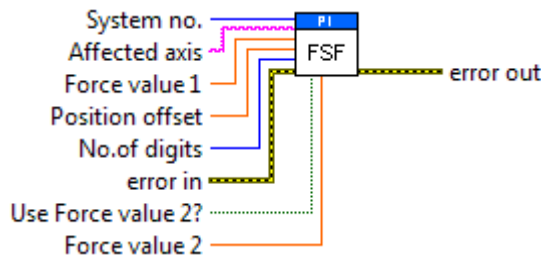


#### 4.10.16. FSF.vi

Performs a find surface procedure.

Approaches the surface with force value 1. After making contact with the surface the parameter for surface position is set. The parameter for pre-position is set accordingly to the value of position offset.

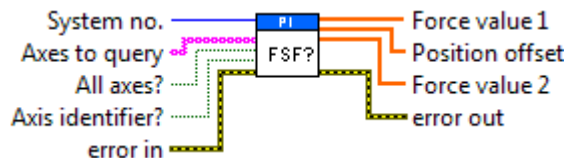
C-413:



#### 4.10.17. FSF?.vi

Returns parameters used for last find surface procedure (FSF).

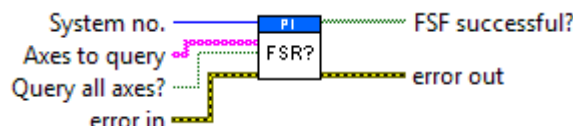
C-413:



#### 4.10.18. FSR?.vi

Indicates whether the find surface procedure for queried axes was successful.

C-413:



#### 4.10.19. GOH.vi

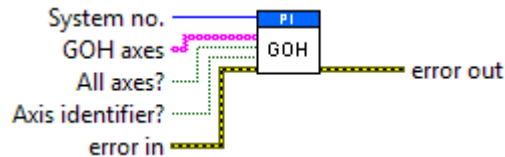
Moves specified axes to their home positions.

C-702, C-843, C-843.PM, C-844, C-848, C-865, C-866, C-867, C-877, C-880, E-710, E-861, E-871, E-873, E-874, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? must be TRUE

C-880K005: VI only supported when called through PI\_Multix.vi

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101.

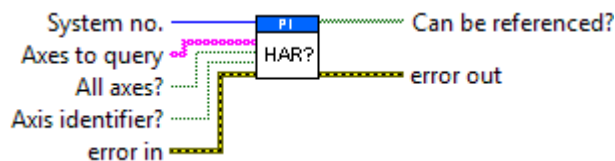
E-517, E-518, E-761, Mercury: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.10.20. HAR?.vi

Indicates if given axis can be referenced with FNL / FPL using the mechanics hard stop.

E-755, E-871, E-873, E-874: If All axes? = TRUE, then Axis identifier? can be FALSE



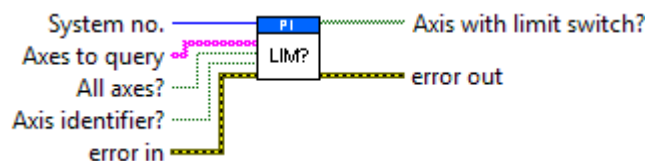
#### 4.10.21. LIM?.vi

Indicates whether queried axes have limit switches or not.

All systems: If All axes? = TRUE, then Axis identifier? can be FALSE

E-712:

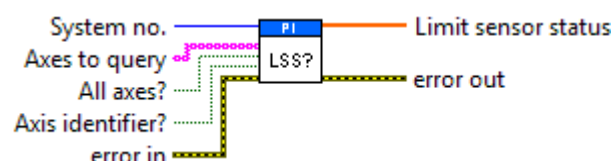
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.10.22. LSS?.vi

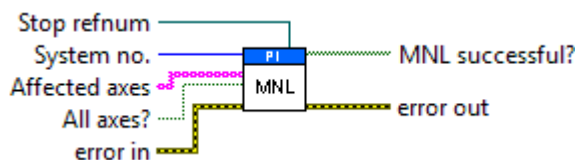
Returns limit sensor status for queried axes.

C-880K005: VI only supported when called through PI\_Multix.vi



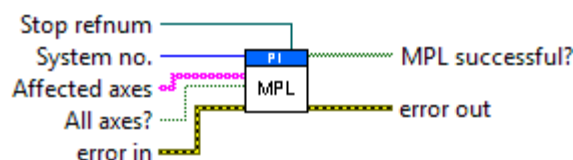
#### 4.10.23. MNL.vi

Moves specified axes to the negative limit switch, waits until this position is reached using #7 polling and indicates whether this was successful or not. VI will also stop if Stop refnum or Local stop is TRUE.



#### 4.10.24. MPL.vi

Moves specified axes to the positive limit switch, waits until this position is reached using #7 polling and indicates whether this was successful or not. VI will also stop if Stop refnum or Local stop is TRUE.



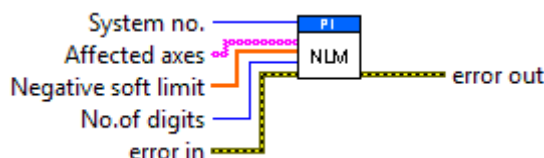
#### 4.10.25. NLM.vi

Sets negative soft limit for the specified axes. No. of digits is the number of digits after the decimal point in the limit values that will be sent.

E-516, E-517, E-518: The NLM command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the device is powered off.

E-755: Command not available for E-755.101

E-761: The NLM command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi with "Affected axes" as an empty array. Parameter changes not saved with WPA will be lost when the E-761 board is powered off or rebooted.



#### 4.10.26. NLM?.vi

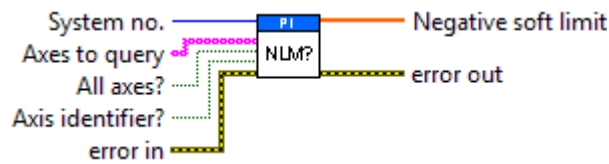
Returns negative soft limit of queried axes.

C-702, C-844, C-848, C-880, E-517, E-518, E-761: If All axes? = TRUE, then Axis identifier? can be FALSE

E-516: If All axes? = TRUE, then Axis identifier? must be TRUE

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101

C-887, F-206, M-8X0: If All axes? = TRUE, then Axis identifier? can be FALSE.



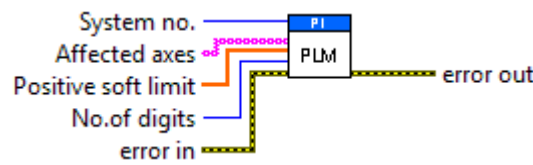
#### 4.10.27. PLM.vi

Sets positive soft limit for the specified axes. No. of digits is the number of digits after the decimal point in the limit values that will be sent.

E-516, E-517, E-518: The PLM command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the device is powered off.

E-755: Command not available for E-755.101

E-761: The PLM command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi with "Affected axes" as an empty array. Parameter changes not saved with WPA will be lost when the E-761 board is powered off or rebooted.



#### 4.10.28. PLM?.vi

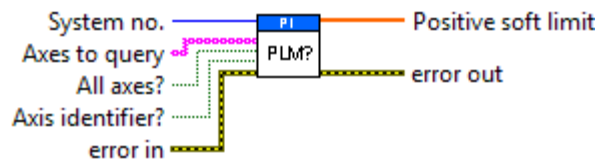
Returns positive soft limit of queried axes.

C-702, C-844, C-848, C-880, E-517, E-518, E-761: If All axes? = TRUE, then Axis identifier? can be FALSE

E-516: If All axes? = TRUE, then Axis identifier? must be TRUE

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101

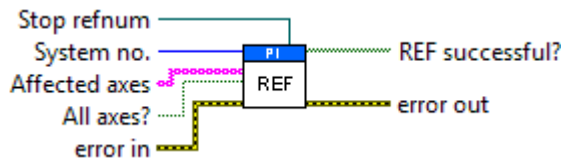
C-887, F-206, M-8X0: If All axes? = TRUE, then Axis identifier? can be FALSE.



#### 4.10.29. REF.vi

Moves the specified axes to the reference position, waits until this position is reached (polling with #7), and indicates whether referencing was successful or not. VI will also stop if Stop refnum or Local stop is TRUE.

C-880K005: VI only supported when called through PI\_Multix.vi



#### 4.10.30. REF?.vi

Indicates whether queried axes have a reference switch or not.

C-702: If All axes = TRUE, then Axis identifier can be FALSE

C-843: If All axes = TRUE, then Axis identifier must be TRUE

C-843.PM: If All axes = TRUE, then Axis identifier must be TRUE

C-844: If All axes = TRUE, then Axis identifier must be TRUE

C-848: If All axes = TRUE, then Axis identifier can be FALSE

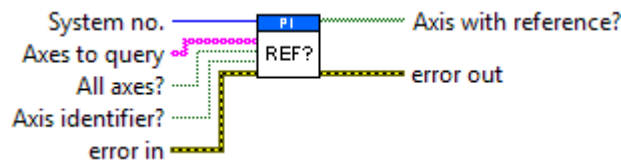
C-865: If All axes = TRUE, then Axis identifier must be TRUE

C-866: If All axes = TRUE, then Axis identifier must be TRUE

C-880: If All axes = TRUE, then Axis identifier can be FALSE

C-880K005: VI only supported when called through PI\_Multix.vi

Mercury: If All axes = TRUE, then Axis identifier can be FALSE



#### 4.10.31. RON.vi

Sets reference mode for given axes. If Without axis ID is TRUE, then Axes to command is ignored and first field of the Reference mode array is used for the reference mode.

If the reference mode of an axis is ON, the axis must be driven to the reference switch (using REF.vi or FRF.vi) or, if no reference switch is available, to a limit switch (positive limit switch: using MPL.vi or FPL.vi; negative limit switch: using MNL.vi or FNL.vi) before any other motion can be commanded in closed-loop operation.

If reference mode is OFF, no referencing is required for the axis. In closed-loop operation, only relative moves can be commanded (using MVR.vi), unless the actual position is set with POS.vi. Afterwards, relative and absolute moves can be commanded.

For stages with neither reference nor limit switch, reference mode is automatically OFF.

##### WARNINGS:

If reference mode is switched off, and relative moves are commanded, stages can be driven into the mechanical hard stop if moving to a position which is outside the travel range!

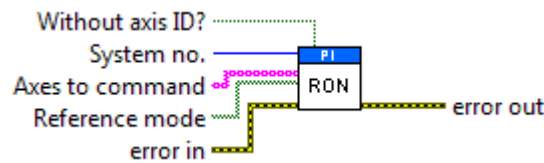
If reference mode is switched off, and the actual position is incorrectly set with POS.vi, stages can be driven into the mechanical hard stop when moving to a position which is thought to be within the travel range of the stage, but actually is not.

C-880K005: VI only supported when called through PI\_Multix.vi



E-712: Check HLP? answer to find out if RON is supported and E-712 user manual to find out for which type of stages RON can be used.

All other systems: Without axis ID = FALSE



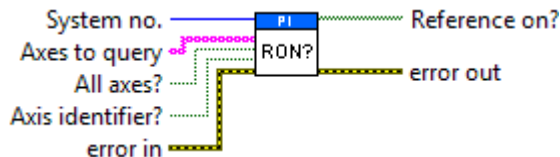
#### 4.10.32. RON?.vi

Indicates whether queried axes have reference mode ON or OFF. See RON.vi above for description of reference mode.

C-880K005: VI only supported when called through PI\_Multix.vi

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

All other systems: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.10.33. SSL.vi

Sets soft limits on or off (set by NLM.vi and PLM.vi) for specified axes. If Without axis ID is TRUE, then Affected axes is ignored and first field of Soft limit mode array is used.

C-702: Without axis ID = FALSE

C-844: Without axis ID = FALSE

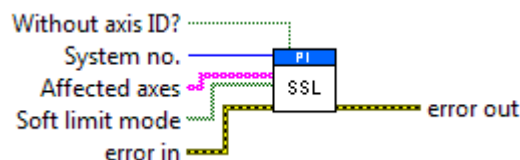
C-848: Without axis ID = FALSE

C-880: Without axis ID = FALSE

E-755: Without axis ID = FALSE. Command not available for E-755.101.

C-886: Without axis ID = FALSE.

C-887, F-206, M-8X0: Without axis ID = FALSE.



#### 4.10.34. SSL?.vi

Returns soft-limit status for queried axes.

C-702: If All axes = TRUE, then Axis identifier can be FALSE

C-844: If All axes = TRUE, then Axis identifier can be FALSE

C-848: If All axes = TRUE, then Axis identifier can be FALSE

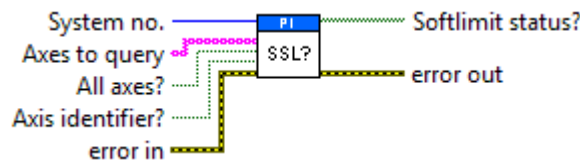
C-880: If All axes = TRUE, then Axis identifier can be FALSE

C-886: If All axes = TRUE, then Axis identifier can be FALSE



E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101.

C-887, F-206, M-8X0: If All axes? = TRUE, then Axis identifier? can be FALSE.



#### 4.10.35. TMN?.vi

Returns minimum (low-end) travel limit (if present, position of negative limit switch, or value of negative soft limit, if set, whichever is higher).

Analog, C-413, C-702, C-848, C-865, C-866, C-867, C-877, C-880, C-884, C-885, C-887, C-891, E-517, E-518, E-709, E-712, E-725, E-727, E-753, E-754, E-761, E-861, E-871, E-873, E-874, F-206, Hydra, Pollux, M-8X0, Mercury, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE

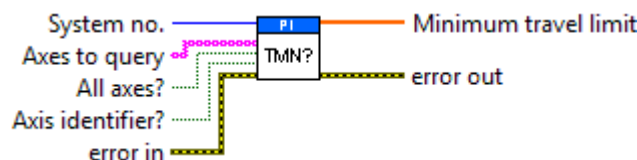
C-843, C-843.PM, C-844, E-710: If All axes? = TRUE, then Axis identifier? must be TRUE

C-413, C-867, C-877, C-884, C-891, E-861, E-871, E-873, E-874, Hydra, Pollux, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE. The minimum commandable position is defined by the MAX\_TRAVEL\_RANGE\_NEG parameter ID 0x30 (SPA).

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101.

E-517, E-761: If All axes? = TRUE, then Axis identifier? can be FALSE. Get the maximum accessible position value, i.e. the value of the "Range max limit" parameter (ID 0x07000001). Note: The maximum position which can be commanded depends either on the "Range max limit" parameter or-if it is smaller than the "Range max limit" parameter value- on the value of the positive soft limit set with PLM.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). TMN? returns the minimum travel range of the axis with all other axes positions being zero. If this is not the case, the available travel range may be less.



#### 4.10.36. TMX?.vi

Returns maximum (high-end) travel limit (if present, position of positive limit switch or value of positive soft limit, if set, whichever is lower).

Analog, C-413, C-702, C-848, C-865, C-866, C-867, C-877, C-880, C-884, C-885, C-887, C-891, E-517, E-518, E-709, E-712, E-725, E-727, E-753, E-754, E-761, E-861, E-871, E-873, E-874, F-206, Hydra, Pollux, M-8X0, Mercury, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE

C-843, C-843.PM, C-844, E-710: If All axes? = TRUE, then Axis identifier? must be TRUE

C-413, C-867, C-877, C-884, E-861, E-871, E-873, E-874, Hydra, Pollux, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE. The maximum commandable position is defined by the MAX\_TRAVEL\_RANGE\_POS parameter ID 0x15 (SPA).

E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Command not available for E-755.101.

E-517, E-761: If All axes? = TRUE, then Axis identifier? can be FALSE. Get the maximum accessible position value, i.e. the value of the "Range max limit" parameter (ID 0x07000001). Note: The maximum position which can be commanded depends either on the "Range max limit" parameter or-if it is smaller than the "Range max limit" parameter value- on the value of the positive soft limit set with PLM.

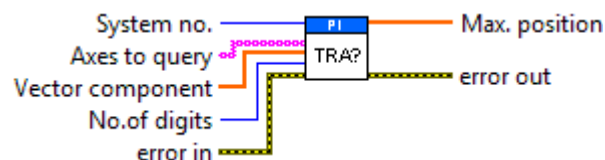
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). TMX? returns the maximum travel range of the axis with all other axes positions being zero. If this is not the case, the available travel range may be less.



#### 4.10.37. TRA?.vi

Returns the maximum absolute position which can be reached from the current position in the given direction. The direction is defined by a vector (X, Y, Z, U, V, W) = (x, y, z, u, v, w). Softlimits are taken into account. In case the parallel kinematics positioner is in motion, an error will be set.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



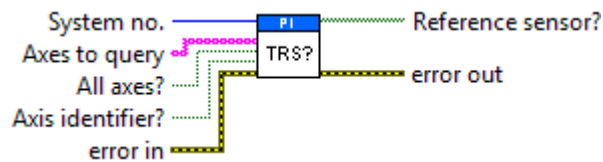
#### 4.10.38. TRS?.vi

Indicates whether or not given axes have a reference sensor with direction sensing.

C-413, C-867, C-877, C-884, C-891, E-755, E-861, E-871, E-873, E-874, Hydra, Pollux, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE

E-712: If All axes? = TRUE, then Axis identifier? can be FALSE.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). If All axes? = TRUE, then Axis identifier? can be FALSE.



## 4.11. Macros.IIb

### 4.11.1. #8.vi

Sends ASCII #8 without Linefeed and returns Macro running? indicating whether a macro is running or not.

E-816: This command cannot be issued to a slave. Check controller manual to find out if #8 is supported.

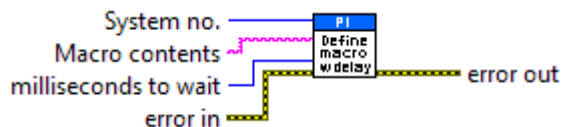
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



### 4.11.2. Define macro contents with delay.vi

Defines macro contents. Each command to be stored in the macro must be written on one line, terminated with the enter key. MAC BEG.vi must be called before running this VI and MAC END.vi must be called afterwards. VI sends Macro contents one line after another separated by the specified wait time.

E-516: Macro will not be saved to FLASH until WPA .vi was run. Changes not saved with WPA are only present in RAM and will be lost when the E-516 is powered off.

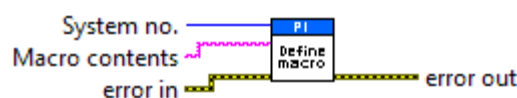


### 4.11.3. Define macro contents.vi

Defines macro contents. Each command to be stored in the macro must be written on one line, terminated with the enter key. MAC BEG.vi must be called before running this VI and MAC END.vi must be called afterwards.

E-816: Macro will not be saved to FLASH until WPA .vi was run. Changes not saved with WPA are only present in RAM and will be lost when the controller is powered off. This command cannot be issued to a slave. Check controller manual to find out if MAC is supported.

M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.11.4. MAC BEG.vi

Begin macro recording. Because controller will not answer VI queries during macro recording phase, command VIs cannot be run after this VI to define the macro.

Run Define macro contents.vi and finish with MAC END.vi to define a macro.

C-702, C-848, C-867, C-877, C-880, C-884, C-886, C-887, C-891, E-516, E-517, E-518, E-727, E-754, E-816, E-871, E-872, E-873, E-874, F-206, M-8X0, Mercury\_GCS: Macro name must be between 1 and 8 characters

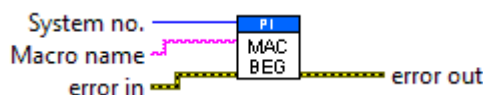
E-516: Macro name must be between 3 and 8 characters. Macro will not be saved to FLASH until WPA .vi was run. Changes not saved with WPA are only present in RAM and will be lost when the E-516 is powered off.

E-816: Macro name must be between 3 and 8 characters. Macro will not be saved to FLASH until WPA .vi was run. Changes not saved with WPA are only present in RAM and will be lost when the controller is powered off. This command cannot be issued to a slave. Check controller manual to find out if MAC BEG is supported.

C-887, F-206: Macro name must be between 3 and 8 characters.

Mercury: For valid macro names see GCS DLL Manual.

C-887, M-8X0: Macro name must be between 3 and 8 characters. Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



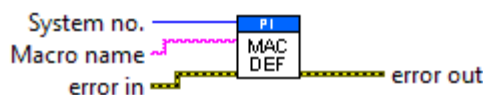
#### 4.11.5. MAC DEF.vi

Define autostart macro. To disable the autostart macro, "Macro name" must be an empty string.

E-516: Autostart macro definition will not be saved to FLASH until WPA.vi was run. Changes not saved with WPA are only present in RAM and will be lost when the controller is powered off.

E-816: Autostart macro definition will not be saved to FLASH until WPA.vi was run. Changes not saved with WPA are only present in RAM and will be lost when the controller is powered off. This command cannot be issued to a slave. Check controller manual to find out if MAC DEF is supported.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

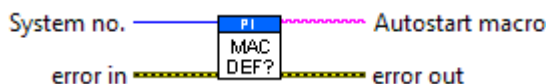


#### 4.11.6. MAC DEF?.vi

Get name of autostart macro.

E-816: This command cannot be issued to a slave. Check controller manual to find out if MAC DEF? is supported.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



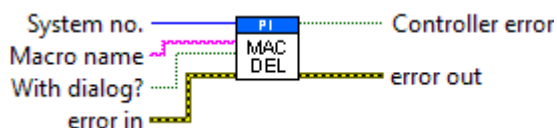
#### 4.11.7. MAC DEL.vi

Delete macro. If "With dialog" is TRUE, a dialog box pops up to confirm the deletion. "Controller error" is TRUE if selected system reports an error code which is not zero.

E-516: Changes not saved with WPA.vi are only present in RAM and will be lost when the controller is powered off.

E-816: Changes not saved with WPA.vi are only present in RAM and will be lost when the controller is powered off. This command cannot be issued to a slave. Check controller manual to find out if MAC BEG is supported.

M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



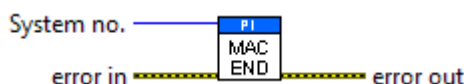
#### 4.11.8. MAC END.vi

Stops current macro recording.

E-516: Macro will not be saved to FLASH until WPA.vi was run. Changes not saved with WPA are only present in RAM and will be lost when the controller is powered off.

E-816: Macro will not be saved to FLASH until WPA .vi was run. Changes not saved with WPA are only present in RAM and will be lost when the controller is powered off. This command cannot be issued to a slave. Check controller manual to find out if MAC END is supported.

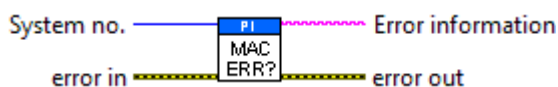
M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.11.9. MAC ERR?.vi

Returns error information from macro execution

C-867, C-887, F-206, M-8X0:



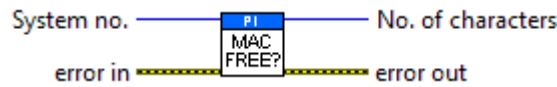
#### 4.11.10. MAC FREE?.vi

Returns the amount of free memory for macro recording (in number of characters).

C-867:

E-816: This command cannot be issued to a slave. Check controller manual to find out if MAC FREE? is supported.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

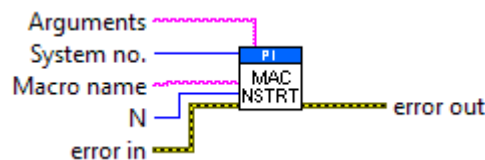


#### 4.11.11. MAC NSTART.vi

Start macro N times.

E-816: This command cannot be issued to a slave. Check controller manual to find out if MAC NSTART is supported. Arguments is not supported.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). Use #8.vi to determine when macro execution has finished.



#### 4.11.12. MAC START.vi

Start macro.

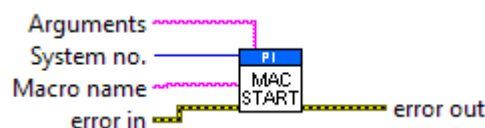
C-702, C-848, C-867, C-877, C-880, C-886, C-891, E-516, E-517, E-518, E-727, E-754, E-861 E-871, E-872, E-873, E-874: Use #8.vi to determine when macro execution has finished. For C-702, C-848, C-880, E-516 and E-517, E-518 Arguments is not supported.

E-816: Use #8.vi to determine when macro execution has finished. This command cannot be issued to a slave. Check controller manual to find out if MAC START is supported. Arguments is not supported.

C-887, F-206: Check HLP?/HELP answer to find out if MAC START and #8 are supported. If yes, use #8.vi to determine when macro execution has finished.

C-887, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). Use #8.vi to determine when macro execution has finished.

Mercury: #8.vi is not supported.

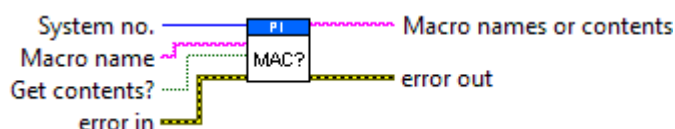


#### 4.11.13. MAC?.vi

If Get contents? is FALSE, returns names of all available macros, if TRUE, returns contents of one specified macro.

E-816: This command cannot be issued to a slave. Check controller manual to find out if MAC? is supported.

M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

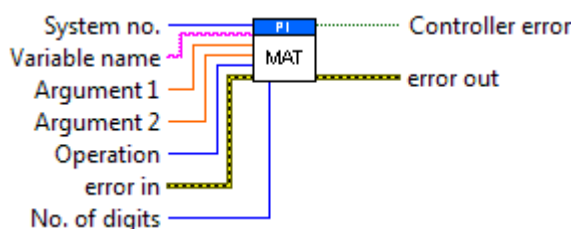


#### 4.11.14. MAT.vi

Carries out a mathematical operation or bit operation and saves the result as a variable.

See controller user manual for details regarding local and global variables.

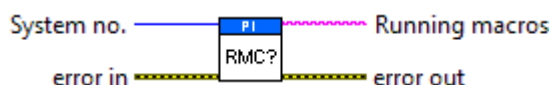
C-884:



#### 4.11.15. RMC?.vi

List macros currently running.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

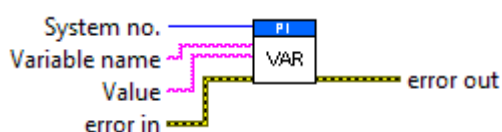


#### 4.11.16. VAR.vi

Sets a variable to a certain value. A variable is present in RAM only. See controller user manual for details regarding local and global variables.

C-887, F-206, M-8X0: For GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

C-867, Mercury\_GCS :



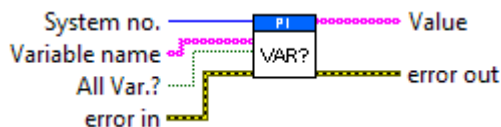


#### 4.11.17. VAR?.vi

Returns the value of given variable name(s). See controller user manual for details regarding local and global variables.

C-887, F-206, M-8X0: For GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

C-867, Mercury\_GCS :



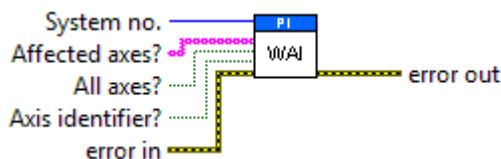
#### 4.11.18. WAI.vi

This VI sends the wait command WAI to the controller. Controller will wait for specified axes to stop before executing any other command. Since the controller does not indicate when the wait phase has finished, WAI.vi does not provide that information.

C-702: If All axes = TRUE, then Axis identifier must be TRUE

C-848: If All axes = TRUE, then Axis identifier must be TRUE

C-880: If All axes = TRUE, then Axis identifier must be TRUE

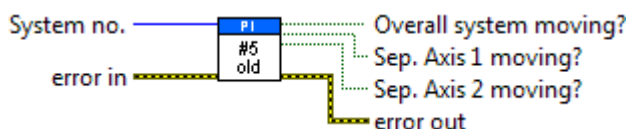


### 4.12. Old commands.llb

#### 4.12.1. #5\_old.vi

Polls the motion status of the F-206/M-8X0 and/or up to 2 additional connected axes by sending the single ASCII character 5. Required by General wait for movement to stop.vi.

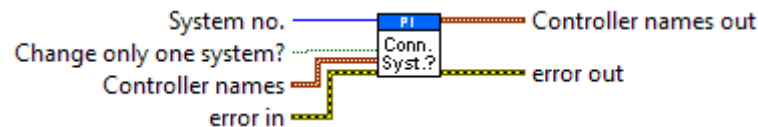
M-8X0: Only answer for overall system is valid.



#### 4.12.2. Define connected systems.vi

Defines connected systems and writes controller names into Global2.vi. This VI is called automatically by PI\_GCS2\_Configuration\_Setup.vi and must be completed successfully before General wait for movement to stop.vi is called. If Change only one system? is FALSE, all four entries from Controller names are written into Global2.vi. If Change only one system? is TRUE, only the entry for the given system number is overwritten in Global2.vi.

Old VI - only for compatibility reasons available. Limited to 4 systems. Use Define connected systems (Array).vi instead.



#### 4.12.3. HELP.vi

Returns help string.

F-206, M-8X0: Only for GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For GCS 2.0 use HLP?.vi instead.



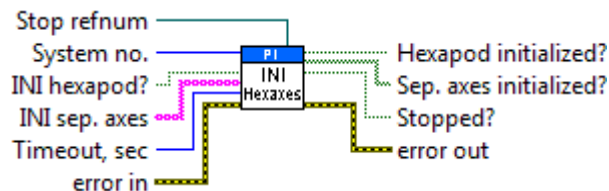
#### 4.12.4. INI hexaxes and wait until finished.vi

Initializes hexapod and/or additional axes and waits until procedure has finished, Stop refnum or Local stop is TRUE or a timeout condition has occurred. When using as a sub-VI, use Stop refnum to stop VI from caller.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

GCS 1.0: Since version 4.0.0 of the F-206 firmware and version 4.5.0 of the M-8X0 firmware it is possible to stop the ini procedure by sending #24.

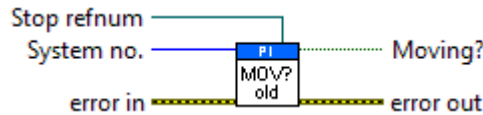
GCS 2.0 or higher: VI calls FRF instead for compatibility.



#### 4.12.5. MOV?\_old.vi

Sends MOV? and waits for response of 0 (not moving). In case of no answer, VI stops after 60 s. VI also stops if Local stop or Stop refnum is TRUE. When using as a sub-VI, use Stop refnum to stop VI from caller.

F-206, M-8X0: Only for GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

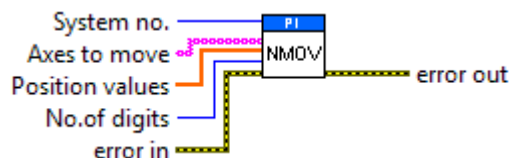


#### 4.12.6. NMOV.vi

Moves specified axes to specified absolute positions. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

E-710: Caution: The use of NMOV.vi is the same as that of the GCS command VI MOV.vi, but it is not a GCS command VI! This function is faster than MOV.vi, but does not check range limits and servo states and does not move the axes synchronously. When the commanded target is outside the range limit, the axis will stop at its physical limit. To go back to normal operation, command the axis to a valid position using MOV.vi. When servo is off, the axis does not move. The commanded target can be queried with MOV?.vi.

Note: Due to the emulation of the native E-710 command set, the execution of MOV.vi is noticeably slower than that of the native firmware commands. Therefore NMOV.vi is provided for applications which require quickest possible response to motion commands.

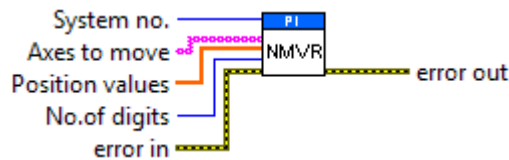


#### 4.12.7. NMVR.vi

Moves specified axes relative to current position. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent. Uses native move command of controller without any security checks.

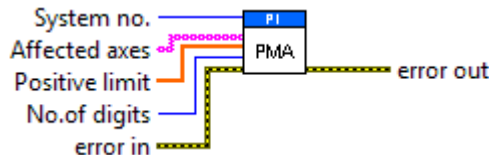
E-710: Caution: The use of NMVR.vi is the same as that of the GCS command VI MVR.vi, but it is not a GCS command VI! This function is faster than MVR.vi, but does not check range limits and servo states and does not move the axes synchronously. When the commanded target is outside the range limit, the axis will stop at its physical limit. To go back to normal operation, command the axis to a valid position using MOV.vi. When servo is off, the axis does not move. The commanded target can be queried with MOV?.vi.

Note: Due to the emulation of the native E-710 command set, the execution of MVR.vi is noticeably slower than that of the native firmware commands. Therefore NMVR.vi is provided for applications which require quickest possible response to motion commands.



#### 4.12.8. PMA.vi

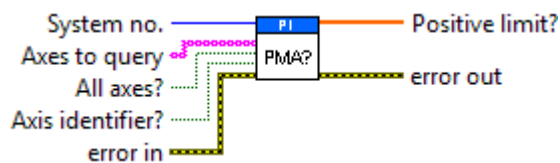
Sets the maximum position limit for specified axes. No. of digits is the number of digits after the decimal point in the limit values that will be sent. Old command-please use PLM if the system supports that command.



#### 4.12.9. PMA?.vi

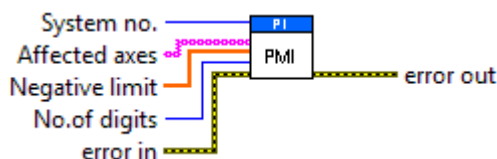
Returns maximum position limit of queried axes. Old command-please use PLM? if the system supports that command.

E-516: If All axes = TRUE, then Axis identifier must be TRUE



#### 4.12.10. PMI.vi

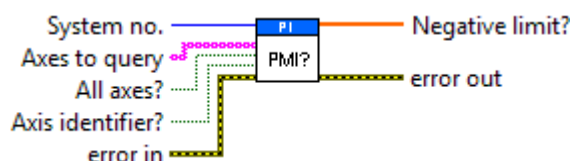
Sets the minimum position limit for specified axes. No. of digits is the number of digits after the decimal point in the limit values that will be sent. Old command-please use NLM if the system supports that command.



#### 4.12.11. PMI?.vi

Returns minimum position limit of queried axes. Old command-please use NLM? if the system supports that command.

E-516: If All axes = TRUE, then Axis identifier must be TRUE

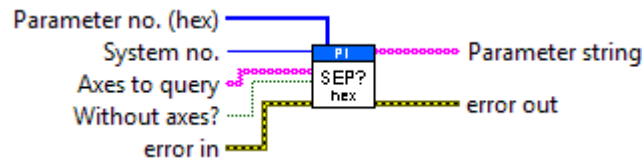


#### 4.12.12. SEP?\_Hex.vi

Returns parameter string from non-volatile memory for queried axis and hex. parameter number. Old VI - please use SEP?.vi instead.

E-710: See E-710 GCS DLL manual for a detailed list of available parameter numbers. Without axes? can be TRUE and will return all available parameter for all axes.

E-761: See E-761 GCS DLL manual for a detailed list of available parameter numbers. Without axes? can be TRUE and will return all available parameter for all axes.

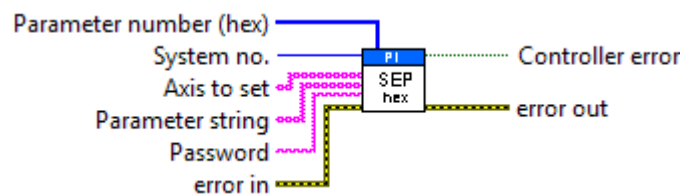


#### 4.12.13. SEP\_Hex.vi

If password is correct, writes specified parameter values for specified hex. parameters of specified axes/channels to EPROM, waits 100 ms and queries ERR?. Controller error is TRUE if selected system reports an error code which is not 0. Old VI - please use SEP.vi instead.

E-710: See E-710 GCS DLL Manual for a detailed list of available parameter numbers. Command is available in command level 1 only (see CCL.vi and CCL?.vi)

E-761: See E-761 GCS DLL Manual for a detailed list of available parameter numbers.



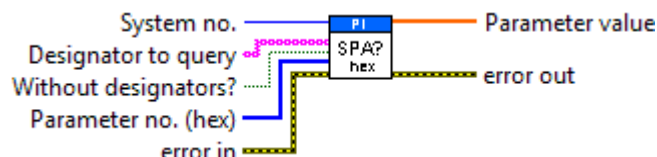
#### 4.12.14. SPA?\_Hex.vi

Returns parameter values for queried designators and hex. parameter numbers. See GCS DLL manual for available parameter numbers. For axis-related parameters, Designator to query is the axis name; for piezo- or sensor-related parameters, the channel number, otherwise a parameter-related one. If "Without designators?" is TRUE, all available parameters for all designators are returned. For parameter numbers which output a string use SPA?\_hex\_String.vi.

Old VI - please use SPA?.vi instead.

E-710: Use HPA?.vi to get valid parameter numbers and see GCS DLL manual for a description of valid parameter numbers.

E-761: See GCS DLL manual for a description of valid parameter numbers.

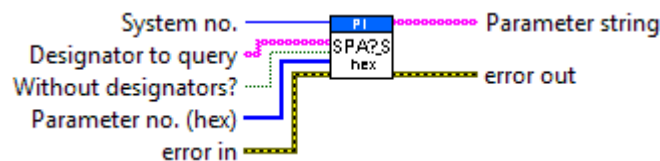


#### 4.12.15. SPA?\_Hex\_String.vi

Returns parameter strings for queried designators and hex. parameter numbers. See GCS DLL manual for available parameter numbers. For axis-related parameters, Designator to query is the axis name; for piezo- or sensor-related parameters, the channel number, otherwise a parameter-related one. If "Without designators?" is TRUE, all available parameters for all designators are returned. For parameter numbers which output a numerical value use SPA?\_hex.vi.

Old VI - please use SPA?.vi instead.

E-710: Use HPA?.vi to get valid parameter numbers.



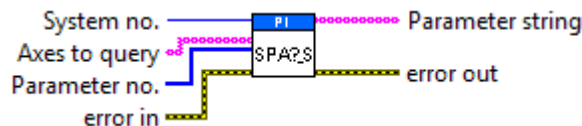
#### 4.12.16. SPA?\_String.vi

Returns parameter strings for queried axes and parameter numbers. See SPA\_String.vi for available parameter numbers. For parameter numbers which output a numerical value use SPA?.vi.

Old VI - please use SPA?.vi instead.

C-843, C-843.PM, C-865: the following parameter number is valid:

60: stage name (maximum 14 characters)



#### 4.12.17. SPA\_Hex.vi

Sets parameters in hex format for commanded designators, waits 100 ms and queries ERR?. See system manual for available parameter numbers and values. No. of digits is the number of digits after the decimal point in the parameter value(s) that will be sent. For axis-related parameters, Designator to set is the axis name; for piezo- or sensor-related parameters, the channel number, otherwise a parameter-related one. Controller error is TRUE if selected system reports an error code which is not 0.

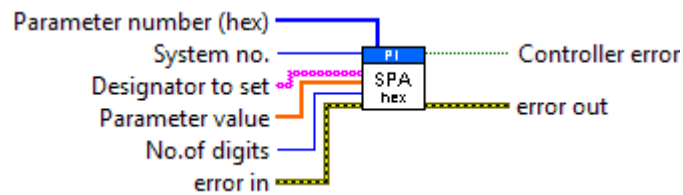
Old VI - please use SPA.vi instead.

E-710:

Please refer to the GCS DLL manual for valid parameter numbers and the stage manual for valid parameter settings.

E-761:

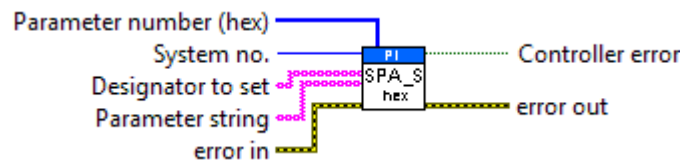
Please refer to the GCS DLL manual for valid parameter numbers and the stage manual for valid parameter settings.



#### 4.12.18. SPA\_Hex\_String.vi

Sets string parameters for commanded designators, waits 100 ms and queries ERR?. For axis-related parameters, Designator to set is the axis name; for piezo- or sensor-related parameters, the channel number, otherwise a parameter-related one. Controller error is TRUE if selected system reports an error code which is not 0. For parameter numbers which require a numerical value as input use SPA\_hex.vi. See GCS DLL manual for available parameter numbers.

Old VI - please use SPA.vi instead.



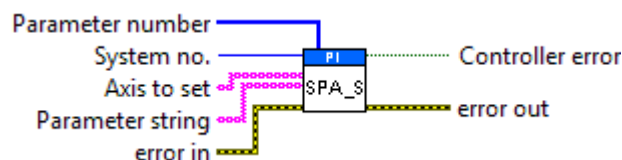
#### 4.12.19. SPA\_String.vi

Sets string parameters for commanded axes, waits 100 ms and queries ERR?. Controller error is TRUE if selected system reports an error code which is not 0. For parameter numbers which require a numerical value as input use SPA.vi.

Old VI - please use SPA.vi instead.

C-843, C-843.PM, C-865: the following parameter number is valid:

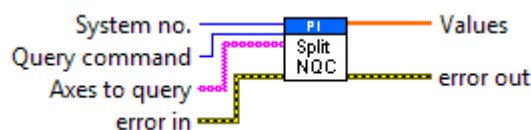
60: stage name (maximum 14 characters)



#### 4.12.20. Split num query command.vi

Splits numerical query command with more than one axis specification into the corresponding one-axis commands and returns answers for all given axes. Supported commands are POS?, MOV?, VOL?, SVA? (more commands can be added).

Old command. Use Split multiple axes command.vi instead.





#### 4.12.21. STOP.vi

Stops single motion (to stop fast scan routines use #24).

F-206, M-8X0: Only for GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For GCS 2.0 or higher, use STP.vi instead.



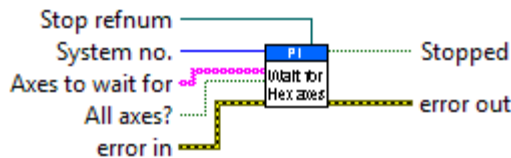
#### 4.12.22. Wait for hexapod system axes to stop.vi

This VI waits for the specified axes of a PI hexapod system (hexapod axes X, Y, Z, U, V, W and separate axes A, B) to stop using #5 polling. If a NanoCube axis (K, L or M) is commanded, the VI will return immediately. If one of the hexapod axes (X, Y, Z, U, V or W) is commanded, it will wait for all six hexapod axes to stop. It returns immediately if a communications error occurred, or if Stop refnum or Local stop is TRUE. When using as a sub-VI, use Stop refnum to stop VI from caller. Required by General wait for movement to stop.vi.

To wait for the hexapod to stop, only one hexapod axis (X, Y, Z, U, V or W) needs to be commanded, because the VI cannot distinguish between the different hexapod axes.

C-887, F-206: Axes to wait for can be any of X, Y, Z, U, V, W, A, B, K, L, M

M-8X0: Axes to wait for can be any of X, Y, Z, U, V, W, A, B



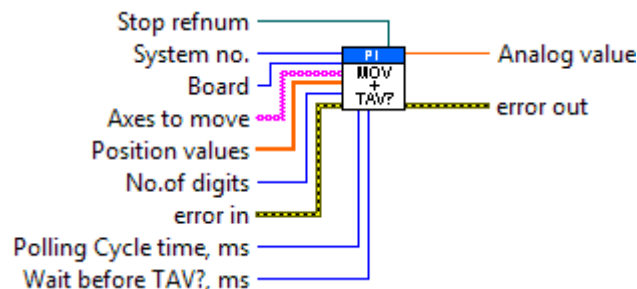
### 4.13. Optical or Analog Input.Ilb

#### 4.13.1. MOV and TAV?.vi

Moves stage to absolute position (MOV.vi), waits for the specified axes to stop (General wait for movement to stop.vi) and queries TAV? (TAV?.vi). A wait time before the TAV? query can be defined. Define connected systems.vi must be run before running this VI. Requires Wait for axes to stop.vi, #5.vi, STA?.vi, #5\_old.vi, ONT?.vi and Wait for hexapod system axes to stop.vi to be present.

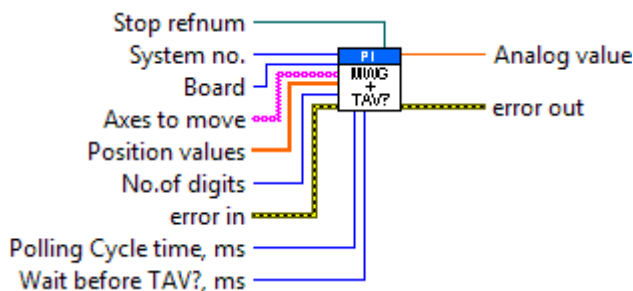
E-712:

E-761: Board = 4. The output is the current voltage at the analog input line, with gain and offset.



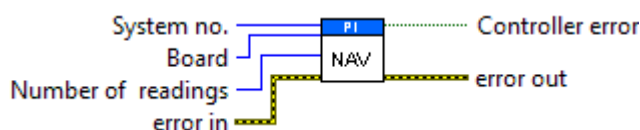
#### 4.13.2. MWG and TAV?.vi

Moves stage to absolute position (MWG.vi), waits for the specified axes to stop using (General wait for movement to stop.vi) and queries TAV? (TAV?.vi). A wait time before the TAV? query can be defined. Define connected systems.vi must be run before running this VI. Requires Wait for axes to stop.vi, #5.vi, STA?.vi, #5\_old.vi and Wait for hexapod system axes to stop.vi to be present. Required by 1D Scan.vi and 2D Scan.vi.



#### 4.13.3. NAV.vi

Sets averaging count for analog readings. Controller error is TRUE if selected system reports an error code which is not 0. For large count values, the communication timeout value may need to be increased to prevent a timeout error during TAV? queries.



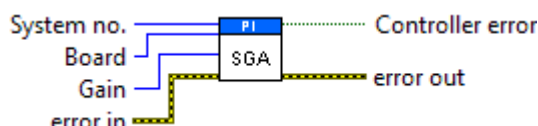
#### 4.13.4. NAV?.vi

Returns averaging-count setting (used for analog readings).



