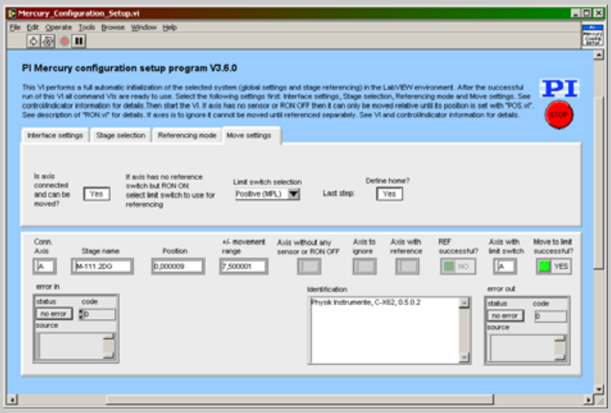
Piezo · Nano · Positioning

MS 149E Software Manual

Mercury LabView Driver Library

Release: 3.6.0   Date: 2006-01-17


Mercury_Configuration_Setup.vi
File Edit Operate Tools Browse Window Help
PI
STOP
Interface settings
Stage selection
Referencing mode
Move settings

PI Mercury configuration setup program V3.6.0

This VI performs a full automatic initialization of the selected system (global settings and stage referencing) in the LabVIEW environment. After the successful run 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. If axis has no sensor or RON OFF then it can only be moved relative until its position is set with "POS.vi". See description of "RON.vi" for details. If axes is to ignore it cannot be moved until referenced separately. See VI and control/indicator information for details.

Is axis connected and can be moved? Yes

If axis has no reference switch but RON ON: select limit switch to use for referencing

Limit switch selection: Positive (MPL)

Last step: Define home? Yes

Conn. Axis	Stage name	Position	+/- movement range	Axis without any sensor or RON OFF	Axis to ignore	Axis with reference	REF successful?	Axis with limit switch	Move to limit successful?
A	M-111.2DG	0,000009	7,500001				NO	A	YES

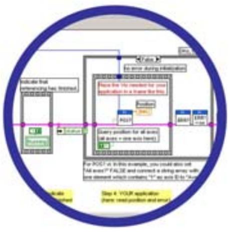
error in: status: no error; code: 0; source:

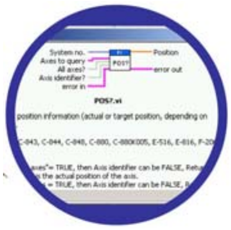
Identification: Physik Instrumente, C-X62, 0.5.0.2

error out: status: no error; code: 0; source:

This document describes software for use with the following products:

- C-862 Mercury™
  Networkable Single-Axis DC-Motor Controller
- C-663 Mercury™ Step
  Networkable Single-Axis Stepper Motor Controller
- C-863 Mercury™
  Networkable Single-Axis DC-Motor Controller





© Physik Instrumente (PI) GmbH & Co. KG
Auf der Römerstr. 1 · 76228 Karlsruhe, Germany
Tel. +49 721 4846-0 · Fax: +49 721 4846-299
info@pi.ws · www.pi.ws

Table of Contents

0. DISCLAIMER 2

1. INTRODUCTION 2

1.1. PI GENERAL COMMAND SET (GCS) 2
1.2. SCOPE OF THIS MANUAL 3
1.3. VI STRUCTURE 3
1.4. WORKING WITH TWO PI PRODUCTS WHICH UNDERSTAND PI'S GENERAL COMMAND SET (GCS) IN LABVIEW 5
1.5. FIRST STEPS FOR GCS-COMPATIBLE PI CONTROLLERS 7

2. LOW LEVEL VIS 20

2.1. COMMUNICATION VIS ("COMMUNICATION.LLB"): 20
2.2. GENERAL COMMAND VIS ("GENERAL COMMAND.LLB"): 24
2.3. SPECIAL COMMANDS ("SPECIAL COMMAND.LLB") 36
2.4. OLD COMMANDS AND COMMANDS WITH ALTERNATE IMPLEMENTATIONS ("OLD COMMANDS.LLB") 44
2.5. LIMIT- AND REFERENCE-SPECIFIC COMMANDS ("LIMITS.LLB") 45
2.6. MACRO FUNCTIONS ("MACROS.LLB") 50
2.7. COMMANDS FOR OPTICAL OR ANALOG SIGNALS ("OPTICAL OR ANALOG INPUT.LLB") 52
2.8. SUPPORT VIS ("SUPPORT.LLB") 53

3. HIGH LEVEL VIS 64

3.1. PI TERMINAL.VI 64
3.2. SIMPLE_OPERATION.VI 67
3.3. MERCURY_SIMPLE_TEST.VI 68
3.4. MERCURY_CONFIGURATION_SETUP.VI 69
3.5. MERCURY_SAMPLE_APPLICATION_1.VI 72
3.6. MERCURY_SAMPLE_APPLICATION_1A.VI 73

4. PI SYSTEMS CURRENTLY SUPPORTED BY THIS DRIVER SET 74

5. APPENDIX A 75

6. INDEX 82

Release: 3.6.0
File:Mercury_GCSLabVIEW_MS149E.doc, 1648128 Bytes

0. Disclaimer

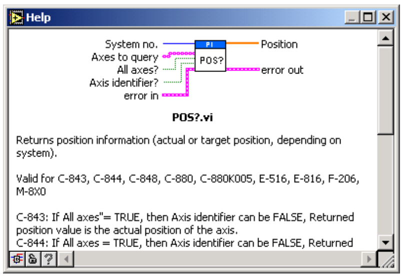
This software is provided “as is”. Physik Instrumente (PI) does not guarantee that this software is free of errors and will not be responsible for any damage arising from the use of this software. The user agrees to use this software on his own responsibility.

1. Introduction

The LabVIEW software 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 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 LabVIEW environment to display the *Context Help* window.

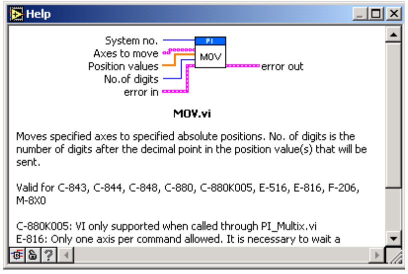


POS?.vi

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

Valid for C-843, C-844, C-848, C-880, C-880K005, E-516, E-816, F-206, M-8X0

C-843: If All axes"= TRUE, then Axis identifier can be FALSE, Returned position value is the actual position of the axis.
C-844: If All axes = TRUE, then Axis identifier can be FALSE, Returned



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.

Valid for C-843, C-844, C-848, C-880, C-880K005, E-516, E-816, F-206, M-8X0

C-880K005: VI only supported when called through PI_Multix.vi
E-816: Only one axis per command allowed. It is necessary to wait a

1.1. PI General Command Set (GCS)

This VI driver set supports the *PI General Command Set*, 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-516 Display Module* or the *C-880 Multi-Axis Controller*, with only one driver set simply by “wiring” the correct command parameters to the associated VIs.

Translation Libraries

To control PI systems that are not originally compatible with the *PI General Command Set*, such as the *E-710 Digital Piezo Controller* or the *C-843 Motion Control Board*, libraries are used to translate *PI General Command Set* commands to other controller-specific languages. The universal library which adds this functionality is GCSTranslator.dll; it must be installed on the computer in the GCS_LabVIEW\Low Level folder, no matter whether the system being controlled is *PI General Command Set* compatible or not. To control certain systems (such as a non-GCS-compatible system or a PC board), one or more system-specific DLLs and data files (e.g. PIStages.dat) must also be installed. If you install this driver set from within the setup program of the PI software CD ROM, this installation is done automatically. If you want to install this driver set manually, please run “GCSLibrarySetup.exe” from the CD-ROM that came with your system. This setup tool makes sure that all necessary libraries and their data files are correctly registered in the Windows™ environment and can be found by the GCS drivers.

If LabVIEW still cannot find PIStages.dat, it may be because it is marked read-only. To see, open Microsoft Explorer, right-click the file PIStages.dat and select *Properties*. Make sure that the *read-only* attribute is not checked.

Once the libraries and data files for the system to control are installed, this LabVIEW driver set can be used to control a non-GCS-compatible system just like any GCS-compatible system, and PCI/ISA-based controller boards by selecting "DLL" as communication interface (see Section "

First Steps for GCS-Compatible PI **Controllers**

" on p. 7 and the "PI Open Interface.vi" / "PI Open Interface of one system.vi" command description on p. 21 / 22).

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-precion entry of gearhead system values. See the DFF.vi and SPA.vi command descriptions, taking special note of the sections referring specifically to your controller.

1.2. Scope of This Manual

This manual covers only VIs which can be used with the product with which it came.

1.3. VI Structure

The folder structure of the LabVIEW drivers consists of the main folder "GCS_LabVIEW" with the sub-folder "Low Level".

The main folder "GCS_LabVIEW" contains a terminal VI, a configuration VI (XXXX_Configuration_Setup.vi with XXXX being the PI product number of your system), a simple test VI, and, if available, 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 or GPIB interfaces
- Defining the parameter IDs of the connected axes
- Sending and receiving ASCII characters to/from the specified system
- Sending system-specific commands (system-specific commands are separated into function-specific LLBs).

Additionally, the sub-folder "Low Level" contains GCSTranslator.dll.

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:

<table>
<tr><td>1</td><td rowspan="3">7</td><td rowspan="3">9</td><td rowspan="3">11</td><td rowspan="3">13</td><td>15</td></tr>
<tr><td>2</td><td>16</td></tr>
<tr><td>3</td><td>17</td></tr>
<tr><td>4</td><td rowspan="3">8</td><td rowspan="3">10</td><td rowspan="3">12</td><td rowspan="3">14</td><td>18</td></tr>
<tr><td>5</td><td>19</td></tr>
<tr><td>6</td><td>20</td></tr>
</table>

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

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 LabVIEW error numbers recommended by National Instruments, but rather those used by PI controllers. As a result, the error texts displayed by 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 Hidden error to indicate that the selected system has been queried for a controller error with „ERR?“ and reported an error number ≠ zero.

See also chapter 5 on p. 75 for a summary of error numbers produced by this driver set.

If you are using LabVIEW 7.1, uncheck *Enable automatic error handling dialogs* in *Tools→Options→New and Changed in 7.x* to prevent that 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, "**XXXX_Configuration_Setup.vi**" (located in the main folder, with XXXX being the PI product number of your system) 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, 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 LabView session can access this data at runtime.

As the initialization is a complex procedure which uses a large number of sub-VIs, **XXXX_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. Advanced users who have a need to do so, should please contact their local sales representative to receive the password.

For testing, the easiest method is to call "PI Terminal.vi", which is located in the "GCS_LabVIEW" main folder. This is a "stand-alone" routine that calls "PI Ask for Communication Parameters.vi" first and then opens the specified ports. It does not, however, define the connected axes of the (motion) systems. A more system-specific sample VI is "XXXX_Simple_Test.vi" (with XXXX being the PI product number of your system), also located in the "GCS_LabVIEW" main folder.