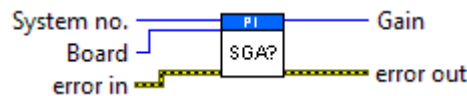
#### 4.13.5. SGA.vi

Sets the gain value for the optical input. See system manual for allowable gain values. Controller error is TRUE if selected system reports an error code which is not 0.



#### 4.13.6. SGA?.vi

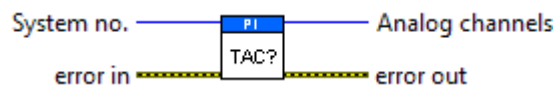
Returns gain-value setting for specified optical input.



#### 4.13.7. TAC?.vi

Returns the number of installed analog channels.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.13.8. TAD?.vi

Returns AD value for the specified sensor number.

C-413 : If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be a "genuine" sensor (sensor integrated in the mechanics) or a "general purpose" analog input.

E-517, E-518: If All sensors? = TRUE, then Sensor identifier? must be FALSE. The input signal channels to be queried with TAD? are the sensor channels of the piezo control electronics, IDs = 1 to 3 (actually available IDs depend on the response to TSC?).

E-709: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be the "genuine" sensor (sensor integrated in the mechanics, identifier = 1) and the "general purpose" analog input (identifier = 2).

E-710: If All sensors? = TRUE, then Sensor identifier? must be TRUE. Command is available for 4-channel controller only and in command level 1 only (see CCL.vi and CCL?.vi)

E-712: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be a "genuine" sensor (sensor integrated in the mechanics) or a "general purpose" analog input.

E-725, E-727: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be a "genuine" sensor (sensor integrated in the mechanics) or a "general purpose" analog input.

E-753: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be the "genuine" sensor (capacitive sensor integrated in the mechanics, identifier = 1) and the "general purpose" analog input (identifier = 2).

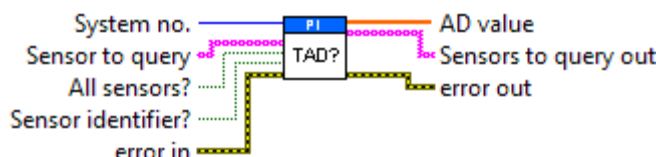
E-754: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be the "genuine" sensor (capacitive sensor integrated in the mechanics, identifier = 1, or incremental sensor, identifier = 3) and the "general purpose" analog input (identifier = 2).

E-755: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Command not available for E-755.101.

E-761: Sensors to query can be 1 to 3. If All sensors? = TRUE, then Sensor identifier? can be FALSE.

E-870: Sensors to query comprises all ADC channels of the device: can be the analog command input (identifier = 1) or other internal on board sources for diagnosis information (identifier = 2...5).

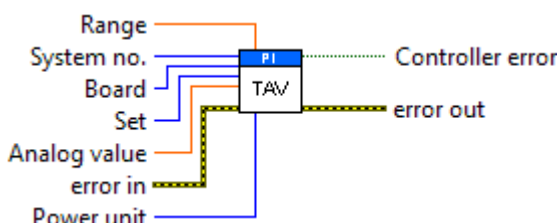
C-887, F-206, M-8X0: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.13.9. TAV.vi

Sets range and power unit for optical head.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.13.10. TAV?.vi

Returns the current analog value in volts, the range of the optical head or the power unit of the Analog value, depending on "Query". Query time for Analog value will depend on "NAV" settings.

C-865, C-866: Query = Value, Board = 1. Range and Power unit are not valid.

C-867, C-884, E-861, E-871, E-873, E-874, Mercury\_GCS: Query = Value, Board = 1 to 4. Range and Power unit are not valid.

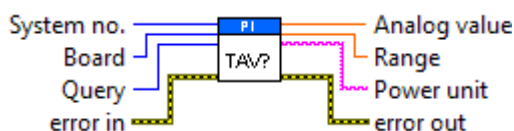
C-880: Query = Value. Range and Power unit are not valid

E-712, E-727:

E-761: Query = Value, Board = 4. Range and Power unit are not valid. The output is the current voltage at the analog input line, with gain and offset.

C-887, F-206, M-8X0: For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), Query can be Value, Range or Power unit. For GCS syntax version = GCS 2.0 or higher, Query = Value

Mercury: Query = Value. Board = analog input channel ID, can be 1-4, 5-7, 8-11 etc., see GCS DLL Manual for details. Range and Power unit are not valid



#### 4.13.11. TNS?.vi

Returns normalized sensor value for the specified sensor number.

C-413 : If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be a "genuine" sensor (sensor integrated in the mechanics) or a "general purpose" analog input.

E-709: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be the "genuine" sensor (sensor integrated in the mechanics, identifier = 1) and the "general purpose" analog input (identifier = 2).

E-710: If All sensors? = TRUE, then Sensor identifier? must be TRUE. Command is available in command level 1 only (see CCL.vi and CCL?.vi)

E-712: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be a "genuine" sensor (sensor integrated in the mechanics) or a "general purpose" analog input.

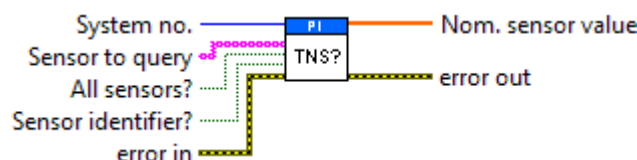
E-725, E-727: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be a "genuine" sensor (sensor integrated in the mechanics) or a "general purpose" analog input.

E-753: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be the "genuine" sensor (capacitive sensor integrated in the mechanics, identifier = 1) and the "general purpose" analog input (identifier = 2).

E-754: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be the "genuine" sensor (capacitive sensor integrated in the mechanics, identifier = 1, or incremental sensor, identifier = 3) and the "general purpose" analog input (identifier = 2).

E-755: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Command not available for E-755.101.

E-761: If All sensors? = TRUE, then Sensor identifier? can be FALSE.



#### 4.13.12. TSC?.vi

Returns the number of available sensor channels.

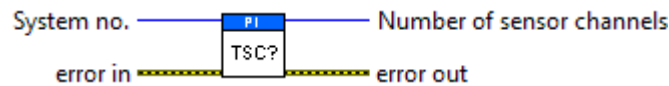
E-517: Using the Sensor Enable parameter, ID 0x02000000, you can change the E-517 configuration in case of hardware changes, e.g. if you install additional sensor and/or amplifier channels in the system. If this parameter is changed, the Number Of Sensor Channels parameter is adapted automatically. E.g. if parameter 0x02000000 is set to "disabled" for a sensor channel, this sensor channel is no longer included in the TSC? response. See "Configure Axes and Channels" in the E-517 User manual for details.

E-709: The response comprises all ADC channels of the device: the "genuine" sensor (sensor integrated in the mechanics) and the "general purpose" analog input.

C-413, E-712, E-725, E-727: The response comprises all ADC channels of the device: "genuine" sensors (sensors integrated in the mechanics) and "general purpose" analog input channels.

E-753: The response comprises all ADC channels of the device: the "genuine" sensor (capacitive sensor integrated in the mechanics) and the "general purpose" analog input.

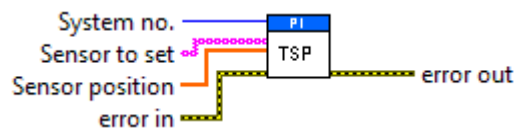
E-754: The response comprises all ADC channels of the device: the "genuine" sensor (capacitive or incremental sensors integrated in the mechanics) and the "general purpose" analog input.



#### 4.13.13. TSP.vi

Sets sensor position for the specified sensor number.

Mercury\_GCS:



#### 4.13.14. TSP?.vi

Returns sensor position for the specified sensor number.

E-517, E-518: If All sensors? = TRUE, then Sensor identifier? can be FALSE. The input signal channels to be queried with TSP? are the sensor channels of the piezo control electronics, IDs = 1 to 3 (actually available IDs depend on the response to TSC?).

E-761: If All sensors? = TRUE, then Sensor identifier? can be FALSE.

E-709: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be the "genuine" sensor (sensor integrated in the mechanics, identifier = 1) and the "general purpose" analog input (identifier = 2).

E-710: If All sensors? = TRUE, then Sensor identifier? must be TRUE. Command is available in command level 1 only (see CCL.vi, CCL?.vi)

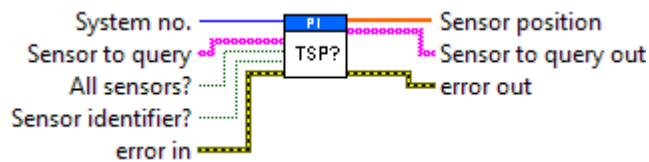
C-413, E-712, E-725, E-727: If All sensors? = TRUE, then Sensor identifier? can be FALSE. Sensors to query comprises all ADC channels of the device: can be a "genuine" sensor (sensor integrated in the mechanics) or a "general purpose" analog input.

E-753: If All sensors? = TRUE, then Sensor identifier? can be FALSE. Sensors to query comprises all ADC channels of the device: can be the "genuine" sensor (capacitive sensor integrated in the mechanics, identifier = 1) and the "general purpose" analog input (identifier = 2).

E-754: If All sensors? = TRUE, then Sensor identifier? must be FALSE. Sensors to query comprises all ADC channels of the device: can be the "genuine" sensor (capacitive sensor integrated in the mechanics, identifier = 1, or incremental sensor, identifier = 3) and the "general purpose" analog input (identifier = 2).

E-755: If All sensors? = TRUE, then Sensor identifier? can be FALSE. Command not available for E-755.101.

Mercury\_GCS: If All sensors? = TRUE, then Sensor identifier? can be FALSE.

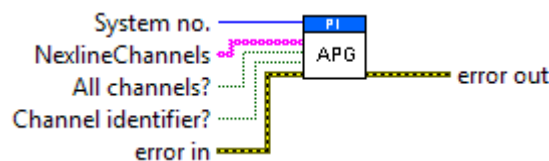


## 4.14. PZT voltage.IIb

### 4.14.1. APG.vi

Auto Piezo Gain calibration of specified Nexline channels.

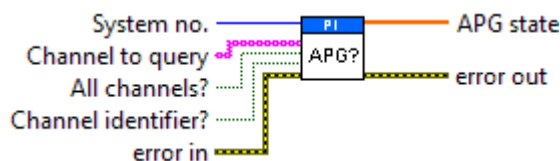
E-755: If All channels = TRUE, then Channel identifier must be FALSE. Command not available for E-755.101. Use #7.vi to check if calibration has finished. To save the calibration parameters to default use PGS.vi. See E-755 user manual for details.



### 4.14.2. APG?.vi

Returns Auto Piezo Gain calibration state for specified Nexline Channels.

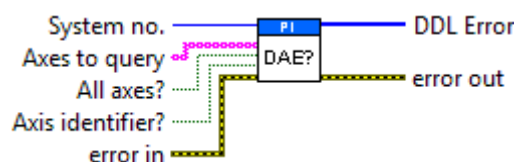
E-755: If All channels? = TRUE, then Channel identifier? must be FALSE. Command not available for E-755.101.



### 4.14.3. DAE?.vi

Returns DDL autoexamined error. See GCS DLL manual for details.

E-710: Only supported by 6-channel version. Only one axis per command allowed. "All axes" must be FALSE. Only valid for axes for which the autoexamine function has been enabled using DAS.vi.



### 4.14.4. DCO.vi

Sets drift compensation mode for given axes.

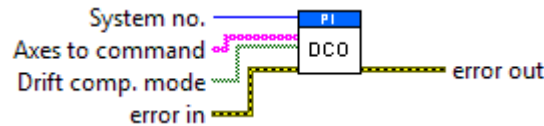
E-516: When activating/deactivating drift compensation with DCO, this setting is automatically saved to flash ROM together with the current settings for the parameters listed below, and they become the new power-on defaults: communication interface, enabled channels and display format, averaging (AVG), velocity control mode (VCO) and velocity (VEL), offset and gain for position and



output voltage display, mode and tolerance for on-target reading (SPA), position limits (PLM, NLM), voltage limits (VMA, VMI), macros and default macro setting.

If current parameter values are incorrect, the system may malfunction. Be sure that you have the correct parameter settings before using DCO.

E-816: Only one axis per command allowed. It is necessary to wait a certain time before sending the next command to prevent it from being lost. See system manual for valid drift compensation modes.



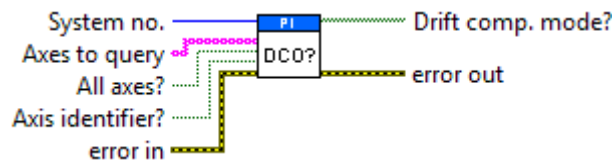
#### 4.14.5. DCO?.vi

Returns drift compensation mode status of queried axes.

E-516: If All axes = TRUE, then Axis identifier must be TRUE

E-517, E-518: If All axes? = TRUE, then Axis identifier? can be FALSE

E-816: All axes? = FALSE, only one axis per command allowed.



#### 4.14.6. DPO.vi

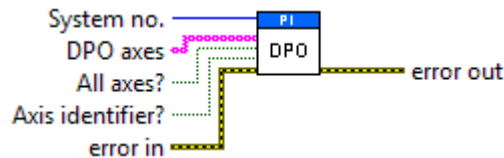
DDL processing parameter correction for specified axis. Calculates internal DDL processing parameters. DPO is required when servo parameters have changed for an axis.

E-710: If All axes = TRUE, then Axis identifier must be TRUE. Command is available in command level 1 only (see CCL.vi and CCL?.vi)

E-712: If All axes? = TRUE, then Axis identifier? can be FALSE

E-725, E-727: If All axes? = TRUE, then Axis identifier? can be FALSE

E-753, E-754: If All axes = TRUE, then Axis identifier? can be FALSE.



#### 4.14.7. DTC.vi

Clears DDL table.

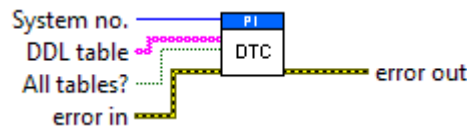
E-710: All tables? must be FALSE.

E-712: All tables? must be FALSE.

E-725, E-727: All tables? must be FALSE.

E-753, E-754: All tables? must be FALSE.

E-761: All tables? must be FALSE.



#### 4.14.8. OAD.vi

Sets driving voltage amplitude of open-loop analog driving for given PiezoWalk channels. No. of digits is the number of digits after the decimal point in the voltage value(s) that will be sent. Hidden error is TRUE if selected system reports error code which is not 0.

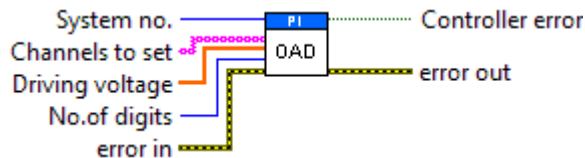
E-861: Servo must be disabled for the commanded axis prior to using this command (open-loop operation). In open-loop operation, an RNP command must be sent each time the motion mode is to be changed from stepping motion (OSM command) to analog motion (OAD command) and vice versa. RNP brings the drive to a full-holding-force, zero-drive-voltage Relaxed state.

The first OAD sent for a NEXACT® linear drive which is in the Relaxed state prepares the drive for analog motion (brings it to the Analog state) before the actual motion is done. Once the drive is in the Analog state, each subsequent OAD motion will be executed immediately.

The first OAD after a change of motion mode and the RNP procedure can take up to four times the slewrate value (parameter ID 0x7000002; can be changed with SPA and SEP).

After open-loop analog motion was done with OAD, an RNP command must be sent before the servo can be switched on with SVO for closed-loop operation.

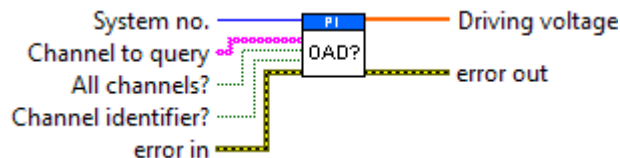
Motion commands like OAD are not allowed when a joystick is active on the axis.



#### 4.14.9. OAD?.vi

Returns last commanded open loop Analog Driving voltage of given PiezoWalk channels.

E-755, E-861, E-872: If All channels? = TRUE, then Channel identifier? must be FALSE



#### 4.14.10. OSM.vi

Open loop step moving of given channels or axes. Use SSA.vi (if available) to set step size.

C-891: Only valid for devices with stepper motors.

E-135: The SSA command is not supported by the E-135. To change the step size, please use the SPA command to change the corresponding parameter (parameter ID 0x23000500).

E-712: To change between nanostepping mode and full-step mode use parameter "PiezoWalk Driving Mode" (ID 0x07001a00, can be changed with SPA and SEP).

The No. of steps to perform can always be given as floating-point number, but depending on the driving mode selection made with the PiezoWalk Driving Mode parameter, processing of the value differs:

Full-step mode: decimal places are discarded. This means that only complete step cycles are performed. Set No. of digits to 0 for speed of transmission.

Nanostepping mode: floating-point numbers are processed. Make sure to increase No. of digits. This means that also parts of a complete step cycle are performed.

Prior to using this command, servo must be disabled for the axis to which the PiezoWalk channel is assigned.

E-755: No. of digits = 0

E-861: The velocity for open-loop step motion depends on the step size and on the step frequency. The step size is given by the amplitude of the transport voltage which can be set with SSA and queried with SSA?. The step frequency can be set with OVL and queried with OVL?.

In open-loop operation, a RNP command must be sent each time the motion mode is to be changed from stepping motion (OSM command) to analog motion (OAD command) and vice versa. RNP brings the drive to a full-holding-force, zero-drive-voltage Relaxed state.

The first OSM sent for a NEXACT® linear drive which is in the Relaxed state prepares the drive for stepping motion (brings it to the In Motion state) before the actual motion is done. Once the drive is in the In Motion state, each subsequent OSM motion will be executed immediately.

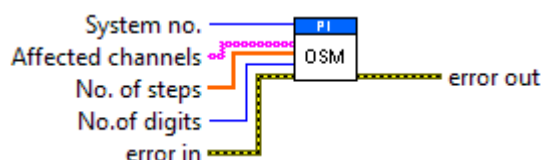
The first OSM after a change of motion mode and the RNP procedure can take up to four times the slewrate value (parameter ID 0x7000002; can be changed with SPA and SEP).

To command parts of a step cycle, make sure to increase "No. of digits".

Motion commands like OSM are not allowed when a joystick is active on the axis.

E-870, E-872: The SSA command is not supported by the E-870 driver electronics. To change the step size, please use the SPA command to change the corresponding parameters.

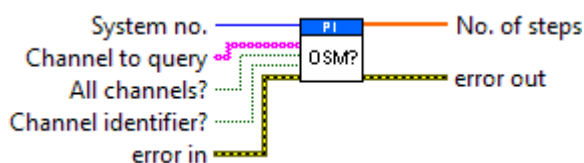
E-871, E-873:



#### 4.14.11. OSM?.vi

Returns the number of steps set by last OSM command for the given channel or axis.

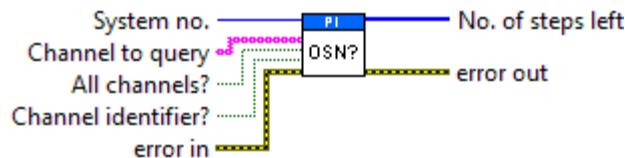
C-891, E-755: If All channels = TRUE, then Axis identifier must be FALSE



#### 4.14.12. OSN?.vi

Returns the number of steps still to be performed for the given channel or axis after the last OSM command.

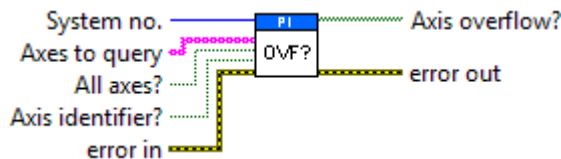
C-891: If All channels = TRUE, then Axis identifier can be FALSE.  
 E-712: If All channels = TRUE, then Axis identifier must be FALSE.  
 E-755: If All channels = TRUE, then Axis identifier must be FALSE  
 E-872: If All channels = TRUE, then Axis identifier must be FALSE.  
 E-871, E-873: If All channels = TRUE, then Axis identifier must be FALSE.  
 E-135, E-874: If All channels = TRUE, then Axis identifier must be FALSE.



#### 4.14.13. OVN?.vi

Returns overflow information for queried axes.

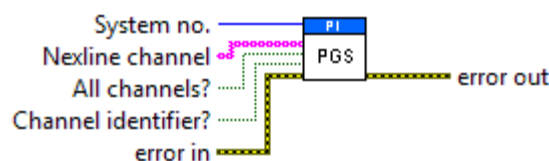
E-516: If All axes? = TRUE, then Axis identifier? must be TRUE  
 C-413, E-517, E-518, E-709, E-712, E-725, E-727, E-753, E-754, E-761: If All axes? = TRUE, then Axis identifier? can be FALSE  
 E-816: All axes? = FALSE, only one axis per command allowed.



#### 4.14.14. PGS.vi

Makes the current settings of the parameters for the Nexline channel piezo gain correction (APG) default by writing them to the EPROM.

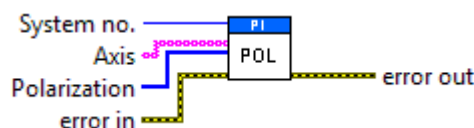
E-755: If All channels? = TRUE, then Channel identifier? must be FALSE.  
 Command is available in command level 1 only (see CCL.vi). Command not available for E-755.101.



#### 4.14.15. POL.vi

Set axis to either 0% or 100% of its travel range. It does not use the previous history of the actuator but always uses the maximum number of pulses / maximum duration of pulses.

E-135:



#### 4.14.16. RNP.vi

Relax piezos of given PiezoWalk® channels without motion and wait until procedure has stopped. The aim of this procedure is to reduce all applied voltages when the target is reached and thus to increase the lifetime of the piezos. To compensate a preload which would lead to a small motion of the slider when the piezo voltage is zero an adjustment voltage can be applied. VI will also stop if Stop refnum or Local stop is TRUE.

E-712: Adj. voltage must be zero to set the voltages to 0. Servo must be disabled for the axis to which the PiezoWalk channel is assigned (open-loop operation). RNP starts the simple relax procedure. To start the iterative relax procedure which is available for axes driven by one PiezoWalk® drive and measured by one sensor only, use SVO.vi to switch servo off for these axes. In nanostepping mode, RNP must be used in the following cases due to the possibility to perform parts of complete steps:

After stepping motion in open-loop operation a relax procedure is required before analog mode motion is possible

Switching to closed-loop operation (set servo on with SVO) is only possible after a relax procedure.

E-755: The RNP procedure is automatically performed when the servo is switched off and when RTO is used.

E-861: Adj. voltage must be zero to set the voltages to 0.

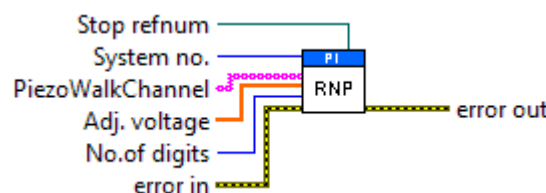
Servo must be disabled for the commanded axis prior to using this command (open-loop operation).

In open-loop operation, an RNP command must be sent each time the motion mode is to be changed from stepping motion (OSM, OMA, OMR) to analog motion (OAD) or vice versa.

After open-loop analog motion was done with OAD, an RNP command must be sent before the servo can be switched on with SVO for closed-loop operation.

The RNP procedure can take up to four times the slewrate value (parameter ID 0x7000002; can be changed with SPA and SEP).

You can query the current state of the system (E-861 and NEXACT® linear drive) using #4 and SRG?.



#### 4.14.17. SVA.vi

Sets absolute PZT voltage for specified axes. Servo must be switched off (using SVO.vi) before using this command. No. of digits is the number of digits after the decimal point in the voltage value(s) that will be sent.

E-517, E-518: Motion commands like SVA are not allowed when the controller is in OFFLINE mode or when the wave generator output is active. When a macro is running on the controller, SVA will be executed not until the macro is finished or stopped. See "Control Value Generation" and "Control Modes" in the controller user manual for details

C-413, E-709: Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-712: For axes driven by conventional piezo actuators, PZT voltage is a dimensionsless value whose range corresponds approximately to the mechanics travel range in  $\mu\text{m}$ .

For axes driven by NEXLINE® drives, PZT voltage is given in volts. SVA affects only the shearing segments of the NEXLINE® stack actuators. Hence SVA will cause motion in analog mode only.

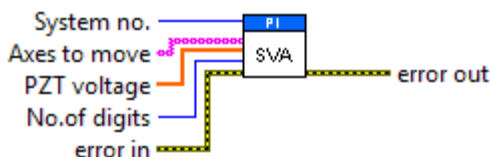
Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-725, E-727, E-753, E-754: PZT voltage is a dimensionsless value whose range corresponds approximately to the mechanics travel range in  $\mu\text{m}$ .

Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-761: PZT voltage is a dimensionsless value whose range corresponds approximately to the mechanics travel range in  $\mu\text{m}$ . Motion commands are not allowed when a wave generator is active.

E-816: Only one axis per command allowed. It is necessary to wait a certain time before sending the next command to prevent it from being lost.



#### 4.14.18. SVA?.vi

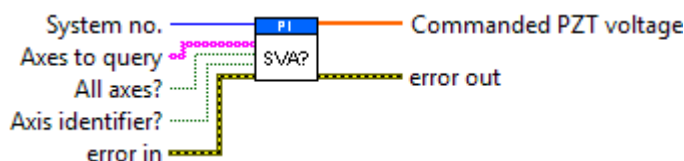
Returns commanded PZT voltage for queried axes.

E-516, E-710: If All axes= TRUE, then Axis identifier? must be TRUE

C-413, E-517, E-518, E-709, E-712, E-725, E-727, E-753, E-754, E-755, E-761, E-874: If All axes= TRUE, then Axis identifier? can be FALSE.

E-712, E-725, E-727, E-753, E-754, E-761: The response is a dimensionsless value whose range corresponds approximately to the mechanics travel range in  $\mu\text{m}$ .

E-816: All axes? = FALSE, only one axis per command allowed.



#### 4.14.19. SVR.vi

Sets relative PZT voltage for specified axes. Servo must be switched off (using SVO.vi) before using this command. No. of digits is the number of digits after the decimal point in the voltage value(s) that will be sent.

E-517, E-518: Motion commands like SVR are not allowed when the controller is in OFFLINE mode or when the wave generator output is active. When a macro is running on the controller, SVR will be executed not until the macro is finished or stopped. See "Control Value Generation" and "Control Modes" in the controller user manual for details.

C-413, E-709: Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

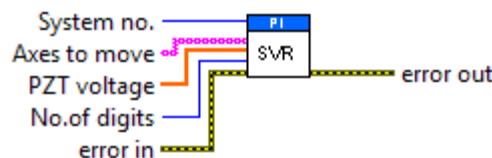
E-712: SVR is valid for axes driven by conventional piezo actuators only. PZT voltage is a dimensionless value whose range corresponds approximately to the mechanics travel range in  $\mu\text{m}$ . Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-725, E-727: PZT voltage is a dimensionless value whose range corresponds approximately to the mechanics travel range in  $\mu\text{m}$ . Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-753, E-754: PZT voltage is a dimensionless value whose range corresponds approximately to the mechanics travel range in  $\mu\text{m}$ . Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

E-761: PZT voltage is a dimensionless value whose range corresponds approximately to the mechanics travel range in  $\mu\text{m}$ . Motion commands are not allowed when a wave generator is active.

E-816: Only one axis per command allowed. It is necessary to wait a certain time before sending the next command to prevent it from being lost.



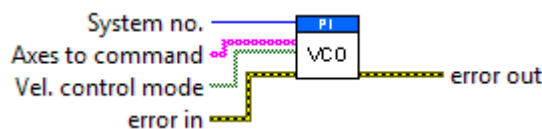
#### 4.14.20. VCO.vi

Sets velocity-control mode for specified axes.

Analogue: Velocity control mode can only be set for all axes equally. Therefore only first field of Vel.control mode is valid.

E-516, E-517, E-518, E-727, E-754: The VCO command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the controller is powered off.

E-761: The VCO command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi with "Affected axes" as an empty array. Parameter changes not saved with WPA will be lost when the PC is powered off or the E-761 is rebooted.



#### 4.14.21. VCO?.vi

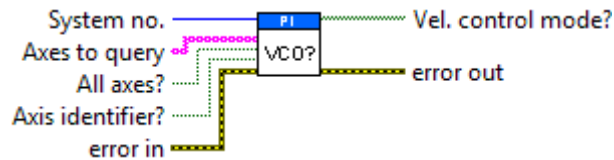
Returns velocity-control mode status for queried axes.

Analogue: If All axes = TRUE, then Axis identifier can be FALSE. Only first field of Vel.control mode is valid.

E-516: If All axes = TRUE, then Axis identifier must be TRUE

E-517, E-518, E-727, E-754, E-761: If All axes = TRUE, then Axis identifier can be FALSE





#### 4.14.22. VMA.vi

Sets upper voltage limit for all PZTs affected by specified axes. No. of digits is the number of digits after the decimal point in the limit value(s) that will be sent.

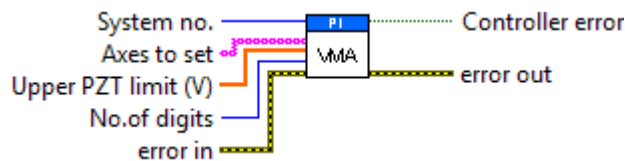
Controller error is TRUE if selected system reports an error code which is not 0.

E-516: The VMA command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the controller is powered off.

E-517, E-518: Axes to set are piezo channel numbers. The VMA command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the controller is powered off.

E-710: Axes to set are piezo numbers. Command is available in command level 1 only (see CCL.vi and CCL?.vi)

E-761: Axes to set are piezo channel numbers, which can be 1 to 4. Use WPA.vi to save currently valid parameters. When booting the E-761, the value is replaced by the "Max. Voltage of Amplifier" parameter, but can be restored with RPA.



#### 4.14.23. VMA?.vi

Returns upper PZT voltage limit for queried axes / piezo channels.

E-516: If All axes = TRUE, then Axis identifier must be TRUE

E-517, E-755: Axes to query are piezo channel numbers. If All axes? = TRUE, then Axis identifier? must be FALSE.

E-761: If All axes = TRUE, then Axis identifier must be FALSE. Axes to query are piezo numbers to query, which can be 1 to 4.



#### 4.14.24. VMI.vi

Sets lower voltage limit for all PZTs affected by specified axes. No. of digits is the number of digits after the decimal point in the limit value(s) that will be sent.

Controller error is TRUE if selected system reports an error code which is not 0.

E-516: The VMI command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on

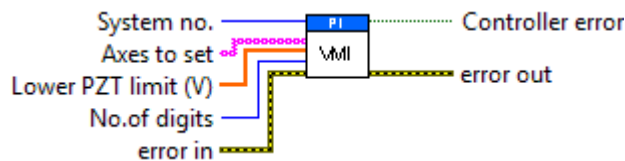
defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the controller is powered off.

E-517, E-518: The VMI command saves the parameters in RAM only. To save the currently valid parameters to flash ROM, where they become the power-on defaults, you must run WPA.vi. Parameter changes not saved with WPA will be lost when the controller is powered off.

Axes to set are piezo channel numbers.

E-710: Axes to set are piezo numbers. Command is available in command level 1 only (see CCL.vi and CCL?.vi)

E-761: Axes to set are piezo channel numbers, which can be 1 to 4. Use WPA.vi to save currently valid parameters. When booting the E-761, the value is replaced by the "Min. Voltage of Amplifier" parameter, but can be restored with RPA.



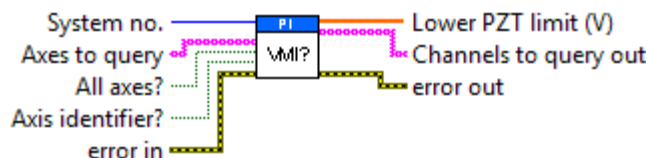
#### 4.14.25. VMI?.vi

Returns lower PZT voltage limit for queried axes / piezo channels.

E-516: If All axes = TRUE, then Axis identifier must be TRUE

E-517, E-518, E-755: Axes to query are piezo channel numbers. If All axes? = TRUE, then Axis identifier? must be FALSE.

E-761: If All axes = TRUE, then Axis identifier must be FALSE. Axes to query are piezo numbers to query, which can be 1 to 4.

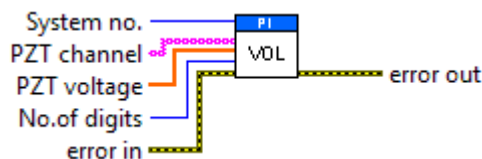


#### 4.14.26. VOL.vi

Sets absolute PZT voltage for specified piezo channel. No. of digits is the number of digits after the decimal point in the voltage value(s) that will be sent. If the commanded voltage exceeds the voltage limits of the piezo channel, then the command is not executed.

Analog system: PZT channel is identical with axis ID.

E-761: PZT channel can be 1 to 4.



#### 4.14.27. VOL?.vi

Returns current PZT voltage for queried axes / piezo channels.

Analog: If All axes? = TRUE, then Axis identifier? can be FALSE. VI reads control voltage.

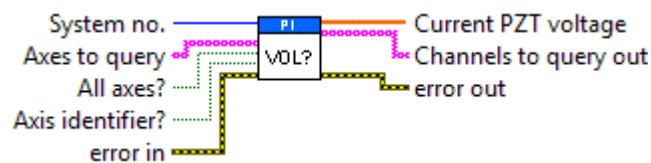
E-516: If All axes? = TRUE, then Axis identifier? must be TRUE

E-710: If All axes? = TRUE, then Axis identifier? must be TRUE. Axes to query are piezo channel numbers.

C-413, E-517, E-518, E-709, E-712, E-725, E-727, E-753, E-754, E-755: If All axes? = TRUE, then Axis identifier? can be FALSE. Axes to query are piezo channel numbers.

E-761: If All axes? = TRUE, then Axis identifier? can be FALSE. Axes to query are piezo channel numbers, which can be 1 to 4.

E-816: All axes? = FALSE, only one axis per command allowed.



### 4.15. Scan support.Ilb

#### 4.15.1. Axis names.vi

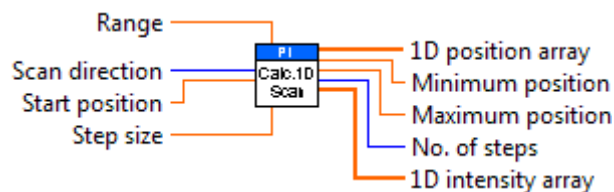
Checks if Names contains three strings for axis names. If this is not the case, it assigns X Values, Y Values and/or Z Values as the missing axis name. Sub-VI for Show\_Save\_Load\_XY\_Data.vi



#### 4.15.2. Calculate 1D scan positions.vi

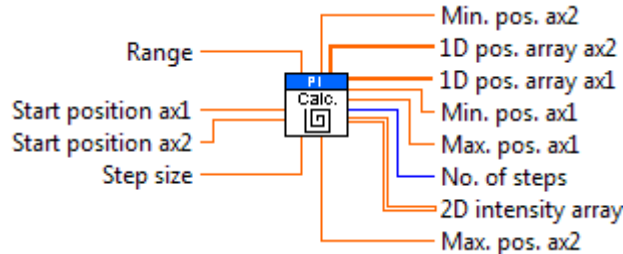
Calculates 1D scan positions according to Scan direction, Start position, Range and Step size and returns Minimum position, Maximum position, No. of steps and initialized 1D position array and 1D intensity array. Scan direction can be

- Left & right (- & +) meaning that the scan starts at (Start position - ½ Range) and stops at (Start position + ½ Range),
- To the left (-) meaning that scan starts (Start position - Range) and stops at Start position, or
- To the right (+) meaning that scan starts at Start position and stops at (Start position + Range).



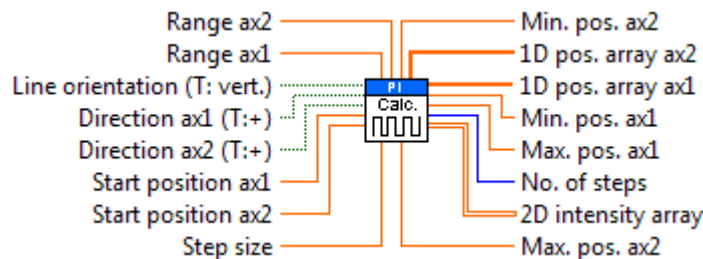
#### 4.15.3. Calculate 2D linear spiral.vi

Calculates position values for a 2D linear spiral scan and initializes intensity array according to Start position of axes 1 and 2, Step size and Range. The VI returns minimum position, maximum position, No. of steps and initialized 1D position array for axes 1 and 2 and a 2D intensity array.



#### 4.15.4. Calculate 2D meander.vi

Calculates position values for a 2D linear meander scan and initializes intensity array according to Start position of axes 1 and 2, Step size, range of axes 1 and 2 and scan directions of axes 1 and 2. The VI returns Minimum position, Maximum position, No. of steps and initialized 1D position array for axes 1 and 2 and a 2D intensity array.

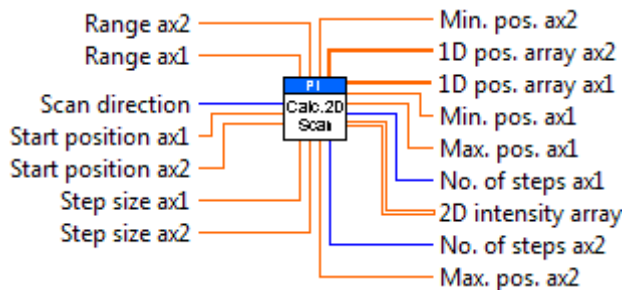


#### 4.15.5. Calculate 2D scan positions.vi

Calculates 2D scan positions according to Scan direction, Start position, Range and Step size of axes 1 and 2 and returns Minimum position, Maximum position, No. of steps and initialized 1D position array for axes 1 and 2 and a 2D intensity array. Scan direction can be

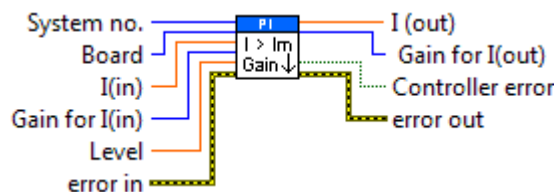
- Scan from middle meaning that the scan starts at (Start position ax1 -  $\frac{1}{2}$  Range ax1, Start position ax2 -  $\frac{1}{2}$  Range ax2) and stops at (Start position ax1 +  $\frac{1}{2}$  Range ax1, Start position ax2 +  $\frac{1}{2}$  Range ax2),
- Scan upper left meaning that scan starts at (Start position ax1 - Range ax1, Start position ax2) and stops at (Start position ax1, Start position ax2 + Range ax2),
- Scan upper right meaning that scan starts at (Start position ax1, Start position ax2) and stops at (Start position ax1 + Range ax1, Start position ax2 + Range ax2),
- Scan lower left meaning that the scan starts at (Start position ax1 - Range ax1, Start position ax2 - Range ax2) and stops at (Start position ax1, Start position ax2),
- Scan lower right meaning that the scan starts at (Start position ax1, Start position ax2 - Range ax2) and stops at (Start position ax1 + Range ax1, Start position ax2),
- Scan left meaning that the scan starts at (Start position ax1 - Range ax1, Start position ax2 -  $\frac{1}{2}$  Range ax2) and stops at (Start position ax1, Start position ax2 +  $\frac{1}{2}$  Range ax2),

- Scan right meaning that the scan starts at (Start position ax1, Start position ax2 -  $\frac{1}{2}$  Range ax2) and stops at (Start position ax1 + Range ax1, Start position ax2 +  $\frac{1}{2}$  Range ax2),
- Scan above meaning that the scan starts at (Start position ax1 -  $\frac{1}{2}$  Range ax1, Start position ax2) and stops at (Start position ax1 +  $\frac{1}{2}$  Range ax1, Start position ax2 + Range ax2),
- Scan below meaning that the scan starts at (Start position ax1 -  $\frac{1}{2}$  Range ax1, Start position ax2 - Range ax2) and stops at (Start position ax1 +  $\frac{1}{2}$  Range ax1, Start position ax2),



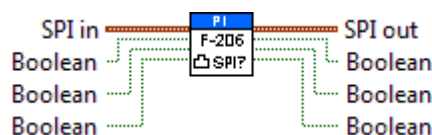
#### 4.15.6. Decrease Gain?.vi

Decreases gain if intensity is higher than Level.



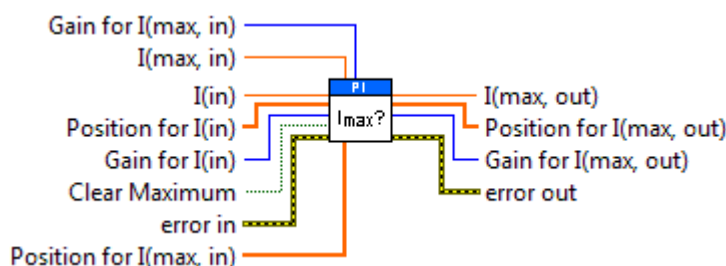
#### 4.15.7. F206\_Choose\_SPI.vi

F-206-specific. Returns coordinates for Pivot Point depending on states of buttons on schematic platform.



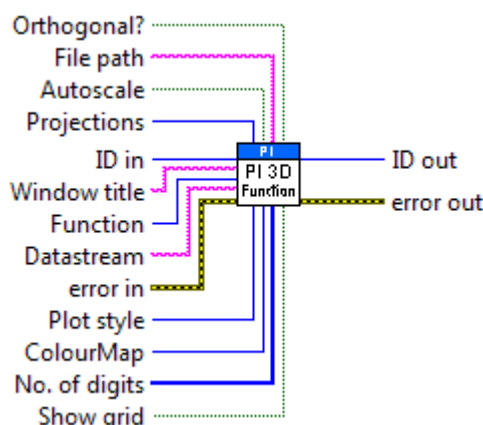
#### 4.15.8. Maximum Intensity?.vi

Checks if the current intensity value (I(in)), multiplied by the current gain value (Gain for I(in)), is larger than the last maximum intensity value (I(max, in)), multiplied by the last gain value (Gain for I(max, in)), and returns the new maximum intensity values (intensity, gain and position). If Clear Maximum is TRUE, I(max, out) and Position for I(max, out) is set to zero.



#### 4.15.9. PIGraph3D\_DLL\_Functions.vi

This VI calls a given function from GCSTranslator.dll. GCSTranslator.dll must be installed, and wxPIGraph3D.dll must be installed in PI\GCSTranslator

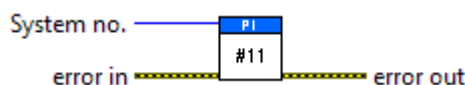


### 4.16. Special command.llb

#### 4.16.1. #11.vi

Returns the number of free memory points for profile definition (e.g. for a dynamic move) by sending the single ASCII character 11.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.16.2. #24.vi

Stops all motion (by sending the single ASCII character 24). #24 sets error code 10, call ERR?.vi to reset error after #24 has been called.

Valid for Analog systems, C-413, C-702, C-843, C-843.PM, C-844, C-848, C-865, C-866, C-867, C-877, C-880, C-880K005, C-884, C-885, C-886, C-887, C-891, E-135, E-516, E-517, E-518, E-709, E-712, E-725, E-727, E-753, E-754, E-755, E-761, E-816, E-861, E-870, E-871, E-872, E-873, E-874, F-206, Hydra, Pollux, M-8X0, Mercury, Mercury\_GCS (but must be present for E-710 also). To support analog interfacing, VI must be present for E-816 also.

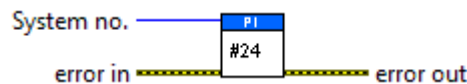
Analog systems: #24 does not set any error code. When used with any digital controller, does not influence connection between selected analog input channel and axis.

C-880K005: VI only supported when called through PI\_Multix.vi

E-761: #24 does not take effect to analog input which is used for "direct" axis control (see the E-761 User manual). To disable "direct" control for an axis, the value of the corresponding "Aux-Input to target factor" parameter (ID 0x06000902) must be set to 0 with SPA.

E-816: This command cannot be issued to a slave. Check controller manual to find out if #24 is supported.

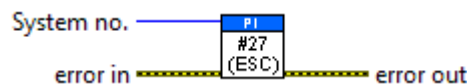
C-887, F-206, M-8X0: Depending on the firmware version on the controller, this command may not take immediate effect for motion initiated by INI or fast scanning commands.



#### 4.16.3. #27.vi

Sends the single ASCII character 27 (ESC, system abort) and sets error flag. Motion stops immediately, all servo registers are reset, the servo-loop is disabled. The controller emits a continuous beep at 300 Hz. For restart, the controller must be manually reset or turned off and on (power switch).

C-887, F-206, M-8X0: Depending on the firmware version on the controller, this command may not take immediate effect for motion initiated by INI or fast scanning commands.



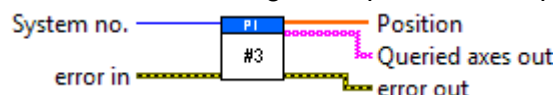
#### 4.16.4. #3.vi

Returns position information. Command is equivalent to POS?.

C-887, F-206: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

M-8X0: Check HLP?/HELP answer to find out if #3 is supported. For GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), reports target position of hexapod axes X, Y, Z, U, V, W. Only for controllers based on C-842.80.

For GCS syntax version = GCS 2.0 or higher, reports current position.

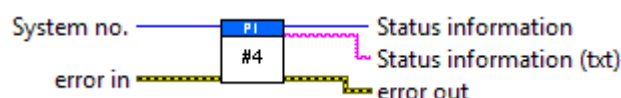


#### 4.16.5. #4.vi

Request status information by sending the single ASCII character 4. Answer is controller specific. See controller user manual for decoding.

C-887, F-206: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

M-8X0: Only for controllers based on C-842.80 or GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).





**4.16.6. #5.vi**

Polls the motion status of the connected axes by sending the single ASCII character 5. Connected axes are read from Global2.vi and displayed on the front panel for assignment. Required by General wait for movement to stop.vi and Wait for axes to stop.vi.

Analog: Motion status can only be determined for all connected axes, not for single axes.

C-887, F-206: For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0) coding in answer is different, please use #5\_old.vi.