1.4. Working with two PI products which understand PI's General Command Set (GCS) in LabVIEW

When installing the LabVIEW programming support for two different PI products, there are two "Low Level" folders installed, one in each product-specific LabVIEW driver set. This is because every product comes with only the VIs which are used with the product. Another product may have different libraries or different library contents due to the product supporting more or fewer functions. When working with two product-specific LabVIEW driver set installations on one computer, it is important to make sure that LabVIEW always uses the right libraries.

a) When working separately with two products, the "Low Level" folder of each product must be located in the same folder as the product-specific main VI which calls sub-VIs from the product-specific driver set. Otherwise LabVIEW will start searching for sub VIs whereever it finds them, which may result in version conflicts and "broken Run" arrows. Please make sure that no VIs are saved under LabVIEWs own "user.lib" sub-folder. If they are LabVIEW will always find them there first, which will cause errors in many cases.

b) When working with two products in parallel, the libraries should be combined. Under LabVIEW 7.1 (but also under older LabVIEW versions), this is quite easy using LabVIEWs "Library manager", which can be found on the "Tools" menu. First, make a backup copy of the older library you want to combine with the newer one. In Library manager, open the newer library (e.g. "Special.llb" from the C-865 release) in the left window, and the older library (e.g. "Special.llb" from the older C-843 release) in the right window. Then choose "Show dates", "Disable files with identical dates", and "Sort alphabetically". This will show you only the VIs which have different file dates or which are only present in one of these libraries, but will not show any identical VIs. Now select all VIs which are still shown on the left window and copy them to the right window. In this way, you get a library with all VIs used by both product driver sets, and the newest version of each. Do this for all libraries in the low level folder. Make sure to work thereafter with the combined libraries instead of the product-specific libraries.

1.5. First Steps for GCS-Compatible PI Controllers

1.5.1. C-843

To control one or more C-843 boards with this driver set, “C843_GCS_DLL.dll”, “MC.dll”, “PiStages.dat” and the C-843 device driver must be installed on your computer. See chapter 1.1 for information about methods for proper installation of the first three items. A description of how to install the C-843 device driver is given in the C-843 User Manual. The following steps must then be performed:

Step 1 (advanced users can skip this step): To check communication with the C-843 board in the host PC, run “C843_Simple_Test.vi”. This VI will return the ID string of the C-843 board and the axis IDs of the connected axes. See chapter 3 for a description of this VI and 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.

Step 2:

WARNING: C843_Configuration_Setup.vi May Cause Move

When you start “C843_Configuration_Setup.vi” with All axes? = TRUE, the VI will automatically determine which axes have a reference switch and which have limit switches and, if the referencing mode of these axes is ON, 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, referencing mode (the mode which tells the controller to reference the stages or not) can be switched off. See description of RON for details and warnings.

Open “C843_Configuration_Setup.vi”. Select your C-843 board (2- or 4-axis version, board number) and leave “Use dialog to define connected stages” = TRUE. Run the VI. In the following screen, specify which stages you have connected to which axes and press OK. This VI performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: the definition of axis IDs, the initialization of the connected stages including referencing (if appropriate) 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 “C843_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can implement “C843_Configuration_Setup.vi” as an initialization VI in your software. See chapter 3 for a detailed description of “C843_Configuration_Setup.vi” and 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.

If you do not find your stage in the drop-down list, press CANCEL. You can then either define a User Stage with the Stage Editor, or you can contact PI to see about getting a new stage list: the "PIStages.dat" file contains all relevant stage parameters.

Default axis names are 1 to 4, but can be changed using “SAI.vi”.

1.5.2. C-843.PM

With this driver set, “C843_PM_GCS_DLL.dll”, “MC.dll”, “PiStages.dat” and the C-843 device driver must be installed on your computer if you wish to control axes on PIs linear piezo motor stages. It can be used with axes on other motorized stages connected to the same or other C-843 boards in the same machine. See chapter 1.1 for information about methods for proper installation of the first three items. A description of how to install the C-843 device driver is given in the C-843 User Manual. The following steps must then be performed:

Step 1 (advanced users can skip this step): To check communication with the C-843 board in the host PC, run “C843_PM_Simple_Test.vi”. This VI will return the ID string of the C-843 board and the axis IDs of the connected axes. See chapter 3 for a description of this VI and 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.

Step 2:

WARNING: C843_PM_Configuration_Setup.vi May Cause Move

When you start “C843_PM_Configuration_Setup.vi” with All axes? = TRUE, the VI will automatically determine which axes have a reference switch and which have limit switches and, if the referencing mode of these axes is ON, 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, referencing mode (the mode which tells the controller to reference the stages or not) can be switched off. See description of RON for details and warnings.

Open “C843_PM_Configuration_Setup.vi”. Select your C-843 board (2- or 4-axis version, board number) and leave “Use dialog to define connected stages” = TRUE. Run the VI. In the following screen, specify which stages you have connected to which axes and press OK. This VI performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: the definition of axis IDs, the initialization of the connected stages including referencing (if appropriate) 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 “C843_PM_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can implement “C843_PM_Configuration_Setup.vi” as an initialization VI in your software. See chapter 3 for a detailed description of “C843_PM_Configuration_Setup.vi” and 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.

If you do not find your stage in the drop-down list, press CANCEL. You can then either define a User Stage with the Stage Editor, or you can contact PI to see about getting a new stage list: the "PIStages.dat" file contains all relevant stage parameters.

Default axis names are 1 to 4, but can be changed using “SAI.vi”.

1.5.3. C-848

Step 1: The C-848 controller is delivered pre-configured. Before you start, please check that the current configuration matches your stage connections. See the C-848 User Manual for a detailed description of this step.

Step 2 (advanced users can skip this step): To check communication between the C-848 controller and the host PC, run “C848_Simple_Test.vi”. This VI will return the ID string of the C-848 controller, the axis IDs of the connected axes, and their current positions. See chapter 3 for a description of this VI and 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. Please make sure that you have all reported axes connected. If you want to work with only some of these axes, remember the IDs of those axes so that you can enter them in the Axes to move control in “C848_Configuration_ Setup.vi”; thereafter the other axes will not be moved when executing “C848_Configuration_Setup.vi”.

Step 3:

WARNING: C848_Configuration_Setup.vi May Cause Move

When you start “C848_Configuration_Setup.vi” with All axes? = TRUE, the VI will automatically determine which axes have a reference switch and which have limit switches and, if the referencing mode of these axes is ON, 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, referencing mode (the mode which tells the controller to reference the stages or not) can be switched off (see chapter 3). See description of RON for details and warnings.

To control one or more C-848 controllers with this driver set, run “C848_Configuration_Setup.vi”. This VI performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: the definition of axis IDs, the initialization of the connected stages including referencing (if appropriate) 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 “C848_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can implement “C848_Configuration_Setup.vi” as an initialization VI in your software. See chapter 3 for a detailed description of “C848_Configuration_Setup.vi” and 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.

Before using a joystick connected to the C-848 controller, calibrate the joystick by running “PI Terminal.vi”. Type “JEN CALIB” and follow the instructions on the screen.

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

1.5.4. C-865

To control one or more C-865s with this driver set, “C865_GCS_DLL.dll”, “ MC_C865.dll ”, and “PiStages.dat” must be installed on your computer. See Section 1.1 for information about methods for proper installation of the first three items. The following steps must then be performed:

Step 1 (advanced users can skip this step): To check communication with the C-865 controller, run “C865_Simple_Test.vi”. This VI will return the ID string of the C-865 controller. See Section 3 for a description of this VI and 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.

Step 2:

WARNING: C865_Configuration_Setup.vi May Cause Move

When you start “C865_Configuration_Setup.vi” with Is axis connected and can be moved? = TRUE, the VI will automatically determine if the connected axis has a reference switch or limit switch and, if the referencing mode of this axis is ON, will move the stage to one of the switches. It is therefore important to make sure that items connected to or mounted on the connected stage will not be damaged by such a move. If referencing is not possible (because the connected stage has no reference or limit switch) or not desired, referencing mode (the mode which tells the controller to reference the stage or not) can be switched off. See description of RON for details and warnings.

Open “C865_Configuration_Setup.vi”. Select the RS-232 settings (port number and appropriate baudrate) and leave Use dialog to define connected stage = TRUE. Run the VI. In the following screen, specify which stage you have connected and press OK. This VI performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: the definition of the axis ID, the initialization of the connected stage, including referencing (if appropriate) 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 “C865_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can implement “C865_Configuration_Setup.vi” as an initialization VI in your software. See Section 3 for a detailed description of “C865_Configuration_Setup.vi” and 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.

If you do not find your stage in the drop-down list, press CANCEL. You can then either define a User Stage with the Stage Editor, or you can contact PI to see about getting a new stage list: the "PIStages.dat" file contains all relevant stage parameters.

Default axis name is “1”, but can be changed using “SAI.vi”.

If the controller does not respond, please reset it using the reset button (unlabeled) on the rear panel of the C-865 controller.

See also “C865_Sample_Application_1.vi” and “C865_Sample_Application_1a.vi” as sample VIs showing how to implement “C865_Configuration_Setup.vi” as the initialization VI for the C-865 in your application.

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

1.5.5. C-880

Step 1: The C-880 controller is delivered pre-configured. Before you start, please check that the current configuration matches your stage connections. See the C-880 User Manual for a detailed description of this step.

Step 2 (advanced users can skip this step): To check communication between the C-880 controller and the host PC, run “C880_Simple_Test.vi”. This VI will return the ID string of the C-880 controller, the axis IDs of the connected axes, and their current positions. See chapter 3 for a description of this VI and 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. Please make sure that you have all reported axes connected. If you want to work with only some of these axes, remember the IDs of those axes so that you can enter them in the Axes to move control in “C880_Configuration_ Setup.vi”; thereafter the other axes will not be moved when executing “C880_Configuration_Setup.vi”.

Step 3:

WARNING: C880_Configuration_Setup.vi May Cause Move

When you start “C880_Configuration_Setup.vi” with All axes? = TRUE, the VI will automatically determine which axes have a reference switch and which have limit switches and, if the referencing mode of these axes is ON, 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, referencing mode (the mode which tells the controller to reference the stages or not) can be switched off (see chapter 3). See description of RON for details and warnings.

To control one or more C-880 controllers with this driver set, run “C880_Configuration_Setup.vi”. This VI performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: the definition of axis IDs, the initialization of the connected stages including referencing (if appropriate) 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 “C880_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can implement “C880_Configuration_Setup.vi” as an initialization VI in your software. See chapter 3 for a detailed description of “C880_Configuration_Setup.vi” and 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.

Before using a joystick connected to the C-880 controller, calibrate the joystick by running “PI Terminal.vi”. Type “JEN CALIB” and follow the instructions on the screen.

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

1.5.6. C-880K005

To control a C-880K005 controller with this driver set, you must perform the following steps:

1. Run "PI Open Interface.vi"
2. Run "Multix axis assignment.vi".
3. Reference all connected stages using "PI_Multix.vi" with Command = REF and All axes? = TRUE. If no referencing is desired, refencing mode can be switched off using Command = RON. With referencing mode off, only relative moves can be commanded (using Command = MVR), unless the actual position is set with Command = POS.vi. Thereafter both relative and absolute moves can be commanded.

OR

1. Run C880K005_Simple_Test.vi
2. Proceed as in step 3 above.

After these steps, a number of selected commands can be used by calling "PI_Multix.vi".

The control concept of the C-880K005 is based on the assumption that several C-880 multi-axis motion controllers are connected to the C-880K005, which is commanded over a single interface from the host PC. In this way, the number of axes to command is not limited by the number of connectors available on a single C-880. To ease handling so many axes, the user does not have to worry about the individual C-880 controllers, but only the C-880K005 controller with axes 1 to N, N being the sum of all axes connected to all interconnected C-880s.

The C-880K005 is called the "controller" and handles the communication to all connected C-880 controllers, which are called VControllers (virtual controllers). "PI_Multix.vi" must be used to send commands. For ease of operation, when running "Multix axis assignment.vi", all connected axes of all connected C-880 controllers are queried and the axis IDs 1 to N are assigned to these axes automatically.