C-887, M-8X0: For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0) coding in answer is different, please use #5\_old.vi.

**4.16.7. #6.vi**

Polls for change in position of the connected axes by sending the single ASCII character 6. After a position change #6 answer is reset to FALSE with next POS? query. Connected axes are read from Global2.vi and displayed on the front panel for assignment.

E-517: #6 can be used in open-loop and closed-loop operation.

The query considers only motion caused by control sources (e.g. move commands), but ignores position changes caused by amplifier noise.

C-887, F-206: For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), Only first field of Pos change? array is valid, refers to whole system.

C-887, M-8X0: For GCS syntax version = GCS 1.0 (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), only first field of Pos change? array is valid, refers to whole system.

**4.16.8. #7.vi**

Sends the single ASCII character 7 and returns the ready status of the controller. Sub-VI for Wait for answer of longlasting command.vi.

C-880K005: VI only supported when called through PI\_Multix.vi

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

E-712: This VI is to be used during reference moves only.



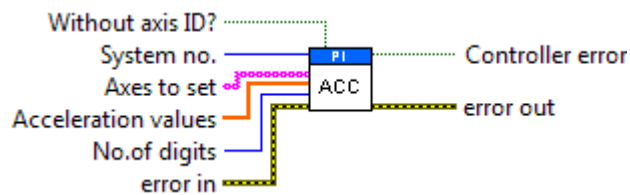
**4.16.9. ACC.vi**

Sets closed-loop acceleration and checks for error. If Without axis ID? is TRUE, then Axes to set is ignored and first field of Acceleration values array is used for acceleration command. Number of digits is the number of digits after the decimal point in the acceleration value(s) that will be sent. Controller error is TRUE if selected system reports error code which is not 0.

C-843, C-867, C-877, C-884, C-891, E-861, E-874, Hydra, Pollux, Mercury\_GCS: Without axis ID? = FALSE. Acceleration unit is mm/s<sup>2</sup>.

C-413: Without axis ID? = FALSE. Acceleration unit is mm/s<sup>2</sup>.

E-873: Without axis ID? = FALSE. Acceleration unit is mm/s<sup>2</sup>.

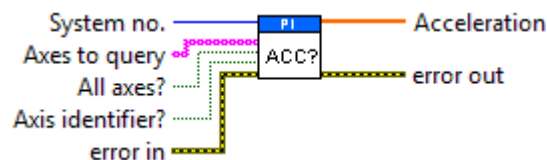
**4.16.10. ACC?.vi**

Returns closed-loop acceleration setting for specified axes.

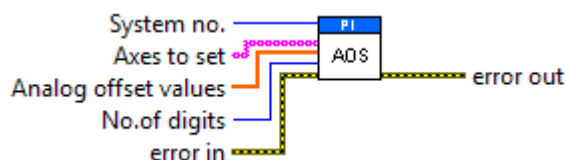
C-843, C-867, C-877, C-884, C-891, E-861, E-874, Hydra, Pollux, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE. Acceleration unit is mm/s<sup>2</sup>.

C-413: If All axes? = TRUE, then Axis identifier? can be FALSE. Acceleration unit is mm/s<sup>2</sup>.

E-873: If All axes? = TRUE, then Axis identifier? can be FALSE. Acceleration unit is mm/s<sup>2</sup>.

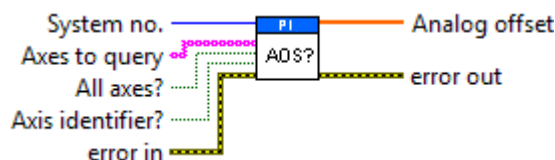
**4.16.11. AOS.vi**

Set Analog Input Offset. This command adds an offset value to an input value of an analog input, which is configured as analog target position for the selected axis. The target position of the selected axis is Target = Analog Input + Offset. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

**4.16.12. AOS?.vi**

Returns analog input offset, which was set by AOS or by a parameter command.

C-413, E-709, E-712, E-725, E-727, E-753, E-754: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.16.13. AVG.vi

Sets averaging time to use for measurements. Controller error is TRUE if selected system reports an error code which is not 0.

E-516: When making settings with AVG, they are automatically saved to flash ROM together with the current settings for the parameters listed below, and they become the new power-on defaults:

communication interface, enabled channels and display format, drift compensation mode (DCO), velocity control mode (VCO) and velocity (VEL), offset and gain for position and output voltage display, mode and tolerance for on-target reading (SPA), position limits (PLM, NLM), voltage limits (VMA, VMI), macros and default macro setting.

If current parameter values are incorrect, the system may malfunction. Be sure that you have the correct parameter settings before using AVG.

E-816: Averaging time (factor) can be 1, 2, 4, 8, 16, 32 or 64. This command cannot be issued to a slave

E-761: Averaging (oversampling) time (factor) can be 4, 8, 16, 32, 64 or 128. The "Sensor sampling time" and "Servo update time" parameters are influenced by AVG. Use WPA.vi to save the values.

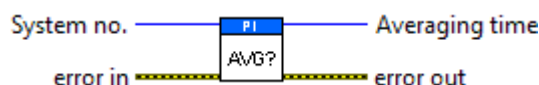


#### 4.16.14. AVG?.vi

Returns current measurement-averaging-time setting.

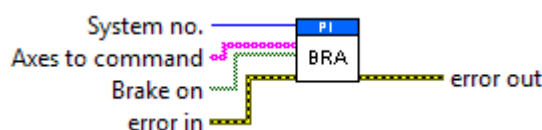
E-816: This command cannot be issued to a slave

E-761: This command returns the averaging (oversampling) factor which influences the "Sensor sampling time" and "Servo update time" parameters.



#### 4.16.15. BRA.vi

Switches brake for given axes on or off.



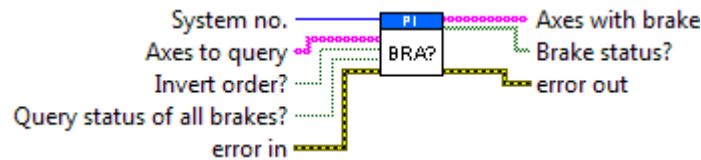
#### 4.16.16. BRA?.vi

Returns list of axes with a brake and status of the brakes for queried axes. Requires SPA?.vi to be present.

C-843, C-843.PM, C-844, Mercury: Invert order should be FALSE. Brake status? is not valid.

C-848, C-880: Invert order should be TRUE. Brake status? is not valid

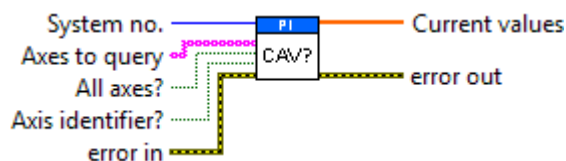
C-884, C-891, Mercury\_GCS: Invert order is not valid. VI returns axes with brakes and their brake status.



#### 4.16.17. CAV?.vi

Get the current value (in physical units) of the variable controlled by the selected control mode.

C-413: If All axes? = TRUE, then Axis identifier? can be FALSE.



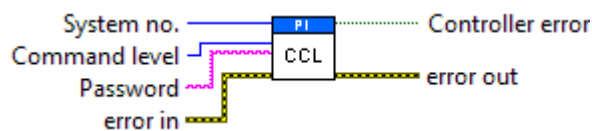
#### 4.16.18. CCL.vi

If password is correct, this vi sets the command level of the controller and queries ERR?. Controller error is TRUE if selected system report error code not equal to zero. Use HLP?.vi to determine which commands are available in the current command level.

C-413, C-867, C-877, C-884, C-891, E-517, E-518, E-709, E-710, E-712, E-725, E-727, E-753, E-754, E-755, E-870, E-871, E-872, E-873, E-874, F-206, M-8X0, Mercury\_GCS: Command level can be 0 (only commands needed for normal operation are available) or 1 (all commands from command level 0 plus special commands for advanced users are available). Password for CCL 1 is "ADVANCED".

C-867, C-884, Mercury\_GCS:

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



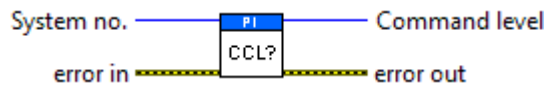
#### 4.16.19. CCL?.vi

Returns the current command level.

C-867, C-884, Mercury\_GCS:

E-816: This command cannot be issued to a slave. Check controller manual to find out if CCL? is supported.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

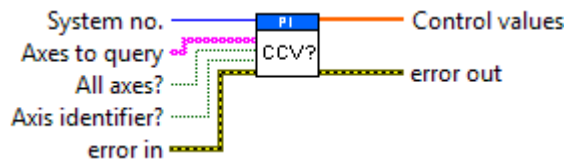


#### 4.16.20. CCV?.vi

Returns last valid control value for currently enabled control method.

C-413: If All axes? = TRUE, then Axis identifier? can be FALSE.

C-867: If All axes? = TRUE, then Axis identifier? can be FALSE.



#### 4.16.21. CLR.vi

Clears axis status for specified axes.

C-702: If All axes = TRUE, then Axis identifier must be TRUE

C-843: If All axes = TRUE, then Axis identifier must be TRUE

C-843.PM: If All axes = TRUE, then Axis identifier must be TRUE

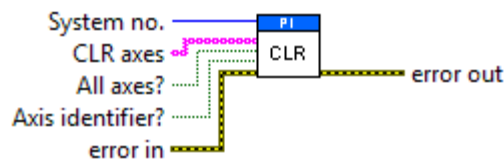
C-848: If All axes = TRUE, then Axis identifier must be TRUE

C-865: If All axes = TRUE, then Axis identifier must be TRUE

C-866: If All axes = TRUE, then Axis identifier must be TRUE

C-880: If All axes = TRUE, then Axis identifier must be TRUE

E-755: If All axes? = TRUE, then Axis identifier? must be TRUE



#### 4.16.22. CMO.vi

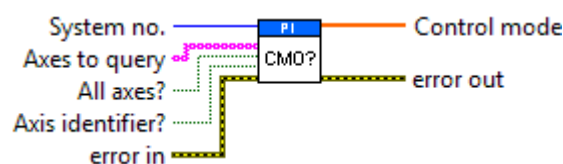
Sets the control mode and checks for error. Controller error is TRUE if selected system reports error code unequal to 0.



#### 4.16.23. CMO?.vi

Returns the current control mode.

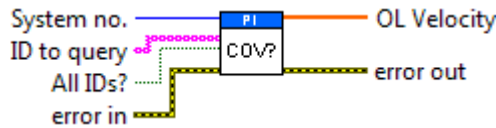
C-413: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.16.24. COV?.vi

Returns current open loop velocity for specified IDs.

E-870: Only one ID to query per command allowed.



#### 4.16.25. CST.vi

Assigns axes to stages and queries ERR?. With this command the stage assignment of the connected axes can be changed. Valid stage names can be listed with VST?.vi.

Axes assignment is written to volatile memory only and will be lost after restarting the controller. To assign the axes permanently, please use the WPA command.

C-877, C-884, E-872, E-873: Command is available via USB interface only.

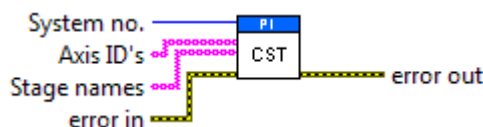
C-891: : Command is available via USB interface only. Axes assignment will be written to non volatile memory and will be available even after restarting the controller. Controller may reboot while CST command is in progress.

E-761: The settings are automatically written to non-volatile memory. Valid stage names are "ID-STAGE" for configured axes (a stage should be connected) and "NOSTAGE" for non-configured axes (no stage should be connected). The axis configuration as "ID-STAGE" is required before you can address any move command to this axis (i.e. to the connected stage).

C-867, E-871, Mercury\_GCS: Command is available via USB, USB DaisyChain or RS232 DaisyChain interface only.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). Only valid for separate axes A and/or B.

Pollux: See HydraPollux\_GCS\_DLL Manual for specialities of the Pollux controller family regarding stage name settings.



#### 4.16.26. CST?.vi

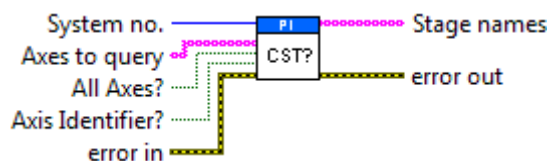
Returns the name of the connected stage for queried axes.

C-702, C-843, C-843.PM, C-844, C-848, C-880, C-887, F-206, M-8X0: If All axes? = TRUE, then Axis identifier? must be TRUE

C-413, C-865, C-866, C-867, C-877, C-884, C-885, C-891, E-135, E-709, E-710, E-712, E-725, E-727, E-753, E-754, E-755, E-761, E-861, E-871, E-873, E-874, Hydra, Mercury, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE

C-887, F-206, M-8X0: Check HLP?/HELP answer to find out if CST? is supported

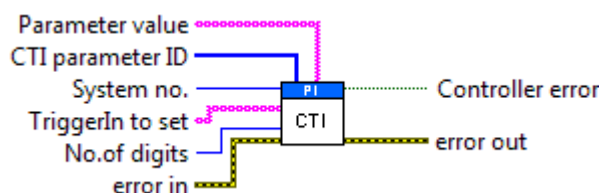
Pollux: If All axes? = TRUE, then Axis identifier? can be FALSE. See HydraPollux\_GCS\_DLL Manual for specialities of the Pollux controller family regarding stage name settings.



#### 4.16.27. CTI.vi

This VI configures the Trigger Input conditions for the given digital input line, waits 100 ms and queries ERR?. The trigger input actions only become active when enabled with TRI. See user manual for available CTI Parameter IDs, parameter numbers and values. No. of digits is the number of digits after the decimal point in the numeric parameter value(s) that will be sent. Controller error is TRUE if selected system reports error code which is not 0.

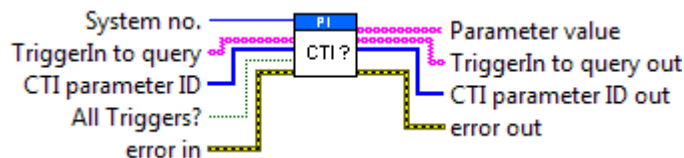
E-709:



#### 4.16.28. CTI?.vi

Returns the Trigger Input configuration for the given trigger input line. See user manual for available TriggerIn IDs and CTI parameter IDs.

E-709:



#### 4.16.29. CTO.vi

This VI configures the trigger output conditions for the given trigger output line, waits 100 ms and queries ERR?. The trigger output will always be periodically. The trigger output only becomes active when enabled with TRO. Note: Do not use DIO when an axis was configured with CTO. See user manual or GCS DLL manual for available parameter numbers and values. No. of digits is the number of digits after the decimal point in the numeric parameter value(s) that will be sent. Controller error is TRUE if selected system reports error code not equal to 0.

C-413: TriggerOut to set can be 1 to 5.

C-702: TriggerOut to set can be 1 to 8. DIO uses A-H to designate the same lines CTO calls 1-8.

C-843: TriggerOut to set can be 1 to 4 (4-axis card) or 2 (2-axis card).

C-866: TriggerOut to set can be 1.

C-867, C-877: Check controller user manual for available output trigger numbers.

C-884, E-871, E-873, E-874: Check C-884 user manual for available output trigger numbers.

E-517, E-518: TriggerOut to set can be 1 to 3. The trigger output is immediately active when the condition given by the CTO configuration is fulfilled (there is no TRO command). If the CTO Parameter ID is "Trigger Mode" and the Parameter



value is "Generator Trigger", then the trigger points in the waveform must be set with TWS and/or with WGO.

The current trigger output configuration is saved with the WPA command, in addition to the current parameter values and other settings.

The width of a trigger pulse is 30  $\mu$ s by default, except with the MinMaxThreshold trigger mode where the pulse width depends on the threshold settings. You can change the default pulse width using the "Pulse Width parameter", ID 0x0E000900. Possible values are in the range of 10 to 150  $\mu$ s.

The assignment of the trigger lines to the axes of the controller is fixed (DIO\_O1 belongs to the first axis (A by default), DIO\_O2 to the second axis (B by default) and DIO\_O3 to the third axis (C by default)).

E-709: TriggerOut to set can be 1 to 2. The trigger output is immediately active when the condition given by the CTO configuration is fulfilled (there is no TRO command). If the CTO Parameter ID is "Trigger Mode" and the Parameter value is "Generator Trigger", then the trigger points in the waveform must be set with TWS.

E-712: TriggerOut to set can be 1 to 7. The trigger output is immediately active when the condition given by the CTO configuration is fulfilled (there is no TRO command). If the CTO Parameter ID is "Trigger Mode" and the Parameter value is "Generator Trigger", then the trigger points in the waveform must be set with TWS. By default, axis 1 is connected to TriggerOut line 1, axis 2 to line 2, axis 3 to line 3, ..., axis n to line n. If the number of TriggerOut lines exceeds the number of axes, the "surplus" lines are all connected to the last axis.

E-725: TriggerOut to set can be 1 to 7. The trigger output is immediately active when the condition given by the CTO configuration is fulfilled (there is no TRO command). If the CTO Parameter ID is "Trigger Mode" and the Parameter value is "Generator Trigger", then the trigger points in the waveform must be set with TWS. By default, axis 1 is connected to TriggerOut line 1, axis 2 to line 2, axis 3 to line 3, ..., axis n to line n. If the number of TriggerOut lines exceeds the number of axes, the "surplus" lines are all connected to the last axis.

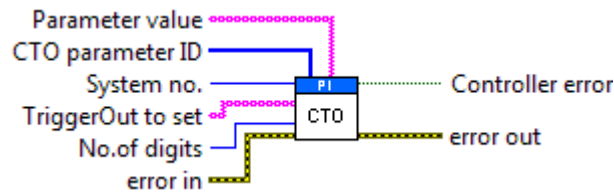
E-727: TriggerOut to set can be 1 to 3. The trigger output is immediately active when the condition given by the CTO configuration is fulfilled (there is no TRO command). If the CTO Parameter ID is "Trigger Mode" and the Parameter value is "Generator Trigger", then the trigger points in the waveform must be set with TWS. By default, axis 1 is connected to TriggerOut line 1, axis 2 to line 2, axis 3 to line 3. If the number of TriggerOut lines exceeds the number of axes, the "surplus" lines are all connected to the last axis.

E-753: TriggerOut to set can be 1. The trigger output is immediately active when the condition given by the CTO configuration is fulfilled (there is no TRO command). If the CTO Parameter ID is "Trigger Mode" and the Parameter value is "Generator Trigger", then the trigger points in the waveform must be set with TWS.

E-754: TriggerOut to set can be 1. The trigger output is immediately active when the condition given by the CTO configuration is fulfilled (there is no TRO command). If the CTO Parameter ID is "Trigger Mode" and the Parameter value is "Generator Trigger", then the trigger points in the waveform must be set with TWS.

Mercury: TriggerOut to set can be 1 to N (N: number of connected axes, if supported by corresponding Mercury firmware).

Mercury\_GCS: Check Mercury user manual for available output trigger numbers.



#### 4.16.30. CTO?.vi

Returns the Trigger Output configuration for the given trigger output line.

C-413: TriggerOut to query can be 1 to 5.

C-702: TriggerOut to query can be 1 to 8.

C-843: TriggerOut to query can be 1 to 4 (4-axis card) or 2 (2-axis card).

C-866: TriggerOut to query can be 1.

C-867, C-877: Check controller user manual for available output trigger numbers.

C-884, E-871, E-873, E-874: Check controller user manual for available output trigger numbers.

E-517, E-518: TriggerOut to query can be 1 to 3.

E-709: TriggerOut to query can be 1 to 2.

E-712: TriggerOut to query can be 1 to 7.

E-725: TriggerOut to query can be 1 to 7.

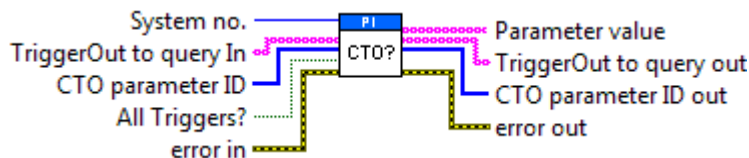
E-727: TriggerOut to query can be 1 to 3.

E-753: TriggerOut to query can be 1.

E-754: TriggerOut to query can be 1.

Mercury: TriggerOut to query can be 1 to N (N: number of connected axes, if supported by corresponding Mercury firmware).

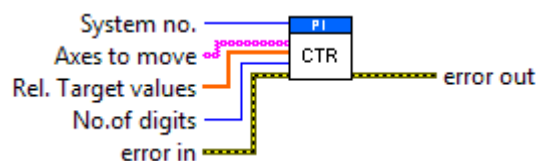
Mercury\_GCS: Check Mercury user manual for available output trigger numbers.



#### 4.16.31. CTR.vi

Set relative target value for currently selected control mode. A motion induced by CTR can be interrupted by #24, STP or HLT. Servo must be enabled when commanding CTR. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

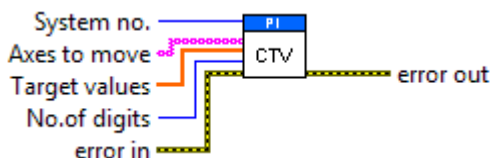
C-413: Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.



#### 4.16.32. CTV.vi

Set target value for currently selected control mode. A motion induced by CTV can be interrupted by #24, STP or HLT. Servo must be enabled when commanding CTV. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

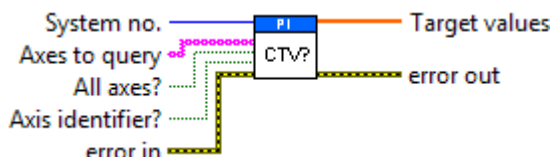
C-413: Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.



#### 4.16.33. CTV?.vi

Returns the last valid target value (in physical units) for the currently selected control mode.

C-413: If All axes? = TRUE, then Axis identifier? can be FALSE.

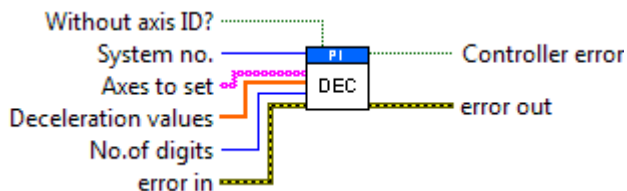


#### 4.16.34. DEC.vi

Sets closed-loop deceleration and checks for error. If Without axis ID? is TRUE, then Axes to set is ignored and first field of Deceleration values array is used for deceleration command. Number of digits is the number of digits after the decimal point in the deceleration value(s) that will be sent. Controller error is TRUE if selected system reports error code which is not 0.

C-843, C-867, C-877, C-884, C-891, E-861, E-874, Mercury\_GCS: Without axis ID? = FALSE. Deceleration unit is mm/s<sup>2</sup>.

E-873: Without axis ID? = FALSE. Deceleration unit is mm/s<sup>2</sup>.

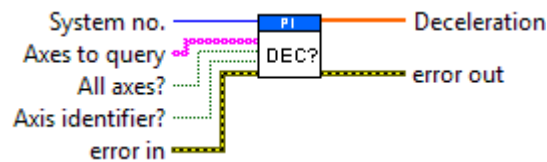


#### 4.16.35. DEC?.vi

Returns closed-loop deceleration setting for specified axes.

C-843, C-867, C-877, C-884, C-891, E-861, E-874, Hydra, Pollux, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE. Deceleration unit is mm/s<sup>2</sup>.

E-873: If All axes? = TRUE, then Axis identifier? can be FALSE. Deceleration unit is mm/s<sup>2</sup>.



#### 4.16.36. DEL.vi

Delays the command interpreter of specified system for given "Delay time". DEL is used within macros primarily. Do not mistake MAC DEL which deletes macros for DEL which delays.

All systems: Delay time unit is ms.

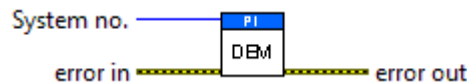
E-816: This command cannot be issued to a slave. Check controller manual to find out if DEL is supported.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.16.37. DEM.vi

Starts demo mode of the system controller. Use #24 to stop demo.

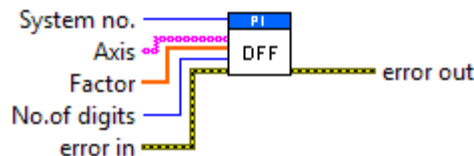


#### 4.16.38. DFF.vi

Defines scale factor which is applied to the basic unit (default is 1). E.g., 25.4 changes the physical unit from mm to inches. No. of digits is the number of digits after the decimal point in the factor value(s) that will be sent.

Example: The physical unit is mm and the scale factor is 1. The current position of a stage is 12. Now the scale factor is set to 3 with DFF. Reading the position gives 4 as result. A relative move of 1.5 causes the stage to move 4.5 mm.

All systems: Factor can only be positive



#### 4.16.39. DFF?.vi

Returns constant unit value for specified axes (e.g. 25.4 for inches).

C-702: If All axes = TRUE, then Axis identifier can be FALSE

C-843: If All axes = TRUE, then Axis identifier must be TRUE

C-843.PM: If All axes = TRUE, then Axis identifier must be TRUE

C-844: If All axes = TRUE, then Axis identifier must be TRUE

C-848: If All axes = TRUE, then Axis identifier can be FALSE

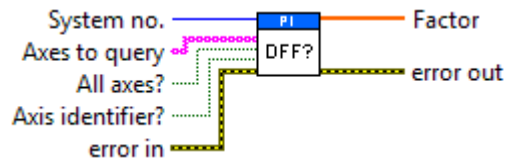
C-865: If All axes = TRUE, then Axis identifier must be TRUE

C-866: If All axes = TRUE, then Axis identifier must be TRUE

C-880: If All axes = TRUE, then Axis identifier can be FALSE

E-710: If All axes = TRUE, then Axis identifier must be TRUE

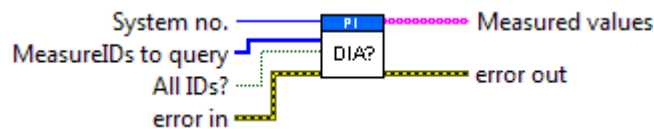
Mercury: If All axes = TRUE, then Axis identifier can be FALSE



#### 4.16.40. DIA?.vi

Returns diagnosis information. Use HDI?.vi to find out valid MeasureIDs and their meanings.

E-870: If All ID's? = FALSE, only one MeasureID to query per command allowed.



#### 4.16.41. DIO.vi

Switches digital outputs on or off. For DO mode format = Boolean, DO mode must be selected for each DO to command. For DO mode format = hexadecimal (decimal), DO mode and DO's to command are not valid and DO pattern must be selected correctly in hexadecimal (decimal) format.

C-413: DO mode format is Boolean, DO's to command can be 1 to 5. DO pattern is not valid.

C-702, C-843, C-843.PM, C-848: DO's to command can be A-H. DO mode format is Boolean, DO pattern is not valid.

C-865, C-866: DO's to command can be A-B. . DO mode format is Boolean, DO pattern is not valid.

C-867, C-891, E-861, E-871, E-873, E-874, Mercury\_GCS: For DO mode format = Boolean, DO's to command can be 1-4 and only one DO per command allowed. For DO mode format = hexadecimal, DO's to command and DO mode are not valid and DO pattern must be set correctly (all DO's are set with one single command).

C-877: For DO mode format = Boolean, DO's to command can be 1-4 and only one DO per command allowed. For DO mode format = hexadecimal, DO's to command and DO mode are not valid and DO pattern must be set correctly (all DO's are set with one single command). Check HLP? answer to find out if DIO is supported

C-884: For DO mode format = Boolean, DO's to command can be 1 to 4. Up to four DO's per command call are allowed. For DO mode format = hexadecimal, DO's to command and DO mode are not valid and DO pattern must be set correctly (all DO's are set with one single command).

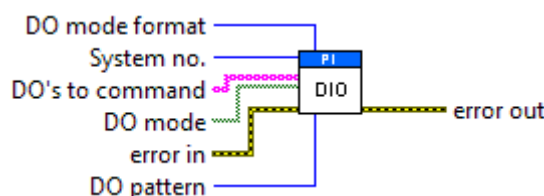
C-880: DO's to command can be A-H (one C-842 inside), A-P (two C-842 inside) etc. DO mode format is Boolean, DO pattern is not valid.

F-206, M-8X0: For DO mode format = Boolean, DO's to command can be 1 to 8. For DO mode format = hexadecimal, DO's to command and DO mode are not valid and DO pattern must be set correctly (all DO's are set with one single command).

Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

Hydra: For DO mode format = Boolean, DO's to command can be 1-3 and only one DO per command allowed. For DO mode format = hexadecimal, DO's to command and DO mode are not valid and DO pattern must be set correctly (all DO's are set with one single command).

Mercury: DO's to command can be A-D, E-H, I-L etc., see GCS DLL Manual for details. DO mode format is Boolean, DO pattern is not valid.



#### 4.16.42. DIO?.vi

Returns digital input values for queried digital inputs. Uses TIO?.vi (GCS 1.0 and 2.0) and TVI?.vi (GCS 1.0 to determine available DI identifiers if All DI's = TRUE and DI identifier = TRUE. If Query pattern? = TRUE, returns binary pattern for the digital input status of all channels.

C-413: If All DI's = TRUE, then DI identifier must be FALSE. Invert order for TVI? is not valid. DI's to query are 1 to . Query pattern? is not valid.

C-702, C-848, C-880: If All DI's = TRUE, then DI identifier can be FALSE and Invert order for TVI? must be TRUE. Query pattern? is not valid.

C-843, C-843.PM, C-865, C-866: If All DI's = TRUE, then DI identifier must be TRUE and Invert order for TVI? must be FALSE. Query pattern? is not valid..

C-867, C-884, E-861, E-871, E-873, E-874, Mercury\_GCS: If All DI's = TRUE, then DI identifier must be FALSE. Invert order for TVI? is not valid. DI's to query are 1-4. Query pattern? can be TRUE.

C-877: If All DI's = TRUE, then DI identifier must be FALSE. Invert order for TVI? is not valid. DI's to query are 1-4. Query pattern? can be TRUE. Check HLP?/HELP answer to find out if DIO is supported

C-891: If All DI's = TRUE, then DI identifier must be FALSE. Invert order for TVI? is not valid. . If All DI's = FALSE, only one DI to query per command allowed and DI's to query can be 1 to 4. Query pattern? can be TRUE.

E-517, E-518: If All DI's = TRUE, then DI identifier can be FALSE. Invert order for TVI? is not valid. DI's to query are 1-3. Query pattern? is not valid.

E-709, E-727: If All DI's = TRUE, then DI identifier can be FALSE. Invert order for TVI? is not valid. Query pattern? is not valid.

E-754: If All DI's = TRUE, then DI identifier can be FALSE. Invert order for TVI? is not valid. DI's to query are 1-2. Query pattern? is not valid.

E-761: All DI's = FALSE. DI's to query are "1". Query pattern? is not valid. Note that the E-761 has no genuine digital input lines, but the analog input is internally interpreted as digital input for triggering tasks (see E-761 user manual), and its signal state can be queried by this command. If the voltage on the analog input is < 0.8 V, the signal is interpreted as LOW, if the voltage is = 2.4 V, the signal is interpreted as HIGH.

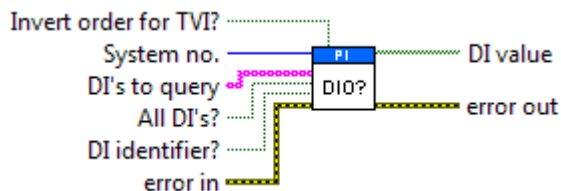
C-887, F-206, M-8X0: If All DI's = TRUE, then DI identifier must be FALSE. Invert order for TVI? is not valid. DI's to query are 1 to 8. Query pattern? can be TRUE.

Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

E-872: If All DI's = TRUE, then DI identifier must be FALSE. Invert order for TVI? is not valid. DI's can be queried with TIO?.vi. Query pattern? can be TRUE.

Hydra: If All DI's = TRUE, then DI identifier must be FALSE. Invert order for TVI? is not valid. DI's to query are 1-2. Query pattern? can be TRUE.

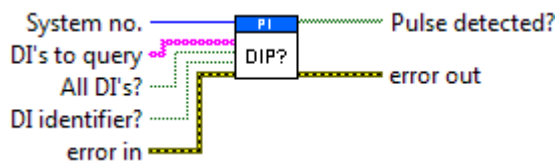
Mercury: All DI's must be FALSE. DI's to query can be A-D, E-H, I-L etc., see GCS DLL Manual for details. Query pattern? is not valid.



#### 4.16.43. DIP?.vi

Returns if a digital pulse was detected since last call of DIP?.

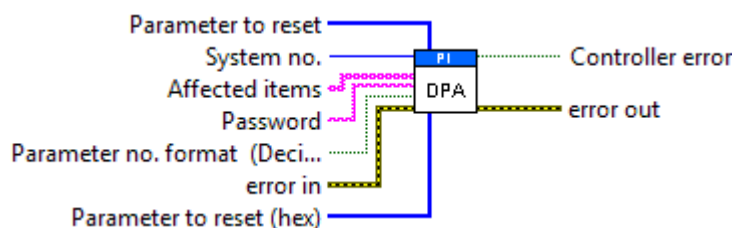
E-816: If All DI's = TRUE, then DI identifier can be FALSE. DI's to query are 1. Check controller manual to find out if DIP? is supported.



#### 4.16.44. DPA.vi

If password is correct, resets parameters or settings to default values, waits 3000 ms and queries ERR?. It does not overwrite settings in the non-volatile memory. For details, see the specific documentation. If parameter no. is in decimal format, use Parameter to reset input, for hexadecimal parameter numbers use Parameter to reset(hex) input and switch Parameter no. format to TRUE. Do not mix decimal and hex. parameter numbers in one call. If Affected items is an empty array, DPA is sent without item and parameter specification.

C-867, C-887, F-206, M-8X0: If Affected items = empty array, the currently valid values of all parameters affected by the specified password are reset. Parameter no. format is FALSE (num.). Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).





#### **4.16.45. DRC.vi**

This VI configures the data recording, waits 100 ms and queries ERR?. See GCS DLL manual or user manual for available recording and trigger options.

GCS 2.0: Trigger option must be 0.

Controller error is TRUE if selected system reports an error code which is not 0.

C-843: Trigger option must be 0. See user manual for available record options.

The C-843 has four data recorder tables. The available points per table depend on the host computer's memory only. Some hardware revisions do not allow the parallel use of DIO and the data recorder. To switch between both, the C-843 needs to be reconnected.

C-866: Trigger option must be 0. See C-866\_GCS\_Commands\_SM150E.pdf for available record options.

C-867: See user manual for available record options. The C-867 has four data recorder tables with 8192 points per table.

C-877: See user manual for available record options, number of data recorder tables and points per table. Trigger option must be 0.

C-884: See user manual for available record options. Trigger option must be 0. The C-884 has eight data recorder tables with 8192 points per table.

C-891: See user manual for available record options. Trigger option must be 0. The C-891 has four data recorder tables with 4096 points per table. The number of tables can be changed to 1, 2 or 8 by setting the appropriate parameter value, see user manual for details.

E-517, E-518: Trigger option must be 0. See user manual for available record options. The number of data recorder tables is 3 with 8192 points per table. The current data recorder configuration is saved with WPA, in addition to the current parameter values and other settings.

E-709: See user manual for available record options. Trigger option must be 0. By default, the number of data recorder tables is 4. It can be reduced by setting the appropriate parameter value, see user manual for details.

E-710: Rec. table and Source ID must be identical.

C-413, E-712, E-725, E-727, E-753, E-754: Trigger option must be 0. See user manual for available record options. By default, the number of data recorder tables is 8. It can be reduced by setting the appropriate parameter value, see user manual for details.

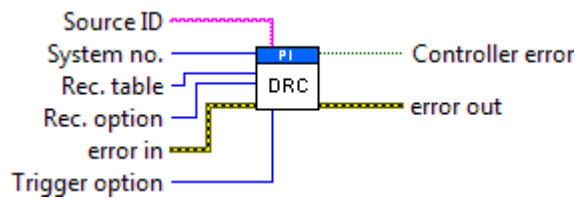
E-861: Trigger option must be 0. See user manual for available record options. The E-861 has two data recorder tables with 1024 points per table.

E-871, E-873: Trigger option must be 0. See user manual for available record options. The E-871, E-873 has two data recorder tables with 1024 points per table.

E-874: Trigger option must be 0. See user manual for available record options. The E-874 has two data recorder tables with 1024 points per table.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). See user manual for available record options, trigger options, number of data recorder tables and points per table. The number of points can be changed by setting the appropriate parameter value, see user manual of the controller for details.

Mercury\_GCS: Trigger option must be 0. See user manual for available record options, number of data recorder tables and points per table.



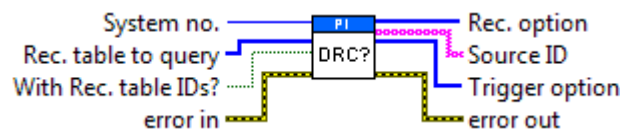
#### 4.16.46. DRC?.vi

Returns the data recording configuration (Source ID and Rec. option) for the queried record table.

GCS 2.0: Trigger option is not valid.

C-413, C-843, C-866. C-867, C-877, C-886, C-887, C-891, E-517, E-518, E-709, E-712, E-725, E-727, E-753, E-754, E-861, E-871, E-873, E-874, F-206, M-8X0, Mercury\_GCS: Trigger option is not valid.

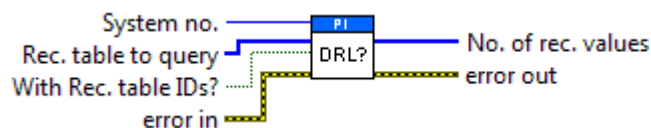
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.16.47. DRL?.vi

This VI returns the number of recorded data values for the given record tables.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.16.48. DRR? and display data.vi

Returns N recorded data points and displays them in a 2D graph by calling Show\_Save\_Load\_XY\_Data.vi. N must be less than or equal to Nmax. For large N values, communication timeout must be set long enough, otherwise a comm.error may occur. If Sample time is zero, it is set to 1.0 for displaying data in the 2D graph only.

Analog: Rec. table IDs, x0, N and Nmax are not valid. Without parameter? must be TRUE.

C-413: X0 >= 1. Nmax = 4096. The 4096 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 8. It can be reduced by setting the appropriate parameter value, see C-413 user manual for details.

C-702: Nmax = 262144. X0 >= 0.

C-843: X0 >= 1. Check C-843 user manual for valid Nmax values. Some hardware revisions don't allow the parallel use of DIO and the data recorder. To switch between both modes the C-843 needs to be reconnected. The number of tables is 4. The available points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 4. It can be reduced with DRC, see C-843 user manual for details. If N = -1 all points of the last record are returned.

C-866:  $X0 \geq 1$ .  $N_{max} = 32,256$ . If  $N = -1$  all points of the last record are returned.

C-867:  $X0 \geq 1$ .  $N_{max} = 8192$ . The number of tables is 4.

C-877:  $X0 \geq 1$ .  $N_{max} = 8192$ . The number of tables is 4.

C-884:  $X0 \geq 1$ .  $N_{max} = 8192$ . The number of tables is 8.

C-891:  $X0 \geq 1$ .  $N_{max} = 16384$ . The C-891 has four data recorder tables with 4096 points per table. It can be changed to 1, 2 or 8 by setting the appropriate parameter value, see user manual for details.

E-517, E-518:  $X0 \geq 1$ .  $N_{max} = 8192$ . The number of tables is 3.

E-709:  $X0 \geq 1$ .  $N_{max} = 4096$ . The 4096 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 4. It can be reduced by setting the appropriate parameter value, see E-709 user manual for details.

E-710:  $X0 \geq 1$ .  $N_{max} = 32,256$ .

E-712:  $X0 \geq 1$ .  $N_{max} = 262,144$ . The 262,144 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 8. It can be reduced by setting the appropriate parameter value, see E-712 user manual for details.

E-725, E-727:  $X0 \geq 1$ .  $N_{max} = 262,144$ . The 262,144 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 8. It can be reduced by setting the appropriate parameter value, see controller user manual for details.

E-753:  $X0 \geq 1$ .  $N_{max} = 65,536$ . The 65,536 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 8. It can be reduced by setting the appropriate parameter value, see E-753 user manual for details.

E-754:  $X0 \geq 1$ .  $N_{max} = 1,048,576$ . The 1,048,576 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 8. It can be reduced by setting the appropriate parameter value, see E-754 user manual for details.

E-755:  $X0 \geq 1$ .  $N_{max} = 4,096$ .

E-761:  $X0 \geq 0$ . Recording takes place for all recorder tables as long as the wave generator is running for an arbitrary axis, when an impulse is started with IMP or when a step is started with STE. The assignment of axis and data sources to the recorder tables is as follows:

table 1: axis 1 actual position

table 2: axis 2 actual position

table 3: axis 3 actual position

table 4: analog input voltage (same value as read with TAV?, i.e. contains gain and offset for the analog input, see E-761 user manual).

The maximum number of data points is 8192 per recorder table.

E-861:  $X0 \geq 1$ .  $N_{max} = 1024$ . Two data recorder tables with 1024 points per table are provided.

E-871, E-873, E-874:  $X0 \geq 1$ .  $N_{max} = 1024$ . The number of tables is 2.

C-887, F-206:  $X0 \geq 1$ . See C-887 user manual for  $N_{max}$  default value (can be changed with SPA.vi). Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). See controller user manual for available data recorder tables and points.

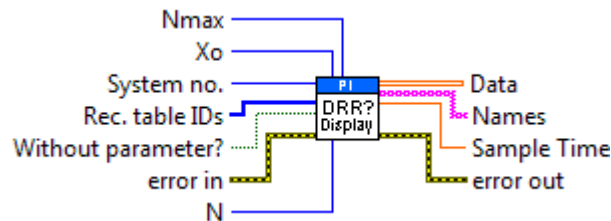
C-887, M-8X0: For GCS syntax version = 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), Rec. table IDs,  $x0$ ,  $N$  and  $N_{max}$  are not valid and Without parameter? must be TRUE.

For GCS syntax version = 2.0,  $X_0 \geq 1$ . See C-887 user manual for Nmax default value (can be changed with SPA.vi).

Only supported if controller is based on C-842.80 board or GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For GCS 1.0, returns 360 motor current values recorded during execution of DRV.

For GCS 2.0 and higher, see controller user manual for available data recorder tables and points.

Mercury\_GCS:  $X_0 \geq 1$ . Nmax = 1024. See controller user manual for available data recorder tables.



#### 4.16.49. DRR?.vi

Returns N recorded data points. If N is greater than Nmax, multiple queries are sent. For large Nmax values, communication timeout must be set long enough, otherwise a communication error may occur.

Analog: Rec. table IDs,  $x_0$ , N and Nmax are not valid. Without parameter? must be TRUE.

C-413:  $X_0 \geq 1$ . Nmax = 4096. The 4096 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 8. It can be reduced by setting the appropriate parameter value, see C-413 user manual for details.

C-702: Nmax = 262144.  $X_0 \geq 0$ .

C-843:  $X_0 \geq 1$ . Check C-843 user manual for valid Nmax values. Some hardware revisions don't allow the parallel use of DIO and the data recorder. To switch between both modes the C-843 needs to be reconnected. The number of tables is 4. The available points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 4. It can be reduced with DRC, see C-843 user manual for details. If N = -1 all points of the last record are returned.

C-866:  $X_0 \geq 1$ . Nmax = 32,256. If N = -1 all points of the last record are returned.

C-867:  $X_0 \geq 1$ . Nmax = 8192. The number of tables is 4.

C-877:  $X_0 \geq 1$ . Nmax = 8192. The number of tables is 4.

C-884:  $X_0 \geq 1$ . Nmax = 8192. The number of tables is 8.

C-891:  $X_0 \geq 1$ . Nmax = 16384. The C-891 has four data recorder tables with 4096 points per table. It can be changed to 1, 2 or 8 by setting the appropriate parameter value, see user manual for details.

E-517, E-518:  $X_0 \geq 1$ . Nmax = 8192. The number of tables is 3.

E-709:  $X_0 \geq 1$ . Nmax = 4096. The 4096 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 4. It can be reduced by setting the appropriate parameter value, see E-709 user manual for details.

E-710:  $X_0 \geq 1$ . Nmax = 32,256.

E-712:  $X_0 \geq 1$ . Nmax = 262,144. The 262,144 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is

8. It can be reduced by setting the appropriate parameter value, see E-712 user manual for details.

E-725, E-727:  $X0 \geq 1$ .  $N_{max} = 262,144$ . The 262,144 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 8. It can be reduced by setting the appropriate parameter value, see controller user manual for details.

E-753:  $X0 \geq 1$ .  $N_{max} = 65,536$ . The 65,536 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 8. It can be reduced by setting the appropriate parameter value, see controller user manual for details.

E-754:  $X0 \geq 1$ .  $N_{max} = 1,048,576$ . The 1,048,576 points are in equal shares assigned to the available data recorder tables. By default, the number of tables is 8. It can be reduced by setting the appropriate parameter value, see E-754 user manual for details.

E-755:  $X0 \geq 1$ .  $N_{max} = 4,096$ .

E-761:  $X0 \geq 0$ . Recording takes place for all recorder tables as long as the wave generator is running for an arbitrary axis, when an impulse is started with IMP or when a step is started with STE. The assignment of axis and data sources to the recorder tables is as follows:

table 1: axis 1 actual position

table 2: axis 2 actual position

table 3: axis 3 actual position

table 4: analog input voltage (same value as read with TAV?, i.e. contains gain and offset for the analog input, see E-761 user manual).

The maximum number of data points is 8192 per recorder table.

E-861:  $X0 \geq 1$ .  $N_{max} = 1024$ . Two data recorder tables with 1024 points per table are provided.

E-871, E-873, E-874:  $X0 \geq 1$ .  $N_{max} = 1024$ . The number of tables is 2.

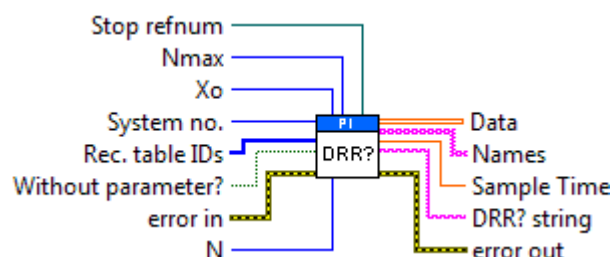
C-887, F-206:  $X0 \geq 1$ . See controller user manual for  $N_{max}$  default value (can be changed with SPA.vi). Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). See controller user manual for available data recorder tables and points.

C-887, M-8X0: For GCS syntax version = 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), Rec. table IDs,  $x_0$ , N and  $N_{max}$  are not valid and Without parameter? must be TRUE.

For GCS syntax version = 2.0,  $X_0 \geq 1$ . See controller user manual for available data recorder tables and points. The maximum number of points to be recorded per table ( $N_{max}$ ) can be changed with SPA.vi.

Only supported if controller is based on C-842.80 board or GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For GCS 1.0, returns 360 motor current values recorded during execution of DRV.

Mercury\_GCS:  $X_0 \geq 1$ .  $N_{max} = 1024$ . See controller user manual for available data recorder tables.



**4.16.50. DRT.vi**

This VI defines a trigger source for data recorder tables, waits 100 ms and queries ERR?. See GCS DLL manual or User manual for available trigger sources and values. Controller error is TRUE if selected system reports error code not equal to 0.

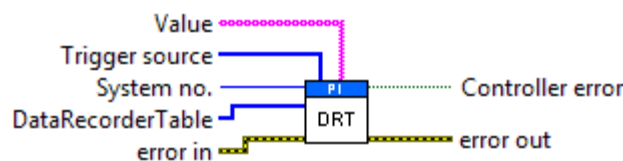
C-866: See C-866\_GCS\_Commands\_SM150E.pdf for available trigger sources and values. DataRecorderTable = 0 (the specified trigger source is set for all data recorder tables).

C-413, C-843, C-867, C-877, C-884, C-891, E-518, E-709, E-712, E-754, E-861, E-871, E-873, E-874, Mercury\_GCS: See user manual for available trigger sources and values. DataRecorderTable = 0 (the specified trigger source is set for all data recorder tables).

E-712, E-725, E-753: See user manual for available trigger sources and values. The specified trigger source is set for all data recorder tables.

E-727, E-754: See user manual for available trigger sources and values. The specified trigger source is set for all data recorder tables.

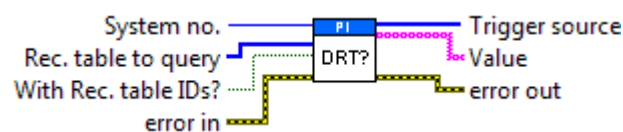
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). See user manual for available trigger sources and values. DataRecorderTable = 0 (the specified trigger source is set for all data recorder tables).

**4.16.51. DRT?.vi**

Returns the Data Recorder Trigger source and value for the queried data recorder tables.

E-712, E-725, E-753:

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

**4.16.52. DRV.vi**

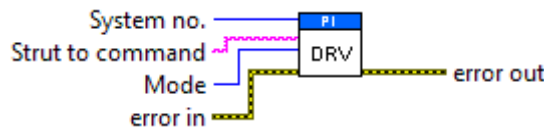
A spindle revolution of one single strut is performed while monitoring the motor current. For system maintenance purposes. User DRR?.vi or DRR? and display data.vi to read values back.

C-887, F-206: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). Requires command level 1 (CCL). Strut to command can be 1 to 6. Mode can be 1 (forward spindle rotation) or -1 (backward rotation)

C-887, M-8X0: Only supported if controller is based on C-842.80 or GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For GCS 2.0, requires command level 1 (CCL). Strut



to command can be 1 to 6. Mode can be 1 (forward spindle rotation) or -1 (backward rotation)



#### 4.16.53. EAX.vi

Enable axis. If disabled, no motion is executed and an error is set. Affected motion sources: motion commands like MOV, MVR, SMO, STE, IMP, SVA, OMA, OMR, MRT, MRW, wave generator commands, macros, analog input, joystick.

Disabling an axis with EAX stops the axis but does not imply that the motor current is zero. If a motion error occurs axis will be disabled automatically (and servo will be switched off).

Enabling an axis with EAX will not enable the servo of this axis (SVO.vi). Disabling an axis with EAX will disable the servo status of this axis.

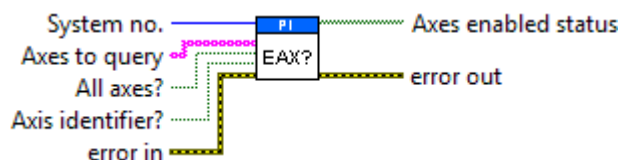
The servo status of an axis (SVO.vi) can only be enabled if the axis is enabled (EAX.vi), but enabling the servo with SVO.vi will not change the EAX status of this axis.



#### 4.16.54. EAX?.vi

Returns enabled status of queried axes.

All systems: If All axes? = TRUE, then Axis identifier? can be FALSE



#### 4.16.55. EGE.vi

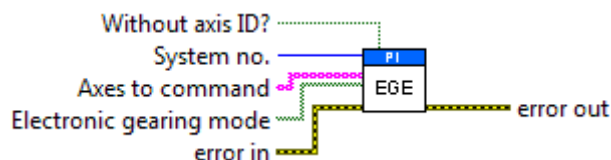
Sets electronic gearing mode for given axes. If Without axis ID is TRUE, then Axes to command is ignored and first field of Electronic gearing mode array is used.

C-843: Without axis ID = FALSE. This command can only be issued to a slave axis. The master-slave assignments are set with MAS.vi. Use MAS?.vi to get the current master axis, i.e. the one to be commanded with MOV.vi.

C-848: Without axis ID = FALSE. This command can only be issued to a slave axis. The master-slave assignments are hardware-dependent. Use MAS?.vi to get the related master axis, i.e. the one to be commanded with MOV.vi.

C-880: Without axis ID = FALSE. This command can only be issued to a slave axis. The master-slave assignments are hardware-dependent. Use MAS?.vi to get the related master axis, i.e. the one to be commanded with MOV.vi.

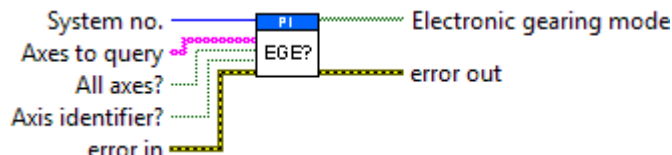




#### 4.16.56. EGE?.vi

Returns electronic gearing mode of queried axes.

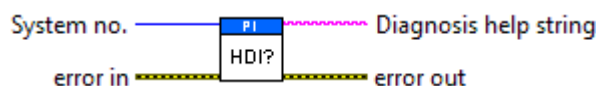
C-843, C-848, C-880: If All axes = TRUE, then Axis identifier can be FALSE



#### 4.16.57. HDI?.vi

Returns help on diagnosis information, received with DIA?.

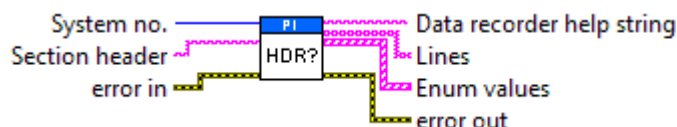
C-867, C-887, C-891:



#### 4.16.58. HDR?.vi

Returns help on data recording: possible parameter values for record sources and record trigger options (DRC, DRT), parameters to set and other information. If Section header contains a valid section name, Lines and Enum values return the corresponding section content.

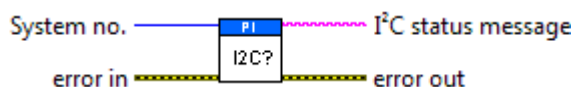
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.16.59. I2C?.vi

Returns the status message of the I<sup>2</sup>C bus.

E-816: Status message consists of status bit and channel ID. This command cannot be issued to a slave.



#### 4.16.60. IMP.vi

Performs a single impulse-move (two equal moves in opposite directions in quick succession) from the current position with specified impulse size (amplitude) and records a fixed number of actual positions at specified intervals thereafter, which can be read out with IMP?.vi (GCS 1.0) or DRR?.vi (GCS 2.0). If supported, "Delay" sets the number of servo loops between each position recording.

E-517, E-518: Controller saves up to 8,192 position values. Typically, IMP is used in open-loop mode. For a single step-move, see STE.vi . Use DRR?.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands like IMP are not allowed when the controller is in OFFLINE mode or when the wave generator output is active. When a macro is running on the controller, IMP will be executed not until the macro is finished or stopped. See "Control Value Generation" and "Control Modes" in the controller User manual for details.

C-413, E-709: Controller saves up to 4096 position values. Typically, IMP is used in open-loop mode. For a single step-move, see "STE.vi". Use DRR?.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

E-710: Delay = 0. Controller saves 32,256 position values. Width of impulse and sampling interval taken from "Table rate" parameter, set with SPA\_Hex.vi. Caution: "Table rate" parameter influences wave generator also, not only IMP. For a single step-move, see STE.vi. Use IMP?.vi to read position values back.

E-712: Delay = 0. Controller saves up to 262,144 position values. Typically, IMP is used in open-loop mode. For a single step-move, see STE.vi. Use DRR?.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-725, E-727: Delay = 0. Controller saves up to 262,144 position values. Typically, IMP is used in open-loop mode. For a single step-move, see STE.vi. Use DRR?.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

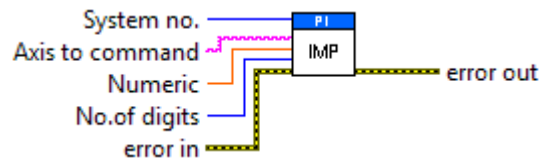
E-753: Controller saves up to 65,536 position values. Typically, IMP is used in open-loop mode. For a single step-move, see STE.vi. Use DRR?.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

E-754: Delay = 0. Controller saves up to 1,048,576 position values. Typically, IMP is used in open-loop mode. For a single step-move, see STE.vi. Use DRR?.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-755: Delay= 0. Controller saves 4,096 position values. Typically, IMP is used in open-loop mode. For a single step-move, see STE.vi. Use DRR?.vi to read recorded values back.

E-761: Delay is impuls width in servo loops. Default value is 0 (for one servo loop). Controller saves 8,192 position values. Typically, IMP is used in open loop mode. For a single step-move, see STE.vi. The number of servo cycles used for data recording depends on the setting made with RTR.vi. Use DRR?.vi or IMP?.vi to read position values back.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). See controller user manual for number of points (can be changed by setting the appropriate parameter value). For a single step-move, see STE.vi. Use DRR?.vi to read recorded values back.



#### 4.16.61. IMP?.vi

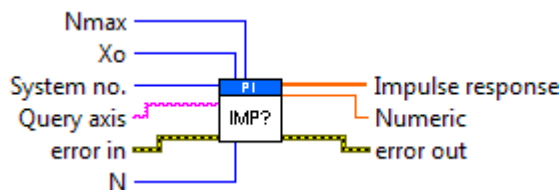
GCS 1.0 controller: Returns N saved impulse response points. N must be less than or equal to Nmax. For large N values, communication timeout must be set long enough, otherwise a comm.error may occur.

GCS 2.0 or higher: Returns IMP settings. Use DRR? to read impulse response points back.

E-517, E-518, E-709, E-754, E-755: xo, N and Nmax are not valid. VI reads impulse size. Impulse response is not valid.

E-710: Nmax = 32256. VI reads impulse response points. Impulse size is not valid.

E-761: Nmax = 8192. VI reads impulse response points. Impulse size is not valid.



#### 4.16.62. INI.vi

Initializes axes. System-specific: see individual GCS-DLL or system manual for details.

C-702: If All axes = TRUE, then Axis identifier must be TRUE

C-843: If All axes = TRUE, then Axis identifier must be TRUE

C-843.PM: If All axes = TRUE, then Axis identifier must be TRUE

C-844: If All axes = TRUE, then Axis identifier must be TRUE, It is necessary to wait a certain time - appr. 4 s - before sending the next command to prevent it from being lost.

C-848: If All axes = TRUE, then Axis identifier must be TRUE

C-865: If All axes = TRUE, then Axis identifier must be TRUE

C-866: If All axes? = TRUE, then Axis identifier? must be TRUE

C-880: If All axes = TRUE, then Axis identifier must be TRUE

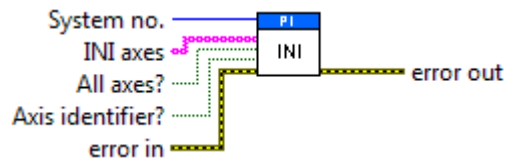
C-885: All axes must be TRUE, Axis identifier must be FALSE

E-710: If All axes = TRUE, then Axis identifier must be TRUE

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For GCS 2.0 or higher, VI calls FRF instead for compatibility. To initialize hexapod, All axes = TRUE, Axis identifier = FALSE, (separate axes A,B (if present) will not be initialized); to initialize axes A,B: All axes = FALSE; VI will not wait for INI procedure to finish. Depending on the firmware version on the controller, motion initiated by INI may not be able to be stopped by STOP, #24 or #27.

Hydra, Pollux: If All axes? = TRUE, then Axis identifier? can be FALSE.

Mercury: If All axes = TRUE, then Axis identifier can be FALSE



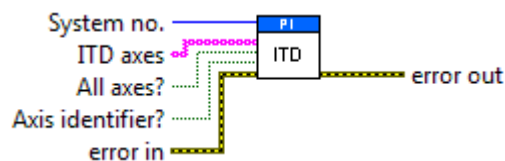
#### 4.16.63. ITD.vi

Sets SPA and DFF values of specified axes to default values.

C-702: If All axes = TRUE, then Axis identifier must be TRUE

C-848: If All axes = TRUE, then Axis identifier must be TRUE

C-880: If All axes = TRUE, then Axis identifier must be TRUE



#### 4.16.64. JOG.vi

Starts motion with the given (constant) velocity values for the specified axes. The sign of the velocity values determines the direction of motion.

When motion started with JOG is executed, the target value is changed continuously according to the given velocity (can be checked with MOV?.vi).

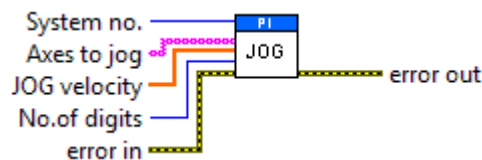
Motion started with JOG is executed in addition to motion started with other move commands (e.g. MOV.vi or MVR.vi ).

As long as motion of the axis is caused by JOG only, the axis stays on target (i.e. ONT?.vi responds with 1 since the target is continuously adapted to the actual motion).

Motion started by JOG is stopped in the following cases:

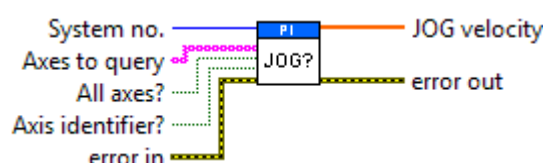
- The velocity is set to 0 with JOG.
- #24, STP or HLT is sent: these commands set the velocity for JOG to 0.
- A travel range limit is reached: the velocity for JOG remains unchanged, no error is set, the target value is set equal to the limit value.

JOG can be changed while the axis is moving.



#### 4.16.65. JOG?.vi

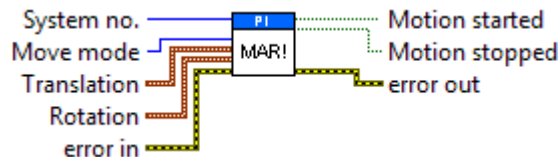
Get the velocity and direction for motion caused by JOG.



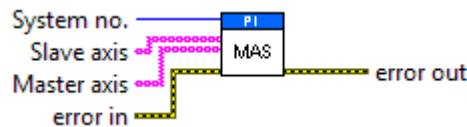
**4.16.66. MAR!.vi**

M-8X0-specific command. Moves stage. Waits for M-8X0 controller to reply 1 when all calculations are done and the movement starts, and another 1 when motion is complete.

C-887, M-8X0: Only for GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). For compatibility, calls MOV.vi for GCS 2.0 or higher (in this case Motion started returns TRUE immediately, Motion stopped is determined by #5 polling).

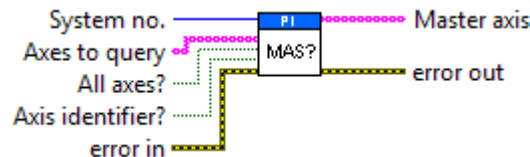
**4.16.67. MAS.vi**

Set master axis.

**4.16.68. MAS?.vi**

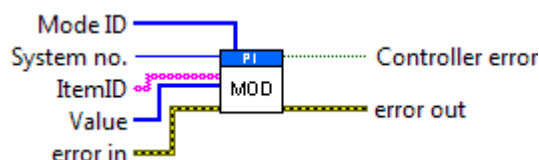
Returns the master axis for the specified axis.

C-843, C-848, C-880: If All axes = TRUE, then Axis identifier can be FALSE.

**4.16.69. MOD.vi**

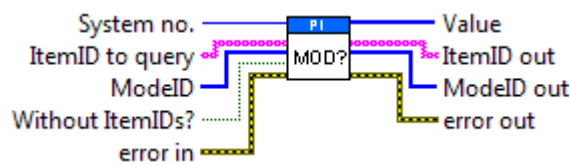
This VI sets modes, waits 100 ms and queries ERR?. See driver manual for available ItemIDs, Mode IDs and values.

E-870: Only one ModeID for only one ItemID per command allowed.

**4.16.70. MOD?.vi**

Returns mode values for queried ItemIDs and ModeIDs. If Without ItemIDs? is TRUE, all available values for all Item/ModeIDs are returned. See driver manual for available ItemIDs and ModeIDs.

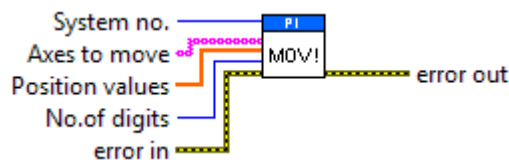
E-870: For Without ItemIDs? = FALSE; only one ModeID for only one ItemID to query per command allowed.



#### 4.16.71. MOV!.vi

Moves specified axes to specified absolute positions and allows setting a new target position during the move without the move being interrupted. Use SCT.vi to set the cycle time for a periodic send. See user manual for details. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

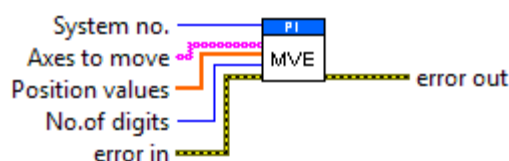
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0). Axes to move can be X,Y,Z,U,V,W. For compatibility, calls MOV.vi for GCS 2.0 or higher. In this case, parameter 0x19001900 must be set to 1 (SPA.vi), see controller user manual for details.



#### 4.16.72. MVE.vi

Vector path move to specified absolute position. Specified axes to move start moving simultaneously. The current settings for velocity, acceleration and deceleration define the maximum possible values. The actual velocity is defined by the slowest axis. No other motion commands are allowed during vector move. No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

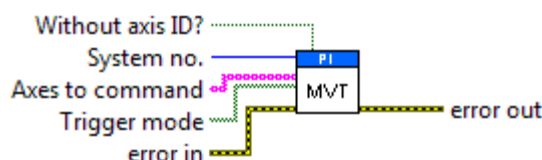
C-867: Check HLP? answer to determine if MVE is supported.



#### 4.16.73. MVT.vi

Sets MVT trigger mode for given axes. When enabled, the target position will be updated (target position = current position + increment value) after every trigger pulse. If Without axis ID is TRUE, then Axes to command is ignored and first field of Trigger mode array is used for trigger mode command.

E-816: Without axis ID = FALSE. Only one axis per command allowed. Check controller manual to find out if MVT is supported. The increment is defined by parameter No. 11.



#### 4.16.74. MVT?.vi

Returns MVT trigger mode of queried axes.

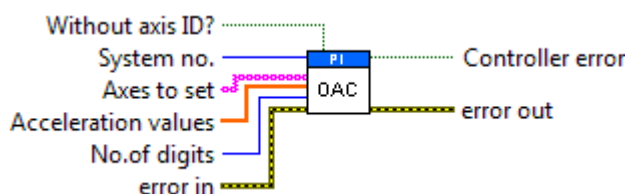
E-816: If All axes? = TRUE, then Axis identifier? can be FALSE. Check controller manual to find out if MVT? is supported.



#### 4.16.75. OAC.vi

Sets open-loop acceleration and checks for error. If Without axis ID? is TRUE, then Axes to set is ignored and first field of Acceleration values array is used for acceleration command. Number of digits is the number of digits after the decimal point in the acceleration value(s) that will be sent. Controller error is TRUE if selected system reports error code which is not 0.

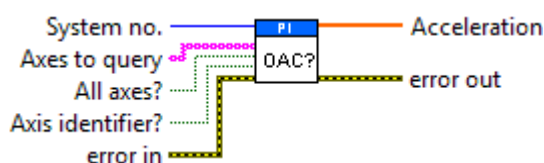
E-861, E-874: Without axis ID? = FALSE. Acceleration unit is steps/s<sup>2</sup>.



#### 4.16.76. OAC?.vi

Returns open-loop acceleration setting for specified axes.

E-861, E-874: If All axes? = TRUE, then Axis identifier? can be FALSE. Velocity unit is steps/s<sup>2</sup>.

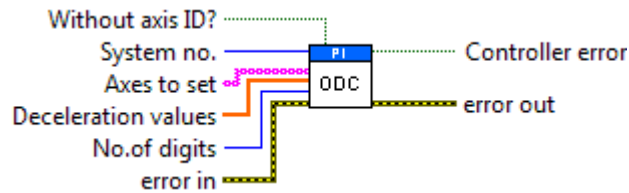


#### 4.16.77. ODC.vi

Sets open-loop deceleration and checks for error. If Without axis ID? is TRUE, then Axes to set is ignored and first field of Deceleration values array is used for deceleration command. Number of digits is the number of digits after the decimal point in the deceleration value(s) that will be sent. Controller error is TRUE if selected system reports error code which is not 0.

E-861, E-874: Without axis ID? = FALSE. Deceleration unit is steps/s<sup>2</sup>.



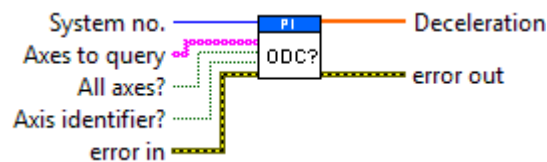


#### 4.16.78. ODC?.vi

Returns open-loop deceleration setting for specified axes.

E-861, E-874: If All axes? = TRUE, then Axis identifier? can be FALSE.

Deceleration unit is steps/s<sup>2</sup>.

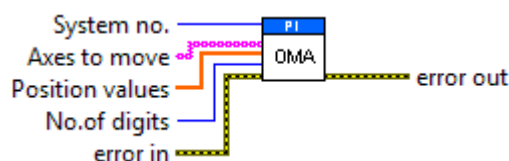


#### 4.16.79. OMA.vi

Open-loop move absolute. Sets new absolute open-loop target position for given axes. Servo must be disabled for all commanded axes prior to using this command.

E-861: This command works only in open-loop operation. With closed-loop systems, use MOV instead to command motion. Do not call OMA or OMR for axes which have no sensor, otherwise stage will run into hard stop. Motion commands are not allowed when a joystick is active on the axis. OMA can not be processed as long as motion commanded by a former OMA or OMR command is still performed.

E-871, E-873: This command works only in open-loop operation. With closed-loop systems, use MOV instead to command motion. Do not call OMA or OMR for axes which have no sensor, otherwise stage will run into hard stop. Motion commands are not allowed when a joystick is active on the axis.

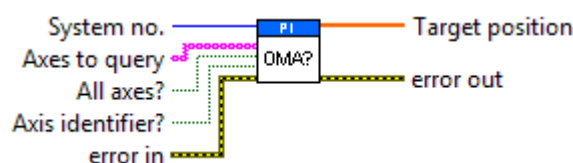


#### 4.16.80. OMA?.vi

Returns commanded open-loop target position.

E-861: If All axes? = TRUE, then Axis identifier? can be FALSE

E-871, E-873: If All axes? = TRUE, then Axis identifier? can be FALSE.

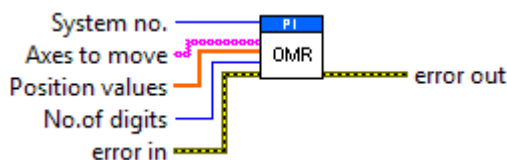


#### 4.16.81. OMR.vi

Open-loop move relative. Sets new relative open-loop target position for given axes. Servo must be disabled for all commanded axes prior to using this command.

E-861: This command works only in open-loop operation. With closed-loop systems, use MVR instead to command motion. Do not call OMA or OMR for axes which have no sensor, otherwise stage will run into hard stop. Motion commands are not allowed when a joystick is active on the axis. OMR can not be processed as long as motion commanded by a former OMR or OMA command is still performed.

E-871, E-873: This command works only in open-loop operation. With closed-loop systems, use MOV instead to command motion. Do not call OMA or OMR for axes which have no sensor, otherwise stage will run into hard stop. Motion commands are not allowed when a joystick is active on the axis.

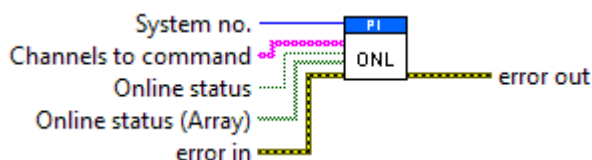


#### 4.16.82. ONL.vi

Switches system online (TRUE) or offline (FALSE). GCS 1.0: Online status is valid for all channels/axes of the system. GCS 2.0: Online status is channel specific.

E-516: Channels to command and Online status (Array) are not valid, use Online status instead to set online status for all channels/axes.

E-517, E-518: Online status is not valid. Use Channels to command and Online status (Array) instead.

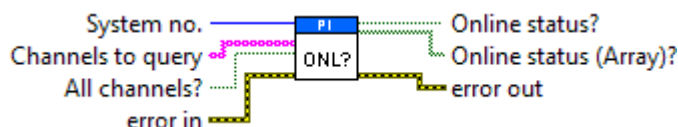


#### 4.16.83. ONL?.vi

Indicates whether system is online (TRUE) or offline (FALSE). GCS 1.0: Online status is valid for all channels/axes of the system. GCS 2.0: Online status is channel specific.

E-516: All channels? must be TRUE. Online status (Array)? is not valid, Online status? contains online status for all channels/axes.

E-517, E-518: All channels? must be FALSE. Online status is channel specific.



#### 4.16.84. OpenStageEditorDialog.vi

This VI calls the PIStageEditor which can be used to view/change stage parameters and/or add new stages. To view parameters of PI stages, select PIStages in Database. To add stages and/or change parameters, select UserStages in Database. GCSTranslator.dll and the corresponding system specific GCS DLL must be installed. The VI reads the system name from Global 2 and calls the corresponding DLL function.



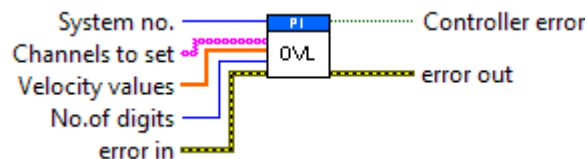
#### 4.16.85. OVL.vi

Sets open-loop velocity in steps/s and checks for error. Number of digits is the number of digits after the decimal point in the velocity value(s) that will be sent. Controller error is TRUE if selected system reports error code which is not 0.

E-861: The velocity for open-loop stepping motion is also influenced by the step amplitude set with SSA.

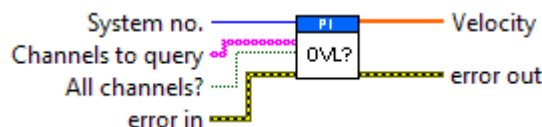
The maximum value which can be set with OVL is given by the Open-loop velocity parameter, ID 0x7000201 (can be changed with SPA and SEP).

E-870: The maximum value which can be set with OVL is given by the PIShift frequency parameter, ID 0x1F0004xx. Changing PIShift Velocity Parameter 0x1F000600 has the same effect like OVL.



#### 4.16.86. OVL?.vi

Returns open-loop velocity in steps/s for specified channels.



#### 4.16.87. POS.vi

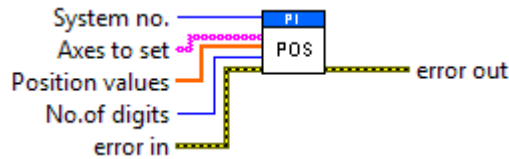
Assigns new position value to current position without moving the stage. Command can only be used when the reference mode is switched off (see RON.vi). No. of digits is the number of digits after the decimal point in the position value(s) that will be sent.

Warning: If the current position is incorrectly set on an axis with reference mode OFF, the stage can be driven into the mechanical hard stop when moving to a position which is thought to be within the travel range of the stage, but actually is not.

C-880K005: VI only supported when called through PI\_Multix.vi

E-755: Command not available for E-755.101

E-861: With open-loop systems, this command is not useful because there is no position sensor.

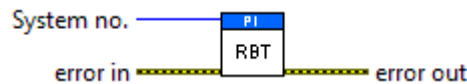


#### 4.16.88. RBT.vi

Reboots the controller. Controller behaves like after a cold start.

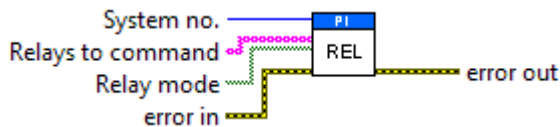
C-887, F-206, M-8X0: For GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

C-885: Reboots master only.



#### 4.16.89. REL.vi

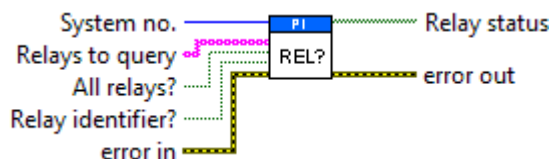
Switches relays on or off. Relay identifiers are A-H for relay board 1, I-P for relay board 2 etc.



#### 4.16.90. REL?.vi

Returns relay status for queried relays.

C-880: If All relays = TRUE, then Relay identifier can be FALSE



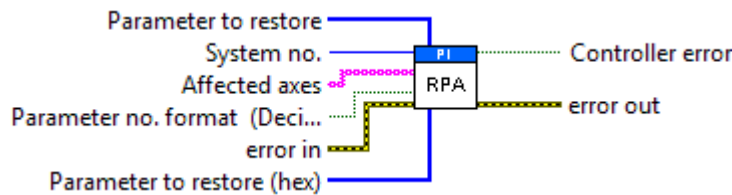
#### 4.16.91. RPA.vi

Replaces the current values of the given parameters to restore for Affected axes in the controller RAM with the values from non-volatile memory, waits 5000 ms and queries ERR?. For axis-related parameters, Affected axes is the axis name, for piezo- or sensor-related parameters, the channel number, otherwise a parameter-related code. If parameter no. is in decimal format, use Parameter to restore input, for hexadecimal parameter numbers use Parameter to restore (hex) input and switch Parameter no. format to TRUE. Do not mix decimal and hex. parameter numbers in one call. See GCS DLL manual for available parameter numbers. If Affected axes is an empty array, RPA is sent without axis (item) and parameter specification and controller restores all values for all axes (items). Controller error is TRUE if selected system reports error code not equal to 0.

E-710, E-761: If Affected axes = empty array, all parameters for all axes are restored. Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see the E7XX\_GCS\_DLL Manual.

C-413, E-517, E-518, E-712, E-725, E-727, E-753, E-754, E-755: If Affected axes = empty array, all parameters for all axes are restored. Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see user manual.

C-867, C-877, C-884, E-709, E-861, E-870, E-871, E-873, E-874, Mercury\_GCS: You can reset either all parameters or one single parameter with RPA. If Affected axes = empty array, all parameters for all axes are restored. Parameter no. format is TRUE (hex.). Use HPA?.vi to get valid parameter numbers or see user manual.



#### 4.16.92. RST.vi

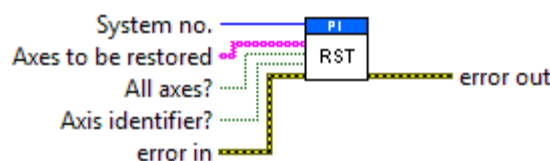
Restores parameters or resets E-816 master unit.

C-702: If All axes = TRUE, then Axis identifier must be TRUE. Restores last saved stage configuration in volatile memory (RAM) for any of the specified axes. Loads back the stage configuration (can be modified with CST) and the motion parameters (can be modified with SPA) which were last saved with SAV.

C-848: If All axes = TRUE, then Axis identifier must be TRUE. Restores parameters of any of the specified axes which have been modified using SPA command to values from last SAV.

C-880: If All axes = TRUE, then Axis identifier must be TRUE. Restores parameters of any of the specified axes which have been modified using SPA command to values from last SAV.

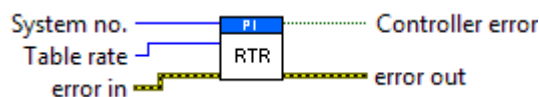
E-816: All axes = TRUE, Axis identifier = FALSE; this command affects the master unit only. (Resets the master unit.)



#### 4.16.93. RTR.vi

This VI sets the table rate and queries ERR?. The table rate is the number of servo-loop cycles to be used in data recording operations. Settings larger than 1 make it possible to cover longer time periods with a limited number of points. Controller error is TRUE if selected system report error code not equal to zero.

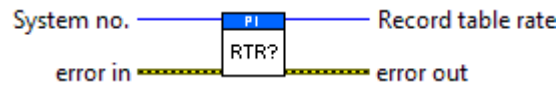
C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



**4.16.94. RTR?.vi**

Returns the current table rate.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

**4.16.95. SAI.vi**

Set axis identifier. With this command the axis identifiers of the connected axes can be changed.

GCS 1.0: Only one character is allowed as axis ID. Valid axis IDs can be listed with TVI?.vi.

GCS 2.0: Characters that Axis IDs can contain can be listed with TVI?.vi.

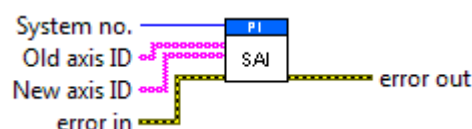
Please run Define connected axes.vi with new axis IDs after renaming axes.

E-517, E-518: Maximum length for axis name is 8 characters. SAI affects the Axis Name parameter, ID 0x07000600, in volatile memory (RAM). To save the currently valid value to non-volatile memory, where it becomes the power-on default, you must use WPA. Changes not saved with WPA will be lost when the controller is powered down or rebooted.

E-709: Maximum length for axis name is 4 characters. SAI affects the Axis Name parameter, ID 0x07000600, in volatile memory (RAM). To save the currently valid value to non-volatile memory, where it becomes the power-on default, you must use WPA. Changes not saved with WPA will be lost when the controller is powered down or rebooted.

C-867, C-877, C-884, C-891, E-761, E-871, E-873, E-874, Mercury\_GCS: The settings are automatically written to non-volatile memory.

Hydra, Pollux: To save the currently valid axis identifier to non-volatile memory, where it becomes the power-on default, you must use WPA. Changes not saved with WPA will be lost when the controller is powered down or rebooted.

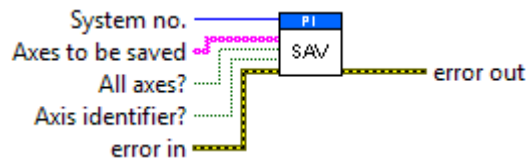
**4.16.96. SAV.vi**

Saves parameters of any of the specified axes which have been modified using SPA command.

C-702: If All axes = TRUE, then Axis identifier must be TRUE. Save configuration of given axis. Saves the current stage configuration (stage-to-axis assignment with CST) and the corresponding motion parameter values (can be modified with SPA) from RAM to non-volatile memory and thus makes them the new power-on default.

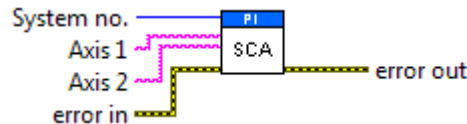
C-848: If All axes = TRUE, then Axis identifier must be TRUE

C-880: If All axes = TRUE, then Axis identifier must be TRUE



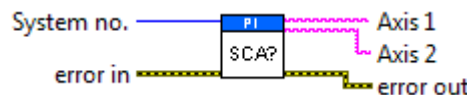
#### 4.16.97. SCA.vi

Enables control of the two specified axes with controller keyboard cursor keys. Axis 1 can be controlled with left-right keys, axis 2 with up-down keys. The step width for each keystroke is set by SST.vi.



#### 4.16.98. SCA?.vi

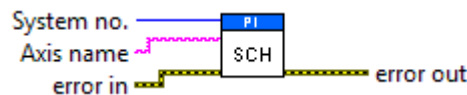
Indicates which axes can be controlled with controller keyboard cursor keys: left-right for axis 1 and up-down for axis 2.



#### 4.16.99. SCH.vi

Set axis (channel) name. See system manual for further description. Run Define connected axes.vi after renaming axes.

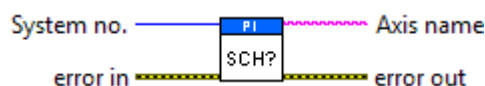
E-816: This command cannot be issued to a slave E-816. Changes must be written to non-volatile memory with WPA.vi. and do not take effect before the next power on or RST.



#### 4.16.100. SCH?.vi

Returns the axis (channel) name of the master unit.

E-816: This command cannot be issued to a slave.



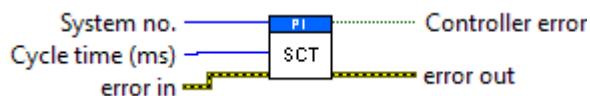
#### 4.16.101. SCT.vi

Sets cycle time for MOV!/MOV command in s. Controller error is TRUE if selected system reports an error code which is not 0.

C-887, F-206, M-8X0: For GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), sets cycle time for MOV! command.



For GCS 2.0 or higher, sets cycle time for MOV command. Parameter 0x19001900 must be set to 1 in this case (SPA.vi), see controller user manual for details.

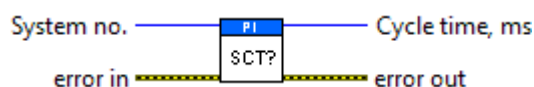


#### 4.16.102. SCT?.vi

Returns current cycle time for MOV! command.

C-887, F-206, M-8X0: For GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), returns cycle time for MOV! command.

For GCS 2.0 or higher, returns cycle time for MOV command. Parameter 0x19001900 must be set to 1 in this case (SPA.vi), see controller user manual for details.



#### 4.16.103. SEP.vi

If Password is correct, this VI sets parameters for commanded axes to EPROM, waits 100 ms and queries ERR?. For axis-related parameters, Axis to set is the axis name; for piezo- or sensor-related parameters, the channel number; otherwise a parameter-related code. If parameter number is in decimal format, use Parameter number input, for hexadecimal parameter numbers use Parameter number (hex.) input and switch Parameter no. format to TRUE. For numeric parameter values use Parameter value input, for parameter strings use Parameter string input and switch Parameter format to TRUE. Do not mix decimal and hex. parameter numbers or numeric and string parameter values in one call. Parameter numbers which can be set depend on current CCL level. Use HPA?.vi to get valid parameter numbers or see GCS DLL manual for available parameter numbers and values. No. of digits is the number of digits after the decimal point in the numeric parameter value(s) that will be sent. Hidden error is TRUE if selected system reports error code which is not 0.

Note: This command sets the same parameters as SPA, but SPA writes them only to volatile memory, while SEP only writes to non-volatile memory. After parameters were set with SEP, use RPA to activate them (write them to volatile memory), or they become active after next power up.

C-413: Parameter no. format is TRUE (hex.). Required command level depends on parameter (CCL.vi) . Do not set more than 4 parameter no. at once.

C-867, C-877, C-891, E-135, E-871, E-872, E-873, E-874, Mercury\_GCS: Parameter no. format is TRUE (hex.). You can write only one single parameter per SEP command. See User manual for more information.

C-884: Parameter no. format is TRUE (hex.). You can write up to four single parameters per SEP command. See User manual for more information.

E-517, E-518: Parameter no. format is TRUE (hex.). Requires command level 1 (CCL.vi) for parameter 0X02000000 (used to enable/disable axes which is only required if the hardware configuration is changed), 0x04000E00 and 0x04000E01 (used to configure unit and format of the LCD display on the E-517 front panel).

E-709: Parameter no. format is TRUE (hex.). You can write only one single parameter per SEP command. Required command level depends on parameter (CCL.vi)

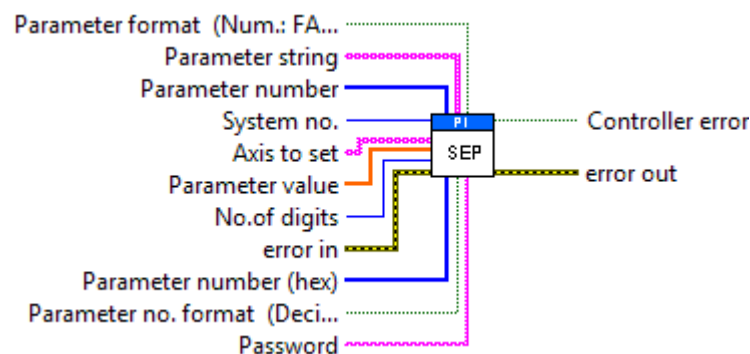
E-710: Parameter no. format is TRUE (hex.). Command is available in command level 1 only (see CCL.vi, CCL?.vi). Command writes parameters to Eprom and RAM

E-755, E-761: Parameter no. format is TRUE (hex.)

E-712, E-725, E-727, E-753, E-754: Parameter no. format is TRUE (hex.). Requires command level 1 (CCL.vi). Do not set more than 10 parameters at once.

E-861: Parameter no. format is TRUE (hex.). You can write only one single parameter per SEP command. The GEMAC parameters (ID 0x7000010 to ID 0x700001F) can not be changed with SEP. Use SPA and WPA instead to save their values to non-volatile memory. See "GEMAC Parameter Adjustment" in the E-861 User manual for more information.

E-870: Parameter no. format is TRUE (hex.). You can write only one single parameter per SEP command.



#### 4.16.104. SEP?.vi

Returns parameter values from non-volatile memory for queried axes and parameter numbers. For axis-related parameters, Axes to query is the axis name; for piezo- or sensor-related parameters, the channel number; otherwise a parameter-related code. If parameter number is in decimal format, use Parameter no. input, for hexadecimal parameter numbers use Parameter no. (hex) input and switch Parameter no. format to TRUE. Do not mix decimal and hex. parameter numbers in one call. If Without axes? is TRUE, all available parameter for all axes/designators are returned. For parameter numbers which output a string use Parameter string output. See GCS DLL Manual for available parameter numbers.

All systems: Use HPA?.vi to get valid parameter numbers. Parameter no. format is TRUE (hex).

C-413: Do not query more than 6 parameter no. at once (except with Without axes? = TRUE).

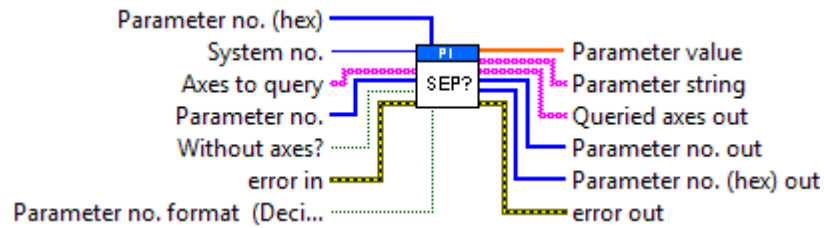
C-867, C-877, C-891, E-135, E-709, E-871, E-872, E-873, E-874, Mercury\_GCS: Only one parameter value for only one axis per command allowed. Use Without axes? = TRUE for a query of all parameters.

C-884: Use Without axes? = TRUE for a query of all parameters.

E-712, E-725, E-727, E-753, E-754: Do not query more than 10 parameter no. at once (except with Without axes? = TRUE).

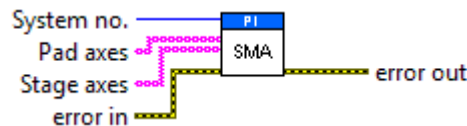
E-861: Only one parameter value for only one axis per command allowed. Use Without axes? = TRUE for a query of all parameters.

E-870: Only one parameter value for only one axis per command allowed. Use Without axes? = TRUE for a query of all parameters.



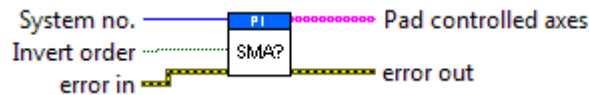
#### 4.16.105. SMA.vi

Enables control of the specified axes with the (optional) manual control pad. Pad axes are axis IDs of the manual control pad (for 1 pad A-F, for 2 pads A-L). Stage axes are the corresponding axis IDs of connected stages to be controlled.



#### 4.16.106. SMA?.vi

Gets axes controlled by one or more (optional) manual control pads.  
C-880: Invert order should be TRUE



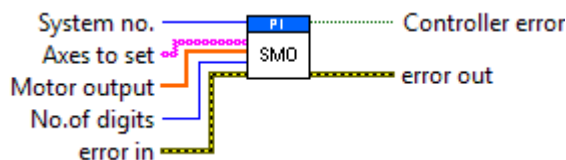
#### 4.16.107. SMO.vi

Sets the motor output directly and checks for error. Command will only be executed if channel is in servo-off mode (SVO.vi).

Caution: In servo-off mode limit switches are not enabled!

No. of digits is the number of digits after the decimal point in the motor output value(s) that will be sent. Controller error is TRUE if selected system reports an error code which is not 0.

All systems: No. of digits is 0.

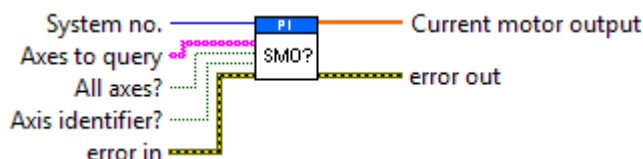


#### 4.16.108. SMO?.vi

Returns the current motor output. In servo-on mode, the actual value as set by the regulator is reported. In servo-off mode, the value set by the last SMO command is reported.

C-702, C-848, C-865, C-866, C-867, C-877, C-880, C-884, C-891, Hydra, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE

C-843, C-843.PM, C-844: If All axes? = TRUE, then Axis identifier? must be TRUE  
E-873: If All axes? = TRUE, then Axis identifier? can be FALSE.