A C-880 connected to the C-880K005 can be directly commanded by setting it active ("ACT.vi"). The C-880K005 communications controller then passes subsequent commands to the active C-880, except for commands which, by their nature, must be directly handled by the C-880K005 (e.g. WAA).

Example: Three C-880 controllers are connected to the C-880K005, and each C-880 controller has 12 axes designated A to L on each of the C-880's. These ID's are taken as VAxis IDs and the axis IDs 1 to 36 are assigned for the C-880K005 controller. To command axis A of C-880 1 (VController 1) and axis B of C-880 3 (VController 3), the user commands axes 1 and 26 in "PI_Multix.vi"; the axis and value parsing is done internally.

See "C880K005_Simple_Test.vi" and "C880K005_Configuration_Setup.vi" for sample programs for the driver configuration of the C-880K005.

1.5.7. E-516

Step 1 (advanced users can skip this step): To check communication with the E-516 controller, run “E516_Simple_Test.vi”. This VI will return the ID string and the help string of the E-516 controller, the available axis IDs and positions of all axes. See Section 3 for a description of this VI and 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.

Step 2:

WARNING: E516_Configuration_Setup.vi May Cause Move

When you start “E516_Configuration_Setup.vi” with Move all axes to middle? = TRUE, or Move all axes to middle? = FALSE and Axes to move = TRUE for some axes, the VI will move all axes or the selected axes to their middle positions. It is therefore important to make sure that items connected to or mounted on the connected stages will not be damaged by such a move.

Open “E516_Configuration_Setup.vi”. First select the interface settings (Interface = “RS232” or “GPIB”, RS232 settings = Portnumber and appropriate Baudrate, or GPIB settings = Bus and Address). Select whether a wave generator output is to be stopped (if you are not sure if there is any wave generator output running, leave this control TRUE) and whether the axes are to be moved to the midpoints of their travel ranges.

Then run the VI. It performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: definition of the axis IDs, the online setting of the controller, the servo setting of the axes 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 “E516_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can include “E516_Configuration_Setup.vi” as an initialization VI in your software. See Section 3 for a detailed description of “E516_Configuration_Setup.vi” and 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.

Default position unit is µm, default velocity unit is µm/ms.

To use wave-generator-specific VI’s, whose names start with “WGWaveEditor_*.vi”, “WGWaveEditor.dll” must be installed on your computer. During the installation of “WGWaveEditor.dll” “NTGraph.ocx” will be installed also. If “NTGraph.ocx” is not registered correctly in the Windows environment, the editor of “WGWaveEditor.dll” will not function.

See “E516_WaveGenerator_Sample_Program.vi” for a sample program using these VIs.

1.5.8. E-710

To control one or more E-710s with this driver set, “E7XX_GCS_DLL.dll” must be installed on your computer. See Section 1.1 for information about methods for a proper installation. The following steps must then be performed:

Step 1 (advanced users can skip this step): To check communication with the E-710 controller, run “E710_Simple_Test.vi”. This VI will return the ID string of the E-710 controller and the available axis IDs. See Section 3 for a description of this VI and 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.

Step 2:

WARNING: E710_Configuration_Setup.vi May Cause Move

When you start “E710_Configuration_Setup.vi” with Perform autozero? = TRUE and/or Move to middle? = TRUE, the VI will perform an automated zero-point calibration of the connected linear axes and/or move these axes to their middle positions. It is therefore important to make sure that items connected to or mounted on the connected stages will not be damaged by such a move.

Open “E710_Configuration_Setup.vi”. First select the interface settings (Interface = “RS232” or “GPIB”, RS232 settings = Portnumber and appropriate Baudrate, or GPIB settings = Bus and Address), and specify if stages are connected. Select whether the automated zero-point calibration is to be performed (linear axes only), whether the Low voltage parameter is to be defined automatically or manually, whether the results of the AutoZero procedure are to be saved in the non-volatile memory of the E-710 controller and whether the axes are to be moved to the midpoints of their travel ranges.

WARNING:

The repeat write times of the internal non-volatile memory of the E-710 controller are limited. Do not save AutoZero results except when necessary. See User Manual for details.

Then run the VI. It performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: the definition of the axis IDs, the initialization of the axes (if appropriate) 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 “E710_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can implement “E710_Configuration_Setup.vi” as an initialization VI in your software. See Section 3 for a detailed description of “E710_Configuration_Setup.vi” and 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.

Default axis IDs are “1”,”2”,”3”,”4”.

Default position unit is µm, default velocity unit is µm/ms.

Due to the emulation of the native E-710 command set, the execution of “MOV.vi” and “MVR.vi” is noticeably slower than that of the native firmware commands. Therefore this driver set provides the special non-GCS motion functions “NMOV.vi” and “NMVR.vi” for the case that your application requires quickest possible response to motion commands. See the VI reference of these two VIs for details.

Before using a joystick connected to the host PC, install joystick driver and calibrate joystick in the Windows Control Panel.

1.5.9. E-761

To control one or more E-761 boards with this driver set, “E7XX_GCS_DLL.dll” must be installed on your computer. See Section 1.1 for information about methods for a proper installation. The following steps must then be performed:

Step 1 (advanced users can skip this step): To check communication with the E-761 controller, run “E761_Simple_Test.vi”. This VI will return the ID string of the E-761 controller and the available axis IDs. See Section 3 for a description of this VI and 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.

Step 2:

WARNING: E761_Configuration_Setup.vi May Cause Move

When you start “E761_Configuration_Setup.vi” with Perform autozero? = TRUE and/or Move to middle? = TRUE, the VI will perform an automated zero-point calibration of the connected linear axes and/or move these axes to their middle positions. It is therefore important to make sure that items connected to or mounted on the connected stages will not be damaged by such a move.

Open “E761_Configuration_Setup.vi”. First select the Board number and specify if stages are connected. Select whether the automated zero-point calibration is to be performed (linear axes only), whether the Low voltage parameter is to be defined automatically or manually, whether the servo status is to be changed to ON, whether a wave generator output is to be stopped (if you are not sure if there is any wave generator output running, leave this control TRUE because AutoZero cannot be performed while a wave generator is running) and whether the axes are to be moved to the midpoints of their travel ranges.

Then run the VI. It performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: the definition of the axis IDs, the initialization of the axes (if appropriate) 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 “E761_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can implement “E761_Configuration_Setup.vi” as an initialization VI in your software. See Section 3 for a detailed description of “E761_Configuration_Setup.vi” and 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.

Default axis IDs are “1”,”2”,”3”. Default piezo channel IDs are “1”, “2”, “3”, “4”. Defauls sensor IDs are “1”, “2”, “3”. Default analog input board no. is “4”.

Default position unit is µm, default velocity unit is µm/ms.

Before using a joystick connected to the host PC, install joystick driver and calibrate joystick in the Windows Control Panel.

1.5.10. E-816

When controlling the E-816, timing problems can occur if several command VIs are run in rapid sequence, resulting in lost commands. To prevent such communication errors, it is recommended that you include a certain wait time between the different programming steps, depending on the command to be executed. This is especially true for commands that need a certain execution time inside the E-816 module, like MOV, MVR, SPA, SVA, SVR, RST, WPA, SWT and WTO. Only one axis per command can be controlled. “Split num query command.vi” can be used to query POS?, MOV?, VOL?, SVA? for multiple axes at a time.

1.5.11. F-206

This driver set (PI General LabVIEW Driver Set) and the F-206 LabVIEW driver set that also comes with the F-206 system are fully compatible and can be used in parallel. The F-206 can be fully controlled with the PI General LabVIEW Driver Set.The axis identifiers of the F-206 (X,Y,Z,U,V,W), NanoCube (K,L,M, if present) and additional axes (A,B, if any) cannot be changed.

Step 1 (advanced users can skip this step): To check communication between the F-206 controller and the host PC, run “F206_Simple_Test.vi”. This VI will return the ID and help strings of the F-206 controller and the axis IDs and stage names of the connected axes (according to your selection of Is a NanoCube present? and How many additional axes are present?). If you have ordered the AC8 option, you can drive up to two additional separate, motor-driven axes (PWM-compatible motors with position control) with the F-206 controller (see also the F-206 User Manual). If you have ordered the NCU option, you can drive a 3-axis piezo stage (“NanoCube”) with the F-206 controller. Before you proceed with step 2, please check that the current configuration matches your stage connections. See chapter 3 for a description of this VI and 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.

Step 2:

WARNING: F206_Configuration_Setup.vi May Cause Move

When you start “F206_Configuration_Setup.vi” with Initialize hexapod? = TRUE and/or Initialize additional axes? = TRUE, the VI will automatically move the Hexapod (and NanoCube, if present) and/or the additional axes to their 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.

To control one or more F-206 controllers with this driver set, run “F206_Configuration_Setup.vi”. This VI performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: the definition of axis IDs, the initialization of the connected stages including referencing (if appropriate) 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 “F206_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can implement “F206_Configuration_Setup.vi” as an initialization VI in your software. See chapter 3 for a detailed description of “F206_Configuration_Setup.vi” and 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.

1.5.12. M-840 / M-850

This driver set (PI General LabVIEW Driver Set) and the M-840 / M-850 LabVIEW driver set that comes with the M-840 / M-850 system are fully compatible and can be used in parallel. The M-840 / M-850 can be fully controlled with the PI General LabVIEW Driver Set and is called “M-8X0” from here on. The axis identifiers of the M-840 / M-850 and additional axes (if any) cannot be changed.

Step 1 (advanced users can skip this step): To check communication between the M-8X0 controller and the host PC, run “M8X0_Simple_Test.vi”. This VI will return the ID and help strings of the M-8X0 controller and the axis IDs and stage names of the connected axes (according to your selection of How many additional axes? are connected to the M-8X0 controller). If you have ordered the AC8 option, you can drive up to two additional separate, motor-driven axes (PWM-compatible motors with position control) with the M-8X0 controller (see also the M-8X0 User Manual). Before you proceed with step 2, please check that the current configuration matches your stage connections. See chapter 3 for a description of this VI and 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.

Step 2:

WARNING: M8X0_Configuration_Setup.vi May Cause Move

When you start “M_8X0_Configuration_Setup.vi” with Initialize hexapod? = TRUE and/or Initialize additional axes? = TRUE, the VI will automatically move the Hexapod and/or the additional axes to their 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.

To control one or more M-8X0 controllers with this driver set, run “M8X0_Configuration_Setup.vi”. This VI performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: the definition of axis IDs, the initialization of the connected stages including referencing (if appropriate) 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 “M8X0_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can implement “M8X0_Configuration_Setup.vi” as an initialization VI in your software. See chapter 3 for a detailed description of “M8X0_Configuration_Setup.vi” and 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.

1.5.13. Mercury

To control one or more Mercury controllers with this driver set, “Mercury_GCS_DLL.dll” and “PiStages.dat” must be installed on your computer. See Section 1.1 for information about methods for proper installation of these items. The following steps must then be performed:

Step 1 (advanced users can skip this step): To check communication with the Mercury controller, run “Mercury_Simple_Test.vi”. This VI will return the ID string of the Mercury controller. See Section 3 for a description of this VI and 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.

Step 2:

WARNING: Mercury_Configuration_Setup.vi May Cause Move

When you start “Mercury_Configuration_Setup.vi” with Is axis connected and can be moved? = TRUE, the VI will automatically determine if the connected axis has a reference switch or limit switch and, if the referencing mode of this axis is ON, will move the stage to one of the switches. It is therefore important to make sure that items connected to or mounted on the connected stage will not be damaged by such a move. If referencing is not possible (because the connected stage has no reference or limit switch) or not desired, referencing mode (the mode which tells the controller to reference the stage or not) can be switched off. See description of RON for details and warnings.