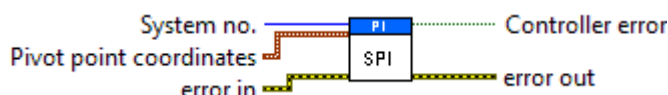


#### 4.16.109. SPI.vi

C-886, C-887, F-206- and M-8X0-specific. Sets the pivot point. Command will only be executed if all angular coordinates of current position are zero. Controller error is TRUE if selected system reports an error code which is not 0.

Can only be used if a coordinate system of types KSF or ZERO is active.

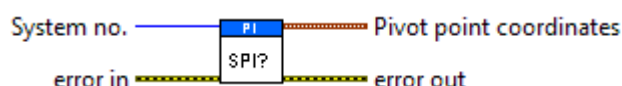
C-886, C-887, F-206, M-8X0: Pivot point coordinates (R,S,T) and (X,Y,Z) are valid and identical.



#### 4.16.110. SPI?.vi

C-886, C-887, F-206- and M-8X0-specific. Returns the pivot point coordinates.

C-887, F-206, M-8X0: Pivot point coordinates (R,S,T) and (X,Y,Z) are valid and identical.



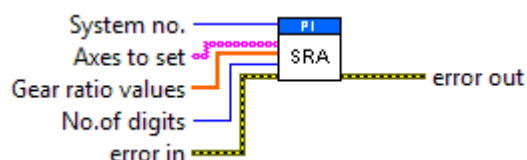
#### 4.16.111. SRA.vi

Sets gear ratio for electronic gearing mode. No. of digits is the number of digits after the decimal point in the gear ratio value(s) that will be sent.

C-843 This command can only be issued to a slave axis. Use MAS to define the master axis.

C-848: This command can only be issued to a slave axis. The master-slave assignments are hardware-dependent.

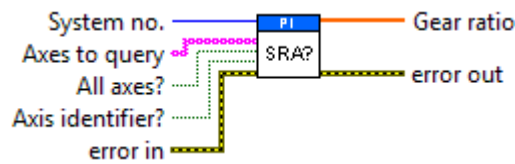
C-880: This command can only be issued to a slave axis. The master-slave assignments are hardware-dependent.



#### 4.16.112. SRA?.vi

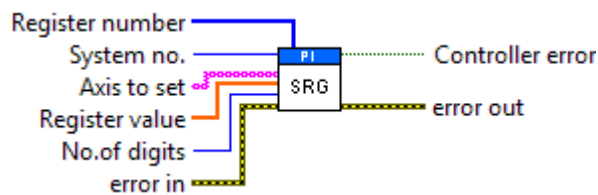
Returns gear ratio values for specified axes.

C-843, C-848, C-880: If All axes = TRUE, then Axis identifier can be FALSE. This command can only be issued to a slave axis.



#### 4.16.113. SRA.vi

Sets register values for commanded axes, waits 100 ms and queries ERR?. See system manual for available parameter numbers and values. No. of digits is the number of digits after the decimal point in the register value(s) that will be sent. Controller error is TRUE if selected system reports an error code which is not 0.



#### 4.16.114. SRG?.vi

Returns register values for queried axes and register numbers. See motion processor or controller user manual for a description of bit-coded answer. If Query all? Is TRUE, Axes to query and Register no. are ignored and command is sent without any parameters to query all register numbers for all axes.

C-843, C-843.PM: Register numbers are:

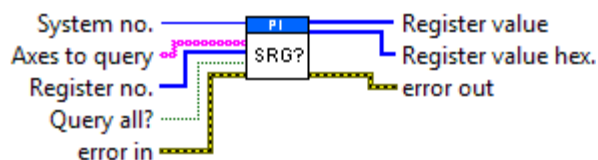
- 1: Event status register
- 2: Activity status register
- 3: Signal status register

C-866: Register numbers are:

- 1: Event status register
- 2: Activity status register
- 3: Signal status register
- 4: Signal sense

C-413, C-867, C-877, C-884, C-885, C-891, E-135, E-712, E-754, E-861, E-871, E-873, E-874, Hydra, Pollux, Mercury, Mercury\_GCS: Register number is:

- 1: Status register



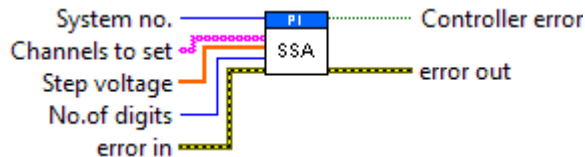
#### 4.16.115. SSA.vi

Sets driving voltage amplitude for all kind of step motion of specified PiezoWalk channels. No. of digits is the number of digits after the decimal point in the Step voltage value(s) that will be sent. Hidden error is TRUE if selected system reports error code unequal to 0.

E-755: After changing the step voltage, APG.vi must be run again.

E-861: SSA changes the value of the Bending Voltage parameter (ID 0x07000003) in volatile memory (can be saved as power-on default with WPA, can also be changed with SPA and SEP).

The SSA setting takes effect for stepping motion in open-loop operation (servo off). Decreasing the step size will decrease the velocity for open-loop stepping motion. The velocity for open-loop stepping motion is also influenced by the open-loop velocity set with OVL.



#### 4.16.116. SSA?.vi

Returns driving voltage amplitude for step mode (step size) of given PiezoWalk channel.

E-755, E-861: If All channels? = TRUE, then Channel identifier? must be FALSE.



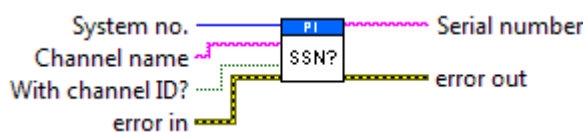
#### 4.16.117. SSN?.vi

Returns controller serial number.

E-816: With channel ID = TRUE. This command cannot be issued to a slave.

All other systems: With channel ID = FALSE

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.16.118. SST.vi

Sets step size for manual control. No. of digits is the number of digits after the decimal point in the step size values that will be sent.

C-702: Sets step size for cursor control (see also SCA.vi).

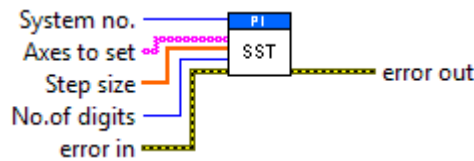
C-848: Sets step size for cursor control (see also SCA.vi) and position control with a joystick connected to the C-848 (see also JEN.vi).

C-867: Step size is applied to a motion caused by an Human Interface (HI) device. HI control must be configured appropriately with HIA (Function = 2 (relative position), DeviceAxis = 3 or 4).

C-880: Sets step size for Manual Control Pad, cursor control (see also SCA.vi) and position control with a joystick connected to the C-880 (see also JEN.vi).

E-871, E-873: Step size is applied to a motion caused by an Human Interface (HI) device. HI control must be configured appropriately with HIA (Function = 2 (relative position), DeviceAxis = 3 or 4)

C-887, F-206, M-8X0: Sets step size for Manual Control Pad. Only axes X,Y,Z,U,V,W allowed.

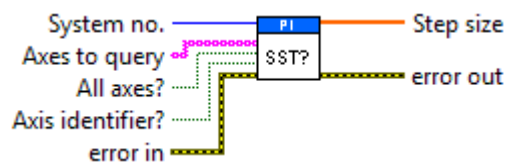


#### 4.16.119. SST?.vi

Returns step-size setting (e.g. for manual control pad, see also SST.vi).

C-702, C-867, C-880, E-871, E-873: If All axes = TRUE, then Axis identifier can be FALSE

C-887, F-206, M-8X0: All axes = TRUE, Axis identifier = FALSE. Only values for axes X,Y,Z,U,V,W are valid



#### 4.16.120. STA?.vi

Returns axis status (integer). Required by General wait for movement to stop.vi and Wait for axes to stop.vi.

C702: If All axes = TRUE, then Axis identifier can be FALSE

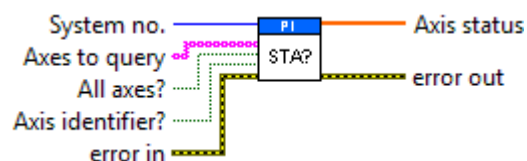
C-848: If All axes = TRUE, then Axis identifier can be FALSE

C-880: If All axes = TRUE, then Axis identifier can be FALSE

C-880K005: VI only supported when called through PI\_Multix.vi

C-886: Command is equivalent to #4. For details see user manual of the controller.

C-887, F-206, M-8X0: All axes? = TRUE, Axis identifier? = FALSE. Command is equivalent to #4. For details see user manual of the controller.



#### 4.16.121. STE.vi

Performs a single step-move from the current position with specified step size (amplitude). If supported, Delay defines the number of servo loops between position recording (GCS 2.0: Delay must be 0). Controller saves a definite number of position values, which can be read out with STE?.vi (GCS 1.0) or DRR? (GCS 2.0). No. of digits is the number of digits after the decimal point in the step size values that will be sent. For an impulse-move, see IMP.vi.

Analog: Delay = 0. Use DRR?.vi or DRR? and display data.vi to read position values back.

C-843: Controller saves up to 32,640 position values for all 4 channels in sum. Delay = 0. Use STE?.vi to read position values back.



C-843.PM: Controller saves up to 32,640 position values for all 4 channels in sum. Delay = 0. Use STE?.vi to read position values back.

C-848: Controller saves 1024 position values. Use STE?.vi to read position values back.

C-865: Controller saves up to 32,640 position values. Delay = 0. Use STE?.vi to read position values back.

C-866: Controller saves up to 32,256 position values. STE will overwrite DRC settings of Rec. table 1 to record actual position values. Use DRC to define additional record options for Rec. table no. 2 to 4. Record table rate is reset to 1 by STE. Use STE?.vi to read position values back or DRR? to read all Rec. tables back. You can also use MVR in combination with DRC to record values of a step motion. Use DRR? to read values back then.

C-867: Controller saves up to 8192 position values.

Motion commands like STE are not allowed when the joystick is active for the axis. Use DRC to define record options. Use DRR?.vi or DRR? and display data.vi to read recorded values back. You can also use MVR in combination with DRC to record values of a step motion.

C-877: See user manual for available record options, number of data recorder tables and points per table. Use DRC to define record options. Use DRR?.vi or DRR? and display data.vi to read recorded values back. You can also use MVR in combination with DRC to record values of a step motion.

C-880: Controller saves 1024 position values. Use STE?.vi to read position values back.

C-884: Controller saves 8192 position values. Motion commands like STE are not allowed when control via a Human Interface Device (HID) is active for the axis of the controller. Use DRC to define record options. Use DRR?.vi or DRR? and display data.vi to read recorded values back. You can also use MVR in combination with DRC to record values of a step motion.

C-891: See Controller user manual for details. Use DRC to define record options. Use DRR?.vi or DRR? and display data.vi to read recorded values back. You can also use MVR in combination with DRC to record values of a step motion.

E-517, E-518: Controller saves up to 8,192 position values. Use DRR?.vi or DRR? and display data.vi to read recorded values back. The number of servocycles used for data recording depends on the setting made with RTR.

Motion commands like STE are not allowed when the controller is in OFFLINEmode or when the wave generator output is active. When a macro is running on the controller, STE will be executed not until the macro is finished or stopped. See "Control Value Generation" and "Control Modes" in the controller user manual for details.

C-413, E-709: Controller saves up to 4096 position values. Use DRR?.vi or DRR? and display data.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

E-710: Controller saves 32256 position values. Delay = 0. "Table Rate" parameter, set with SPA.vi, is used as sampling interval instead of Delay. Caution: "Table Rate" parameter influences wave generator also, not only STE. Use STE?.vi to read position values back.

E-712: Controller saves up to 262,144 position values. Use DRR?.vi or DRR? and display data.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands are not

allowed when a wave generator is active or the analog input is used for target generation.

E-725, E-727: Controller saves up to 262,144 position values. Use DRR?.vi or DRR? and display data.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands are not allowed when a wave generator is active or the analog input is used for target generation.

E-753: Controller saves up to 65,536 position values. Use DRR?.vi or DRR? and display data.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

E-754: Controller saves up to 1,048,576 position values. Use DRR?.vi or DRR? and display data.vi to read recorded values back. The number of servo cycles used for data recording depends on the setting made with RTR. Motion commands are not allowed when the wave generator is active or the analog input is used for target generation.

E-755: Delay = 0. Controller saves 4096 position values. Use DRR?.vi or DRR? and display data.vi to read recorded values back.

E-761: Controller saves 8192 position values. Delay = 0. The number of servo cycles used for data recording depends on the setting made with RTR.vi. Use DRR?.vi or STE?.vi to read position values back.

E-861: Step response measurements provide meaningful results only in closed-loop operation. Controller saves up to 1,024 position values.

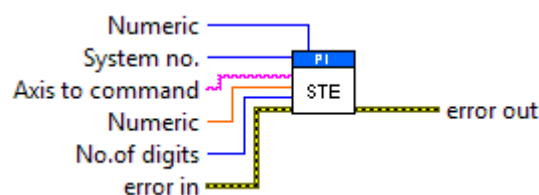
Motion commands like STE are not allowed when the joystick is active for the axis. Use DRC to define record options. Use DRR?.vi or DRR? and display data.vi to read recorded values back. You can also use MVR in combination with DRC to record values of a step motion.

E-871, E-873, E-874: Controller saves up to 1024 position values.

Motion commands like STE are not allowed when the joystick is active for the axis. Use DRC to define record options. Use DRR?.vi or DRR? and display data.vi to read recorded values back. You can also use MVR in combination with DRC to record values of a step motion.

C-887, F-206, M-8X0: Controller saves 1,024 position values (number can be changed by setting the appropriate parameter value, see user manual). Use DRR?.vi or DRR? and display data.vi to read recorded values back. Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

Mercury\_GCS: Controller saves up to 1,024 position values. Motion commands like STE are not allowed when the joystick is active for the axis. Use DRC to define record options. Use DRR?.vi or DRR? and display data.vi to read recorded values back. You can also use MVR in combination with DRC to record values of a step motion.



#### 4.16.122. STE?.vi

GCS 1.0 controller: Returns N saved step response points. N must be less than or equal to Nmax. For large N values, communication timeout must be set long enough, otherwise a comm.error may occur.

GCS 2.0 or higher: Returns STE settings. Use DRR? to read step response points back.

C-843: VI reads step response points. Step size is not valid. Nmax = 32640.

C-843.PM: VI reads step response points. Step size is not valid. Nmax = 32640.

C-848: VI reads step response points. Step size is not valid. Nmax = 1024.

C-866: VI reads step response points. Step size is not valid. Nmax = 32256.

C-880: VI reads step response points. Step size is not valid. Nmax = 1024.

E-517, E-518: VI reads STE settings. Step response is not valid. xo, N and Nmax are not valid.

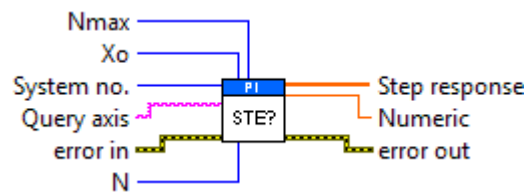
E-709: VI reads STE settings. Step response is not valid. xo, N and Nmax are not valid.

E-710: VI reads step response points. Step size is not valid. Nmax = 8192.

E-754: VI reads STE settings. Step response is not valid. xo, N and Nmax are not valid.

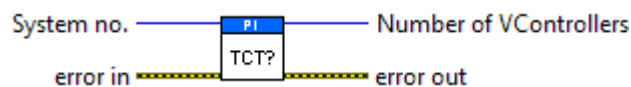
E-755: VI reads STE settings. Step response, xo, N and Nmax are not valid.

E-761: VI reads step response points. Step size is not valid. Nmax = 8192.



#### 4.16.123. TCT?.vi

Returns the number of virtual controllers connected to the communications controller identified by System no.

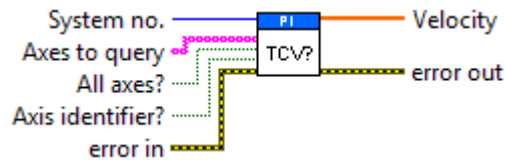


#### 4.16.124. TCV?.vi

Returns current commanded velocity for specified axes. In contrast to VEL?, during acceleration and deceleration TCV? reports the value currently produced by the trajectory generator. The sign of the returned value depends on the commanded direction of motion. If the velocity of the axis is controlled by an axis of a Human Interface Device (i.e. HI control is activated, see HIA and HIN), TCV? returns the current value according to the deflection of the HID axis, and VEL? returns the maximum velocity set by the corresponding parameter or VEL.

C-867, Mercury\_GCS: If All axes? = TRUE, then Axis identifier? can be FALSE. Velocity unit is mm/s.

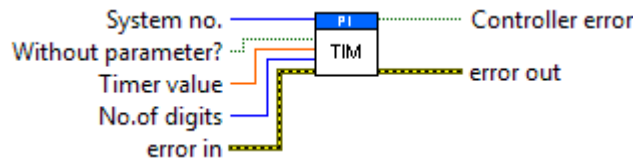
C-877, C-884, C-891: If All axes? = TRUE, then Axis identifier? can be FALSE. Velocity unit is mm/s.



#### 4.16.125. TIM.vi

Sets the milliseconds timer to a given value. Number of digits is the number of digits after the decimal point in the Timer value that will be sent. Without parameter the timer is reset to zero. Controller error is TRUE if selected system reports error code is not 0.

C-867, C-884:



#### 4.16.126. TIM?.vi

GCS 1.0: Returns controller system time (integer). GCS 2.0 or higher: Reports the number of milliseconds (float) since start-up or last TIM command (if supported).

C-702: Reports system time as number of servo-loop counts (independent of servo-control mode) since firmware program start; duration of one loop is 100  $\mu$ s.

C-848: Reports system time as number of servo-loop counts (independent of servo-control mode) since firmware program start; duration of one loop is 400  $\mu$ s.

C-867, C-884: Reports the number of milliseconds (float) since start-up or last TIM command. System time is not valid.

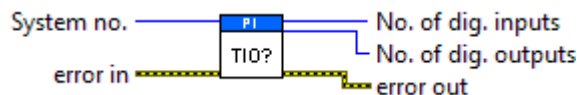
C-880: Reports system time as number of servo-loop counts (independent of servo-control mode) since firmware program start; duration of one loop is 400  $\mu$ s.



#### 4.16.127. TIO?.vi

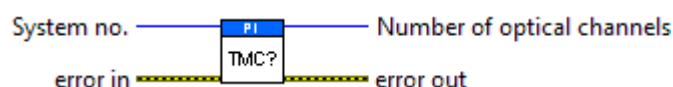
Returns the number of digital inputs and outputs available in the controller.

E-761: The E-761 has no genuine digital input and output lines, but the analog input is internally interpreted as digital input for triggering tasks (see E-761 user manual), and its signal state can be queried by DIO?.vi.



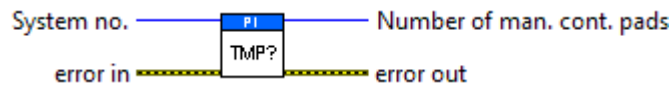
#### 4.16.128. TMC?.vi

Returns the number of motor controller boards installed in the controller.



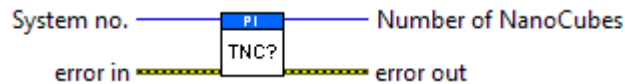
#### 4.16.129. TMP?.vi

Returns the number of manual control pads installed in the controller.



#### 4.16.130. TNC?.vi

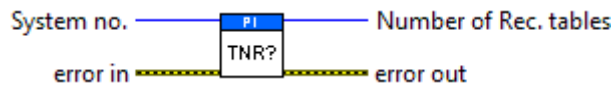
Returns the number of NanoCube control boards installed in the controller.



#### 4.16.131. TNR?.vi

Returns the number of recording tables.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.16.132. TPC?.vi

Returns the number of available piezo channels.

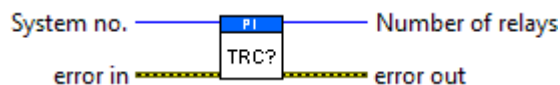
E-517, E-518: Using the Sensor Enable parameter, ID 0x02000000, you can change the controller configuration in case of hardware changes, e.g. if you install additional sensor and/or amplifier channels in the system. If this parameter is changed, the Number Of Piezo Channels parameter is adapted automatically. E.g. if parameter 0x02000000 is set to "disabled" for a sensor channel, the corresponding piezo channel is disabled too and no longer included in the TPC? response. See "Configure Axes and Channels" in the controller user manual for details.

E-709, E-712, E-725, E-727: Returns all Output Signal Channels (piezo channels + analog output channels).



#### 4.16.133. TRC?.vi

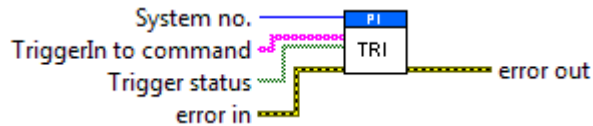
Returns the number of relays available in the controller (one relay board supports 8 relays).



#### 4.16.134. TRI.vi

Enables or disables the Trigger Input mode which was set with CTI. TRI has no effect on DIO?, i.e. DIO? will always answer the current logical state of the input line level even if TRI is enabled.

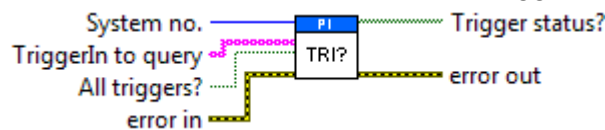
C-709: Check controller user manual for available input trigger numbers.



#### 4.16.135. TRI?.vi

Returns the trigger input mode enable status for the given trigger input lines.

E-709: Check controller user manual for available input trigger numbers.



#### 4.16.136. TRO.vi

Enables or disables the trigger output mode which was set with CTO. Do not use DIO on output lines for which the trigger output is activated with TRO.

C-702: TriggerOut to command can be 1 to 8.

C-843: TriggerOut to command can be 1 to 4 (4-axis card) or 2 (2-axis card).

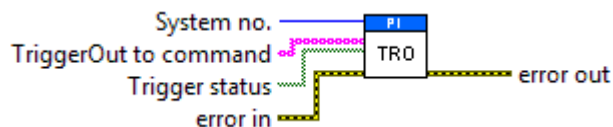
C-866: TriggerOut to command can be 1.

C-867, C-877: Check controller user manual for available output trigger numbers.

C-413, C-884, E-871, E-873, E-874 : Check controller user manual for available output trigger numbers.

Mercury: TriggerOut to command can be 1 to N (N: number of connected axes, if supported by corresponding Mercury firmware).

Mercury\_GCS: Check Mercury user manual for available output trigger numbers.



#### 4.16.137. TRO?.vi

Returns the trigger output mode enable status for the given trigger output lines.

C-702: TriggerOut to query can be 1 to 8.

C-843: TriggerOut to query can be 1 to 4 (4-axis card) or 2 (2-axis card).

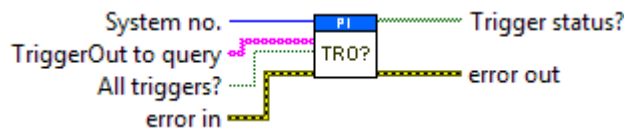
C-866: TriggerOut to query can be 1.

C-867, C-877: Check C-867 user manual for available output trigger numbers.

C-413, C-884, E-871, E-873, E-874 : Check controller user manual for available output trigger numbers.

Mercury: TriggerOut to query can be 1 to N (N: number of connected axes, if supported by corresponding Mercury firmware).

Mercury\_GCS: Check Mercury user manual for available output trigger numbers.



#### 4.16.138. TVI?.vi

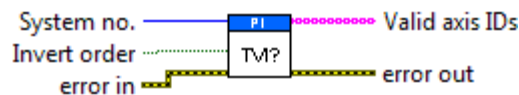
GCS 1.0: Get valid axis identifiers. Should be called before axes are renamed with SAI.vi.

GCS 2.0: Get valid characters for axis IDs.

C-702, C-848, C-880: Invert order should be TRUE. Returns valid axis identifiers.

C-843, C-843.PM, C-844, C-865, C-866, E-710, E-761, Mercury: Invert order must be FALSE. Returns valid axis identifiers.

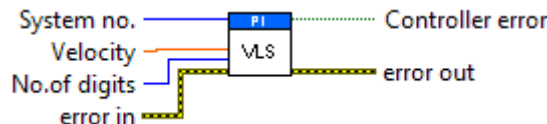
C-867, C-877, C-884, C-891, E-135, E-517, E-518, E-871, E-872, E-873, E-874, Hydra, Pollux, Mercury\_GCS: Invert order must be FALSE. Returns valid characters for axis IDs.



#### 4.16.139. VLS.vi

Sets platform velocity and checks for error. Number of digits is the number of digits after the decimal point in the velocity value(s) that will be sent. Controller error is TRUE if selected system reports error code which is not 0.

C-887, F-206, M-8X0: Only valid for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.16.140. VLS?.vi

Returns platform velocity setting.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).



#### 4.16.141. VST?.vi

Returns the names of all stages which can be connected to the controller.

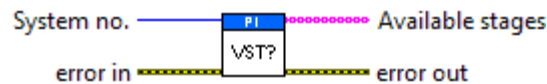
C-877, C-884, C-891, E-872, E-873, E-874: Command is available via USB interface only.

C-867, E-871: Command is available via USB, USB DaisyChain or RS232 DaisyChain interface only.

C-887, F-206, M-8X0: Only for GCS syntax version = GCS 2.0 or higher (check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

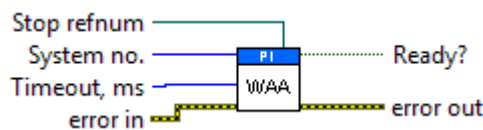


Mercury\_GCS: Command is available via USB interface only.



#### 4.16.142. WAA.vi

Wait for all axes with timeout. This VI returns as soon as the movement of all connected axes has finished, Local stop or Stop refnum is TRUE or after the specified timeout has elapsed. It returns Ready? = TRUE if motion was completed, Ready? = FALSE if timeout elapsed or VI was stopped. When using as a sub-VI, use Stop refnum to stop VI from caller.



#### 4.16.143. WPA.vi

If password is correct, this vi writes current settings of the given parameter numbers for Affected axes to non-volatile memory of the controller, waits 3000 ms (E-725, E-727: waits for controller ready by polling with #7) and queries ERR?. For axis-related parameters, Affected axes is the axis name, for piezo- or sensor-related parameters, the channel number, otherwise a parameter-related code. If parameter no. is in decimal format, use Parameter to save input, for hexadecimal parameter numbers use Parameter to save (hex) input and switch Parameter no. format to TRUE. Do not mix decimal and hex. parameter numbers in one call. See GCS DLL manual for available parameter numbers. If Affected axes is an empty array, WPA is sent without axis (item) and parameter specification.

##### WARNING:

If current parameter values are incorrect, the system may malfunction. Be sure that you have the correct parameter settings before using the WPA command.

C-413: Affected axes = empty array, the currently valid values of all parameters affected by the specified password are saved (see below). Parameter no. format is TRUE (hex).

The WPA command saves the currently valid parameter values to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the C-413 is powered off or rebooted. Parameters can be changed in volatile memory with SPA, AOS, ATZ, CMO, RTR, VEL, WOS and WTR. Depending on the parameter to be saved, it may be necessary to switch to command level 1 (CCL.vi).

C-867, C-877, C-891: Affected axes = empty array, the currently valid values of all parameters affected by the specified password are saved (see below). Parameter no. format is TRUE (hex). The WPA command saves the currently valid parameter values to non-volatile memory, where they become the power-on defaults.

Settings not saved with WPA will be lost when the controller is powered off or rebooted. Parameters can be changed in volatile memory with SPA, ACC, DEC and VEL. WPA must be used without specifying any arguments except of the password, the currently valid values of all parameters affected by the specified password are saved.

C-884: Affected axes = empty array, the currently valid values of all parameters affected by the specified password are saved (see below). Parameter no. format is

TRUE (hex). Depending on the password, the WPA command saves the currently valid parameter values and/or the current HI device configuration to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the controller is powered off or rebooted.

Valid passwords and affected settings:

100: all parameters, settings of HDT, HIA, HIT

101: all parameters

HID: settings of HDT, HIA, HIT

Parameters can be changed in volatile memory with SPA, ACC, DEC and VEL.

WPA must be used without specifying any arguments except of the password.

C-886: See the user manual of the controller for valid passwords and affected settings.

C-887, F-206, M-8X0: Check HLP?/HELP answer to find out if WPA is supported. Only for GCS syntax version = GCS 2.0 or higher (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0).

Affected axes and Parameter to save is only valid for Password = 100 or 101 (If Affected axes = empty array, all parameters for all axes are saved). Parameter no. format is TRUE (hex).

Depending on the password, the WPA command saves the currently valid parameter values and/or the current coordinate systems settings or stage assignments to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the controller is powered off or rebooted.

Valid passwords and affected settings:

SKS: coordinate system settings A12: stage assignments for separate axes A and B 101: all parameters 100: all parameters and settings of SKS and A12

Parameters can be changed in volatile memory with SPA.

E-516: Affected axes and Parameter to save = empty array. E-516: The WPA command saves the currently valid parameters listed below to flash ROM, where they become the power-on defaults. Parameter changes not saved with WPA will be lost when the E-516 is powered off. Communication interface, enabled channels and display format, averaging (AVG), drift compensation mode (DCO), velocity control mode (VCO) and velocity (VEL), offset and gain for position and output voltage display, mode and tolerance for on-target reading (SPA), position limits (NLM, PLM), voltage limits (VMA, VMI), macros and default macro setting.

E-517, E-518: If Affected axes = empty array, all parameters for all axes are saved. Parameter no. format is TRUE (hex). The WPA command saves the currently valid parameter values to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the controller is powered off or rebooted. Parameters can be changed in volatile memory with SPA, IFC, SAI, DFH, VMI, VMA, WAV, WGC, WOS, WTR, RTR, and VEL. Furthermore, WPA saves the current settings of NLM, PLM (position soft limits), VCO (velocity control mode), DRC (data recorder configuration), CTO (trigger output configuration) and CSV (E-517 only. GCS syntax version, i.e. E-517 or E-516 mode; requires command level 1 (CCL.vi).

E-709: If Affected axes = empty array, all parameters for all axes are saved. Parameter no. format is TRUE (hex). The WPA command saves the currently valid parameter values to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the E-709 is powered off or rebooted. Parameters can be changed in volatile memory with SPA, AOS, ATZ, IFC, RTR, VEL, WOS and WTR. Depending on the parameter to be saved, it may be necessary to switch to command level 1 (CCL.vi).

E-710: If Affected axes = empty array, all parameters for all axes are saved. See E-710 GCS DLL Manual for a list of available parameters. Command is available in command level 1 only (see CCL.vi and CCL?.vi). Parameter no. format is TRUE (hex.).

E-712: If Affected axes = empty array, all parameters for all axes are saved. Parameter no. format is TRUE (hex). The WPA command saves the currently valid parameter values to non- volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the E-712 is powered off or rebooted. Parameters can be changed in volatile memory with SPA, AOS, ATZ, DPO, IFC, RTR, VEL, WOS and WTR. Requires command level 1 (CCL.vi). Do not save more than 10 parameters at once.

E-725, E-727: The WPA command saves the currently valid parameter values to non- volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the E-725, E-727 is powered off or rebooted. Parameters can be changed in volatile memory with SPA, AOS, ATZ, DPO, IFC, RTR, VEL, WOS and WTR. Requires command level 1 (CCL.vi). As the WPA command takes up to 90 seconds to finish execution, WPA.vi polls for the controller ready signal (#7) before returning.

E-753: If Affected axes = empty array, all parameters for all axes are saved. Parameter no. format is TRUE (hex). The WPA command saves the currently valid parameter values to non- volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the E-753 is powered off or rebooted. Parameters can be changed in volatile memory with SPA, AOS, ATZ, DPO, IFC, RTR, VEL, WOS and WTR. Requires command level 1 (CCL.vi).

E-754: If Affected axes = empty array, all parameters for all axes are saved. Parameter no. format is TRUE (hex). The WPA command saves the currently valid parameter values to non- volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the E-754 is powered off or rebooted. Parameters can be changed in volatile memory with SPA, AOS, ATZ, DPO, IFC, RTR, VEL, WOS and WTR.

E-755: If Affected axes = empty array, all parameters for all axes are saved. Parameter no. format is TRUE (hex). The WPA command saves the currently valid parameter values to non- volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the E-755 is powered off or rebooted. Parameters can be changed in volatile memory with SPA, APG, BDR and SSA.

E-761: If Affected axes = empty array, all parameters for all axes (items) are saved. See E-7XX GCS DLL Manual for a list of available parameters. Parameter no. format is TRUE (hex.). The WPA command saves the currently valid parameter values and the additional settings listed below to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the PC is powered off or the E-761 is rebooted. Additional settings saved with WPA: Velocity control mode (VCO), position limits (NLM, PLM).

E-816: Affected axes and Parameter to save = empty array. This command cannot be issued to a slave.

E-861: The WPA command saves the currently valid parameter values to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the E-861 is powered off or rebooted.

The password for writing to non-volatile memory depends on the parameter and can be "100" or "4711". See the parameter list in "Controller Parameters" in the E-861 user manual for the password assignment.

When WPA is used without specifying any arguments except of the password, the currently valid values of all parameters affected by the specified password are

saved. Otherwise only one single parameter can be saved per WPA command. Parameters can be changed in volatile memory with SPA, SSA, ACC, DEC, VEL, OVL, OAC and ODC. If Affected axes = empty array, the currently valid values of all parameters affected by the specified password are saved (see below).

Parameter no. format is TRUE (hex).

E-870, E-872: If Affected axes = empty array, all parameters for all axes are saved. Parameter no. format is TRUE (hex). The WPA command saves the currently valid parameter values to non-volatile memory, including the joystick configuration, where they become the power-on defaults. Settings not saved with WPA will be lost when the controller is powered off or rebooted. Parameters can be changed in volatile memory with SPA.

E-871, E-873: If Affected axes = empty array, all parameters for all axes are saved. Parameter no. format is TRUE (hex). Depending on the password, the WPA command saves the currently valid parameter values and/or the current HI device configuration to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the controller is powered off or rebooted.

Valid passwords and affected settings:

100: all parameters, settings of HDT, HIA, HIT

101: all parameters

HID: settings of HDT, HIA, HIT

Parameters can be changed in volatile memory with SPA.

WPA must be used without specifying any arguments except of the password.

E-874: If Affected axes = empty array, all parameters for all axes are saved. Parameter no. format is TRUE (hex). Depending on the password, the WPA command saves the currently valid parameter values and/or the current HI device configuration to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the controller is switched off or rebooted.

Valid passwords and affected settings:

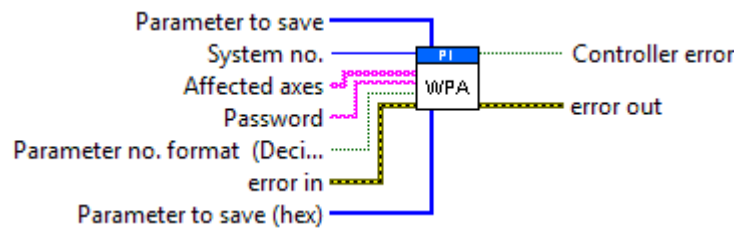
100: all parameters

Parameters can be changed in volatile memory with SPA.

WPA must be used without specifying any arguments except of the password.

Hydra, Pollux: Affected axes = empty array, the currently valid values of all parameters affected by the specified password are saved (see below). Parameter no. format is TRUE (hex). The WPA command saves the currently valid parameter values to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the controller is powered off or rebooted. Parameters can be changed in volatile memory with SPA, ACC, and VEL. WPA must be used without specifying any arguments except of the password, the currently valid values of all parameters affected by the specified password are saved.

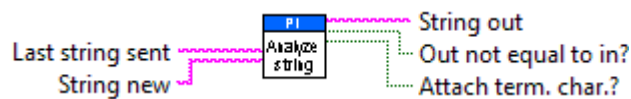
Mercury\_GCS: Affected axes = empty array, the currently valid values of all parameters affected by the specified password are saved (see below). Parameter no. format is TRUE (hex). The WPA command saves the currently valid parameter values to non-volatile memory, where they become the power-on defaults. Settings not saved with WPA will be lost when the Mercury is powered off or rebooted. Parameters can be changed in volatile memory with SPA, ACC, DEC and VEL. WPA must be used without specifying any arguments except of the password, the currently valid values of all parameters affected by the specified password are saved.



## 4.17. Support.Ilb

### 4.17.1. Analyse input string for terminal.vi

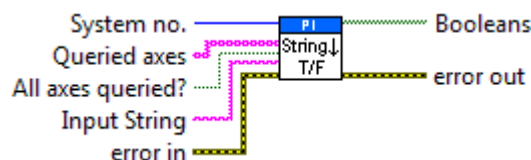
This VI is a sub-VI for PI Terminal.vi. It analyses String new and returns it in String out if it is not empty and does not contain a "#" at the beginning. In case of an empty new string, Last string sent is returned. If String new contains a "#" character, the corresponding ASCII character is returned.



### 4.17.2. Assign booleans from string to axes.vi

This VI assigns numerical values from input string to boolean values for queried axes. If All axes is TRUE, connected axes are read from Global2.vi and displayed on the front panel for assignment.

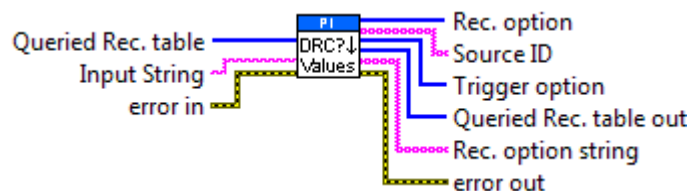
Example: An input string like "A=0SpaceLinefeedB=1Linefeed" or "0SpaceLinefeed1Linefeed" will be converted to an output array consisting of two values "FALSE; TRUE".



### 4.17.3. Assign DRC values.vi

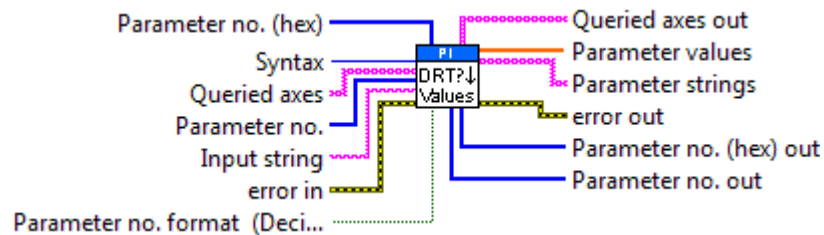
This VI assigns values (Source ID and Rec. option) from input string to queried Rec. tables. Sub-VI for DRC?.vi.

GCS 2.0: Trigger option is not valid.



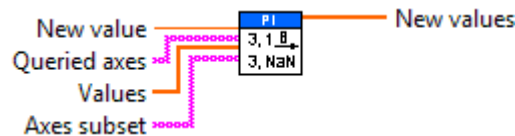
### 4.17.4. Assign DRT values from string.vi

This VI assigns numerical values / strings from input string to queried axes and parameter numbers. Sub-VI for DRT?.vi.



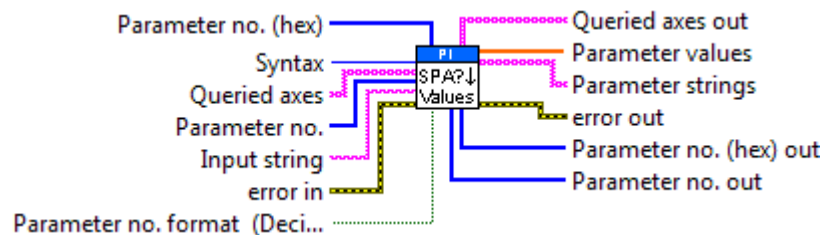
#### 4.17.5. Assign NaN for chosen axes.vi

This VI returns "NaN" for the given axes subset.



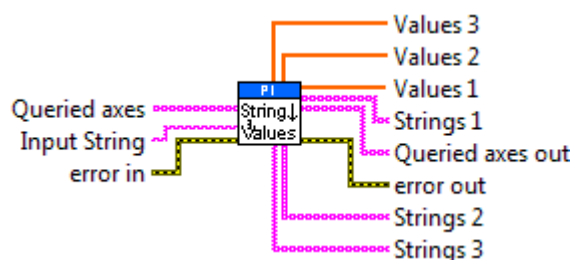
#### 4.17.6. Assign SPA values from string to axes.vi

This VI assigns numerical values / strings from input string to queried axes and hex. parameter numbers. Sub-VI for SPA?.vi.



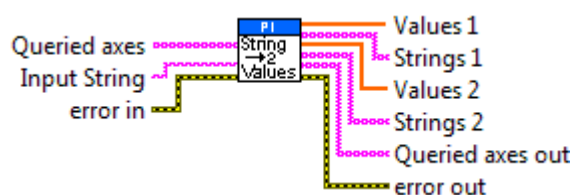
#### 4.17.7. Assign three values from string to axes.vi

This VI assigns three values and/or strings from input string to queried axes. Sub-VI for MVS.vi.



#### 4.17.8. Assign two values from string to axes.vi

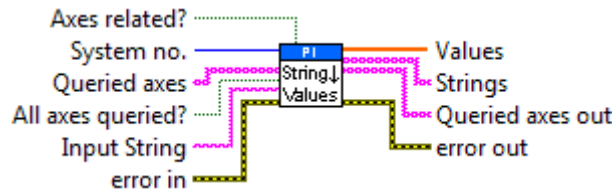
This VI assigns two values and/or strings from input string to queried axes.





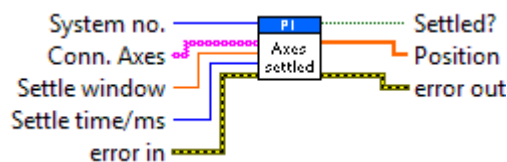
#### 4.17.9. Assign values from string to axes.vi

This VI assigns numerical values and/or single lines from input string to queried axes. If All axes is TRUE, connected axes are read from Global2.vi and displayed on the front panel for assignment.



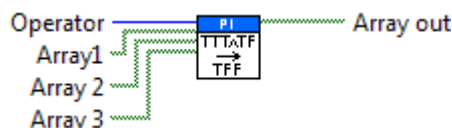
#### 4.17.10. Axes\_settled.vi

Returns the axes settled state which is a software mapping of the ONT? firmware command. Positions must be within the settle window during the settle time to be settled. This VI can take up to the time specified by the Settle time control.



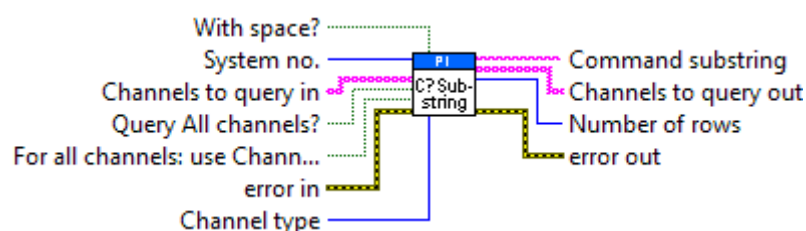
#### 4.17.11. Boolean array calculations.vi

This VI performs a boolean operation of up to three boolean input arrays. The difference to LabVIEW's own boolean operators is that the input arrays can have different sizes. The missing elements are considered to be FALSE elements and the resulting array contains the maximum number of elements.



#### 4.17.12. Build channel query command substring.vi

This VI builds a query command substring for channel query commands. If Query all channels? is true, channels to command are determined in a controller specific way and returned in Channels to query out, otherwise Channels to query out is identical with Channels to query in. Number of rows is size of the Channels to query out array. If Channel identifier? is FALSE, command substring is an empty string (e.g. for systems which accept commands like VMA? without channel IDs). If With space? is true, a space character is added between the channel identifiers.

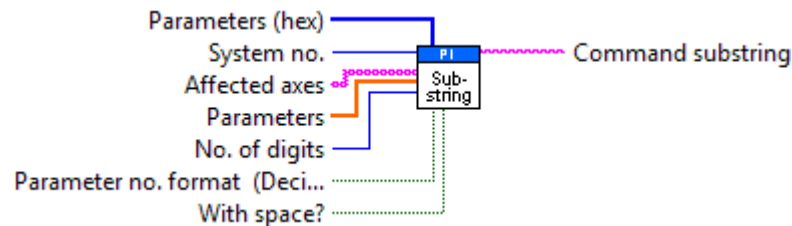




#### 4.17.13. Build command substring.vi

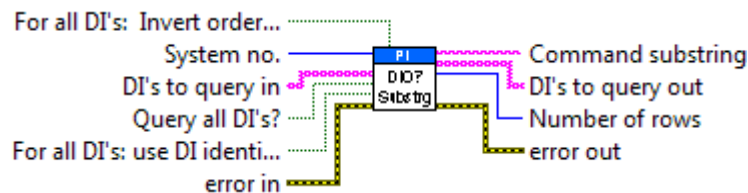
This VI builds a command substring by combining axis identifier and parameter. If parameter no. is in decimal format, use Parameters input, for hexadecimal parameter numbers use Parameters (hex) input and switch Parameter no. format to TRUE. Do not mix decimal and hex. parameter numbers in one call. No. of digits is the number of digits after the decimal point in the parameter value(s) that will be sent.

Example: For Affected axes = A; B, Parameters = 1.2342; 2.3 and No. of digits = 3 the resulting string is "SpaceA1.234SpaceB2.300".



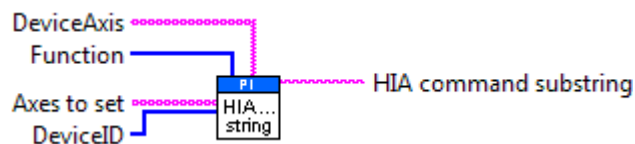
#### 4.17.14. Build DIO? query command substring.vi

This VI builds a DIO? query command substring. If Query all DI's is TRUE, available analog inputs are read using TIO? and DI identifiers are assigned using TVI? (valid identifiers are assigned to available DI's in ascending order) (GCS 1.0) or 1 to x with x being the number of available analog inputs (GCS 2.0). Number of rows is the size of the DI's to query out array. If DI identifier is FALSE, command substring is an empty string.