Open “Mercury_Configuration_Setup.vi”. Select the RS-232 settings (port number and appropriate baudrate, even if using USB) and leave Use dialog to define connected stage = TRUE. Run the VI. In the following screen, specify which stage you have connected and press OK. This VI performs all steps necessary for a full configuration of the driver VIs in the LabVIEW environment: the definition of the axis ID, the initialization of the connected stage, including referencing (if appropriate) 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 “Mercury_Configuration_Setup.vi”; otherwise all global settings will be lost and the driver VIs will not work. When programming your application, you can implement “Mercury_Configuration_Setup.vi” as an initialization VI in your software. See Section 3 for a detailed description of “Mercury_Configuration_Setup.vi” and 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.

If you do not find your stage in the drop-down list, press CANCEL. You can then either define a User Stage with the Stage Editor, or you can contact PI to see about getting a new stage list: the "PIStages.dat" file contains all relevant stage parameters.

The default axis name is determined by the device address (see Mercury User Manual), typically “A” – “P”, and can be changed using “SAI.vi”.

See also “Mercury_Sample_Application_1.vi” and “Mercury_Sample_Application_1a.vi” as sample VIs showing how to implement “Mercury_Configuration_Setup.vi” as the initialization VI for the Mercury in your application.

2. Low Level VIs

The following low-level VIs can be found in the “Low Level” folder:

2.1. Communication VIs (“Communication.llb”):

2.1.1. Close connection if open.vi

Valid for C-843, C-843.PM, C-844, C-865, E-710, E-761, Mercury (but must be present in Communication.llb for all other systems also)

Input System number (1), Error in (no error)

Output Was connected? (T/F), Error out

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

2.1.2. GCSTranslator DLL Functions.vi

Valid for C-843, C-843.PM, C-844, C-865, E-710, E-761, Mercury (but must be present in Communication.llb for all other systems also)

Input System number (1), Function (C844_IsDLLAvailable), String buffer (empty string), String input (empty string), Error in (no error)

Output DLL I32 Return value, Numerical output, Boolean output (T/F), String output, Error out

Remarks 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.

2.1.3. Global1.vi

Valid for All systems

Input None

Output None

Remarks A global variable which contains communication setup information.

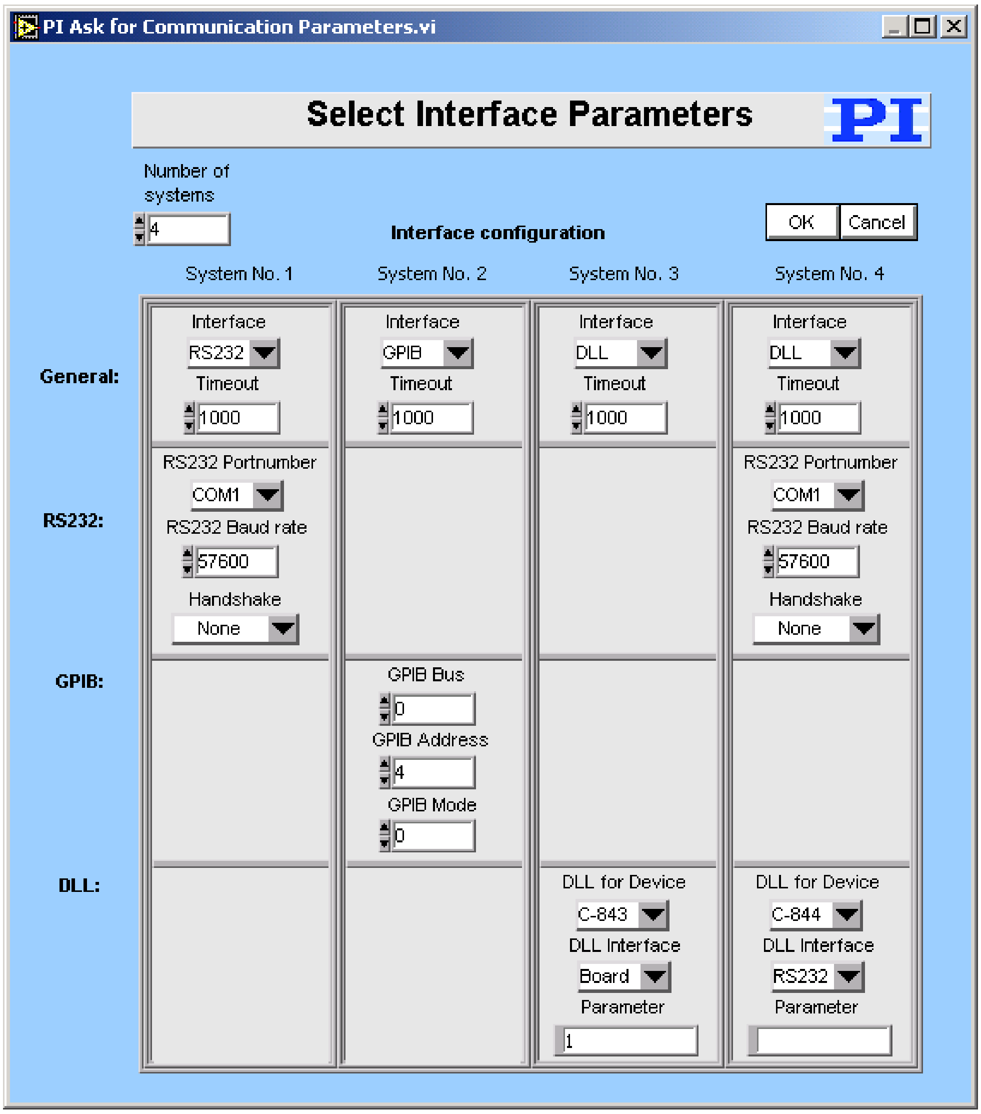
2.1.4. PI Ask for Communication Parameters.vi

Valid for All systems

Input None

Output Number of systems, Cancel (T/F), Interface configuration, DLL interface configuration, Flow control

Remarks A user-interface VI for setting up communications parameters (RS-232 or GPIB, number of systems, baudrate, timeout etc.).



Select Interface Parameters

Number of systems: 4

Interface configuration

		System No. 1	System No. 2	System No. 3	System No. 4
General:	Interface	RS232	GPIB	DLL	DLL
	Timeout	1000	1000	1000	1000
RS232:	RS232 Portnumber	COM1			COM1
	RS232 Baud rate	57600			57600
	Handshake	None			None
GPIB:	GPIB Bus		0		
	GPIB Address		4		
	GPIB Mode		0		
DLL:	DLL for Device			C-843	C-844
	DLL Interface			Board	RS232
	Parameter			1	

2.1.5. PI Open Interface.vi

Valid for All systems

Input Number of systems (1), Interface configuration (RS232, 5000, COM1, 57600), DLL Interface configuration (C-843, Board, 1), Flow control (All FALSE, x13, x11, x0)

Output Error out

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

C-843: Interface = DLL, DLL for Device = C-843, DLL Interface = Board, Parameter = Board number (1 for first C-843 board).

C-843.PM: Interface = DLL, DLL for Device = C-843.PM, DLL Interface = Board, Parameter = Board number (1 for first C-843 board).

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

C-880: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE.

C-848: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE.

C-880K005: Interface = RS232, Input and output HW handshake must be FALSE.

E-516: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE.

E-710: Interface = DLL, DLL for Device = E-710, DLL Interface = RS232 or GPIB, Parameter = empty string

E-761: Interface = DLL, DLL for Device = E-761, DLL Interface = Board, Parameter = Board number (1 for first E-761 board).

E-816: Interface = RS232 (supports only RS-232 communication), Input and output HW handshake must be TRUE.

F-206: Interface = RS232 or GPIB, 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.

M-8X0: Interface = RS232 or GPIB. RS232: Input and output handshake settings must be FALSE.

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)

2.1.6. PI Open Interface of one system.vi

Valid for: All systems

Input: System Number (1), Interface configuration (RS232, 5000, COM1, 57600), DLL Interface configuration (C-843, Board, 1), Flow control (All FALSE, x13, x11, x0)

Output: Error out

Remarks: Establishes communication with one connected system. **This VI is called automatically by “XXXX_Configuration_Setup.vi” (with XXXX being the PI product number of your system) 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-843: Interface = DLL, DLL for Device = C-843, DLL Interface = Board, Parameter = Board number (1 for first C-843 board).

C-843.PM: Interface = DLL, DLL for Device = C-843.PM, DLL Interface = Board, Parameter = Board number (1 for first C-843 board).

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

C-880: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE.

C-848: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE.

C-880K005: Interface = RS232, Input and output HW handshake must be FALSE.

E-516: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE.

E-710: Interface = DLL, DLL for Device = E-710, DLL Interface = RS232 or GPIB, Parameter = empty string

E-761: Interface = DLL, DLL for Device = E-761, DLL Interface = Board, Parameter = Board number (1 for first E-761 board).

E-816: Interface = RS232 (supports only RS-232 communication), Input and output HW handshake must be TRUE.

F-206: Interface = RS232 or GPIB, 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.

M-8X0: Interface = RS232 or GPIB. RS232: Input and output handshake settings must be FALSE.

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)

2.1.7. PI Receive String.vi

Valid for	All systems
Input	System number (1), Error in (no error)
Output	String read, Bytes read, Error out
Remarks	Read string from selected system.

2.1.8. PI ReceiveNCharacters RS232.vi

Valid for	C-848, C-880, C-880K005, E-516, E-816, F-206, M-8X0 (but must be present in Communication.llb for all other systems also)
Input	System number (1), Bytes to read (1), Error in (no error)
Output	String read, Bytes read, Error out
Remarks	Sub-vi for “PI Receive String” for RS 232 communication

2.1.9. PI ReceiveString GPIB.vi

Valid for	C-848, C-880, E-516, F-206, M-8X0 (but must be present in Communication.llb for all other systems also)
Input	System number (1), Error in (no error)
Output	String read, Bytes read, Error out
Remarks	Sub-vi for “PI Receive String” for GPIB communication.

2.1.10. PI Send String.vi

Valid for	All systems
Input	System number (1), String to send (empty string), Attach linefeed? (T), Error in (no error)
Output	Error out

Remarks Sends command with or without trailing linefeed to selected system.

2.1.11. SetQMC_SetQMCA.vi

Valid for C-843, C-843.PM, C-865

Input System number (1), Function (SetQMC), Command (0), Axis (0), Parameter (0), Parameter 1 (0), Parameter 2 (0), Error in (no error)

Output DLL I32 Return value, Error out

Remarks 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.

2.2. General Command VIs (“General command.llb”):

2.2.1. *IDN?.vi

Valid for All systems

Input System number (1), Error in (no error)

Output Identification, Error out

Remarks Returns system identification string.

2.2.2. Controller names.ctl

Valid for All systems

Input None

Output None

Remarks Type definition for control Controller names.

2.2.3. Define connected axes.vi

Valid for All systems

Input System number (1), Read from controller?(F), Invert order?(F), Connected axes (empty string array), Error in (no error)

C-843: Read from controller = TRUE, Invert order = FALSE

C-843.PM: Read from controller = TRUE, Invert order = FALSE

C-844: Read from controller = TRUE, Invert order = FALSE

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

C-865: Read from controller = TRUE, Invert order = FALSE

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

E-516: Read from controller = TRUE, Invert order = FALSE

E-710: Read from controller = TRUE, Invert order = FALSE

E-761: Read from controller = TRUE, Invert order = FALSE

E-816: Read from controller = TRUE, Invert order = FALSE

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

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

Mercury: Read from controller = TRUE, Invert order = FALSE

Output Connected axes out, Error out

Remarks Writes connected axes into Global2.vi. **This VI is called automatically by "XXXX_Configuration_Setup.vi" (with XXXX being the PI product number of your system) and must be completed successfully before any other axis-specific command VI is called.** Requires "SAI?.vi" to be present.