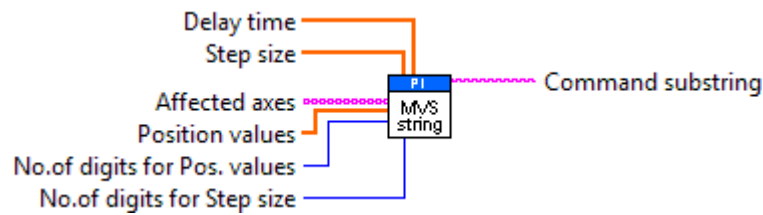
#### 4.17.15. Build HIA command substring.vi

This VI builds a command substring for HIA type commands. Sub-VI for HIA.vi, HIS.vi.



#### 4.17.16. Build MVS command substring.vi

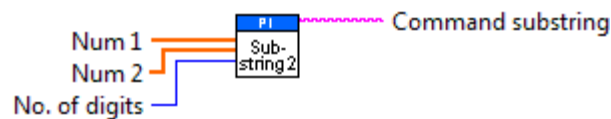
This VI builds a command substring for the MVS command by combining axis identifier and three parameters.



#### 4.17.17. Build num command substring.vi

This VI builds a command substring by combining Num 1, Space and Num 2. No. of digits is the number of digits after the decimal point in the Num 1/2 value(s) that will be sent.

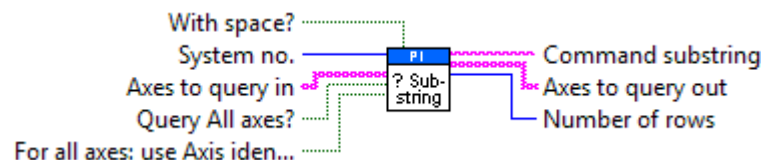
Example: For Num 1 = 1.24; 3.25456, Num 2 = 5.0; 7.4321 and No. of digits = 3 the resulting string is "Space1.240Space5.000Space3.255Space7.432".



#### 4.17.18. Build query command substring.vi

This VI builds a query command substring. If All axes is TRUE, connected axes are read from Global2.vi and returned in Axes to query out, otherwise Axes to query out is identical with Axes to query in. Number of rows is size of the Axes to query out array. If Axis identifier is FALSE, command substring is an empty string (e.c. for systems which accept commands like POS? without axis IDs). If With space? is true or system supports GCS 2.0, a space character is added between the axes identifiers.

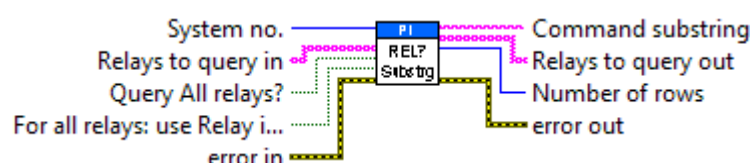
Example: If axes A;B;C;D are connected to the system to command, Axes to query in is A;B;D, Query all axes? is TRUE and Use Axis identifier is TRUE, resulting Command substring is "ABCD", Number of rows is 4 and Axes to query out is A;B;C;D.



#### 4.17.19. Build REL? query command substring.vi

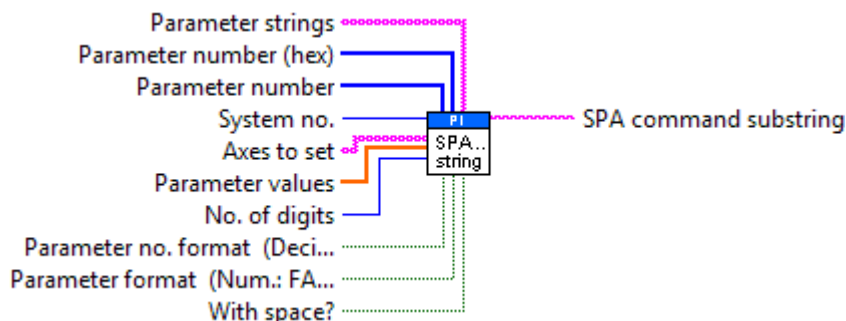
This VI builds a REL? query command substring. If Query all relays is TRUE, available relays are read using TRC?.vi and relay identifiers are assigned using TVI?.vi (valid identifiers are assigned to available relays in ascending order). Number of rows is size of the Relays to query out array. If Relay identifier is FALSE, Command substring is an empty string.

Example: If one relay board is installed in the system to command, Relays to query in is A;B;D, Query all relays? is TRUE and Use Relay identifier is TRUE, resulting Command substring is "ABCDEFGH", Number of rows is 8 and Relays to query out is A;B;C;D;E;F;G;H.



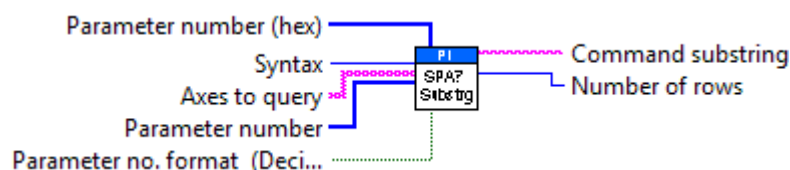
#### 4.17.20. Build SPA command substring.vi

This VI builds a command substring for the SPA command. No. of digits is the number of digits after the decimal point in the parameter value(s) that will be sent. Sub-VI for SPA.vi, CTO.vi, WTR.vi.



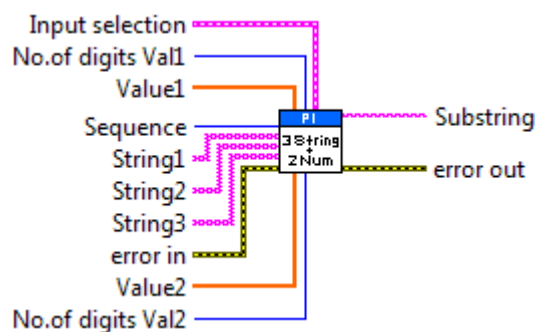
#### 4.17.21. Build SPA query command substring.vi

This VI builds an SPA? Command substring. Axes and parameters are combined into a substring. Number of rows is size of Axes to query array. Sub-VI for SPA?.vi and SEP?.vi.



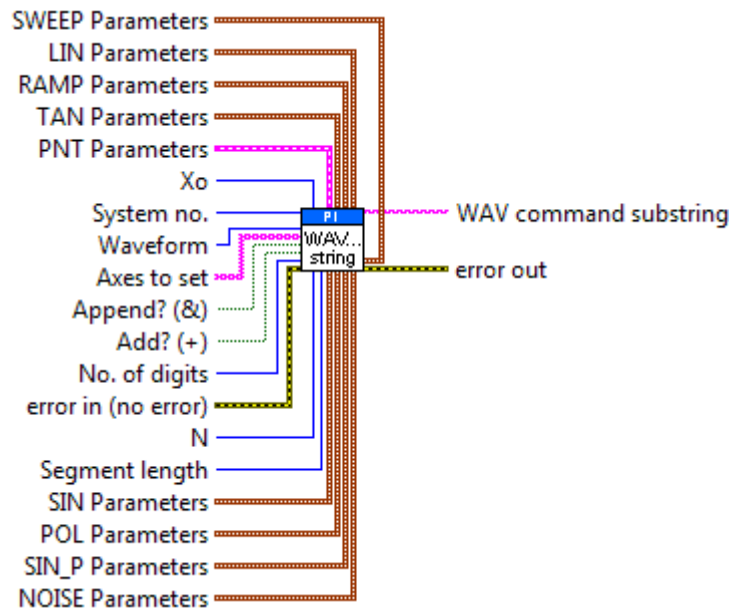
#### 4.17.22. Build stringplusnum substring.vi

This VI builds a command substring by combining up to three strings and two values. Sub-VI for FCG.vi, CTC.vi and DAS.vi



#### 4.17.23. Build WAV command substring.vi

This VI builds a command substring for the WAV command. No. of digits is the number of digits after the decimal point in the parameter value(s) that will be sent. Sub-VI for WAV.vi.



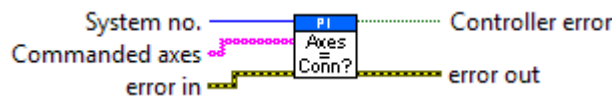
#### 4.17.24. Combine axes arrays.vi

This VI combines axes from Axes subset 1 and Axes subset 2 and returns the combined and sorted axes array plus axes which had double entries.



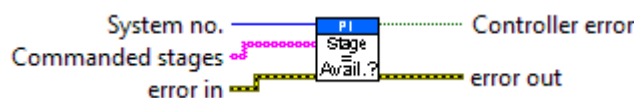
#### 4.17.25. Commanded axes connected?.vi

This VI checks if Commanded axes are a subset of all connected axes (read from Global2 (Array).vi) and returns Controller error TRUE if this is not the case. Connected axes are defined by Define connected axes.vi which is called by PI\_GCS2\_Configuration\_Setup.vi automatically. White space strings in Commanded axes are ignored.



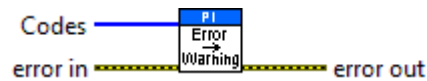
#### 4.17.26. Commanded stage name available?.vi

This VI checks if Commanded stages is a subset of all available stages and returns Controller error TRUE if this is not the case. Available stages are defined by VST?.vi.

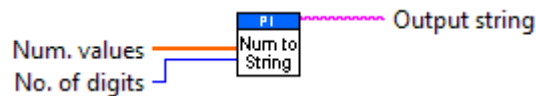


**4.17.27. Convert error to warning.vi**

If "code" is one of the code numbers given in "Codes", resets error status to "no error" and adds "Warning: " to "source".

**4.17.28. Convert num array to string.vi**

This VI converts an array of numerical values to a space separated output string. The difference to LabVIEW's native Array to Spreadsheet String function is that no carriage return or newline is added.

**4.17.29. Convert num value to syntax selection.vi**

This VI converts a numerical value to the corresponding GCS syntax version.

**4.17.30. Count occurrences in string.vi**

This VI counts, how often an expression occurs in a string.

**4.17.31. Create Controller Names.vi**

Create a controller name array from given controller name.

C-887: add additional names for this controller.

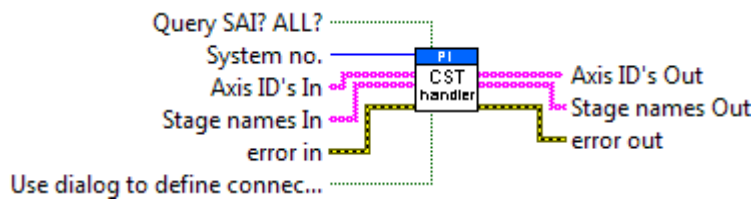
**4.17.32. CST handler.vi**

Defines connected stages for axes, depending on several selections. Requires SAI.vi, SAI?.vi, CST?.vi and ERR?.vi. For Query SAI? ALL? = TRUE, VI determines connected axes by querying SAI? ALL. For Query SAI? = FALSE; Axis ID's IN must contain the correct identifiers of all connected axes. Use dialog to define connected stages? determines if a dialog pops up for the stage selection. If this control is set to FALSE, Stage names In must contain the correct stage names for all connected axes. Empty fields are substituted with "NOSTAGE".

C-877, C-884, C-891: Command is available via USB interface only.

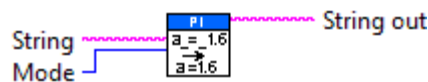
C-867, E-871, E-873: Command is available via USB, USB DaisyChain or RS232 DaisyChain interface only.

Pollux: See HydraPollux\_GCS\_DLL manual for specialities of the Pollux controller family regarding stage name settings.



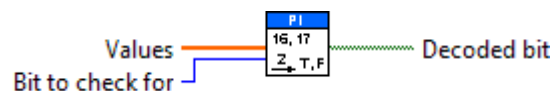
#### 4.17.33. Cut out additional spaces.vi

Searches for spaces in String and cuts them out if the following character is not LF.



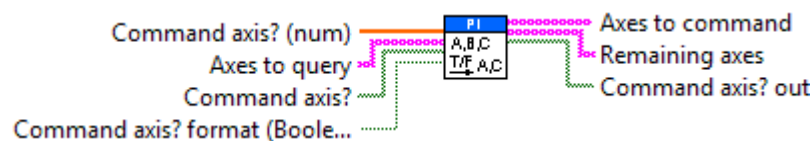
#### 4.17.34. Decode status bit.vi

This VI checks the status bit value for a certain bit and returns Decoded\_bit indicating whether that bit has a value of TRUE or FALSE.



#### 4.17.35. Define axes to command from boolean array.vi

This VI returns only those axes IDs from the Axes to query array in the Axes to command array which have a boolean value TRUE in the Command axis? array or a numerical value of 1 in the Command axis? (num) array, depending on Command axis? format, and returns all remaining axes in the Remaining axes array.



#### 4.17.36. Define connected stages with dialog.vi

This VI reads the connected axes from Global2.vi and the available stages using VST?.vi and returns the stage names in a ring control. If there are no entries in that ring control, PISTages.dat was not found, or is marked read-only. Run GCSLibrarySetup.exe to register PISTages.dat correctly. In Microsoft Explorer, right-click the file PISTages.dat and select "Properties". Make sure that the "read-only" attribute is not checked.

If OK is pressed, the chosen assignment in Connected stages is written into Stage selection. If Cancel is pressed, the contents of Stage selection are not changed. For axes without a stage connected, choose "NOSTAGE".

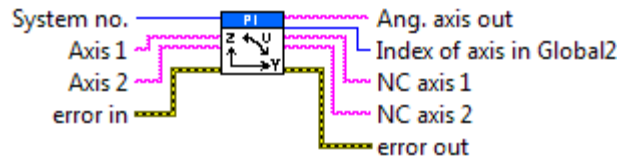
C-877, C-884, E-871, E-873: VI is valid via USB interface only.

C-867: VI is valid via RS232 DaisyChain, USB or USB DaisyChain only.



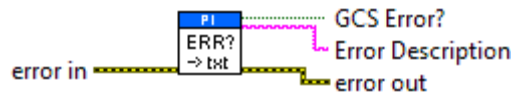
#### 4.17.37. Determine angular axis.vi

Determines angular axis spanned by Axis 1 and Axis 2, and NanoCube axis identifiers corresponding with Axis 1 and Axis 2.



#### 4.17.38. GCSTranslateError.vi

Returns if error in contains a GCS error code and if this is the case, it displays the corresponding error message and appends it to "source" in error out.



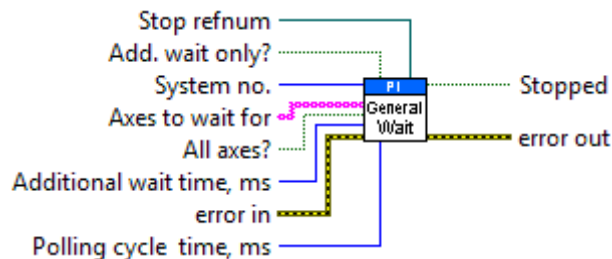
#### 4.17.39. General wait for movement to stop.vi

This VI waits for the specified axes to stop. An additional wait time can be specified. The wait method depends on the system to command. PI\_GCS2\_Configuration\_Setup.vi must be run before running this vi. If Add. wait only? is TRUE, VI waits the given Additional wait time only. Requires Wait for axes to stop.vi, #5.vi, STA?.vi, #5\_old.vi, ONT?.vi and Wait for hexapod system axes to stop.vi to be present. VI does not time out, so when using as a sub-VI, use Stop refnum to stop VI from caller.

Analog: If VOL was commanded or velocity control mode (VCO) is OFF, Add. wait only? must be TRUE, otherwise FALSE.

E-816: All axes? = FALSE. only one axis per command allowed.

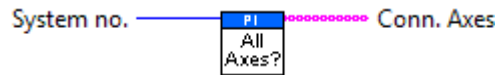
C-887, F-206, M-8X0: For GCS syntax version = GCS 1.0 (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0), VI will not wait for INI procedure to complete.





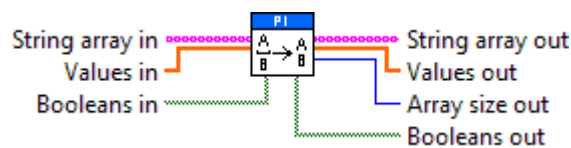
#### 4.17.40. Get all axes.vi

This VI reads all connected axes for given system from Global2 (Array).vi. Connected axes are defined by Define connected axes.vi, which is called by PI\_GCS2\_Configuration\_Setup.vi automatically.



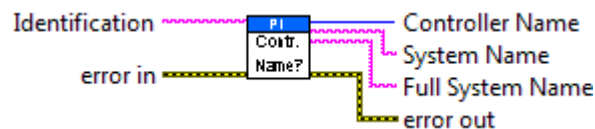
#### 4.17.41. Get arrays without blanks.vi

Returns the string array and related values and boolean arrays without white space string fields.



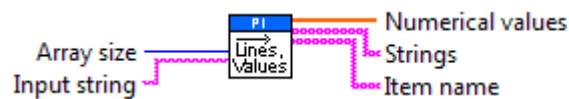
#### 4.17.42. Get Controller Name from IDN.vi

This VI extracts the system name from the \*IDN? answer.



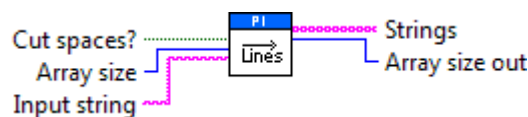
#### 4.17.43. Get lines and values from string.vi

This VI returns numerical values and single lines from input string without any axis assignment. If number of lines/values (Array size) is known, algorithm is faster, otherwise Array size = 0 should be used. Sub-VI for VST?.vi and STE?.vi.



#### 4.17.44. Get lines from string.vi

This VI returns single lines from input string. If number of lines (Array size) is known, algorithm is faster, otherwise Array size = 0 should be used. Sub-VI for VST?.vi and Define macro contents (with or without) delay.vi.



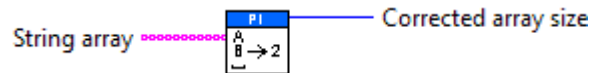
#### 4.17.45. Get Slave Device Info.vi

This VI queries VER? and lists all connected DaisyChain slave devices.



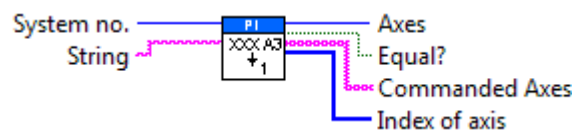
#### 4.17.46. Get string array size without blanks.vi

This VI returns the size of a string array without counting white space strings.



#### 4.17.47. Get total number of commanded axes.vi

This VI returns how many axes to command String contains and if that number is equal to the total number of connected axes. Additionally it returns all commanded axes and their index in the Connected Axes array.



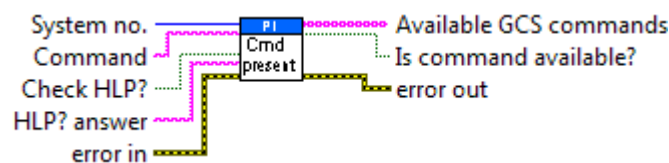
#### 4.17.48. HasCommand?.vi

Polymorphic VI instance for Is command present in HLP? answer.vi.

This VI checks whether the specified command is present in the HLP? answer.

If Check HLP? is TRUE, HLP?.vi will be called to get HLP? information.

If Check HLP? is FALSE, a valid HLP? answer has to be provided in HLP? answer.



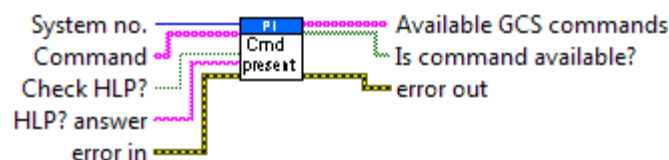
#### 4.17.49. HasCommandArray?.vi

Polymorphic VI instance for Is command present in HLP? answer.vi.

This VI checks whether the specified commands are present in the HLP? answer.

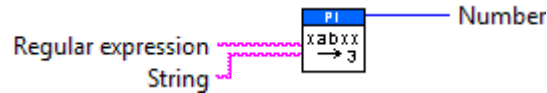
If Check HLP? is TRUE, HLP?.vi will be called to get HLP? information.

If Check HLP? is FALSE, a valid HLP? answer has to be provided in HLP? answer.



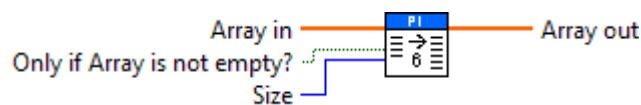
#### 4.17.50. How often does string contain regular expression?.vi

This VI returns a count of the occurrences of a regular expression in a string.



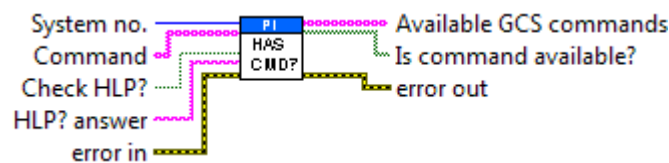
#### 4.17.51. Increase array size.vi

If size of Array in is smaller than Size, this VI increases the size of Array in to Size. If Array in is an empty array and Only if Array is not empty? is FALSE, VI builds an array of zeros with the size of Size.



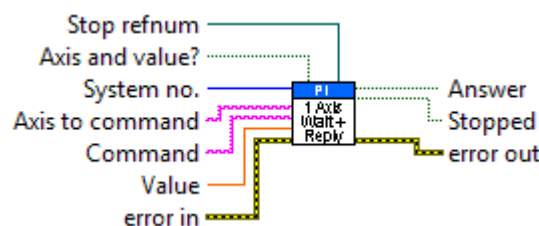
#### 4.17.52. Is command present in HLP answer?.vi

This VI checks whether the specified commands are present in the HLP? answer. If Check HLP? is TRUE, HLP?.vi will be called to get HLP? information. If Check HLP? is FALSE, a valid HLP? answer has to be provided in HLP? answer.



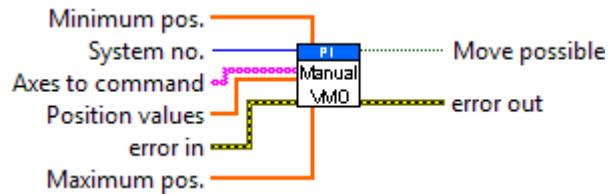
#### 4.17.53. Longlasting one-axis command.vi

This VI sends a command (like REF, MNL or MPL), polls with #7 for controller-ready signal and returns original (boolean) command response.

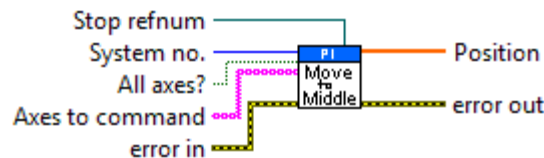


**4.17.54. Manual VMO.vi**

Virtual movement. Indicates whether a move to the specified position is possible or not by checking if the commanded position value is within the given position range. Stage will not be moved.

**4.17.55. Move axes to their middle position.vi**

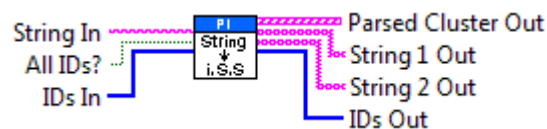
Moves axes to the middle position of their travel range, waits until motion has finished and queries position. VI also stops if Stop refnum or Local stop is TRUE. When using as a sub-VI, use Stop refnum to stop VI from caller.

**4.17.56. Parse KLS? type answer by type.vi**

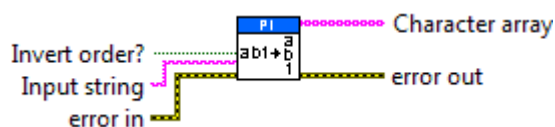
Filters the response of KLS? type answers by the selected type and returns all names related to that type.

**4.17.57. Parse\_FRH\_Type\_String.vi**

Returns ID, String 1 and String 2 for String In. String In must be in FRH? reply format (ID1 = String1 TAB String2 Space LF ID2 = String 3 TAB String 4 Space LF end of help LF). If All IDs? is FALSE, parsed strings are returned for IDs In only.

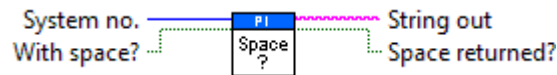
**4.17.58. Return single characters from string.vi**

Get single characters from input string.



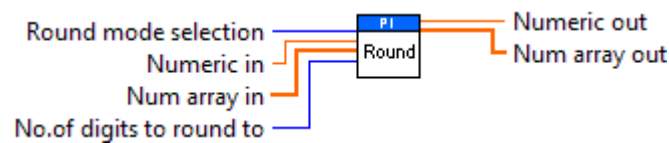
#### 4.17.59. Return space.vi

This VI returns a space character in String out if With space? is TRUE or GCS syntax version is higher than 1.0.



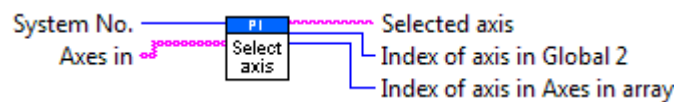
#### 4.17.60. Round with options.vi

Rounds Numeric in and Num array according to No. of digits to round to and Round mode selection.



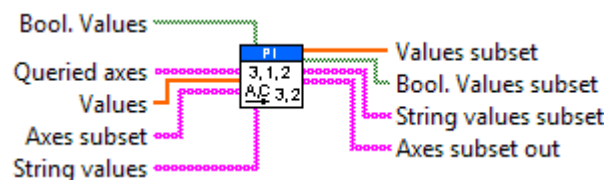
#### 4.17.61. Select axis.vi

This VI reads all connected axes from Global2 and writes them into a menu ring control for selection. The selected axis and its index in Global2 are returned.



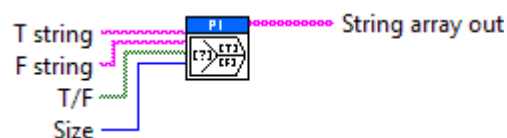
#### 4.17.62. Select values for chosen axes.vi

This VI returns only values for the given axes subset.



#### 4.17.63. Select with boolean array input.vi

This VI returns a string array of a given size with T string and F string, depending on the boolean value at the corresponding index of T/F.



**4.17.64. Selection to String array.vi**

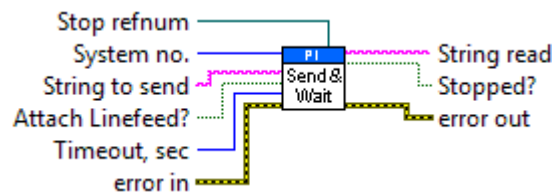
This VI returns a string array which contains strings according to the selected value of String input.

Example: For Selection array = (2,0,1) and String input = (A,B,C) the resulting String array is (C,A,B).

**4.17.65. Send string and wait for answer or timeout.vi**

Sends a string and waits until an answer has come or a timeout condition has occurred. Timeout is independent of the global communication timeout value. Sends #24 if VI was stopped. Needed by INI hexaxes and wait until finished.vi.

C-887, F-206, M-8X0: INI procedure of GCS 1.0 firmware (Check with CSV?.vi. If CSV?.vi is not supported, syntax version is GCS 1.0) can be stopped with #24 since firmware version 4.0.0 (F-206) and 4.5.0 (M-8X0).

**4.17.66. Sep axes ini result in order.vi**

Returns result of Sep. axes initialized? according to order of Sep. axes to initialize.

**4.17.67. Set RON and return RON status.vi**

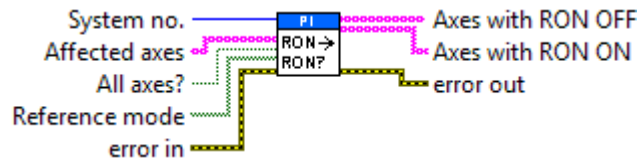
Sets RON mode ON or OFF and returns which of the connected axes have RON mode ON and which have RON mode OFF.

If reference mode is OFF, no referencing is required for the axis. Only relative moves can be commanded (using MVR.vi), unless the actual position is set with POS.vi. Afterwards, relative and absolute moves can be commanded.

For stages with neither reference nor limit switch, reference mode is automatically OFF.

**WARNING!** If reference mode is switched off, and relative moves are commanded, stages can be driven into the mechanical hard stop if moving to a position which is outside the travel range!

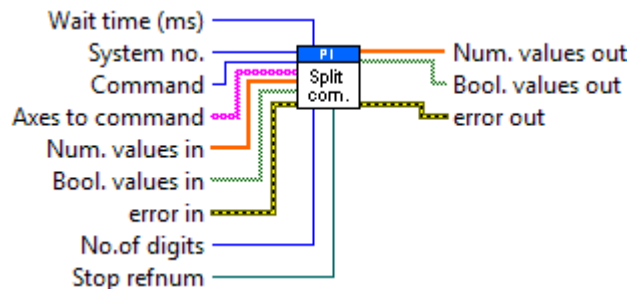
If reference mode is switched off, and the actual position is incorrectly set with POS.vi, stages can be driven into the mechanical hard stop when moving to a position which is thought to be within the travel range of the stage, but actually is not.



#### 4.17.68. Split multiple axes command.vi

Splits command with more than one axis specification into the corresponding one-axis commands and returns answers for all given axes (if any). Wait time (ms) defines the time to wait between single commands. When using as a sub-VI, connect Stop refnum terminal to stop VI from caller.

Supported commands are DCO, DCO?, MOV, MOV?, MVR, ONT?, OVF?, POS?, SVA, SVA?, SVO, SVO?, SVR, VOL? and General Wait for movement to stop (GWait).



#### 4.17.69. String with ASCII code conversion.vi

Converts each ASCII control code from Input string to "\x" with x being the ASCII code of the corresponding character for better readability of log files.



#### 4.17.70. Subtract axes array subset from axes array.vi

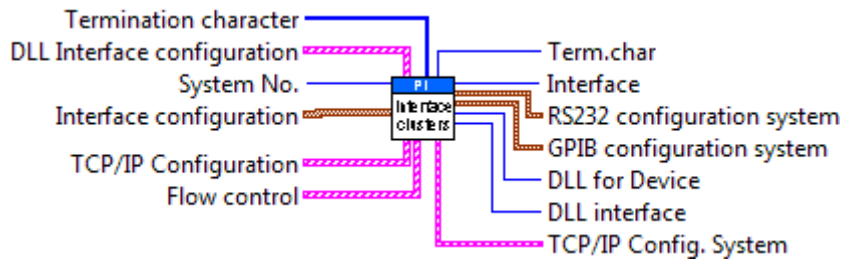
This VI returns only these axes IDs from the Axes to query array which are not present in the Axes subset array. If no axes IDs are returned, All present? is TRUE. Needed by Define axes to command from boolean array.vi.



#### 4.17.71. Unbundle/bundle interface clusters for PI Terminal.vi

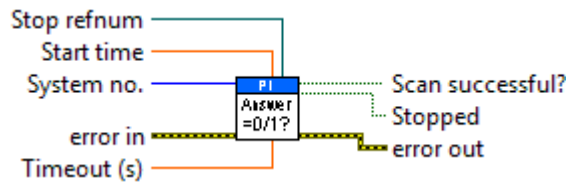
This VI is a sub-VI for PI Terminal.vi. It unbundles Interface configuration and DLL Interface configuration and returns the cluster contents in a different composition which is used by PI Terminal.vi.





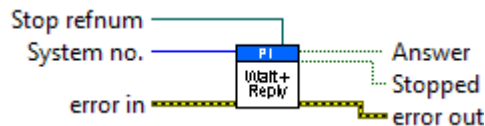
#### 4.17.72. Wait for answer 0 or 1 without polling.vi

This VI waits until answer 0 or 1 comes or timeout is reached without single character polling. VI also stops if Stop refnum or Local stop is TRUE. When using as a sub-VI, use Stop refnum to stop VI from caller.



#### 4.17.73. Wait for answer of longlasting command.vi

This VI waits for the answer of commands like REF, MPL, MNL or scanning routines using #7 polling and stops if answer has come, Stop refnum or Local stop is TRUE, or if a communications error occurred. Sub-VI for long-lasting, one-axis commands and controller-algorithm commands. Requires #7.vi to be present. When using as a sub-VI, use Stop refnum to stop VI from caller.

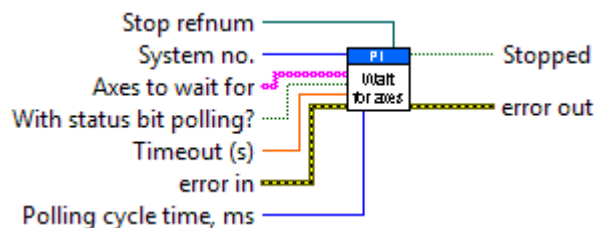


#### 4.17.74. Wait for axes to stop.vi

This VI waits for the specified axes to stop using #5 polling. It also stops if a communication error occurred, Stop refnum or Local stop is TRUE. Requires STA?.vi to be present. Required by General wait for movement to stop.vi. When using as a sub-VI, use Stop refnum to stop VI from caller.

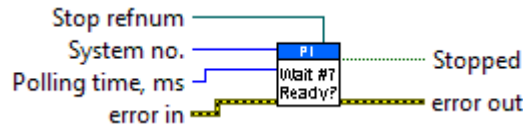
C-880: With status bit polling? = TRUE

All other systems: With status bit polling? = FALSE



#### 4.17.75. Wait for controller ready.vi

This VI waits for controller ready signal using #7 polling and stops also if Stop refnum or Local stop is TRUE, or if a communications error occurred. Requires #7.vi to be present. When using as a sub-VI, use Stop refnum to stop VI from caller.



### 4.18. WaveGenerator.IIb

#### 4.18.1. #9.vi

Polls to determine whether a wave generator is running for any of the connected axes by sending the single ASCII character 9. Connected axes are read from Global2.vi and displayed on the front panel for assignment.

C-843: #9 polls the state of the User Profile mode.



#### 4.18.2. DDL.vi

Loads the specified values to the specified DDL table. Please refer to the controller user manual for units and restrictions.

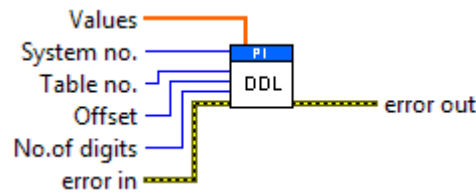
E-710: Table number can be 1 to 8. First stored value in table has index 1, index 0 must be DDL repeat number. Maximum 32768 values can be stored.

E-712: Table number can be 1 to the maximum number of axes (query with TLT? for the number of DDL tables). First stored value in table has index 1, so Offset must be greater than zero. Maximum 262,144 values can be stored. Due to the maximum numbers of parameters per command only 30 DDL table values can be set per one command call.

E-725, E-727: Table number can be 1 to the maximum number of axes (query with TLT? for the number of DDL tables). First stored value in table has index 1, so Offset must be greater than zero. Maximum 262,144 values can be stored. Due to the maximum numbers of parameters per command only 30 DDL table values can be set per one command call.

E-753: Table number can be 1. First stored value in table has index 1, so Offset must be greater than zero. Maximum 65,536 values can be stored. Due to the maximum numbers of parameters per command only 30 DDL table values can be set per one command call.

E-754: Table number can be 1. First stored value in table has index 1, so Offset must be greater than zero. Maximum 1,046,076 values can be stored. Due to the maximum numbers of parameters per command only 30 DDL table values can be set per one command call.



#### 4.18.3. DDL?.vi

Returns N values from specified DDL Table (in the controller memory). For large N values, communication timeout must be set long enough, otherwise a comm. error may occur.

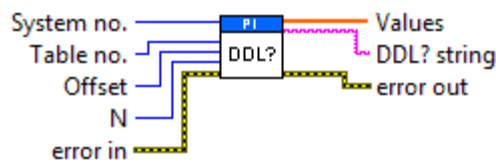
E-710: Table number can be 1 to 8. First stored value in table has index 1, index 0 is the DDL repeat number. Maximum 32768 values can be stored.

E-712: Table number can be 1 to the maximum number of axes (query with TLT? for the number of DDL tables). Offset is start point and must be greater than zero, because first stored value in table has index 1. Maximum 262,144 values can be stored.

E-725, E-727: Table number can be 1 to the maximum number of axes (query with TLT? for the number of DDL tables). Offset is start point and must be greater than zero, because first stored value in table has index 1. Maximum 262,144 values can be stored.

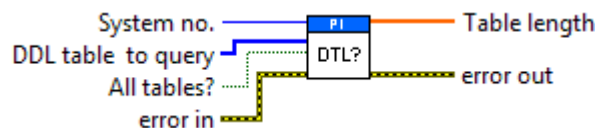
E-753: Table number can be 1. Offset is start point and must be greater than zero, because first stored value in table has index 1. Maximum 65,536 values can be stored.

E-754: Table number can be 1. Offset is start point and must be greater than zero, because first stored value in table has index 1. Maximum 1,046,076 values can be stored.



#### 4.18.4. DTL?.vi

Returns DDL table length.



#### 4.18.5. GWD?.vi

Returns N waveform sequence points. Nmax is the maximum number of points that can be read at once. If  $N > N_{max}$ , then the VI will query GWD? more than once to read all N values.

C-413: Query axis is identical with wave table number, which can be 1 to 8. Nmax must be less than or equal to the length of the defined wave. Xo must be greater than zero.

C-886: Query axis is identical with wave table number. Nmax must be less than or equal to the length of the defined wave. Xo must be greater than zero.

C-887: Query axis is identical with wave table number, which can be 1 to 100. Nmax must be less than or equal to the length of the defined wave. Xo must be greater than zero.

C-891: Query axis is identical with wave table number, which can be 1 to 40. Nmax must be less than or equal to the length of the defined wave. Xo must be greater than zero. The GWD? command is present for compatibility reasons only. The response to GWD? only contains the header, but no waveform values.

E-516: Nmax = 5, N can be 8192 maximum

E-517, E-518: Query axis is identical with wave table number, which can be 1 to 3. Nmax must be less than or equal to the length of the defined wave. Xo must be greater than zero. The response to GWD? does not contain any offset to the wave generator output set with WOS.

E-709: Query axis is identical with wave table number, which can be 1 to 4. Nmax must be less than or equal to the length of the defined wave. The recommended Nmax value is 1000. For larger values it might be necessary to increase the timeout value in E709\_Configuration\_Setup.vi. Xo must be greater than zero.

E-710: Query axis is identical with wave table number. Nmax must be less than the length of the defined wave.

E-712: Query axis is identical with wave table number, which can be 1 to 90. Nmax must be less than or equal to the length of the defined wave. Xo must be greater than zero.

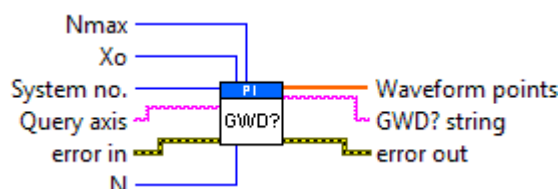
E-725: Query axis is identical with wave table number, which can be 1 to 30. Nmax must be less than or equal to the length of the defined wave. Xo must be greater than zero.

E-727: Query axis is identical with wave table number, which can be 1 to 40. Nmax must be less than or equal to the length of the defined wave. Xo must be greater than zero.

E-753: Query axis is identical with wave table number, which can be 1 to 10. Nmax must be less than or equal to the length of the defined wave. Xo must be greater than zero.

E-754: Query axis is identical with wave table number, which can be 1 to 10. Nmax must be less than or equal to the length of the defined wave. Xo must be greater than zero.

E-761: Query axis is identical with wave table number, which can be 1 to 4. Nmax must be less than the length of the defined wave. The content of a wave table is not completely erased when a new waveform is written to this table. Only the number of points given by the new waveform is written beginning with the first point in the table, but any subsequent data points will keep the old values from the former waveform. This affects only the response to the GWD? query and not the wave generator output which will only send the new waveform points.



#### 4.18.6. Send WG commands and wait for answers.vi

This VI sends the wave generator commands from Command array to the given system and waits for the controller to reply. Use WGWaveEditor\_with\_parameter.vi, or use the WGWaveEditorI\_call\_Editor.vi, and

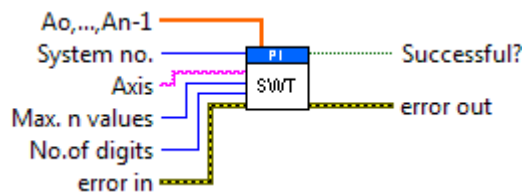
WGWaveEditor\_read\_commands.vi to receive the strings for Command array from WGWaveEditor.dll. Controller error is TRUE if selected system reports an error code which is not zero.



#### 4.18.7. SWT.vi

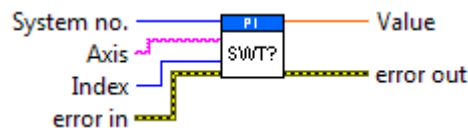
This VI defines a waveform for one axis. Data points are transferred point by point. Max. n values is the maximum number of points that can be transferred. No. of digits is the number of digits after the decimal point in the data point value(s) that will be sent. Please refer to the system manual for units and restrictions.

E-816: Max. n values = 64



#### 4.18.8. SWT?.vi

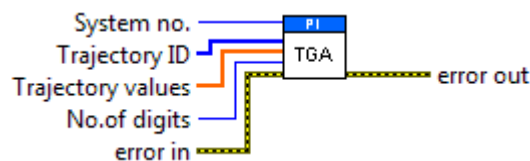
Returns wave table point with given index for given axis.



#### 4.18.9. TGA.vi

Append values to trajectories. For further details, see "Trajectories for Motion Paths" in the controller user manual.

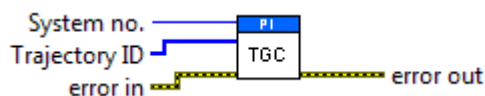
C-867, C-884:



#### 4.18.10. TGC.vi

Clear all values of trajectories. For further details, see "Trajectories for Motion Paths" in the controller user manual.

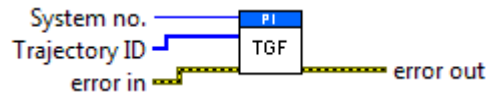
C-867, C-884:



#### 4.18.11. TGF.vi

Finalize trajectories. For further details, see "Trajectories for Motion Paths" in the controller user manual.

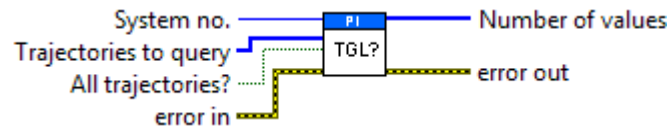
C-867, C-884:



#### 4.18.12. TGL?.vi

Get number of values stored in trajectories. For further details, see “Trajectories for Motion Paths” in the controller user manual.

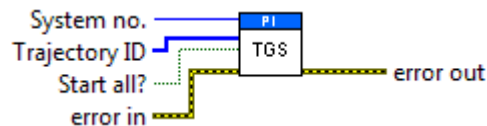
C-867, C-884:



#### 4.18.13. TGS.vi

Start trajectories. For further details, see “Trajectories for Motion Paths” in the controller user manual.

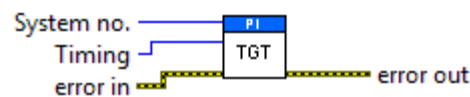
C-867, C-884:



#### 4.18.14. TGT.vi

Set trajectory timing. For further details, see “Trajectories for Motion Paths” in the controller user manual.

C-867, C-884:



#### 4.18.15. TGT?.vi

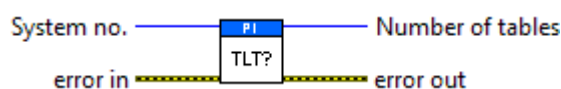
Get trajectory timing. For further details, see “Trajectories for Motion Paths” in the controller user manual.

C-867, C-884:



#### 4.18.16. TLT?.vi

Returns the number of DDL tables available in the controller.



#### 4.18.17. TWC.vi

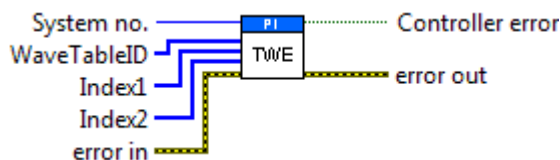
Clears the trigger settings for the waveform.



#### 4.18.18. TWE.vi

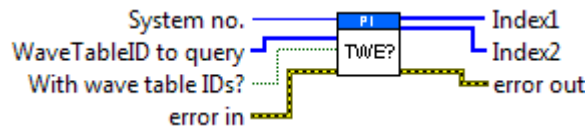
Defines the edges of a trigger signal which is to be output in conjunction with the wave generator output. The edges of the trigger signal are defined via the index of the corresponding waveform points of the given wave table. Controller error is TRUE if selected system reports error code  $\neq 0$ .

E-518: WaveTableID can be 1 to 3



#### 4.18.19. TWE?.vi

Reads back the trigger definition set with TWE.



#### 4.18.20. TWG?.vi

Returns the number of available wave generators.

C-413: Number of wave generators = 2

C-887: Number of wave generators = No. of connected axes.

C-891: Number of wave generators = 1

E-517, E-518: Number of wave generators = 3

E-709: Number of wave generators = 1

E-710: Number of wave generators = 2

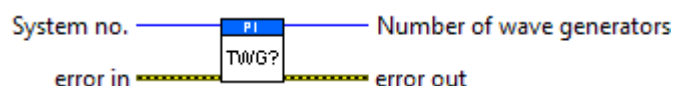
E-712: Number of wave generators = No. of connected axes

E-725, E-727: Number of wave generators = No. of connected axes

E-753: Number of wave generators = 1

E-754: Number of wave generators = 1

E-761: Number of wave generators = 4





#### 4.18.21. TWS.vi

GCS 1.0: Sets trigger values for point(s) on the waveform and checks for error. Trigger value is bit-mapped.

GCS 2.0: Defines the trigger state of a certain trigger line for a certain waveform point to high or low.

Controller error is TRUE if selected system reports error code not equal to 0.