2.2.4. Define connected systems.vi

Valid for All systems

Input Controller names (cluster of 4 Enum controls, none), Change only one system? (F), System number (1), Error in (no error)

Output Controller names out, Error out

Remarks Defines connected systems and writes controller names into Global2.vi. **This VI is called automatically by "XXXX_Configuration_Setup.vi" (with XXXX being the PI product number of your system) 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".

2.2.5. ERR?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, E-816, F-206, M-8X0, Mercury

Input System number (1), Error in (no error)

Output Hidden error (T/F), Error out

Remarks Returns error information. Hidden error is TRUE if selected system reports error code ≠ 0. See appendix A of this manual for a list of PI error codes and use "GCSTranslateError.vi" to translate error codes into error descriptions programmatically.

2.2.6. Global2.vi

Valid for	All systems
Input	Connected axes (empty string array), Connected systems (Cluster of 4 enum controls, none)
Output	None
Remarks	A global variable which contains identifiers for all connected axes of all connected systems and the names of all connected systems.

2.2.7. HLP?.vi

Valid for	C-843, C-843.PM, C-848, C-865, C-880, E-516, E-710, E-761, Mercury
Input	System number (1), Error in (no error)
Output	Help string, Error out
Remarks	Returns help string.

2.2.8. HLT.vi

Valid for	C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, E-761, Mercury
Input	System number (1), Affected axes (empty string array), All axes? (F), Axis identifier? (T), Error in (no error) 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-865: If All axes? = TRUE, then Axis identifier? must be TRUE C-880: If All axes? = TRUE, then Axis identifier? must be TRUE E-710: If All axes? = TRUE, then Axis identifier? must be TRUE E-761: If All axes? = TRUE, then Axis identifier? can be FALSE Mercury: If All axes? = TRUE, then Axis identifier? can be FALSE
Output	Error out
Remarks	Stops motion of specified axes.

2.2.9. MOV.vi

Valid for	C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, E-816, F-206, M-8X0, Mercury
Input	System number (1), Axes to move (empty string array), Position values (empty num. array, 0), No. of digits (4), Error in (no error) C-880K005: VI only supported when called through PI_Multix.vi 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. F-206: No mix between F-206 axes X,Y,Z,U,V,W and separate axes A,B allowed
Output	Error out
Remarks	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: See also “NMOV.vi” in “Old commands.llb”.

2.2.10. MOV?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-516, E-710, E-761, E-816, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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-880: If All axes? = TRUE, then Axis identifier? can be FALSE

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-761: If All axes? = TRUE, then Axis identifier? can be FALSE

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

F-206: Command has different implementation, please use MOV?_old.vi

M-8X0: Command has different implementation, please use MOV?_old.vi

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

Output Target position, Error out

Remarks Returns commanded target position.

2.2.11. MVR.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, E-816, Mercury

Input System number (1), Axes to move (empty string array), Position values (empty num. array, 0), No. of digits (4), Error in (no error)

C-880K005: VI only supported when called through PI_Multix.vi

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.

Output Error out

Remarks 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.

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

2.2.12. ONT?.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880, E-516, E-761, E-816, Mercury (but must be present for all other systems also)

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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

E-516: If All axes? = TRUE, then Axis identifier? can be FALSE.

E-761: If All axes? = TRUE, then Axis identifier? can be FALSE.

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

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

Output Axis on target? (T/F), Error out

Remarks Indicates whether or not queried axis is at target position.

2.2.13. POS?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, E-816, F-206, M-8X0, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

C-843: If All axes?= TRUE, then Axis identifier? can be FALSE

C-843.PM: 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-865: 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

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-761: If All axes? = TRUE, then Axis identifier? can be FALSE

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

F-206: If All axes? = TRUE, then Axis identifier? must be TRUE

M-8X0: If All axes? = TRUE, then Axis identifier? must be TRUE

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

Output Position, Error out

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

C-843: Returned position value is the actual position of the axis.

C-843.PM: Returned position value is the actual position of the axis.

C-844: Returned position value is the actual position of the axis.

C-848: Returned position value is the actual position of the axis.

C-865: Returned position value is the actual position of the axis.

C-880: Returned position value is the actual position of the axis.

E-516: Returned position value is the actual position of the axis.

E-710: Returned position value is the actual position of the axis.

E-761: Returned position value is the actual position of the axis.

E-816: Returned position value is the actual position of the axis.

F-206: Returned position value is the commanded target position for the axis.

M-8X0: Returned position value is the commanded target position for the axis.

Mercury: Returned position value is the actual position of the axis.

2.2.14. SAI?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, E-816, Mercury (but must be present in "General command.llb" for all other systems also)

Input System number (1), Invert order? (F), SAI? ALL (F), Error in (no error)

C-843: Invert order should be FALSE, SAI? ALL is supported

C-843.PM: Invert order should be FALSE, SAI? ALL must be FALSE

C-844: Invert order should be FALSE, SAI? ALL must be FALSE

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

C-865: Invert order should be FALSE, SAI? ALL is supported

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-710: Invert order should be FALSE, SAI? ALL is supported

E-761: Invert order should be FALSE, SAI? ALL is supported

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

Mercury: Invert order should be FALSE, SAI? ALL is supported

Output Connected axes, Error out

Remarks Returns axis identifiers of all configured axes and writes them into Global2.vi. If SAI? ALL is TRUE, all physically available axes are returned, no matter if configured or not. Required by "Define connected axes.vi"

2.2.15. SPA.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, E-816, Mercury

Input System number (1), Axis to set (empty string array), Parameter number (empty num. array, 0), Parameter number (hex) (empty hex. array, 0), Parameter value (empty num. array, 0), No. of digits (4), Parameter string (empty string array), Parameter no. format (Decimal: FALSE) (F), Parameter format (Num.: FALSE) (F), Error in (no error)

C-843:

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).

C-843.PM: See C-843 for warnings.

C-848: See C-880 for warnings.

C-865: See C-843 for warnings.

C-880:

WARNING

This command is 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).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)
- 7: motor bias (-32767 to 32767)
- 8: maximum position error (0 to 32767)
- 9: maximum value for the motor output (0 to 32767)
- 10: maximum velocity (allowed range depends on stage)
- 11: maximum allowed acceleration (allowed range depends on stage)
- 13: maximum allowed Jerk (allowed range depends on stage)
- 14, 15: reserved

C-880K005: VI only supported when called through PI_Multix.vi. See C-880 for warnings and description of parameter numbers.

E-516:

WARNING

This command is for setting hardware-specific calibration parameters, except parameter number 268500993. Incorrect values may lead to improper operation.

The following parameter numbers are valid:

- 7: Ksen (Coefficient of Sensor K_s). When sensor output change is 1V, the position change of stage is K_s (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-710: Parameter no. format is TRUE (hex.) Use “HPA?.vi” to get valid parameter numbers or see the E7XX_GCS_DLL Manual.

WARNING

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

E-761: Parameter no. format is TRUE (hex.). Use “HPA?.vi” to get valid parameter numbers or see the E7XX_GCS_DLL Manual. See E-710 for warnings.

E-816: See E-516 for warnings and a description of parameter numbers. Each command limited to setting one parameter for only one axis.

Mercury: Parameter no. format is FALSE (decimal). See C-843 for warning.

Output Hidden error (T/F), Error out

Remarks Sets parameters, 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. 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 ≠ 0.

C-843: 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).

E-516: 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 E-516 is powered off.

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

E-710: 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 E-710 is powered off.

E-761: 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 rebootet.

Mercury: The SPA command saves the parameters in RAM only. Use PIStageEditor.exe to change parameters or add new stages to the data base permanently.

2.2.16. SPA?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, E-816, Mercury

Input System number (1), Axes to query (empty string array), Parameter no. format (Decimal: FALSE) (F), Without axes? (F), Parameter no. (empty num. array, 0), Parameter no. (hex) (empty hex. array, 0), Error in (no error)

C-865: Parameter number 25 is read-only.

C-880: 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

E-710: Use “HPA?.vi” to get valid parameter numbers.

E-761: Use “HPA?.vi” to get valid parameter numbers.

E-816: Only one parameter value for only one axis per command allowed.

Mercury: Parameter no. format is FALSE (decimal).

Output Parameter value, Parameter string, Error out

Remarks Returns parameter values for queried items and parameter numbers. For axis-related parameters, Axis 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. 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.

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

C-843: The following parameter number outputs a string:
60: stage name (maximum 14 characters)

C-843.PM: The following parameter number outputs a string:
60: stage name (maximum 14 characters)

C-865: The following parameter number outputs a string:
60: stage name (maximum 14 characters)

Mercury: The following parameter number outputs a string:
60: stage name (maximum 14 characters)

2.2.17. STP.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-516, E-761, Mercury (but must be present for E-710 also)

Input System number (1), Affected axes? (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

C-843: All axes? = TRUE, Axis identifier? = FALSE

C-843.PM: All axes? = TRUE, Axis identifier? = FALSE

C-844: All axes? = TRUE, Axis identifier? = FALSE

C-848: All axes? = TRUE, Axis identifier? = FALSE

C-865: All axes? = TRUE, Axis identifier? = FALSE

C-880: All axes? = TRUE, Axis identifier? = FALSE

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

E-761: If All axes? = TRUE, then Axis identifier? can be FALSE

Mercury: All axes? = TRUE, Axis identifier? = FALSE

Output Error out

Remarks 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”.

2.2.18. SVO.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-516, E-710, E-761, E-816, F-206, M-8X0, Mercury

Input System number (1), Without axis ID?(F), Axes to command (empty string array), Servo mode (empty bool. array, F), Error in (no error)

C-843: Without axis ID = FALSE

C-843.PM: Without axis ID = FALSE

C-844: Without axis ID = FALSE

C-848: Without axis ID = FALSE

C-865: Without axis ID = FALSE

C-880: Without axis ID = FALSE

E-516: Without axis ID = FALSE

E-710: Without axis ID = FALSE

E-761: Without axis ID = FALSE

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

F-206: Without axis ID = TRUE, only first field of Servo mode array is valid

M-8X0: Without axis ID = TRUE, only first field of Servo mode array is valid

Mercury: Without axis ID = FALSE

Output Error out

Remarks 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.

2.2.19. SVO?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-516, E-710, E-761, E-816, F-206, M-8X0, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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-880: If All axes? = TRUE, then Axis identifier? can be FALSE

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-761: If All axes? = TRUE, then Axis identifier? can be FALSE

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

F-206: All axes? = TRUE, Axis identifier? = FALSE

M-8X0: All axes? = TRUE, Axis identifier? = FALSE

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

Output Servo status (T/F), Error out

F-206: Only first field of servo status array is valid

M-8X0: Only first field of servo status array is valid

Remarks Returns servo status of queried axes.

2.2.20. VEL.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, F-206, M-8X0, Mercury

Input System number (1), Without axis ID? (F), No. of digits (4), Axes to set (empty string array), Velocity values (empty num. array, 0), Error in (no error)

C-843: Without axis ID? = FALSE

C-843.PM: Without axis ID? = FALSE

C-844: Without axis ID? = FALSE

C-848: Without axis ID? = FALSE

C-865: Without axis ID? = FALSE

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

C-880K005: VI only supported when called through PI_Multix.vi

E-516: Without axis ID? = FALSE

E-710: Without axis ID? = FALSE. Velocity unit is µm/ms.

E-761: Without axis ID? = FALSE. Velocity unit is µm/ms.

F-206: F-206 platform velocity: Without axis ID? = TRUE; velocity of axes A and/or B: Without axis ID? = False; axes K,L,M: command not valid

M-8X0: M-8X0 platform velocity: Without axis ID? = TRUE; velocity of axes A and/or B: Without axis ID? = False

Mercury: Without axis ID? = FALSE

Output Error out, Hidden error

Remarks 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. Hidden error is TRUE if selected system reports error code ≠ 0.

E-516: 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-761: 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 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 rebootet.

2.2.21. VEL?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, F-206, M-8X0, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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-880: If All axes? = TRUE, then Axis identifier? can be FALSE

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. Velocity unit is µm/ms.

E-761: If All axes? = TRUE, then Axis identifier? can be FALSE. Velocity unit is µm/ms.

F-206: Velocity of F-206: All axes? = TRUE AND Axis identifier? = FALSE; velocity of axes A,B: All axes? must be FALSE; axes K,L,M: command not valid

M-8X0: Velocity of M-8X0: All axes? = TRUE AND Axis identifier? = FALSE; velocity of axes A,B: All axes? must be FALSE

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

Output Velocity, Error out

C-880: NanoCube axes will report velocity = 0

F-206: F-206 velocity: only first field of velocity array is valid

M-8X0: M-8X0 velocity: only first field of velocity array is valid

Remarks Returns velocity setting for specified axes.

2.2.22. VER?.vi

Valid for C-848, C-880, E-516, E-761, F-206, M-8X0, Mercury

Input System number (1), Error in (no error)

Output Firmware version, Error out

Remarks Returns firmware string.

2.3. Special commands (“Special command.llb”)

2.3.1. #24.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-761, F-206, M-8X0, Mercury (but must be present for E-710 also)

Input System number (1), Error in (no error)

C-880K005: VI only supported when called through PI_Multix.vi

Output Error out

Remarks Stops motion by sending the single ASCII character 24.

2.3.2. #5.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-761, Mercury (but must be present for all other systems also)

Input System number (1), Error in (no error)

Output Axis moving? (T/F), Error out

Remarks 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”.

F-206: Different coding in answer, please use #5_old.vi

M-8X0: Different coding in answer, please use #5_old.vi

2.3.3. #7.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-710, E-761, Mercury (but must be present for E-516, F-206, M-8X0 also)

Input System number (1), Error in (no error)

C-880K005: VI only supported when called through PI_Multix.vi

Output Ready? (T/F), String read, Error out

Remarks Sends the single ASCII character 7 and returns the ready status of the controller. Sub-VI for “Wait for answer of longlasting command.vi”.

2.3.4. BRA.vi

Valid for C-843, C-843.PM, C-844, C-848, C-880, Mercury

Input System number (1), Axes to command (empty string array), Brake on (empty bool. array, F), Error in (no error)

Output Error out

Remarks Switches brake for given axes on or off.

2.3.5. BRA?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-880, Mercury

Input System number (1), Invert order? (F), Error in (no error)

C-843: Invert order should be FALSE

C-843.PM: Invert order should be FALSE

C-844: Invert order should be FALSE

C-848: Invert order should be TRUE

C-880: Invert order should be TRUE

Mercury: Invert order should be FALSE

Output Axes with brake, Error out

Remarks Returns list of axes with a brake.

2.3.6. CLR.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880 (but must be present for Mercury, also)

Input System number (1), CLR axes (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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-865: 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

Output Error out

Remarks Clears axis status for specified axes.

2.3.7. CST.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, E-761, Mercury

Input System number (1), Axis ID's (empty string array), Stage names (empty string array), Error in (no error)

Output Error out

Remarks 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.

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).

2.3.8. CST?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, E-761, F-206, M-8X0, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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? must be TRUE

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

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

E-761: If All axes? = TRUE, then Axis identifier? can be FALSE

F-206: If All axes? = TRUE, then Axis identifier? must be TRUE

M-8X0: If All axes? = TRUE, then Axis identifier? must be TRUE

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

Output Stage names, Error out

Remarks Returns the name of the connected stage for queried axes.

2.3.9. DEL.vi

Valid for C-848, C-880, E-516, Mercury

Input System number (1), Delay time (1), Error in (no error)

C-848: Delay time unit is ms.

C-880: Delay time unit is ms.

E-516: Delay time unit is ms.

Mercury: Delay time unit is ms.

Output Error out

Remarks Delays macro execution for given time and specified system.

2.3.10. DFF.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, Mercury

Input System number (1), Axis (empty string array), Factor (empty num. array, 0), No. of digits (4), Error in (no error)

Output Error out

Remarks 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.

C-843: Factor can only be positive.

C-843.PM: Factor can only be positive.

C-848: Factor can only be positive.

C-865: Factor can only be positive.

C-880: Factor can only be positive.

E-710: Factor can only be positive.

2.3.11. DFF?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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-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

Output Factor, Error out

Remarks Returns constant unit value for specified axes (e.g. 25.4 for inches).

2.3.12. DIO.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880, Mercury

Input System number (1), DO's to command (empty string array), DO mode (empty bool. array, F), Error in (no error)

C-843: DO's to command can be A-H

C-843.PM: DO's to command can be A-H

C-848: DO's to command can be A-H

C-865: DO's to command can be A-B

C-880: DO's to command can be A-H (one C-842 inside), A-P (two C-842 inside) etc.

Mercury: DO's to command can be A-D, E-H, I-L etc., see GCS DLL Manual for details.

Output Error out

Remarks Switches digital outputs on or off.

2.3.13. DIO?.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880, E-761, Mercury

Input System number (1), DI's to query (empty string array), All DI's? (F), DI identifier? (T), Invert order for TVI? (T), Error in (no error)

C-843: If All DI's = TRUE, then DI identifier must be TRUE and Invert order for TVI? must be FALSE.

C-843.PM: If All DI's = TRUE, then DI identifier must be TRUE and Invert order for TVI? must be FALSE.

C-848: If All DI's = TRUE, then DI identifier can be FALSE and Invert order for TVI? must be TRUE

C-865: If All DI's = TRUE, then DI identifier must be TRUE and Invert order for TVI? must be FALSE.

C-880: If All DI's = TRUE, then DI identifier can be FALSE and Invert order for TVI? must be TRUE

E-761: If All DI's = TRUE, then DI identifier can be FALSE and Invert order for TVI? must be FALSE. DI's to query are "1".

Mercury: If All DI's = TRUE, then DI identifier can be FALSE and Invert order for TVI? must be FALSE. DI's to query can be A-D, E-H, I-L etc., see GCS DLL Manual for details.

Output DI value (T/F), Error out

Remarks Returns digital input values for queried digital inputs. Uses “TIO?.vi” and “TVI?.vi” to determine available DI identifiers if All DI’s = TRUE.

E-761: 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 < 1.25 V, the signal is interpreted as LOW, if the voltage is ≥ 1.25 V, the signal is interpreted as HIGH.

2.3.14. INI.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, F-206, M-8X0, Mercury

Input System number (1), INI axes (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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? must be TRUE

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

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

F-206: Initialize F-206: All axes? = TRUE, Axis identifier? = FALSE (separate axes A,B (if present) will not be initialized); initialize axes A,B: All axes? = FALSE, axes K,L,M: command not valid (NanoCube is initialized with F-206); VI will not wait for INI procedure to finish

M-8X0: Initialize M-8X0: All axes? = TRUE, Axis identifier? = FALSE, (separate axes A,B (if present) will not be initialized); initialize axes A,B: All axes? = FALSE; VI will not wait for INI procedure to finish. Use “INI hexaxes and wait until finished.vi” to initialize hexapod and/or separate axes and wait for the procedure to finish.

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

Output Error out

Remarks Initializes axes. System-specific: see individual GCS-DLL Manual for details.

C-844: It is necessary to wait a certain time – appr. 4 s – before sending the next command to prevent it from being lost.

2.3.15. POS.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880, C-880K005, Mercury

Input System number (1), Axes to set (empty string array), Position values (empty num. array, 0), No. of digits (4), Error in (no error)

C-880K005: VI only supported when called through PI_Multix.vi

Output Error out

Remarks Assigns new position value to current position without moving the stage. 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.

2.3.16. SAI.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, E-761, Mercury

Input System number (1), Old axis ID (empty string array), New axis ID (empty string array), Error in (no error)

Output Error out

Remarks Set axis identifier. With this command the axis identifiers of the configured axes can be changed. Only one character is allowed as axis ID. Valid axis IDs can be listed with TVI?.vi. Please run “Define connected axes.vi” with new axis IDs after renaming axes.

E-761: The settings are automatically written to non-volatile memory.

2.3.17. SRG?.vi

Valid for C-843, C-843.PM, Mercury

Input System number (1), Axes to query (empty string array), Register no. (empty num. array, 0), Error in (no error)

C-843, C-843.PM: Register numbers are:

- 1: Event status register
- 2: Activity status register
- 3: Signal status register

Mercury: Register number is:

- 3: Signal status register

Output Parameter value, Error out

Remarks Returns register values for queried axes and register numbers. See User Manual for a description of bit-coded answer.

2.3.18. STA?.vi

Valid for C-848, C-880, C-880K005 (but must be present in Special command.llb for all other systems also)

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

C-848: If <u>All axes? </u>= TRUE, then <u>Axis identifier? </u>can be FALSE

C-880: If <u>All axes? </u>= TRUE, then <u>Axis identifier? </u>can be FALSE

C-880K005: VI only supported when called through PI_Multix.vi

Output Axis status, Error out

C-848, C-880:

The status word for each axis is a 16-bit register containing the following information (bit encoding is 0 = LSB, 15 = MSB):

Bit #	Description
0	Motion complete flag. This bit is set (1) when the axis trajectory has completed. This flag is only valid for the S-curve, trapezoidal, and velocity

contouring profile modes.

1 Wrap-around condition flag. This bit is set (1) when the axis has reached one end of its travel range and has wrapped to the other end of the travel range. Specifically, when traveling in a positive direction past the position +1,073,741,823, the axis will wrap to position -1,073,741,824, and vice-versa. The bit can be reset with the CLR command.

2 Breakpoint reached flag. This bit is set (1) when one of the breakpoint conditions has occurred.

3 Index pulse received flag. This bit is set (1) when an index pulse has been received.

4 Motion error flag. This bit is set (1) when the maximum position error is exceeded. This bit can only be reset when the axis is no longer in a motion error condition

5 Positive limit switch flag. This bit is set (1) when the positive limit switch goes active.

6 Negative limit switch flag. This bit is set (1) when the negative limit switch goes active.

7 Command error flag. This bit is set (1) when an erroneous command has been received by the motion control chip.

8* Servo-control on/off status (1 indicates on, 0 indicates off).

9* Axis on/off status (1 indicates on, 0 indicates off). The C-848 always has the axis ON.

10* In-motion flag. This bit is continuously updated and indicates whether or not the axis is in motion: 1 indicates axis is in motion, 0 not in motion.

11* Reserved (may contain 0 or 1)

12*,13* Current axis # (13 bit = high bit, 12 bit = low bit). Axis encoding is as follows:

Bit 13	Bit12	MC Axis	C-848 Axis
0	0	1	A
0	1	2	B
1	0	3	C
1	1	4	D

14,15 Reserved (may contain 0 or 1)

C-880K005:

The status word for each axis is a 16-bit register containing the following information (bit encoding is 0 = LSB, 15 = MSB):

Bit # Description

0 Motion complete flag. Set to 1 when motion is completed. SetMotionCompleteMode determines if this bit is based on the trajectory generator position or the encoder position.