E-517, E-518: Trigger value is the trigger line to use, can be 1 to 3. Wavepoint can be 1 to 8,192. The power-on default state of all points is low. Afterwards, the signal state of the trigger output line can be switched to "low" for all points using the TWC command. It is recommended to use TWC before trigger actions are set with TWS. During the waveform output, the TWS settings will only be used if the configuration of the appropriate trigger line was set with CTO before (set Parameter value "Generator Trigger" for CTO Parameter ID "Trigger Mode"). During the waveform output, the TWS settings will only be used if the configuration of the appropriate trigger line was set with CTO before (set Parameter value "Generator Trigger" for CTO Parameter ID "Trigger Mode"). If you start the wave generator with the WGO start options given by bit 3, 4 or 5, the corresponding pulses are output in addition to the output pulses defined with TWS. The number of wave generator output cycles during which trigger pulses are to be output can be set using the Number Of Trigger Cycles parameter, ID 0x18000100. The width of a trigger pulse is 30  $\mu$ s by default. You can change the default pulse width using the Pulse Width parameter, ID 0x0E000900. Possible values are in the range of 10 to 150  $\mu$ s.

E-709: Trigger value is the trigger line to use, can be 1 or 2. Line 2 is identical with line 1 here as there is only one trigger table for all wave tables. Wavepoint can be 1 to 16,332. Switch can be 0 (low) or 1 (high). The power-on default state of all points is low. Afterwards, the signal state of the trigger output line can be switched to "low" for all points using the TWC command. It is recommended to use TWC before trigger actions are set with TWS. During the waveform output, the TWS settings will only be used if the configuration of the appropriate trigger line was set with CTO before (set Parameter value "Generator Trigger" for CTO Parameter ID "Trigger Mode"). Only one trigger line per command allowed.

E-710: Wavepoint can be 1-16000. Switch is not valid. Trigger value is bitmapped:

- Bit 0: trigger line 1: 0 not active, 1 active
- Bit 1: trigger line 2: 0 not active, 1 active
- Bit 2: trigger line 3: 0 not active, 1 active
- Bit 3: trigger line 4: 0 not active, 1 active
- Bit 8: If = 0, then the Trigger values apply to corresponding Wavepoints only.

If = 1, then the Trigger value applies to all points between the last point set by this command and the corresponding Wavepoint point.

E-712: Trigger value is the trigger line to use, can be 1 to 7. Wavepoint can be 1 to 262,144. Switch can be 0 (low) or 1 (high). The power-on default state of all points is low. Afterwards, the signal state of the trigger output line can be switched to "low" for all points using the TWC command. It is recommended to use TWC before trigger actions are set with TWS. During the waveform output, the TWS settings will only be used if the configuration of the appropriate trigger line was set with CTO before (set Parameter value "Generator Trigger" for CTO Parameter ID "Trigger Mode"). Due to the maximum number of parameters per command 10 trigger lines can be set per command call.

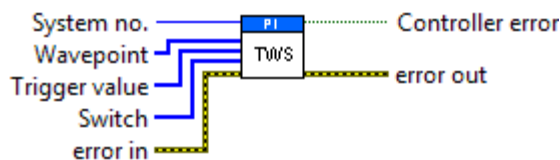
E-725: Trigger value is the trigger line to use, can be 1 to 7. Wavepoint can be 1 to 262,144. Switch can be 0 (low) or 1 (high). The power-on default state of all points is low. Afterwards, the signal state of the trigger output line can be switched to

"low" for all points using the TWC command. It is recommended to use TWC before trigger actions are set with TWS. During the waveform output, the TWS settings will only be used if the configuration of the appropriate trigger line was set with CTO before (set Parameter value "Generator Trigger" for CTO Parameter ID "Trigger Mode").

E-727: Trigger value is the trigger line to use, can be 1 to 3. Wavepoint can be 1 to 262,144. Switch can be 0 (low) or 1 (high). The power-on default state of all points is low. Afterwards, the signal state of the trigger output line can be switched to "low" for all points using the TWC command. It is recommended to use TWC before trigger actions are set with TWS.

E-753: Trigger value is the trigger line to use, can be 1. Wavepoint can be 1 to 65,536. Switch can be 0 (low) or 1 (high). During the waveform output, the TWS settings will only be used if the configuration of the appropriate trigger line was set with CTO before (set Parameter value "Generator Trigger" for CTO Parameter ID "Trigger Mode").

E-754: Trigger value is the trigger line to use, can be 1. Wavepoint can be 1 to 1,048,576. Switch can be 0 (low) or 1 (high). During the waveform output, the TWS settings will only be used if the configuration of the appropriate trigger line was set with CTO before (set Parameter value "Generator Trigger" for CTO Parameter ID "Trigger Mode").



#### 4.18.22. TWS?.vi

Returns trigger points set by TWS. N must be less than or equal to Nmax. For large N values, communication timeout must be set long enough, otherwise a comm. error may occur.

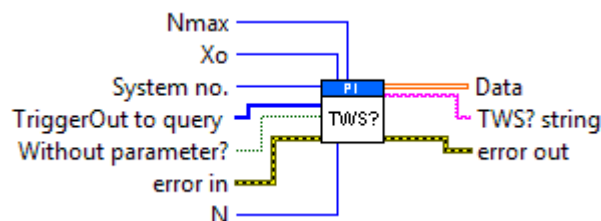
E-517, E-518: Nmax = 8192.

E-712: Nmax = 262,144.

E-725, E-727: Nmax = 262,144.

E-753: Nmax = 262,144.

E-754: Nmax = 1,048,576.

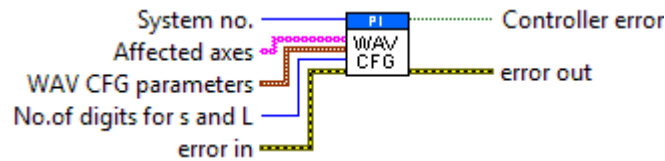


#### 4.18.23. WAV CFG.vi

This VI configures a wave generator for one or more axes. Controller error is TRUE if selected system reports an error code which is not zero. Please refer to the system manual for a description of wave generator configuration parameters, units and restrictions.

E-516, E-761: Only one axis per command allowed.

E-517, E-518: Affected axis is identical with wave table number, which can be 1 to 3. Only one wave table per command allowed.



#### 4.18.24. WAV.vi

Defines or modifies a stored waveform for one or more axes. Please refer to the GCS DLL manual or to the user manual for a description of waveform parameters, units and restrictions. No. of digits is the number of digits after the decimal point in the parameter value(s) that will be sent. Controller error is TRUE if selected system reports error code not equal to 0.

C-413: Affected axis is identical with wave table number. 8 waves can be defined in total. Use WSL.vi to connect wave table to wave generator. See user manual for details.

Waveform can be: SIN\_P, RAMP, LIN or PNT.

C-886: Affected axis is identical with wave table number. 100 waves can be defined in total. Use WSL.vi to connect wave table to wave generator.

Waveform can be: SIN\_P, PNT, LIN, or RAMP. Check MAN? WAV answer to find out which waveforms are supported.

C-887: Affected axis is identical with wave table number. 100 waves can be defined in total. Use WSL.vi to connect wave table to wave generator.

Waveform can be: SIN\_P, PNT, LIN, or RAMP. Check MAN? WAV answer to find out which waveforms are supported..

C-891: Affected axis is identical with wave table number. 40 waves can be defined in total. Use WSL.vi to connect wave table to wave generator. Waveform can be: SIN\_P, RAMP, or LIN.

E-516: Waveform can be: SIN, TAN, POL or PNT.

The following parameters are equivalent (see GCS DLL manual):

Affected axes = szAxes, Xo = nStart, N = nLength, Add? = Add,

PNT Parameters = pPoints (PNT parameters is an array consisting of single data points which will be sent in groups of Nmax for PNT data points.)

E-517, E-518: Affected axis is identical with wave table number, which can be 1 to 3. One wave table number per command call allowed only.

Waveform can be: SIN\_P, RAMP or LIN. For SIN, TAN, POL and PNT Segment length must be zero.

E-709: Affected axis is identical with wave table number. 6 waves can be defined in total. Use WSL.vi to connect wave table to wave generator. Waveforms are automatically stored in non-volatile memory. Note that the number of write cycles is limited. See user manual for details.

Waveform can be: SIN\_P, RAMP, or LIN.

E-710: Affected axis is identical with wave table number.

Waveform can be: SIN\_P, RAMP, LIN or PNT.

The length of the wave to define must be less than the value defined with WMS.vi.

E-712: Affected axis is identical with wave table number. 90 waves can be defined in total. Use WSL.vi to connect wave table to wave generator.

Waveform can be: SIN\_P, RAMP, LIN, NOISE or PNT.

E-725: Affected axis is identical with wave table number. 30 waves can be defined in total. Use WSL.vi to connect wave table to wave generator.

Waveform can be: SIN\_P, RAMP, LIN, NOISE or PNT.

E-727: Affected axis is identical with wave table number. 40 waves can be defined in total. Use WSL.vi to connect wave table to wave generator.

Waveform can be: SIN\_P, RAMP, LIN, NOISE or PNT.

E-753: Affected axis is identical with wave table number. 10 waves can be defined in total. Use WSL.vi to connect wave table to wave generator.

Waveform can be: SIN\_P, RAMP, LIN, NOISE or PNT.

E-754: Affected axis is identical with wave table number. 10 waves can be defined in total. Use WSL.vi to connect wave table to wave generator.

Waveform can be: SIN\_P, PNT, RAMP, LIN, NOISE or SWEEP.

Valid for SWEEP: Affected axes, Append?, SWEEP Parameters. See controller user manual for details.

E-761: Affected axis is identical with wave table number.

Waveform can be: SIN\_P, RAMP, LIN, PNT or POL.

General: The following parameters are equivalent (see GCS DLL manual):

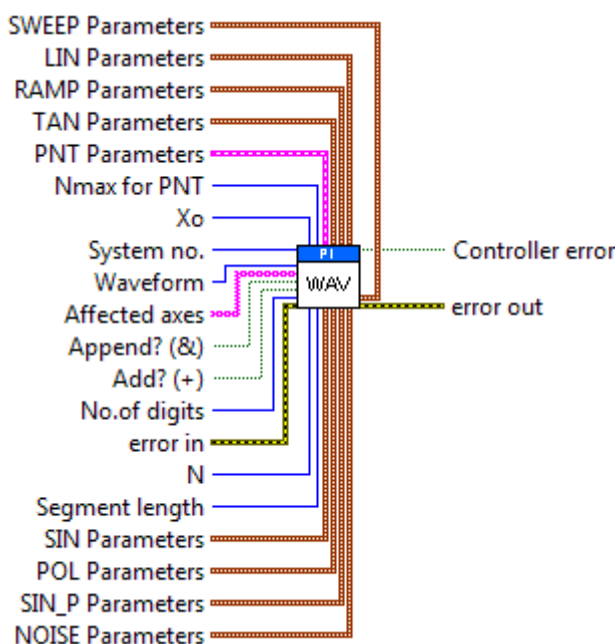
Affected axes = szWaveTableIds, Xo = iOffsetOfFirstPointInWaveTable, N = iNumberOfPoints, Append? = iAddAppendWave, C = iCenterPointOfWave, J = iNumberOfSpeedUpDownPointsInWave, A = dAmplitudeOfWave, O = dOffsetOfWave (is only valid if a wave segment is being concatenated to an existing wave, i.e. Append? = TRUE), Segment length = iSegmentLength, PNT Parameters = pdWavePoints (PNT parameters is an array consisting of single data points which will be sent in groups of Nmax for PNT data points.)

Valid for SIN: Affected axes, No. of digits, Xo, N, Add?, SIN Parameters (A, N, Xo, Phi and B)

Valid for LIN: Affected axes, No. of digits, Xo, N, Append?, Segment length, LIN Parameters (J, A and O)

Valid for TAN: Affected axes, No. of digits, Xo, N, Add?, TAN Parameters (A, N, Xo, Phi and B)

Valid for PNT: see above.

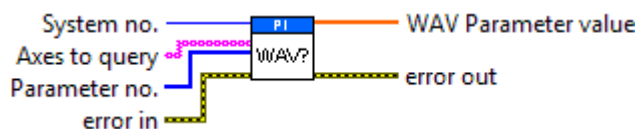


#### 4.18.25. WAV?.vi

Returns waveform parameter values for queried axes and parameter numbers.

All systems: Affected axis is identical with wave table number. The following parameter number is valid:

- 1: Number of waveform points for currently defined wave



#### 4.18.26. WAV\_old.vi

Defines or modifies a stored waveform for one or more axes. Please refer to the GCS DLL manual or to the user manual for a description of waveform parameters, units and restrictions. No. of digits is the number of digits after the decimal point in the parameter value(s) that will be sent. Controller error is TRUE if selected system reports error code not equal to 0.

E-516: Waveform can be: SIN, TAN, POL or PNT.

The following parameters are equivalent (see GCS DLL manual):

Affected axes = szAxes, Xo = nStart, N = nLength, Add? = Add,

PNT Parameters = pPoints (PNT parameters is an array consisting of single data points which will be sent in groups of Nmax for PNT data points.)

(A, N, Xo, Phi and B)

(A, N, Xo, Phi and B)

(Xo, Ao to A5)

E-517: Affected axis is identical with wave table number, which can be 1 to 3. One wave table number per command call allowed only.

Waveform can be: SIN, TAN, POL, or PNT, SIN\_P, RAMP or LIN.

The following parameters are equivalent (see GCS DLL manual):

Affected axes = szWaveTableIds, Xo = iOffsetOfFirstPointInWaveTable,

N = iNumberOfPoints, Append? = iAddAppendWave,

C = iCenterPointOfWave, J = iNumberOfSpeedUpDownPointsInWave,

A = dAmplitudeOfWave, O = dOffsetOfWave,

Segment length = iSegmentLength, PNT Parameters = pdWavePoints (PNT parameters is an array consisting of single data points which will be sent in groups of Nmax for PNT data points.)

(A, N, Xo, Phi and B). Segment length must be zero.

(A, N, Xo, Phi and B). Segment length must be zero.

(Xo, Ao to A5). Segment length must be zero.

Valid for SIN\_P: Affected axes, No. of digits, Xo, N, Append?, Segment length, SIN\_P Parameters (C, A and O)

Valid for LIN: Affected axes, No. of digits, Xo, N, Append?, Segment length, LIN Parameters (J, A and O)

E-709: Affected axis is identical with wave table number. 6 waves can be defined in total. Use WSL.vi to connect wave table to wave generator. Waveforms are automatically stored in non-volatile memory. Note that the number of write cycles is limited. See user manual for details.

Waveform can be: SIN\_P, RAMP, or LIN.

The following parameters are equivalent (see GCS DLL manual):

Affected axes = szWaveTableIds, Xo = iOffsetOfFirstPointInWaveTable,

N = iNumberOfPoints, Append? = iAddAppendWave,

C = iCenterPointOfWave, J = iNumberOfSpeedUpDownPointsInWave,

A = dAmplitudeOfWave, O = dOffsetOfWave,

Segment length = iSegmentLength

Valid for RAMP: Affected axes, No. of digits, Xo, N, Append?, Segment length, RAMP Parameters (C, J, A and O)

E-710: Waveform can be: SIN\_P, RAMP, LIN or PNT. Affected axis is identical with wave table number.

The following parameters are equivalent (see GCS DLL manual):

Affected axes =szWaveTableIds, Xo =iOffsetOfFirstPointInWaveTable, N =

iNumberOfPoints, Append? = iAddAppendWave, C =iCenterPointOfWave, J =

iNumberOfSpeedUpDownPointsInWave, A = dAmplitudeOfWave, O

=dOffsetOfWave, Segment length = iSegmentLength, PNT Parameters =

pdWavePoints (PNT parameters is an array consisting of single data points which will be sent in groups of Nmax for PNT data points.)

O (dOffsetOfWave) is only valid if a wave segment is being concatenated to an existing wave (Append? = TRUE)

Valid for RAMP: Affected axes, No. of digits, Xo, N, Append?, Segment length, RAMP Parameters (C,J,A and O)

Valid for PNT: Affected axes, No. of digits, Xo, N, Append?, Segment length, Nmax for PNT, PNT Parameters (Ao to An-1). Nmax for PNT must be less than or equal to Wave storage max. value set with WMS.vi. Segment length must be zero for PNT.

The length of the wave to define must be less than the value defined with WMS.vi

E-712: Affected axis is identical with wave table number. 90 waves can be defined in total. Use WSL.vi to connect wave table to wave generator.

Waveform can be: SIN\_P, RAMP, LIN or PNT.

The following parameters are equivalent (see GCS DLL manual):

Affected axes = szWaveTableIds, Xo = iOffsetOfFirstPointInWaveTable,

N = iNumberOfPoints, Append? = iAddAppendWave,

C = iCenterPointOfWave, J = iNumberOfSpeedUpDownPointsInWave,

A = dAmplitudeOfWave, O = dOffsetOfWave,

Segment length = iSegmentLength, PNT Parameters = pdWavePoints (PNT parameters is an array consisting of single data points which will be sent in groups of Nmax for PNT data points.)

Valid for RAMP: Affected axes, No. of digits, Xo, N, Append?, Segment length, RAMP Parameters (C, J, A and O)

Valid for PNT: Affected axes, No. of digits, Xo, N, Append?, Segment length, Nmax for PNT, PNT Parameters (Ao to An-1). Nmax for PNT must be 20 (262,144 is maximum size for all wave tables in total but input buffer size is 256 characters). Segment length must be zero for PNT.

E-725: Affected axis is identical with wave table number. 30 waves can be defined in total. Use WSL.vi to connect wave table to wave generator.

Waveform can be: SIN\_P, RAMP, LIN or PNT.

The following parameters are equivalent (see GCS DLL manual):

Affected axes = szWaveTableIds, Xo = iOffsetOfFirstPointInWaveTable,

N = iNumberOfPoints, Append? = iAddAppendWave,



C = iCenterPointOfWave, J = iNumberOfSpeedUpDownPointsInWave,

A = dAmplitudeOfWave, O = dOffsetOfWave,

Segment length = iSegmentLength, PNT Parameters = pdWavePoints (PNT parameters is an array consisting of single data points which will be sent in groups of Nmax for PNT data points.)

Valid for RAMP: Affected axes, No. of digits, Xo, N, Append?, Segment length, RAMP Parameters (C, J, A and O)

Valid for PNT: Affected axes, No. of digits, Xo, N, Append?, Segment length, Nmax for PNT, PNT Parameters (Ao to An-1). Nmax for PNT must be 20 (262,144 is maximum size for all wave tables in total but input buffer size is 256 characters). Segment length must be zero for PNT.

E-753: Affected axis is identical with wave table number. 10 waves can be defined in total. Use WSL.vi to connect wave table to wave generator.

Waveform can be: SIN\_P, RAMP, LIN or PNT.

The following parameters are equivalent (see GCS DLL manual):

Affected axes = szWaveTableIds, Xo = iOffsetOfFirstPointInWaveTable,

N = iNumberOfPoints, Append? = iAddAppendWave,

C = iCenterPointOfWave, J = iNumberOfSpeedUpDownPointsInWave,

A = dAmplitudeOfWave, O = dOffsetOfWave,

Segment length = iSegmentLength, PNT Parameters = pdWavePoints (PNT parameters is an array consisting of single data points which will be sent in groups of Nmax for PNT data points.)

Valid for RAMP: Affected axes, No. of digits, Xo, N, Append?, Segment length, RAMP Parameters (C, J, A and O)

Valid for PNT: Affected axes, No. of digits, Xo, N, Append?, Segment length, Nmax for PNT, PNT Parameters (Ao to An-1). Nmax for PNT must be less than 20 (65,536 is maximum size for all wave tables in total but input buffer size is 256). Segment length must be zero for PNT.

E-761: Waveform can be: SIN\_P, RAMP, LIN, PNT or POL. Affected axis is identical with wave table number.

The following parameters are equivalent (see GCS DLL manual):

Affected axes = szWaveTableIds, Xo = iOffsetOfFirstPointInWaveTable, N = iNumberOfPoints, Append? = iAddAppendWave,

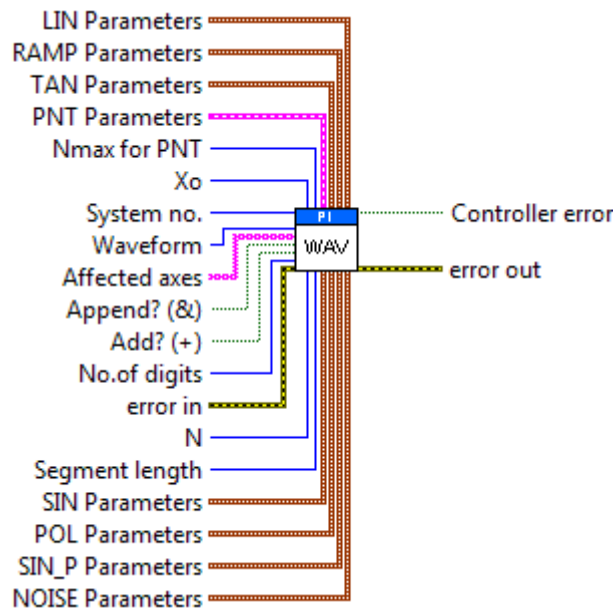
C = iCenterPointOfWave, J = iNumberOfSpeedUpDownPointsInWave, A = dAmplitudeOfWave, O = dOffsetOfWave, Segment length = iSegmentLength, PNT Parameters = pdWavePoints (PNT parameters is an array consisting of single data points which will be sent in groups of Nmax for PNT data points.)

Valid for RAMP: Affected axes, No. of digits, Xo, N, Append?, Segment length, RAMP Parameters (C,J,A and O)

Valid for PNT: Affected axes, No. of digits, Xo, N, Append?, Segment length, Nmax for PNT, PNT Parameters (Ao to An-1). Nmax for PNT must be less than or equal to Wave storage max. value set with WMS.vi and can be 50 maximum. Segment length must be zero for PNT.

(Xo, Ao to A5)





#### 4.18.27. WCL.vi

Clears waveform associated with specified axis.

C-413: WCL axes is identical with wave table number which can be 1 to 8. All axes? must be FALSE.

C-886: WCL axes is identical with wave table number which can be 1 to 100. All axes? must be FALSE.

C-887: WCL axes is identical with wave table number which can be 1 to 100. All axes? must be FALSE.

E-517, E-518: WCL axes is identical with wave table number which can be 1 to 3. All axes? must be FALSE.

E-709: WCL axes is identical with wave table number which can be 1 to 6. All axes? must be FALSE.

E-710: Does also clear DDL table. All axes must be FALSE, only one axis per command allowed.

E-712: WCL axes is identical with wave table number which can be 1 to 90. All axes? must be FALSE.

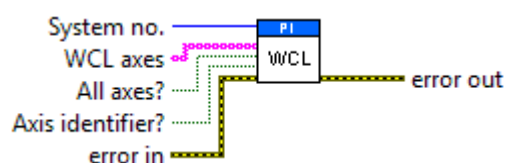
E-725: WCL axes is identical with wave table number which can be 1 to 30. All axes? must be FALSE.

E-727: WCL axes is identical with wave table number which can be 1 to 40. All axes? must be FALSE.

E-753: WCL axes is identical with wave table number which can be 1 to 10. All axes? must be FALSE.

E-754: WCL axes is identical with wave table number which can be 1 to 10. All axes? must be FALSE.

E-761: All axes must be FALSE.



**4.18.28. WGC.vi**

Sets number of wave generator cycles. Controller error is TRUE if selected system reports an error code which is not zero.

C-413: Affected axes is identical with wave generator ID, which can be 1 to 2. If Cycles = 0 the wave is performed periodically until it is stopped otherwise.

C-886: Affected axes is identical with wave generator ID, which can be 1 to No. of connected axes. If Cycles = 0 the wave is performed periodically until it is stopped otherwise. Cycles set for any wave generator ID are valid for all wave generators (cannot be set differently).

C-887: Affected axes is identical with wave generator ID, which can be 1 to No. of connected axes. If Cycles = 0 the wave is performed periodically until it is stopped otherwise. Cycles set for any wave generator ID are valid for all wave generators (cannot be set differently).

C-891: Affected axes is identical with wave generator ID, which can be 1. If Cycles = 0 the wave is performed periodically until it is stopped otherwise.

E-517, E-518: Affected axes is identical with wave generator ID, which can be 1 to 3. If Cycles = 0 the wave is performed periodically until it is stopped otherwise.

WGC saves the new value in RAM only. The WPA command saves the currently valid value to non-volatile memory, where it becomes the power-on default.

Settings not saved with WPA will be lost on power down or reboot.

E-709: Affected axes is identical with wave generator ID, which can be 1. If Cycles = 0 the wave is performed periodically until it is stopped otherwise.

E-712: Affected axes is identical with wave generator ID, which can be 1 to No. of connected axes. If Cycles = 0 the wave is performed periodically until it is stopped otherwise. Cycles set for any wave generator ID are valid for all wave generators (cannot be set differently).

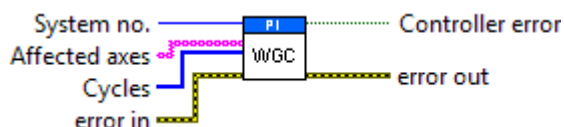
E-725, E-727: Affected axes is identical with wave generator ID, which can be 1 to No. of connected axes. If Cycles = 0 the wave is performed periodically until it is stopped otherwise. Cycles set for any wave generator ID are valid for all wave generators (cannot be set differently).

E-753: Affected axes is identical with wave generator ID, which can be 1. If Cycles = 0 the wave is performed periodically until it is stopped otherwise.

E-754: Affected axes is identical with wave generator ID, which can be 1. If Cycles = 0 the wave is performed periodically until it is stopped otherwise.

E-761: Affected axes is identical with wave table number, which can be 1 to 4. If Cycles = 0 the wave is performed periodically until it is stopped otherwise. E-761: WGC saves the new value in RAM only. The WPA command saves the currently valid value to non-volatile memory, where it becomes the power-on default.

Settings not saved with WPA will be lost when the PC is powered off or the E-761 is rebooted.

**4.18.29. WGC?.vi**

Returns number of wave generator cycles set for specified axes.

C-413: Axes to query is identical with wave generator ID, which can be 1 to 2. All axes? = must be FALSE.

C-886: Axes to query is identical with wave generator ID, which can be 1 to No. of connected axes. All axes? = must be FALSE.

C-887: Axes to query is identical with wave generator ID, which can be 1 to No. of connected axes. All axes? = must be FALSE.

C-891: Axes to query is identical with wave generator ID, which can be 1. All axes? = must be FALSE.

E-516: If All axes = TRUE, then Axis identifier must be TRUE

E-517, E-518: Axes to query is identical with wave generator ID, which can be 1 to 3. All axes? must be FALSE.

E-709: Axes to query is identical with wave generator ID, which can be 1. All axes? must be FALSE.

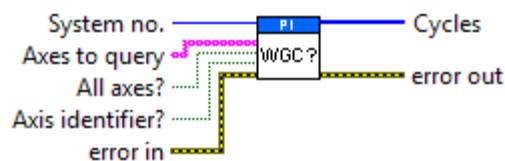
E-712: Axes to query is identical with wave generator ID, which can be 1 to No. of connected axes. All axes? must be FALSE.

E-725, E-727: Axes to query is identical with wave generator ID, which can be 1 to No. of connected axes. All axes? must be FALSE.

E-753: Axes to query is identical with wave generator ID, which can be 1. All axes? must be FALSE.

E-754: Axes to query is identical with wave generator ID, which can be 1. All axes? must be FALSE.

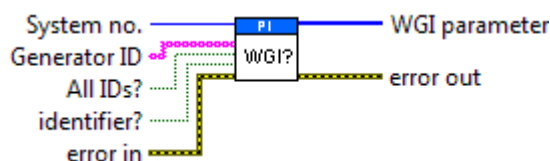
E-761: Axes to query is identical with wave table number, which can be 1 to 4. All axes must be FALSE



#### 4.18.30. WGI?.vi

Returns the index of the currently output wave point.

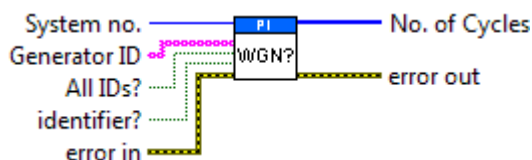
E-709: Generator ID can be 1. If All IDs? = TRUE, then identifier? must be FALSE.



#### 4.18.31. WGN?.vi

Returns the number of finished wave cycles since last WGO. Only starting a wave generator by sending the corresponding WGO command resets the counter to 0. The counter is halted (but not reset) if the wave generator is stopped by sending the corresponding WGO or #24 command or by a hardware trigger. A counter overflow does not set any error. The counter overflow value depends on the hardware, for details see user manual.

E-709: If All IDs? = TRUE, Identifier must be FALSE.



#### 4.18.32. WGO.vi

Enables, disables, and sets wave generator output mode. Parameter is bit-mapped. Controller error is TRUE if selected system reports an error code which is not zero.

C-413: Affected axis is identical with wave generator number, which can be 1 to 2. See C-413 user manual for a description of valid parameters. Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (4096 in total). Starting the wave generator is not allowed when the analog input is used for target generation.

C-886: Affected axis is identical with wave generator number, which can be 1 to No. of connected axes. See controller user manual for a description of valid parameters. Recording ends when the record table content has reached the maximum number of points (1 Million in total). Starting the wave generator is not allowed when axes are moving.

C-887: Affected axis is identical with wave generator number, which can be 1 to No. of connected axes. See controller user manual for a description of valid parameters. Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (4 Million in total). Starting the wave generator is not allowed when axes are moving.

C-891: Affected axis is identical with wave generator number, which can be 1. See controller user manual for a description of valid parameters. Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (16384 in total).

E-516: See user manual for parameter definition.

E-517, E-518: Affected axis is identical with wave generator number, which can be 1 to 3. See controller user manual for a description of valid parameters. Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (8,192 in total).

E-709: Affected axis is identical with wave generator number, which can be 1. See E-709 user manual for a description of valid parameters. Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (4096 in total). Starting the wave generator is not allowed when the analog input is used for target generation.

E-710: Affected axis is identical with wave generator number. Only two wave generators can run simultaneously. See E7XX\_GCS\_DLL Manual for parameter definition.

E-712: Affected axis is identical with wave generator number, which can be 1 to No. of connected axes. See E-712 user manual for a description of valid parameters. Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (262,144 in total). Starting the wave generator is not allowed when the analog input is used for target generation.

E-725: Affected axis is identical with wave generator number, which can be 1 to 3. See E-725 user manual for a description of valid parameters. Use DRC to configure data recording. Recording ends when the record table content has

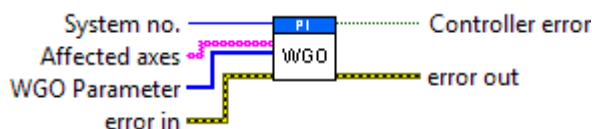
reached the maximum number of points (262,144 in total). Starting the wave generator is not allowed when the analog input is used for target generation.

E-727: Affected axis is identical with wave generator number, which can be 1 to 4. See E-727 user manual for a description of valid parameters. Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (262,144 in total). Starting the wave generator is not allowed when the analog input is used for target generation.

E-753: Affected axis is identical with wave generator number, which can be 1. See E-753 user manual for a description of valid parameters. Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (65,536 in total). Starting the wave generator is not allowed when the analog input is used for target generation.

E-754: Affected axis is identical with wave generator number, which can be 1. See controller user manual for a description of valid parameters. Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (1,048,576 in total). Starting the wave generator is not allowed when the analog input is used for target generation.

E-761: Affected axis is identical with wave generator number, which can be 1 to 4. Four wave generators can run simultaneously. See E7XX\_GCS\_DLL Manual for parameter definition. Each time the wave generator is started recording starts automatically as follows: recorder table 1: axis 1 actual position, recorder table 2: axis 2 actual position, recorder table 3: axis 3 actual position, recorder table 4: analog input voltage (same value as read with TAV?.vi, i.e. contains gain and offset for the analog input, see E-761 user manual). Recording ends when the record table content has reached the maximum number of points (8192 per table).



#### 4.18.33. WGO?.vi

Returns WGO parameter for specified axes.

C-413: Axis to query is identical with wave generator number, which can be 1 to 2. If All axes? = TRUE, then Axis identifier? must be FALSE.

C-886: Axis to query is identical with wave generator number, which can be 1 to No. of connected axes. If All axes? = TRUE, then Axis identifier? must be FALSE.

C-887: Axis to query is identical with wave generator number, which can be 1 to No. of connected axes. If All axes? = TRUE, then Axis identifier? must be FALSE.

C-891: Axis to query is identical with wave generator number, which can be 1. If All axes? = TRUE, then Axis identifier? must be FALSE.

E-516: If All axes = TRUE, then Axis identifier must be FALSE

E-517, E-518: Axis to query is identical with wave generator number, which can be 1 to 3. If All axes? = TRUE, then Axis identifier? must be FALSE.

E-709: Axis to query is identical with wave generator number, which can be 1. If All axes? = TRUE, then Axis identifier? must be FALSE.

E-710: Affected axis is identical with wave generator number. If All axes = TRUE, then Axis identifier must be FALSE

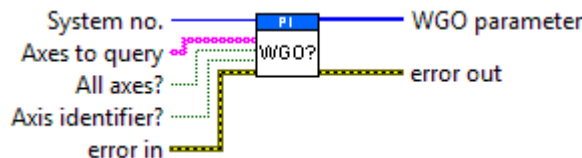
E-712: Axis to query is identical with wave generator number, which can be 1 to No. of connected axes. If All axes? = TRUE, then Axis identifier? must be FALSE.

E-725, E-727: Axis to query is identical with wave generator number, which can be 1 to No. of connected axes. If All axes? = TRUE, then Axis identifier? must be FALSE.

E-753: Axis to query is identical with wave generator number, which can be 1. If All axes? = TRUE, then Axis identifier? must be FALSE.

E-754: Axis to query is identical with wave generator number, which can be 1. If All axes? = TRUE, then Axis identifier? must be FALSE.

E-761: Affected axis is identical with wave generator number, which can be 1 to 4. All axes must be FALSE



#### 4.18.34. WGR.vi

Starts a new recording.

C-413: Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (4096 in total).

C-886: Recording ends when the record table content has reached the maximum number of points (8192 per table).

C-887: Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (4 Million in total).

C-891: Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (16384 in total).

E-517, E-518: Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (8,192 in total). If more than one wave generator is running, recording starts at the waveform start point which occurs first.

E-709: Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (4096 in total).

E-712: Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (262,144 in total).

E-725, E-727: Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (262,144 in total).

E-753: Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (65,536 in total).

E-754: Use DRC to configure data recording. Recording ends when the record table content has reached the maximum number of points (1,048,576 in total).

E-761: Recording will be started for all recorder tables when the wave generator is running for an arbitrary axis. The assignment of axis and data sources to the recorder tables is as follows:

table 1: axis 1 actual position

table 2: axis 2 actual position

table 3: axis 3 actual position

table 4: analog input voltage (same value as read with TAV?.vi, i.e. contains gain and offset for the analog input, see E-761 user manual). Recording starts always with the next start point of the waveform, i.e. there might be a short delay between sending WGR and the start of the record. If more than one wave generator is



running, recording starts at the waveform start point which occurs first. Recording ends when the record table content has reached the maximum number of points (8192 per table).

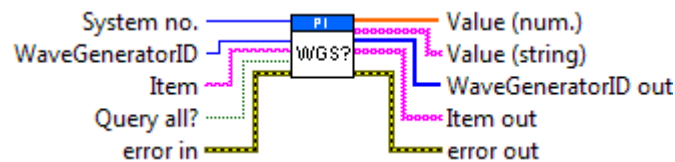


#### 4.18.35. WGS?.vi

Get status information of wave generator.

If Query all? = FALSE, a valid wave generator ID has to be specified.

Item is only taken into account if Query all? = FALSE and a valid wave generator ID has been specified. If Item is an empty string, all items of the given wave generator ID are queried.



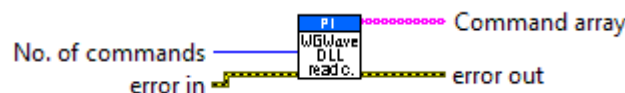
#### 4.18.36. WGWaveEditor\_call\_Editor.vi

This VI calls a wave editor dialog. WGWaveEditor.dll must be installed before running this VI. If Filename, Parameter set name or Affected axes are empty strings, axes, parameter file or set can be chosen in the editor directly. No. of commands is the number of commands available to be received with WGWaveEditor\_read\_commands.vi.



#### 4.18.37. WGWaveEditor\_read\_commands.vi

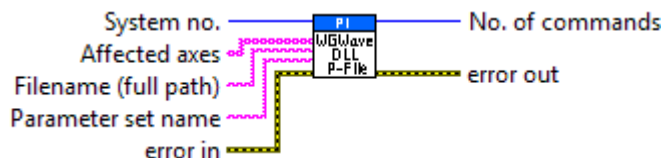
This VI returns an array of commands generated by WGWaveEditor.dll, which must be installed before running this VI. Use the return value from WGWaveEditor\_with\_parameter.vi or WGWaveEditor\_call\_Editor.vi for No. of commands.



#### 4.18.38. WGWaveEditor\_with\_parameter\_file.vi

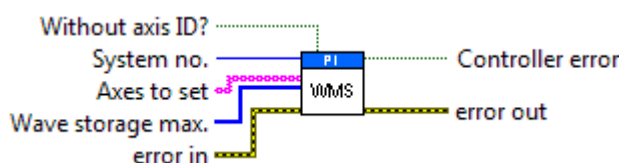
This VI starts the wave editor with a given file name and parameter set name. WGWaveEditor.dll must be installed before running this VI. If Filename or Parameter set name is an empty string, a wave editor dialog window is opened. No. of commands is the number of commands available to be received with WGWaveEditor\_read\_commands.vi.





#### 4.18.39. WMS.vi

Sets maximum size of wave-point storage available for given axes and checks for error. If Without axis ID? is TRUE, then Axes to set is ignored and first field of Wave storage max. array is used as number of wave points for every axis. Controller error is TRUE if selected system reports error code which is not 0.  
E-710: Without axis ID? = FALSE. A maximum number of 63488 samples can be defined for all axes in total.



#### 4.18.40. WMS? vi

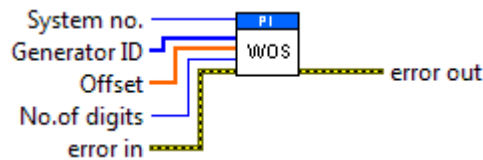
Returns maximum size of wave-point storage for specified axes.  
C-886: Axes to query are identical with wave table numbers, which can be 1 to 100. All axes? must be FALSE.  
C-887: Axes to query are identical with wave table numbers, which can be 1 to 100. All axes? must be FALSE.  
E-517, E-518: Axes to query are identical with wave table numbers, which can be 1 to 3. All axes must be FALSE.  
E-710: If All axes = TRUE, then Axis identifier must be TRUE  
E-761: All axes must be FALSE. Axis to query is identical with wave table number, which can be 1 to 4.



#### 4.18.41. WOS.vi

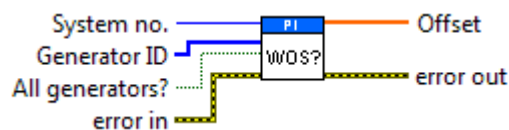
Sets wave generator output offset. The current wave generator output is created by adding the output offset value to the current wave value:  
Generator Output = Output Offset + Current Wave Value.  
The value is also modified by the generator when the WGO command was executed with iStartMod bit 8 set: At the end of each cycle the output offset value is equate with the current Generator Output. No. of digits is the number of digits after the decimal point in the offset value(s) that will be sent.  
C-413: Generator ID can be 1 to 2.  
E-517, E-518: Generator ID can be 1 to 3. You can set the offset also using the CFG wave type (see WAV command) or by directly changing the Wave Offset parameter, ID 0x1300010b, with SPA.

- E-709: Generator ID can be 1
- E-712: Generator ID can be 1 to No. of connected axes
- E-725, E-727: Generator ID can be 1 to No. of connected axes
- E-753: Generator ID can be 1
- E-754: Generator ID can be 1



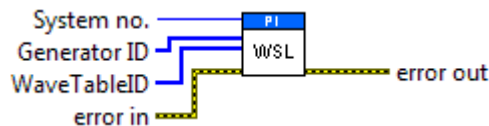
#### 4.18.42. WOS?.vi

- Returns wave generator output offset.
- C-413: Generator ID can be 1 to 2.
- E-517, E-518: Generator ID can be 1 to 3.
- E-709: Generator ID can be 1
- E-712: Generator ID can be 1 to No. of connected axes
- E-725, E-727: Generator ID can be 1 to No. of connected axes
- E-753: Generator ID can be 1
- E-754: Generator ID can be 1



#### 4.18.43. WSL.vi

- Connects a wave table to a generator or disconnects the selected generator from any wave table.
- C-413: Generator ID can be 1 to 2, WaveTableID can be 1 to 8.
- C-886: Generator ID can be 1 to No. of connected axes, WaveTableID can be 1 to 100.
- C-887: Generator ID can be 1 to No. of connected axes, WaveTableID can be 1 to 100.
- C-891: Generator ID can be 1, WaveTableID can be 1 to 2.
- E-709: Generator ID can be 1, WaveTableID can be 1 to 6.
- E-517: Generator ID can be 1 to 3. WaveTableID can be 1 to 3.
- E-518: Generator ID can be 1 to 3. WaveTableID can be 1 to 3.
- E-712: Generator ID can be 1 to No. of connected axes, WaveTableID can be 1 to 90
- E-725: Generator ID can be 1 to No. of connected axes, WaveTableID can be 1 to 30
- E-727: Generator ID can be 1 to No. of connected axes, WaveTableID can be 1 to 40
- E-753: Generator ID can be 1, WaveTableID can be 1 to 10
- E-754: Generator ID can be 1, WaveTableID can be 1 to 10



#### 4.18.44. WSL?.vi

Returns current setting of wave table selection. If WaveTableID is zero no wave table is connected to the generator.

C-413: Generator ID can be 1 to 2.

C-886: Generator ID can be 1 to No. of connected axes.

C-887: Generator ID can be 1 to No. of connected axes.

C-891: Generator ID can be 1.

E-709: Generator ID can be 1

E-517: Generator ID can be 1 to 3.

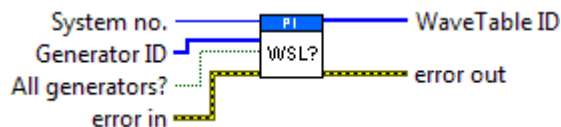
E-51: Generator ID can be 1 to 3.

E-712: Generator ID can be 1 to No. of connected axes

E-725, E-727: Generator ID can be 1 to No. of connected axes

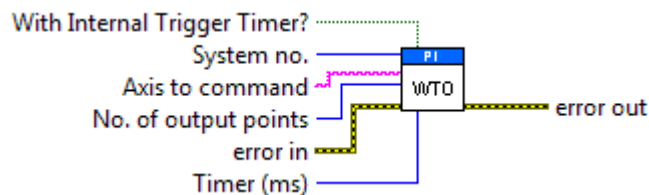
E-753: Generator ID can be 1

E-754: Generator ID can be 1



#### 4.18.45. WTO.vi

Sets the wave table output. If No. of output points = 0, wave table output is disabled, otherwise No. of output points will be sent to the given axis. No. of output points must be less than or equal to 64. If With Internal Trigger Timer? is TRUE and Timer (ms) is greater than zero, the internal trigger is used additionally to the external trigger line (if any), but the external trigger resets the internal trigger timer.



#### 4.18.46. WTR.vi

Set wave generator table rate. This VI will change the sampling interval of the wave generator. When Generator ID is zero all generators are selected, otherwise each generator is selected individually. Table Rate sets the duration of the wave table points as a multiple of the servo interval time. When a Table rate higher than 1 is set, Interpolation type can be used to join each wave table point by an interpolation.

C-413 : The following Interpolation types can be selected:

0 = no interpolation

Generator ID must be 0 because values can only be set for all generators.

C-886: The following Interpolation types can be selected:

0 = no interpolation, 1 = straight line

Table rate values and Interpolation type for any Generator ID are valid for all wave generators (cannot be set differently).

C-887: The following Interpolation types can be selected:

0 = no interpolation, 1 = straight line

Table rate values and Interpolation type for any Generator ID are valid for all wave generators (cannot be set differently).

C-891: The following Interpolation types can be selected:

0 = no interpolation, 1 = straight line.

Generator ID can be 0 or 1 as there is only one generator.

E-517, E-518: Interpolation type must be 0. Generator ID can be 1 to 3. WTR sets the value of the Wave Generator Table Rate parameter, ID 0x13000109, in volatile memory. You can set the wave table rate also using the CFG wave type (see WAV command) or by directly changing the parameter with SPA.

E-709: The following Interpolation types can be selected:

0 = no interpolation

Generator ID can be 0 or 1 as there is only one generator.

E-712: The following Interpolation types can be selected:

0 = no interpolation, 1 = straight line

Generator ID must be 0 because values can only be set for all generators.

E-725, E-727: The following Interpolation types can be selected:

0 = no interpolation, 1 = straight line

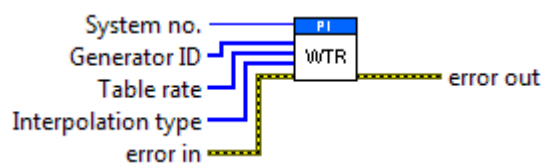
Generator ID must be 0 because values can only be set for all generators.

E-753: The following Interpolation types can be selected:

0 = no interpolation, 1 = straight line. Generator ID must be 0 because values can only be set for all generators.

E-754: The following Interpolation types can be selected:

0 = no interpolation, 1 = straight line. Generator ID must be 0 because values can only be set for all generators.



#### 4.18.47. WTR?.vi

Returns the current settings of the wave generator table rate.

C-413 : Generator ID can be 1 to 2

C-886: Values queried for one Generator ID are valid for all generators.

C-887: Values queried for one Generator ID are valid for all generators.

C-891: Generator ID can be 1.

E-517, E-518: Generator ID can be 1 to 3.

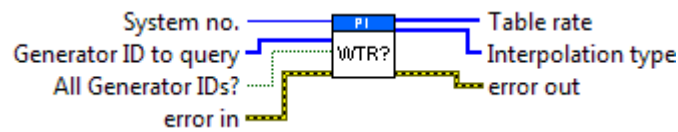
E-709: Generator ID can be 0 or 1 because there is only one generator.