1 Wrap-around condition flag. This bit is set (1) when the actual (encoder) position wraps from maximum allowed position to minimum or vice versa.

2 Breakpoint 1 reached flag. This bit is set (1) when breakpoint 1 is triggered.

3 Capture received flag. This bit is set (1) when a position caputre occures.

4 Motion error flag. This bit is set (1) when a motion error occurs

5 Positive limit switch flag. This bit is set (1) when the positive limit switch goes

active.

6 Negative limit switch flag. This bit is set (1) when the negative limit switch goes active.

7 Instruction error flag. This bit is set (1) when an instruction error occurs.

8-10 Reserved, may be 0 or 1.

11 Commutation error flag. This bit is set (1) when a commutation error occurs.

12-13 Reserved, may be 0 or 1.

14 Breakpoint 2 reached flag. This bit is set (1) when breakpoint 2 is triggered.

15 Reserved, may be 0 or 1.

Remarks Returns axis status (integer). Required by “General wait for movement to stop.vi” and “Wait for axes to stop.vi”.

2.3.19. TIO?.vi

Valid for C-843, C-843.PM, C-848, C-880, E-761, Mercury

Input System number (1), Error in (no error)

Output No. of dig. inputs, No. of dig. outputs, Error out

Remarks 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 the DIO? command.

2.3.20. TVI?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, E-761, Mercury

Input System number (1), Invert order, Error in (no error)

C-843: Invert order must be FALSE

C-843.PM: Invert order must be FALSE

C-844: Invert order must be FALSE

C-848: Invert order should be TRUE

C-865: Invert order must be FALSE

C-880: Invert order should be TRUE

E-710: Invert order must be FALSE

E-761: Invert order must be FALSE

Mercury: Invert order must be FALSE

Output Valid axis IDs, Error out

Remarks Get valid axis identifiers. Should be called before axes are renamed with SAI.vi.

2.3.21. VST?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, E-761, Mercury

Input System number (1), Error in (no error)

Output Available stages, Error out

Remarks Returns the names of all stages which can be connected to the controller.

2.4. Old commands and commands with alternate implementations (“Old commands.llb”)

2.4.1. #5_old.vi

Valid for F-206, M-8X0 (but must be present for all other systems also)

Input System number (1), Error in (no error)

Output Overall system moving? (T/F), Sep. Axis 1 moving? (T/F), Sep. Axis 2 moving? (T/F), Error out

Remarks 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”.

2.4.2. Wait for hexapod system axes to stop.vi

Valid for F-206, M-8X0 (but must be present for all other systems also)

Input System number (1), All axes? (T), Axes to wait for (empty string array), Stop refnum (F), Local stop (F), Error in (no error)

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.

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

Output Error out

Remarks 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 occured, or if Local stop or Stop refnum is TRUE. When using as a sub-VI, use Refnum stop to stop VI from caller. Required by “General wait for movement to stop.vi”.

2.5. Limit- and reference-specific commands ("Limits.llb")

2.5.1. DFH.vi

Valid for: C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-710, E-761, Mercury

Input: System number (1), DFH axes (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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? must be TRUE

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

C-880K005: VI only supported when called through PI_Multix.vi

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

E-761: If All axes? = TRUE, then Axis identifier? must be TRUE

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

Output: Error out

Remarks: Defines the 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, REF, MNL, MPL or similar commands.

E-761: The values are saved with WPA 100.

2.5.2. DFH?.vi

Valid for: C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-710, E-761, Mercury

Input: System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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-880: If All axes? = TRUE, then Axis identifier? can be FALSE

C-880K005: VI only supported when called through PI_Multix.vi

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

E-761: If All axes? = TRUE, then Axis identifier? must be TRUE

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

Output: Home position, Error out

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

2.5.3. GOH.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-710, E-761, Mercury

Input System number (1), GOH axes (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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? must be TRUE

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

C-880K005: VI only supported when called through PI_Multix.vi

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

E-761: If All axes? = TRUE, then Axis identifier? can be FALSE

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

Output Error out

Remarks Moves specified axes to their home positions.

2.5.4. LIM?.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

C-843: If All axes? = TRUE, then Axis identifier? can be FALSE

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

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

C-865: If All axes? = TRUE, then Axis identifier? can be FALSE

C-880: If All axes? = TRUE, then Axis identifier? can be FALSE

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

Output Axis with limit switch? (T/F), Error out

Remarks Indicates whether queried axes have limit switches or not.

2.5.5. MNL.vi

Valid for C-843, C-843.PM, C-844, C-865, C-848, C-880, Mercury

Input System number (1), All axes? (F), Affected axes (empty string array), Stop refnum (F), Local stop (F), Error in (no error)

Output MNL successful? (T/F), Error out

Remarks 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.

2.5.6. MPL.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, Mercury

Input System number (1), All axes? (F), Affected axes (empty string array), Stop refnum (F), Local stop (F), Error in (no error)

Output MPL successful? (T/F), Error out

Remarks 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.

2.5.7. REF.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, Mercury

Input System number (1), All axes? (F), Affected axes (empty string array), Stop refnum (F), Local stop (F), Error in (no error)

C-880K005: VI only supported when called through PI_Multix.vi

Output REF successful? (T/F), Error out

Remarks 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. The home position is reset to default by REF.

2.5.8. REF?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

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-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

Output Axis with reference? (T/F), Error out

Remarks Indicates whether queried axes have a reference switch or not.

2.5.9. RON.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880, C-880K005, Mercury

Input System number (1), Without axis ID?(F), Axes to command (empty string array), Reference mode (empty bool. array, F), Error in (no error)

C-843: Without axis ID = FALSE

C-843.PM: Without axis ID = FALSE

C-848: Without axis ID = FALSE

C-865: Without axis ID = FALSE

C-880: Without axis ID = FALSE

C-880K005: VI only supported when called through PI_Multix.vi

Mercury: Without axis ID = FALSE

Output Error out

Remarks 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, if no reference switch is available, to a limit switch (using “MPL.vi” or “MNL.vi”) before any other motion can be commanded.

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.

WARNINGS:

If reference mode is switched off, and relative movements 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.

2.5.10. RON?.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880, C-880K005, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

C-843: If All axes? = TRUE, then Axis identifier? can be FALSE

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

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

C-865: 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

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

Output Reference on? (T/F), Error out

Remarks Indicates whether queried axes have reference mode ON or OFF. See “RON.vi” above for description of reference mode.

2.5.11. TMN?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, E-761, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

C-843, 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-880: If All axes? = TRUE, then Axis identifier? can be FALSE

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

E-761: If All axes? = TRUE, then Axis identifier? can be FALSE

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

Output Minimum travel limit, Error out

Remarks Returns minimum (low-end) travel limit (position of negative limit switch, or value of negative soft limit, if set, whichever is higher).

E-761: Get the minimum accessible position value, i.e. the value of the "Range min limit" parameter (ID 0x07000000). Note: The minimum position which can be commanded depends either on the "Range min limit" parameter or—if it is greater than the "Range min limit" parameter value— on the value of the negative soft limit set with NLM.

2.5.12. TMX?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, E-761, Mercury

Input System number (1), Axes to query (empty string array), All axes? (F), Axis identifier? (T), Error in (no error)

C-843, 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-880: If All axes? = TRUE, then Axis identifier? can be FALSE

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

E-761: If All axes? = TRUE, then Axis identifier? can be FALSE

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

Output Maximum travel limit, Error out

Remarks Returns maximum (high-end) travel limit (position of positive limit switch or value of positive soft limit, if set, whichever is lower).

E-761: 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.

2.6. Macro Functions ("Macros.llb")

2.6.1. Define macro contents.vi

Valid for	C-844, C-848, C-880, F-206, Mercury
Input	System number (1), Macro contents (empty string), Error in (no error)
Output	Error out
Remarks	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. Macros are stored as entered and may be affected by any change of scale factor before execution.

2.6.2. MAC BEG.vi

Valid for	C-844, C-848, C-880, E-516, F-206, Mercury
Input	System number (1), Macro name (empty string), Error in (no error) E-516: Macro name must be between 3 and 8 characters. F-206: Macro name must be between 3 and 8 characters. Mercury: For valid Macro names see GCS DLL Manual.
Output	Error out
Remarks	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. 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.

2.6.3. MAC DEL.vi

Valid for	C-844, C-848, C-880, E-516, F-206, Mercury
Input	System number (1), Macro name (empty string), With dialog? (T), Error in (no error)
Output	Hidden error (T/F), Error out
Remarks	Delete macro. If "With dialog" is TRUE, a dialog box pops up to confirm the deletion. Hidden error is TRUE if selected system reports error code ≠ 0. E-516: Changes not saved with WPA.vi are only present in RAM and will be lost when the E-516 is powered off.

2.6.4. MAC END.vi

Valid for	C-844, C-848, C-880, E-516, F-206, Mercury
Input	System number (1), Error in (no error)
Output	Error out
Remarks	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 E-516 is

powered off.

2.6.5. MAC NSTART.vi

Valid for	E-516, Mercury
Input	System number (1), Macro name (empty string), N (1), Error in (no error)
Output	Error out
Remarks	Start macro N times.

2.6.6. MAC START.vi

Valid for	C-844, C-848, C-880, E-516, F-206, Mercury
Input	System number (1), Macro name (empty string), Error in (no error)
Output	Error out
Remarks	Start macro. C-848: Use #8.vi to determine when macro execution has finished. C-880: Use #8.vi to determine when macro execution has finished. E-516: Use #8.vi to determine when macro execution has finished.

2.6.7. MAC?.vi

Valid for	C-844, C-848, C-880, E-516, F-206, Mercury
Input	System number (1), Get contents? (F), Macro name (empty string), Error in (no error)
Output	Macro names or contents, Error out
Remarks	If Get contents is FALSE, returns names of all available macros, if TRUE, returns contents of one specified macro.

2.7. Commands for Optical or Analog Signals (“Optical or Analog Input.llb”)

2.7.1. MOV and TAV?.vi

Valid for C-865, C-880, E-761, F-206, M-8X0, Mercury

Input System number (1), Board (1), No. of digits (4), Wait before TAV?, ms (1), Polling cycle time, ms (1), Axes to move (empty string array), Position values (empty num. array), Error in (no error)

Output Analog value, Error out

Remarks 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.

2.7.2. TAC?.vi

Valid for C-880, Mercury

Input System number (1), Error in (no error)

Output Analog channels, Error out

Remarks Returns the number of installed analog channels.

2.7.3. TAV?.vi

Valid for C-865, C-880, E-761, F-206, M-8X0, Mercury

Input System number (1), Board (1), Query (Value), Error in (no error)

C-880: Query = Value

C-880: Query = Value

E-761: Query = Value, Board = 4.

F-206: Query can be Value, Range or Power unit.

M-8X0: Query can be Value, Range or Power unit.

Mercury: Query = Value. Board = analog input channel ID, can be 1-4, 5-7, 8-11 etc., see GCS DLL Manual for details

Output Analog value, Range, Power unit, Error out

C-865: Range and Power unit are not valid

C-880: Range and Power unit are not valid

E-761: Range and Power unit are not valid

Mercury: Range and Power unit are not valid

Remarks 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.

E-761: The output is the current voltage at the analog input line, with gain and offset.

2.8. Support VIs (“Support.llb”)

Support VIs are sub-VIs for command VIs which make certain programming tasks more convenient. They can also be used for building main programs.

Caution: Please do not change these VIs, as that might cause the command VIs that use them to fail.