E-712: Generator ID must be 0 because values are valid for all generators.

E-725, E-727: Generator ID must be 0 because values are valid for all generators.

E-753: Generator ID must be 0 because values are valid for all generators.

E-754: Generator ID must be 0 because values are valid for all generators.



## 5. Error Codes

Error codes can result from a PI error message or LabVIEW internal error code. Therefore, see the National Instruments error codes in addition to the lists below.

### Controller errors

0	PI_CNTR_NO_ERROR	No error
1	PI_CNTR_PARAM_SYNTAX	Parameter syntax error
2	PI_CNTR_UNKNOWN_COMMAND	Unknown command
3	PI_CNTR_COMMAND_TOO_LONG	Command length out of limits or command buffer overrun
4	PI_CNTR_SCAN_ERROR	Error while scanning
5	PI_CNTR_MOVE_WITHOUT_REF_OR_NO_SERVO	Unallowable move attempted on unreferenced axis, or move attempted with servo off
6	PI_CNTR_INVALID_SGA_PARAM	Parameter for SGA not valid
7	PI_CNTR_POS_OUT_OF_LIMITS	Position out of limits
8	PI_CNTR_VEL_OUT_OF_LIMITS	Velocity out of limits
9	PI_CNTR_SET_PIVOT_NOT_POSSIBLE	Attempt to set pivot point while U,V and W not all 0
10	PI_CNTR_STOP	Controller was stopped by command
11	PI_CNTR_SST_OR_SCAN_RANGE	Parameter for SST or for one of the embedded scan algorithms out of range
12	PI_CNTR_INVALID_SCAN_AXES	Invalid axis combination for fast scan
13	PI_CNTR_INVALID_NAV_PARAM	Parameter for NAV out of range
14	PI_CNTR_INVALID_ANALOG_INPUT	Invalid analog channel
15	PI_CNTR_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
16	PI_CNTR_INVALID_STAGE_NAME	Invalid stage name
17	PI_CNTR_PARAM_OUT_OF_RANGE	Parameter out of range

18	PI_CNTR_INVALID_MACRO_NAME	Invalid macro name
19	PI_CNTR_MACRO_RECORD	Error while recording macro
20	PI_CNTR_MACRO_NOT_FOUND	Macro not found
21	PI_CNTR_AXIS_HAS_NO_BRAKE	Axis has no brake
22	PI_CNTR_DOUBLE_AXIS	Axis identifier specified more than once
23	PI_CNTR_ILLEGAL_AXIS	Illegal axis or channel
24	PI_CNTR_PARAM_NR	Incorrect number of parameters
25	PI_CNTR_INVALID_REAL_NR	Invalid floating point number
26	PI_CNTR_MISSING_PARAM	Parameter missing
27	PI_CNTR_SOFT_LIMIT_OUT_OF_RANGE	Soft limit out of range
28	PI_CNTR_NO_MANUAL_PAD	No manual pad found
29	PI_CNTR_NO_JUMP	No more step-response values
30	PI_CNTR_INVALID_JUMP	No step-response values recorded
31	PI_CNTR_AXIS_HAS_NO_REFERENCE	Axis has no reference sensor
32	PI_CNTR_STAGE_HAS_NO_LIM_SWITCH	Axis has no limit switch
33	PI_CNTR_NO_RELAY_CARD	No relay card installed
34	PI_CNTR_CMD_NOT_ALLOWED_FOR_STAGE	Command not allowed for selected stage(s)
35	PI_CNTR_NO_DIGITAL_INPUT	No digital input installed
36	PI_CNTR_NO_DIGITAL_OUTPUT	No digital output configured
37	PI_CNTR_NO_MCM	No more MCM responses
38	PI_CNTR_INVALID_MCM	No MCM values recorded
39	PI_CNTR_INVALID_CNTR_NUMBER	Controller number invalid
40	PI_CNTR_NO_JOYSTICK_CONNECTED	No joystick configured
41	PI_CNTR_INVALID_EGE_AXIS	Invalid axis for electronic gearing, axis can not be slave
42	PI_CNTR_SLAVE_POSITION_OUT_OF_RANGE	Position of slave axis is out of range
43	PI_CNTR_COMMAND_EGE_SLAVE	Slave axis cannot be commanded directly when electronic gearing is enabled
44	PI_CNTR_JOYSTICK_CALIBRATION_FAILED	Calibration of joystick failed
45	PI_CNTR_REFERENCING_FAILED	Referencing failed
46	PI_CNTR_OPM_MISSING	OPM (Optical Power Meter) missing
47	PI_CNTR_OPM_NOT_INITIALIZED	OPM (Optical Power Meter) not

		initialized or cannot be initialized
48	PI_CNTR_OPM_COM_ERROR	OPM (Optical Power Meter) Communication Error
49	PI_CNTR_MOVE_TO_LIMIT_SWITCH_FAILED	Move to limit switch failed
50	PI_CNTR_REF_WITH_REF_DISABLED	Attempt to reference axis with referencing disabled
51	PI_CNTR_AXIS_UNDER_JOYSTICK_CONTROL	Selected axis is controlled by joystick
52	PI_CNTR_COMMUNICATION_ERROR	Controller detected communication error
53	PI_CNTR_DYNAMIC_MOVE_IN_PROCESS	Command is not allowed while the affected axis is in motion.
54	PI_CNTR_UNKNOWN_PARAMETER	Unknown parameter
55	PI_CNTR_NO_REP_RECORDED	No commands were recorded with REP
56	PI_CNTR_INVALID_PASSWORD	Password invalid
57	PI_CNTR_INVALID_RECORDER_CHAN	Data recorder table does not exist
58	PI_CNTR_INVALID_RECORDER_SRC_OPT	Source option does not exist; number too low or too high
59	PI_CNTR_INVALID_RECORDER_SRC_CHAN	Source ID (channel or axis) too low or too high
60	PI_CNTR_PARAM_PROTECTION	Protected Param: current Command Level (CCL) too low
61	PI_CNTR_AUTOZERO_RUNNING	Command execution not possible while Autozero is running
62	PI_CNTR_NO_LINEAR_AXIS	Autozero requires at least one linear axis
63	PI_CNTR_INIT_RUNNING	Initialization still in progress
64	PI_CNTR_READ_ONLY_PARAMETER	Parameter is read-only
65	PI_CNTR_PAM_NOT_FOUND	Parameter not found in non- volatile memory
66	PI_CNTR_VOL_OUT_OF_LIMITS	Voltage out of limits
67	PI_CNTR_WAVE_TOO_LARGE	Not enough memory available for requested wave curve
68	PI_CNTR_NOT_ENOUGH_DDL_MEMORY	Not enough memory available for DDL table; DDL can not be started
69	PI_CNTR_DDL_TIME_DELAY_TOO_LARGE	Time delay larger than DDL table; DDL can not be started
70	PI_CNTR_DIFFERENT_ARRAY_LENGTH	The requested arrays have different lengths; query them separately
71	PI_CNTR_GEN_SINGLE_MODE_RESTART	Attempt to restart the generator while it is running in single step



		mode
72	PI_CNTR_ANALOG_TARGET_ACTIVE	Motion commands and wave generator activation are not allowed when analog target is active
73	PI_CNTR_WAVE_GENERATOR_ACTIVE	Motion commands are not allowed when wave generator output is active; use WGO to disable generator output
74	PI_CNTR_AUTOZERO_DISABLED	No sensor channel or no piezo channel connected to selected axis (sensor and piezo matrix)
75	PI_CNTR_NO_WAVE_SELECTED	Generator started (WGO) without having selected a wave table (WSL).
76	PI_CNTR_IF_BUFFER_OVERRUN	Interface buffer did overrun and command couldn't be received correctly
77	PI_CNTR_NOT_ENOUGH_RECORDED_DATA	Data Record Table does not hold enough recorded data
78	PI_CNTR_TABLE_DEACTIVATED	Data Record Table is not configured for recording
79	PI_CNTR_OPENLOOP_VALUE_SET_WHEN_SERVO_ON	Open-loop commands (SVA, SVR) are not allowed when servo is on
80	PI_CNTR_RAM_ERROR	Hardware error affecting RAM
81	PI_CNTR_MACRO_UNKNOWN_COMMAND	Not macro command
82	PI_CNTR_MACRO_PC_ERROR	Macro counter out of range
83	PI_CNTR_JOYSTICK_ACTIVE	Joystick is active
84	PI_CNTR_MOTOR_IS_OFF	Motor is off
85	PI_CNTR_ONLY_IN_MACRO	Macro-only command
86	PI_CNTR_JOYSTICK_UNKNOWN_AXIS	Invalid joystick axis
87	PI_CNTR_JOYSTICK_UNKNOWN_ID	Joystick unknown
88	PI_CNTR_REF_MODE_IS_ON	Move without referenced stage
89	PI_CNTR_NOT_ALLOWED_IN_CURRENT_MOTION_MODE	Command not allowed in current motion mode
90	PI_CNTR_DIO_AND_TRACING_NOT_POSSIBLE	No tracing possible while digital IOs are used on this HW revision. Reconnect to switch operation mode.
91	PI_CNTR_COLLISION	Move not possible, would cause collision
92	PI_CNTR_SLAVE_NOT_FAST_ENOUGH	Stage is not capable of following the master. Check the gear ratio(SRA).

93	PI_CNTR_CMD_NOT_ALLOWED_WHILE_AXIS_IN_MOTION	This command is not allowed while the affected axis or its master is in motion.
94	PI_CNTR_OPEN_LOOP_JOYSTICK_ENABLED	Servo cannot be switched on when open-loop joystick control is enabled.
95	PI_CNTR_INVALID_SERVO_STATE_FOR_PARAMETER	This parameter cannot be changed in current servo mode.
96	PI_CNTR_UNKNOWN_STAGE_NAME	Unknown stage name
97	PI_CNTR_INVALID_VALUE_LENGTH	Invalid length of value (too much characters)
98	PI_CNTR_AUTOZERO_FAILED	AutoZero procedure was not successful
99	PI_CNTR_SENSOR_VOLTAGE_OFF	Sensor voltage is off
100	PI_LABVIEW_ERROR	PI LabVIEW driver reports error. See source control for details.
200	PI_CNTR_NO_AXIS	No stage connected to axis
201	PI_CNTR_NO_AXIS_PARAM_FILE	File with axis parameters not found
202	PI_CNTR_INVALID_AXIS_PARAM_FILE	Invalid axis parameter file
203	PI_CNTR_NO_AXIS_PARAM_BACKUP	Backup file with axis parameters not found
204	PI_CNTR_RESERVED_204	PI internal error code 204
205	PI_CNTR_SMO_WITH_SERVO_ON	SMO with servo on
206	PI_CNTR_UUDECODE_INCOMPLETE_HEADER	uudecode: incomplete header
207	PI_CNTR_UUDECODE_NOTHING_TO_DECODE	uudecode: nothing to decode
208	PI_CNTR_UUDECODE_ILLEGAL_FORMAT	uudecode: illegal UUE format
209	PI_CNTR_CRC32_ERROR	CRC32 error
210	PI_CNTR_ILLEGAL_FILENAME	Illegal file name (must be 8-0 format)
211	PI_CNTR_FILE_NOT_FOUND	File not found on controller
212	PI_CNTR_FILE_WRITE_ERROR	Error writing file on controller
213	PI_CNTR_DTR_HINDERS_VELOCITY_CHANGE	VEL command not allowed in DTR Command Mode
214	PI_CNTR_POSITION_UNKNOWN	Position calculations failed
215	PI_CNTR_CONN_POSSIBLY_BROKEN	The connection between controller and stage may be broken
216	PI_CNTR_ON_LIMIT_SWITCH	The connected stage has driven into a limit switch, some controllers need CLR to resume operation
217	PI_CNTR_UNEXPECTED_STRUT_STOP	Strut test command failed because of an unexpected strut stop

218	PI_CNTR_POSITION_BASED_ON_ESTIMATION	While MOV! is running position can only be estimated!
219	PI_CNTR_POSITION_BASED_ON_INTERPOLATION	Position was calculated during MOV motion
220	PI_CNTR_INTERPOLATION_FIFO_UNDERRUN	FIFO buffer underrun during interpolation
221	PI_CNTR_INTERPOLATION_FIFO_OVERFLOW	FIFO buffer overflow during interpolation
230	PI_CNTR_INVALID_HANDLE	Invalid handle
231	PI_CNTR_NO_BIOS_FOUND	No bios found
232	PI_CNTR_SAVE_SYS_CFG_FAILED	Save system configuration failed
233	PI_CNTR_LOAD_SYS_CFG_FAILED	Load system configuration failed
301	PI_CNTR_SEND_BUFFER_OVERFLOW	Send buffer overflow
302	PI_CNTR_VOLTAGE_OUT_OF_LIMITS	Voltage out of limits
303	PI_CNTR_OPEN_LOOP_MOTION_SET_WHEN_SERVO_ON	Open-loop motion attempted when servo ON
304	PI_CNTR_RECEIVING_BUFFER_OVERFLOW	Received command is too long
305	PI_CNTR_EEPROM_ERROR	Error while reading/writing EEPROM
306	PI_CNTR_I2C_ERROR	Error on I2C bus
307	PI_CNTR_RECEIVING_TIMEOUT	Timeout while receiving command
308	PI_CNTR_TIMEOUT	A lengthy operation has not finished in the expected time
309	PI_CNTR_MACRO_OUT_OF_SPACE	Insufficient space to store macro
310	PI_CNTR_EUI_OLDVERSION_CFGDATA	Configuration data has old version number
311	PI_CNTR_EUI_INVALID_CFGDATA	Invalid configuration data
333	PI_CNTR_HARDWARE_ERROR	Internal hardware error
400	PI_CNTR_WAV_INDEX_ERROR	Wave generator index error
401	PI_CNTR_WAV_NOT_DEFINED	Wave table not defined
402	PI_CNTR_WAV_TYPE_NOT_SUPPORTED	Wave type not supported
403	PI_CNTR_WAV_LENGTH_EXCEEDS_LIMIT	Wave length exceeds limit
404	PI_CNTR_WAV_PARAMETER_NR	Wave parameter number error
405	PI_CNTR_WAV_PARAMETER_OUT_OF_LIMIT	Wave parameter out of range
406	PI_CNTR_WGO_BIT_NOT_SUPPORTED	WGO command bit not supported
500	PI_CNTR_EMERGENCY_STOP_BUTTON_ACTIVATED	The \"red knob\" is still set and disables system
501	PI_CNTR_EMERGENCY_STOP_BUTTON_WAS_ACTIVATED	The \"red knob\" was activated and

	TIVATED	still disables system - reanimation required
502	PI_CNTR_REDUNDANCY_LIMIT_EXCEEDED	Position consistency check failed
503	PI_CNTR_COLLISION_SWITCH_ACTIVATED	Hardware collision sensor(s) are activated
504	PI_CNTR_FOLLOWING_ERROR	Strut following error occurred, e.g. caused by overload or encoder failure
505	PI_CNTR_SENSOR_SIGNAL_INVALID	One sensor signal is not valid
506	PI_CNTR_SERVO_LOOP_UNSTABLE	Servo loop was unstable due to wrong parameter setting and switched off to avoid damage.
507	PI_CNTR_LOST_SPI_SLAVE_CONNECTION	digital connection to external spi slave device is lost
508	PI_CNTR_MOVE_ATTEMPT_NOT_PERMITTED	Move attempt not permitted due to customer or limit settings
509	PI_CNTR_TRIGGER_EMERGENCY_STOP	Emergency stop caused by trigger input
530	PI_CNTR_CS_DOES_NOT_EXIST	A command refers to a coordinate system that does not exist
531	PI_CNTR_PARENT_CS_DOES_NOT_EXIST	A command refers to a coordinate system that has no parent node
532	PI_CNTR_CS_IN_USE	Attempt to delete or change a coordinate system that is in use
533	PI_CNTR_CS_DEFINITION_IS_CYCLIC	Definition of a coordinate system is cyclic
536	PI_CNTR_HEXAPOD_IN_MOTION	Coordinate system cannot be defined as long as Hexapod is in motion
537	PI_CNTR_CS_TYPE_CANNOT_BE_ENABLED	Coordinate system type is not intended for manual enabling
539	PI_CNTR_CS_PARENT_IDENTICAL_TO_CHILD	A coordinate system cannot be linked to itself
540	PI_CNTR_CS_DEFINITION_INCONSISTENT	Coordinate system definition is erroneous or not complete (replace or delete it)
542	PI_CNTR_CS_NOT_IN_SAME_CHAIN	The coordinate systems are not part of the same chain
543	PI_CNTR_CS_MEMORY_FULL	Unused coordinate system must be deleted before new coordinate system can be stored
544	PI_CNTR_SPI_COMMAND_NOT_SUPPORTED	With this coordinate system type SPI usage is not supported
545	PI_CNTR_SOFTLIMITS_INVALID	Soft limits invalid due to changes in coordinate system

546	PI_CNTR_CS_WRITE_PROTECTED	Coordinate system is write protected
547	PI_CNTR_CS_CONTENT_FROM_CONFIG_FILE	Coordinate system cannot be changed because its content is loaded from a configuration file
548	PI_CNTR_CS_CANNOT_BE_LINKED	Coordinate system may not be linked
549	PI_CNTR_KSB_CS_ROTATION_ONLY	A KSB-type coordinate system can only be rotated by multiples of 90 degrees
551	PI_CNTR_CS_DATA_CANNOT_BE_QUERIED	This query is not supported for this coordinate system type
552	PI_CNTR_CS_COMBINATION_DOES_NOT_EXIST	This combination of work and tool coordinate systems does not exist
553	PI_CNTR_CS_COMBINATION_INVALID	The combination must consist of one work and one tool coordinate system
554	PI_CNTR_CS_TYPE_DOES_NOT_EXIST	This coordinate system type does not exist
555	PI_CNTR_UNKNOWN_ERROR	BasMac: unknown controller error
556	PI_CNTR_CS_TYPE_NOT_ENABLED	No coordinate system of this type is enabled
557	PI_CNTR_CS_NAME_INVALID	Name of coordinate system is invalid
558	PI_CNTR_CS_GENERAL_FILE_MISSING	File with stored CS systems is missing or erroneous
559	PI_CNTR_CS_LEVELING_FILE_MISSING	File with leveling CS is missing or erroneous
601	PI_CNTR_NOT_ENOUGH_MEMORY	Not enough memory
602	PI_CNTR_HW_VOLTAGE_ERROR	Hardware voltage error
603	PI_CNTR_HW_TEMPERATURE_ERROR	Hardware temperature out of range
604	PI_CNTR_POSITION_ERROR_TOO_HIGH	Position error of any axis in the system is too high
606	PI_CNTR_INPUT_OUT_OF_RANGE	Maximum value of input signal has been exceeded
607	PI_CNTR_NO_INTEGER	Value is not integer
608	PI_CNTR_FAST_ALIGNMENT_PROCESS_IS_NOT_RUNNING	Fast alignment process cannot be paused because it is not running
609	PI_CNTR_FAST_ALIGNMENT_PROCESS_IS_NOT_PAUSED	Fast alignment process cannot be restarted/resumed because it is not paused
650	PI_CNTR_UNABLE_TO_SET_PARAM_WITH_SPA	Parameter could not be set with SPA - SEP needed?

651	PI_CNTR_PHASE_FINDING_ERROR	Phase finding error
652	PI_CNTR_SENSOR_SETUP_ERROR	Sensor setup error
653	PI_CNTR_SENSOR_COMM_ERROR	Sensor communication error
654	PI_CNTR_MOTOR_AMPLIFIER_ERROR	Motor amplifier error
655	PI_CNTR_OVER_CURR_PROTEC_TRIGGERED_BY_I2T	Overcurrent protection triggered by I2T-module
656	PI_CNTR_OVER_CURR_PROTEC_TRIGGERED_BY_AMP_MODULE	Overcurrent protection triggered by amplifier module
657	PI_CNTR_SAFETY_STOP_TRIGGERED	Safety stop triggered
658	PI_SENSOR_OFF	Sensor off?
659	PI_CNTR_PARAM_CONFLICT	Parameter could not be set. Conflict with another parameter.
700	PI_CNTR_COMMAND_NOT_ALLOWED_IN_EXTERNAL_MODE	Command not allowed in external mode
710	PI_CNTR_EXTERNAL_MODE_ERROR	External mode communication error
715	PI_CNTR_INVALID_MODE_OF_OPERATION	Invalid mode of operation
716	PI_CNTR_FIRMWARE_STOPPED_BY_CMD	Firmware stopped by command (#27)
717	PI_CNTR_EXTERNAL_MODE_DRIVER_MISSING	External mode driver missing
718	PI_CNTR_CONFIGURATION_FAILURE_EXTERNAL_MODE	Missing or incorrect configuration of external mode
719	PI_CNTR_EXTERNAL_MODE_CYCLETIME_INVALID	External mode cycletime invalid
720	PI_CNTR_BRAKE_ACTIVATED	Brake is activated
725	PI_CNTR_DRIVE_STATE_TRANSITION_ERROR	Drive state transition error
731	PI_CNTR_SURFACEDETECTION_RUNNING	Command not allowed while surface detection is running
732	PI_CNTR_SURFACEDETECTION_FAILED	Last surface detection failed
733	PI_CNTR_FIELDBUS_IS_ACTIVE	Fieldbus is active and is blocking GCS control commands
1000	PI_CNTR_TOO_MANY_NESTED_MACROS	Too many nested macros
1001	PI_CNTR_MACRO_ALREADY_DEFINED	Macro already defined
1002	PI_CNTR_NO_MACRO_RECORDING	Macro recording not activated
1003	PI_CNTR_INVALID_MAC_PARAM	Invalid parameter for MAC
1004	PI_CNTR_MACRO_DELETE_ERROR	Deleting macro failed
1005	PI_CNTR_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)

1006	PI_CNTR_INVALID_IDENTIFIER	Invalid identifier (invalid special characters, ...)
1007	PI_CNTR_UNKNOWN_VARIABLE_OR_ARGUMENT	Variable or argument not defined
1008	PI_CNTR_RUNNING_MACRO	Controller is (already) running a macro
1009	PI_CNTR_MACRO_INVALID_OPERATOR	Invalid or missing operator for condition. Check necessary spaces around operator.
1010	PI_CNTR_MACRO_NO_ANSWER	No answer was received while executing WAC/MEX/JRC/...
1011	PI_CMD_NOT_VALID_IN_MACRO_MODE	Command not valid during macro execution
1012	PI_CNTR_ERROR_IN_MACRO	Error occurred during macro execution
1024	PI_CNTR_MOTION_ERROR	Motion error: position error too large, servo is switched off automatically
1025	PI_CNTR_MAX_MOTOR_OUTPUT_REACHED	Maximum motor output reached
1063	PI_CNTR_EXT_PROFILE_UNALLOWED_CMD	User Profile Mode: Command is not allowed, check for required preparatory commands
1064	PI_CNTR_EXT_PROFILE_EXPECTING_MOTION_ERROR	User Profile Mode: First target position in User Profile is too far from current position
1065	PI_CNTR_PROFILE_ACTIVE	Controller is (already) in User Profile Mode
1066	PI_CNTR_PROFILE_INDEX_OUT_OF_RANGE	User Profile Mode: Block or Data Set index out of allowed range
1071	PI_CNTR_PROFILE_OUT_OF_MEMORY	User Profile Mode: Out of memory
1072	PI_CNTR_PROFILE_WRONG_CLUSTER	User Profile Mode: Cluster is not assigned to this axis
1073	PI_CNTR_PROFILE_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
1090	PI_CNTR_TOO_MANY_TCP_CONNECTIONS_OPEN	There are too many open tcpip connections
2000	PI_CNTR_ALREADY_HAS_SERIAL_NUMBER	Controller already has a serial number
4000	PI_CNTR_SECTOR_ERASE_FAILED	Sector erase failed
4001	PI_CNTR_FLASH_PROGRAM_FAILED	Flash program failed
4002	PI_CNTR_FLASH_READ_FAILED	Flash read failed
4003	PI_CNTR_HW_MATCHCODE_ERROR	HW match code missing/invalid
4004	PI_CNTR_FW_MATCHCODE_ERROR	FW match code missing/invalid



4005	PI_CNTR_HW_VERSION_ERROR	HW version missing/invalid
4006	PI_CNTR_FW_VERSION_ERROR	FW version missing/invalid
4007	PI_CNTR_FW_UPDATE_ERROR	FW update failed
4008	PI_CNTR_FW_CRC_PAR_ERROR	FW Parameter CRC wrong
4009	PI_CNTR_FW_CRC_FW_ERROR	FW CRC wrong
5000	PI_CNTR_INVALID_PCC_SCAN_DATA	PicoCompensation scan data is not valid
5001	PI_CNTR_PCC_SCAN_RUNNING	PicoCompensation is running, some actions can not be executed during scanning/recording
5002	PI_CNTR_INVALID_PCC_AXIS	Given axis can not be defined as PPC axis
5003	PI_CNTR_PCC_SCAN_OUT_OF_RANGE	Defined scan area is larger than the travel range
5004	PI_CNTR_PCC_TYPE_NOT_EXISTING	Given PicoCompensation type is not defined
5005	PI_CNTR_PCC_PAM_ERROR	PicoCompensation parameter error
5006	PI_CNTR_PCC_TABLE_ARRAY_TOO_LARGE	PicoCompensation table is larger than maximum table length
5100	PI_CNTR_NEXLINE_ERROR	Common error in Nexline firmware module
5101	PI_CNTR_CHANNEL_ALREADY_USED	Output channel for Nexline can not be redefined for other usage
5102	PI_CNTR_NEXLINE_TABLE_TOO_SMALL	Memory for Nexline signals is too small
5103	PI_CNTR_RNP_WITH_SERVO_ON	RNP can not be executed if axis is in closed loop
5104	PI_CNTR_RNP_NEEDED	relax procedure (RNP) needed
5200	PI_CNTR_AXIS_NOT_CONFIGURED	Axis must be configured for this action
5300	PI_CNTR_FREQU_ANALYSIS_FAILED	Frequency analysis failed
5301	PI_CNTR_FREQU_ANALYSIS_RUNNING	Another frequency analysis is running
6000	PI_CNTR_SENSOR_ABS_INVALID_VALUE	Invalid preset value of absolute sensor
6001	PI_CNTR_SENSOR_ABS_WRITE_ERROR	Error while writing to sensor
6002	PI_CNTR_SENSOR_ABS_READ_ERROR	Error while reading from sensor
6003	PI_CNTR_SENSOR_ABS_CRC_ERROR	Checksum error of absolute sensor
6004	PI_CNTR_SENSOR_ABS_ERROR	General error of absolute sensor
6005	PI_CNTR_SENSOR_ABS_OVERFLOW	Overflow of absolute sensor position

## Interface errors

0	COM_NO_ERROR	No error occurred during function call
-1	COM_ERROR	Error during com operation (could not be specified)
-2	SEND_ERROR	Error while sending data
-3	REC_ERROR	Error while receiving data
-4	NOT_CONNECTED_ERROR	Not connected (no port with given ID open)
-5	COM_BUFFER_OVERFLOW	Buffer overflow
-6	CONNECTION_FAILED	Error while opening port
-7	COM_TIMEOUT	Timeout error
-8	COM_MULTILINE_RESPONSE	There are more lines waiting in buffer
-9	COM_INVALID_ID	There is no interface or DLL handle with the given ID
-10	COM_NOTIFY_EVENT_ERROR	Event/message for notification could not be opened
-11	COM_NOT_IMPLEMENTED	Function not supported by this interface type
-12	COM_ECHO_ERROR	Error while sending "echoed" data
-13	COM_GPIB_EDVR	IEEE488: System error
-14	COM_GPIB_ECIC	IEEE488: Function requires GPIB board to be CIC
-15	COM_GPIB_ENOL	IEEE488: Write function detected no listeners
-16	COM_GPIB_EADR	IEEE488: Interface board not addressed correctly
-17	COM_GPIB_EARG	IEEE488: Invalid argument to function call
-18	COM_GPIB_ESAC	IEEE488: Function requires GPIB board to be SAC
-19	COM_GPIB_EABO	IEEE488: I/O operation aborted
-20	COM_GPIB_ENEB	IEEE488: Interface board not found
-21	COM_GPIB_EDMA	IEEE488: Error performing DMA
-22	COM_GPIB_EOIP	IEEE488: I/O operation started before previous operation completed
-23	COM_GPIB_ECAP	IEEE488: No capability for intended

		operation
-24	COM_GPIB_EFSO	IEEE488: File system operation error
-25	COM_GPIB_EBUS	IEEE488: Command error during device call
-26	COM_GPIB_ESTB	IEEE488: Serial poll-status byte lost
-27	COM_GPIB_ESRQ	IEEE488: SRQ remains asserted
-28	COM_GPIB_ETAB	IEEE488: Return buffer full
-29	COM_GPIB_ELCK	IEEE488: Address or board locked
-30	COM_RS_INVALID_DATA_BITS	RS-232: 5 data bits with 2 stop bits is an invalid combination, as is 6, 7, or 8 data bits with 1.5 stop bits
-31	COM_ERROR_RS_SETTINGS	RS-232: Error configuring the COM port
-32	COM_INTERNAL_RESOURCES_ERROR	Error dealing with internal system resources (events, threads, ...)
-33	COM_DLL_FUNC_ERROR	A DLL or one of the required functions could not be loaded
-34	COM_FTDIUSB_INVALID_HANDLE	FTDIUSB: invalid handle
-35	COM_FTDIUSB_DEVICE_NOT_FOUND	FTDIUSB: device not found
-36	COM_FTDIUSB_DEVICE_NOT_OPENED	FTDIUSB: device not opened
-37	COM_FTDIUSB_IO_ERROR	FTDIUSB: IO error
-38	COM_FTDIUSB_INSUFFICIENT_RESOURCES	FTDIUSB: insufficient resources
-39	COM_FTDIUSB_INVALID_PARAMETER	FTDIUSB: invalid parameter
-40	COM_FTDIUSB_INVALID_BAUD_RATE	FTDIUSB: invalid baud rate
-41	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_ERASE	FTDIUSB: device not opened for erase
-42	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_WRITE	FTDIUSB: device not opened for write
-43	COM_FTDIUSB_FAILED_TO_WRITE_DEVICE	FTDIUSB: failed to write device
-44	COM_FTDIUSB_EEPROM_READ_FAILED	FTDIUSB: EEPROM read failed
-45	COM_FTDIUSB_EEPROM_WRITE_FAILED	FTDIUSB: EEPROM write failed
-46	COM_FTDIUSB_EEPROM_ERASE_FAILED	FTDIUSB: EEPROM erase failed
-47	COM_FTDIUSB_EEPROM_NOT_PRESENT	FTDIUSB: EEPROM not present
-48	COM_FTDIUSB_EEPROM_NOT_PROGRAMMED	FTDIUSB: EEPROM not programmed
-49	COM_FTDIUSB_INVALID_ARGS	FTDIUSB: invalid arguments
-50	COM_FTDIUSB_NOT_SUPPORTED	FTDIUSB: not supported
-51	COM_FTDIUSB_OTHER_ERROR	FTDIUSB: other error
-52	COM_PORT_ALREADY_OPEN	Error while opening the COM port:

		was already open
-53	COM_PORT_CHECKSUM_ERROR	Checksum error in received data from COM port
-54	COM_SOCKET_NOT_READY	Socket not ready, you should call the function again
-55	COM_SOCKET_PORT_IN_USE	Port is used by another socket
-56	COM_SOCKET_NOT_CONNECTED	Socket not connected (or not valid)
-57	COM_SOCKET_TERMINATED	Connection terminated (by peer)
-58	COM_SOCKET_NO_RESPONSE	Can't connect to peer
-59	COM_SOCKET_INTERRUPTED	Operation was interrupted by a nonblocked signal
-60	COM_PCI_INVALID_ID	No device with this ID is present
-61	COM_PCI_ACCESS_DENIED	Driver could not be opened (on Vista: run as administrator!)
-62	COM_SOCKET_HOST_NOT_FOUND	Host not found
-63	COM_DEVICE_CONNECTED	Device already connected

#### DLL errors

-1001	PI_UNKNOWN_AXIS_IDENTIFIER	Unknown axis identifier
-1002	PI_NR_NAV_OUT_OF_RANGE	Number for NAV out of range--must be in [1.10000]
-1003	PI_INVALID_SGA	Invalid value for SGA--must be one of 1, 10, 100, 1000
-1004	PI_UNEXPECTED_RESPONSE	Controller sent unexpected response
-1005	PI_NO_MANUAL_PAD	No manual control pad installed, calls to SMA and related commands are not allowed
-1006	PI_INVALID_MANUAL_PAD_KNOB	Invalid number for manual control pad knob
-1007	PI_INVALID_MANUAL_PAD_AXIS	Axis not currently controlled by a manual control pad
-1008	PI_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g., reference move, fast scan algorithm)
-1009	PI_THREAD_ERROR	Internal error--could not start thread
-1010	PI_IN_MACRO_MODE	Controller is (already) in macro mode--command not valid in macro mode
-1011	PI_NOT_IN_MACRO_MODE	Controller not in macro mode--command not valid unless macro

		mode active
-1012	PI_MACRO_FILE_ERROR	Could not open file to write or read macro
-1013	PI_NO_MACRO_OR_EMPTY	No macro with given name on controller, or macro is empty
-1014	PI_MACRO_EDITOR_ERROR	Internal error in macro editor
-1015	PI_INVALID_ARGUMENT	One or more arguments given to function is invalid (empty string, index out of range, ...)
-1016	PI_AXIS_ALREADY_EXISTS	Axis identifier is already in use by a connected stage
-1017	PI_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
-1018	PI_COM_ARRAY_ERROR	Could not access array data in COM server
-1019	PI_COM_ARRAY_RANGE_ERROR	Range of array does not fit the number of parameters
-1020	PI_INVALID_SPA_CMD_ID	Invalid parameter ID given to SPA or SPA?
-1021	PI_NR_AVG_OUT_OF_RANGE	Number for AVG out of range--must be >0
-1022	PI_WAV_SAMPLES_OUT_OF_RANGE	Incorrect number of samples given to WAV
-1023	PI_WAV_FAILED	Generation of wave failed
-1024	PI_MOTION_ERROR	Motion error: position error too large, servo is switched off automatically
-1025	PI_RUNNING_MACRO	Controller is (already) running a macro
-1026	PI_PZT_CONFIG_FAILED	Configuration of PZT stage or amplifier failed
-1027	PI_PZT_CONFIG_INVALID_PARAMS	Current settings are not valid for desired configuration
-1028	PI_UNKNOWN_CHANNEL_IDENTIFIER	Unknown channel identifier
-1029	PI_WAVE_PARAM_FILE_ERROR	Error while reading/writing wave generator parameter file
-1030	PI_UNKNOWN_WAVE_SET	Could not find description of wave form. Maybe WG.INI is missing?
-1031	PI_WAVE_EDITOR_FUNC_NOT_LOADED	The WGWaveEditor DLL function was not found at startup
-1032	PI_USER_CANCELLED	The user cancelled a dialog
-1033	PI_C844_ERROR	Error from C-844 Controller
-1034	PI_DLL_NOT_LOADED	DLL necessary to call function not loaded, or function not found in DLL

-1035	PI_PARAMETER_FILE_PROTECTED	The open parameter file is protected and cannot be edited
-1036	PI_NO_PARAMETER_FILE_OPENED	There is no parameter file open
-1037	PI_STAGE_DOES_NOT_EXIST	Selected stage does not exist
-1038	PI_PARAMETER_FILE_ALREADY_OPENED	There is already a parameter file open. Close it before opening a new file
-1039	PI_PARAMETER_FILE_OPEN_ERROR	Could not open parameter file
-1040	PI_INVALID_CONTROLLER_VERSION	The version of the connected controller is invalid
-1041	PI_PARAM_SET_ERROR	Parameter could not be set with SPA--parameter not defined for this controller!
-1042	PI_NUMBER_OF_POSSIBLE_WAVES_EXCEEDED	The maximum number of wave definitions has been exceeded
-1043	PI_NUMBER_OF_POSSIBLE_GENERATORS_EXCEEDED	The maximum number of wave generators has been exceeded
-1044	PI_NO_WAVE_FOR_AXIS_DEFINED	No wave defined for specified axis
-1045	PI_CANT_STOP_OR_START_WAV	Wave output to axis already stopped/started
-1046	PI_REFERENCE_ERROR	Not all axes could be referenced
-1047	PI_REQUIRED_WAVE_NOT_FOUND	Could not find parameter set required by frequency relation
-1048	PI_INVALID_SPP_CMD_ID	Command ID given to SPP or SPP? is not valid
-1049	PI_STAGE_NAME_ISNT_UNIQUE	A stage name given to CST is not unique
-1050	PI_FILE_TRANSFER_BEGIN_MISSING	A uuencoded file transferred did not start with "begin" followed by the proper filename
-1051	PI_FILE_TRANSFER_ERROR_TEMP_FILE	Could not create/read file on host PC
-1052	PI_FILE_TRANSFER_CRC_ERROR	Checksum error when transferring a file to/from the controller
-1053	PI_COULDNT_FIND_PISTAGES_DAT	The PiStages.dat database could not be found. This file is required to connect a stage with the CST command
-1054	PI_NO_WAVE_RUNNING	No wave being output to specified axis
-1055	PI_INVALID_PASSWORD	Invalid password
-1056	PI_OPM_COM_ERROR	Error during communication with OPM (Optical Power Meter), maybe no OPM connected

-1057	PI_WAVE_EDITOR_WRONG_PARAMNUM	WaveEditor: Error during wave creation, incorrect number of parameters
-1058	PI_WAVE_EDITOR_FREQUENCY_OUT_OF_RANGE	WaveEditor: Frequency out of range
-1059	PI_WAVE_EDITOR_WRONG_IP_VALUE	WaveEditor: Error during wave creation, incorrect index for integer parameter
-1060	PI_WAVE_EDITOR_WRONG_DP_VALUE	WaveEditor: Error during wave creation, incorrect index for floating point parameter
-1061	PI_WAVE_EDITOR_WRONG_ITEM_VALUE	WaveEditor: Error during wave creation, could not calculate value
-1062	PI_WAVE_EDITOR_MISSING_GRAPH_COMPONENT	WaveEditor: Graph display component not installed
-1063	PI_EXT_PROFILE_UNALLOWED_CMD	User profile mode: command is not allowed, check for required preparatory commands
-1064	PI_EXT_PROFILE_EXPECTING_MOTION_ERROR	User profile mode: first target position in user profile is too far from current position
-1065	PI_EXT_PROFILE_ACTIVE	Controller is (already) in user profile mode
-1066	PI_EXT_PROFILE_INDEX_OUT_OF_RANGE	User profile mode: block or data set index out of allowed range
-1067	PI_PROFILE_GENERATOR_NO_PROFILE	ProfileGenerator: No profile has been created yet
-1068	PI_PROFILE_GENERATOR_OUT_OF_LIMITS	ProfileGenerator: Generated profile exceeds limits of one or both axes
-1069	PI_PROFILE_GENERATOR_UNKNOWN_PARAMETER	ProfileGenerator: Unknown parameter ID in Set/Get Parameter command
-1070	PI_PROFILE_GENERATOR_PAR_OUT_OF_RANGE	ProfileGenerator: Parameter out of allowed range
-1071	PI_EXT_PROFILE_OUT_OF_MEMORY	User profile mode: out of memory
-1072	PI_EXT_PROFILE_WRONG_CLUSTER	User profile mode: cluster is not assigned to this axis
-1073	PI_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
-1074	PI_INVALID_DEVICE_DRIVER_VERSION	The installed device driver doesn't match the required version. Please see the documentation to determine the required device driver version.
-1075	PI_INVALID_LIBRARY_VERSION	The library used doesn't match the required version. Please see the documentation to determine the required library version.
-1076	PI_INTERFACE_LOCKED	The interface is currently locked by



		another function. Please try again later.
-1077	PI_PARAM_DAT_FILE_INVALID_VERSION	Version of parameter DAT file does not match the required version. Current files are available at <a href="http://www.pi.ws">www.pi.ws</a> .
-1078	PI_CANNOT_WRITE_TO_PARAM_DAT_FILE	Cannot write to parameter DAT file to store user defined stage type.
-1079	PI_CANNOT_CREATE_PARAM_DAT_FILE	Cannot create parameter DAT file to store user defined stage type.
-1080	PI_PARAM_DAT_FILE_INVALID_REVISION	Parameter DAT file does not have correct revision.
-1081	PI_USERSTAGES_DAT_FILE_INVALID_REVISION	User stages DAT file does not have correct revision.
-1082	PI_SOFTWARE_TIMEOUT	Timeout Error. Some lengthy operation did not finish within expected time.
-1083	PI_WRONG_DATA_TYPE	A function argument has an unexpected data type.
-1084	PI_DIFFERENT_ARRAY_SIZES	Length of data arrays is different.
-1085	PI_PARAM_NOT_FOUND_IN_PARAM_DAT_FILE	Parameter value not found in parameter DAT file.
-1086	PI_MACRO_RECORDING_NOT_ALLOWED_IN_THIS_MODE	Macro recording is not allowed in this mode of operation.
-1087	PI_USER_CANCELLED_COMMAND	Command cancelled by user input.
-1088	PI_TOO_FEW_GCS_DATA	Controller sent too few GCS data sets
-1089	PI_TOO_MANY_GCS_DATA	Controller sent too many GCS data sets
-1090	PI_GCS_DATA_READ_ERROR	Communication error while reading GCS data
-1091	PI_WRONG_NUMBER_OF_INPUT_ARGUMENTS	Wrong number of input arguments.
-1092	PI_FAILED_TO_CHANGE_CCL_LEVEL	Change of command level has failed.
-1093	PI_FAILED_TO_SWITCH_OFF_SERVO	Switching off the servo mode has failed.
-1094	PI_FAILED_TO_SET_SINGLE_PARAMETER_WHILE_PERFORMING_CST	A parameter could not be set while performing CST: CST was not performed (parameters remain unchanged).
-1095	PI_ERROR_CONTROLLER_REBOOT	Connection could not be reestablished after reboot.
-1096	PI_ERROR_AT_QHPA	Sending HPA? or receiving the response has failed.

-1097	PI_QHPA_NONCOMPLIANT_WITH_GCS	HPA? response does not comply with GCS2 syntax.
-1098	PI_FAILED_TO_READ_QSPA	Response to SPA? could not be received.
-1099	PI_PAM_FILE_WRONG_VERSION	Version of PAM file cannot be handled (too old or too new)
-1100	PI_PAM_FILE_INVALID_FORMAT	PAM file does not contain required data in PAM-file format
-1101	PI_INCOMPLETE_INFORMATION	Information does not contain all required data
-1102	PI_NO_VALUE_AVAILABLE	No value for parameter available
-1103	PI_NO_PAM_FILE_OPEN	No PAM file is open
-1104	PI_INVALID_VALUE	Invalid value
-1105	PI_UNKNOWN_PARAMETER	Unknown parameter
-1106	PI_RESPONSE_TO_QSEP_FAILED	Response to SEP? could not be received.
-1107	PI_RESPONSE_TO_QSPA_FAILED	Response to SPA? could not be received.
-1108	PI_ERROR_IN_CST_VALIDATION	Error while performing CST: One or more parameters were not set correctly.
-1109	PI_ERROR_PAM_FILE_HAS_DUPLICATE_ENTRY_WITH_DIFFERENT_VALUES	PAM file has duplicate entry with different values.
-1110	PI_ERROR_FILE_NO_SIGNATURE	File has no signature
-1111	PI_ERROR_FILE_INVALID_SIGNATURE	File has invalid signature
-1112	PI_ERROR_CANNOT_DETERMINE_ACTUAL_END_OF_TRAVEL_WHILE_PLATFORM_IS_MOVING	Cannot determine actual end of travel range while platform is moving.
-10000	PI_PARAMETER_DB_INVALID_STAGE_TYPE_FORMAT	PI stage database: String containing stage type and description has invalid format.
-10001	PI_PARAMETER_DB_SYSTEM_NOT_AVAILABLE	PI stage database: Database does not contain the selected stage type for the connected controller.
-10002	PI_PARAMETER_DB_FAILED_TO_ESTABLISH_CONNECTION	PI stage database: Establishing the connection has failed.
-10003	PI_PARAMETER_DB_COMMUNICATION_ERROR	PI stage database: Communication was interrupted (e.g. because database was deleted).
-10004	PI_PARAMETER_DB_ERROR_WHILE_QUERYING_PARAMETERS	PI stage database: Querying data failed.
-10005	PI_PARAMETER_DB_SYSTEM_ALREADY_EXISTS	PI stage database: System already exists. Rename stage and try again.

-10006	PI_PARAMETER_DB_QHPA_CONTANS_UNKNOWN_PAM_IDS	PI stage database: Response to HPA? contains unknown parameter IDs.
-10007	PI_PARAMETER_DB_AND_QHPA_ARE_INCONSISTENT	PI stage database: Inconsistency between database and response to HPA?.
-10008	PI_PARAMETER_DB_SYSTEM_COULD_NOT_BE_ADDED	PI stage database: Stage has not been added.
-10009	PI_PARAMETER_DB_SYSTEM_COULD_NOT_BE_REMOVED	PI stage database: Stage has not been removed.
-10010	PI_PARAMETER_DB_CONTROLLER_DB_PARAMETERS_MISMATCH	Controller does not support all stage parameters stored in PI stage database. No parameters were set.
-10011	PI_PARAMETER_DB_DATABASE_IS_OUTDATED	The version of PISTAGES3.DB stage database is out of date. Please update via PIUpdateFinder. No parameters were set.
-10012	PI_PARAMETER_DB_AND_HPA_MISMATCH_STRICT	Mismatch between number of parameters present in stage database and available in controller interface. No parameters were set.
-10013	PI_PARAMETER_DB_AND_HPA_MISMATCH_LOOSE	Mismatch between number of parameters present in stage database and available in controller interface. Some parameters were ignored.
-10014	PI_PARAMETER_DB_FAILED_TO_SET_PARAMETERS_CORRECTLY	One or more parameters could not be set correctly on the controller.
-10015	PI_PARAMETER_DB_MISSING_PARAMETER_DEFINITIONS_IN_DATABASE	One or more parameter definitions are not present in stage database. Please update PISTAGES3.DB via PIUpdateFinder. Missing parameters were ignored.



End of document