2.8.1. Analyse input string for terminal.vi

Valid for All systems

Input String new (empty string), Last string sent (empty string)

Output String out, Out not equal to in? (T/F)

Remarks 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.

2.8.2. Assign booleans from string to axes.vi

Valid for All Systems

Input System number (1), Queried axes (empty string array), All axes queried? (F), Input string (empty string), Error in (no error)

Output Booleans(T/F), Error out

Remarks 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”.

2.8.3. Assign NaN for chosen axes.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880, F-206, M-8X0, Mercury

Input Queried axes (empty string array), Values (empty num. array), Axes subset (empty string array)

Output New values,

Remarks This VI returns “NaN” for the given axes subset.

2.8.4. Assign SPA values from string to axes.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, E-816, Mercury

Input Input string (empty string), Queried axes (empty string array), Parameter number (empty num. array, 0), Error in (no error)

Output Parameter values, Parameter strings, Error out

Remarks This VI assigns numerical values / strings from input string to queried axes and parameter numbers. Sub-VI for “SPA?.vi”, “SPA?_String.vi” and “SEP?.vi”.

2.8.5. Assign SPA_Hex values from string to axes.vi

Valid for	E-710, E-761 (but must be present for C-843, c-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-816 and Mercury also)
Input	Input string (empty string), Queried axes (empty string array), Parameter number (hex) (empty num. array, 0), Error in (no error)
Output	Parameter values, Parameter strings, Error out
Remarks	This VI assigns numerical values / strings from input string to queried axes and hex. parameter numbers. Sub-VI for “SPA?.vi”, “SPA?_Hex.vi”, “SPA?_Hex_String.vi”.

2.8.6. Assign values from string to axes.vi

Valid for	All systems
Input	System number (1), Queried axes (empty string array), All axes queried? (F), Input string (empty string), Error in (no error)
Output	Values, Strings, Error out
Remarks	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.

2.8.7. Build command substring.vi

Valid for	All systems
Input	Affected axes (empty string array), No. of digits (4), Parameters (empty num. array, 0), Parameters (hex.) (empty hex. array), Parameter no. format (Decimal: FALSE) (F)
Output	Command substring
Remarks	This VI builds a command substring by combining axis identifier and parameter. If parameter number 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”.

2.8.8. Build DIO? query command substring.vi

Valid for	C-843, C-843.PM, C-848, C-865, C-880, E-761, Mercury
Input	System number (1), DI’s to query in (empty string array), Query all DI’s? (F), DI identifer? (T), Invert order for TVI? (T), Error in (no error)
Output	Command substring, DI’s to query out, Number of rows, Error out
Remarks	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). 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.

2.8.9. Build num command substring.vi

Valid for	All systems
Input	No. of digits (4), Num 1 (empty num. array, 0), Num 2 (empty num. array, 0)
Output	Command substring
Remarks	This VI builds a command substring by combining Num1, Space and Num2. 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"

2.8.10. Build query command substring.vi

Valid for	All systems
Input	System number (1), Axes to query in (empty string array), Query all axes? (F), With space? (F), Axis identifer? (T),
Output	Command substring, Axes to query out, Number of rows
Remarks	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, 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. If With space? is TRUE, the resulting Command substring is “A B C D”.

2.8.11. Build SPA command substring.vi

Valid for	C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-816, Mercury (but must be present for E-710 and E-761, also)
Input	Axes to set (empty string array), No. of digits (4), Parameter number (empty num. array, 0), Parameter values (empty num. array, 0)
Output	SPA command substring
Remarks	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”.

2.8.12. Build SPA query command substring.vi

Valid for	C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-710, E-761, E-816, Mercury
Input	Axes to query (empty string array), Parameter number (empty num. array, 0)
Output	Command substring, Number of rows
Remarks	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” .

2.8.13. Build SPA_Hex command substring.vi

Valid for E-710, E-761 (but must be present for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-816 and Mercury also)

Input Axes to set (empty string array), No. of digits (4), Parameter number (hex) (empty num. array, 0), Parameter values (empty num. array, 0)

Output SPA_Hex command substring

Remarks This VI builds a command substring for the SPA_Hex 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” and “SPA_Hex.vi”.

2.8.14. Build SPA_Hex query command substring.vi

Valid for E-710, E-761 (but must be present for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-816 and Mercury also)

Input Axes to query (empty string array), Parameter number (hex) (empty num. array, 0)

Output Command substring, Number of rows

Remarks This VI builds an SPA?_Hex 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 “SPA?_Hex.vi” .

2.8.15. Build SPA_Hex_String command substring.vi

Valid for E-710, E-761 (but must be present for C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-516, E-816 and Mercury also)

Input Axes to set (empty string array), Parameter number (hex) (empty num. array, 0), Parameter values (empty string array)

Output SPA_Hex_String command substring

Remarks This VI builds a command substring for the SPA_Hex_String command. Sub-VI for “SPA.vi” and “SPA_Hex_String.vi”.

2.8.16. Build SPA_String command substring.vi

Valid for C-843, C-843.PM, C-865, E-710, Mercury (but must be present for C-844, C-848, C-880, C-880K005, E-516, E-761 and E-816 also)

Input Axes to set (empty string array), Parameter number (empty num. array, 0), Parameter values (empty string array)

Output SPA_String command substring

Remarks This VI builds a command substring for the SPA_String and SEP command. Sub-VI for “SPA.vi”, “SPA_String.vi” and “SEP.vi”.

2.8.17. Build stringplusnum substring.vi

Valid for All systems

Input Sequence (String1String2String3Value1Value2), String1 (empty string array), String2 (empty string array), String3 (empty string array), Value1 (empty num. array, 0), Value2 (empty num. array, 0), No. of digits Value1 (6), No. of digits Value2 (6), Input selection (T,T,T,T,F), Error in (no error)

Output Substring, Error out

Remarks This vi builds a command substring by combining up to three strings and two values in the given order.

2.8.18. Commanded axes connected?.vi

Valid for All systems

Input System number (1), Commanded axes (empty string array), Error in (no error)

Output Hidden error (T/F), Error out

Remarks This VI checks if Commanded axes are a subset of all connected axes (read from “Global2.vi”) and returns Hidden error TRUE if this is not the case. Connected axes are defined by “Define connected axes.vi”. White space strings in Commanded axes are ignored.

2.8.19. Commanded stage name available?.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, E-761, Mercury

Input System number (1), Commanded stages (empty string array), Error in (no error)

Output Hidden error (T/F), Error out

Remarks This VI checks if Commanded stages is a subset of all available stages and returns Hidden error TRUE if this is not the case. Available stages are defined by “VST?.vi”.

2.8.20. Convert num array to string.vi

Valid for All systems

Input Number of digits (4), Num. values (empty num. array)

Output Output string

Remarks 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.

2.8.21. Count occurrences in string.vi

Valid for All systems

Input Input string (empty string), Expression (empty string)

Output Occurrences

Remarks This VI counts, how often an expression occurs in a string.

2.8.22. Define axes to command from boolean array.vi

Valid for All systems

Input Axes to query (empty string array), Command axis? (empty bool. array, F)

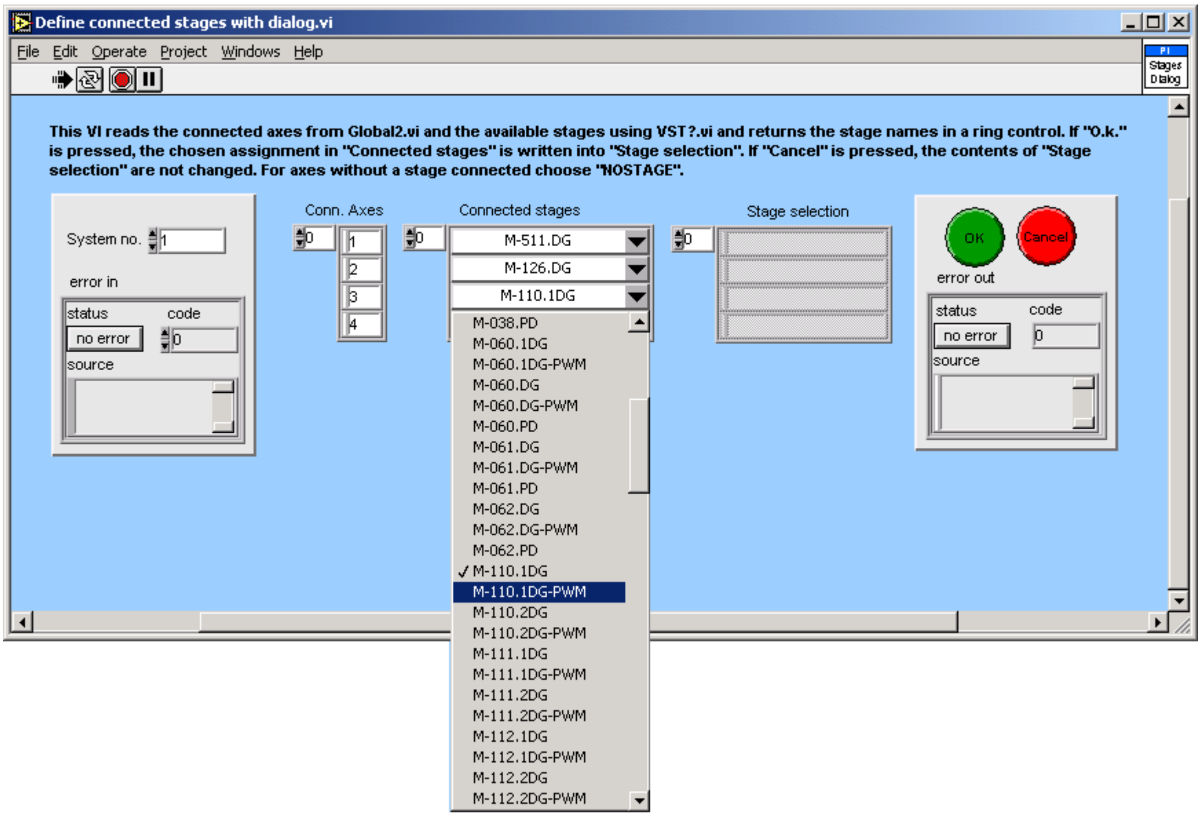
Output Axes to command, Remaining axes

Remarks This VI returns only those axis IDs from the Axes to query array in the Axes to command array which have a boolean value TRUE in the Command axis? array, and all remaining axes in the Remaining axes array.

2.8.23. Define connected stages with dialog.vi

Valid for	C-843, C-843.PM, C-844, C-865, Mercury
Input	System number (1), Error in (no error)
Output	Stage selection, Error out
Remarks	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.** See chapter 1.1 for information about methods for proper registration of PIStages.dat. 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”.



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 "O.k." 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".

System no. 1

error in: status no error, code 0, source

Conn. Axes: 0; 1, 2, 3, 4

Connected stages: 0; M-511.DG, M-126.DG, M-110.1DG

Stage selection: 0

error out: status no error, code 0, source

2.8.24. GCSTranslateError.vi

Valid for	All systems
Input	Error in (no error)
Output	Error out, GCS Error?, Error description
Remarks	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.

2.8.25. General wait for movement to stop.vi

Valid for	All systems

Input System no. (1), Axes to wait for (empty string array), All axes? (T), Polling cycle time, ms (1), Additional wait time, ms (0), Error in (no error)

E-710: VI will not wait, use appropriate Additional wait time to wait for commanded moves.

F-206: VI will not wait for INI procedure to complete.

M-8X0: VI will not wait for INI procedure to complete.

Output Error out

Remarks 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. “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.

2.8.26. Get arrays without blanks.vi

Valid for All systems

Input String array in (empty string array), Values in (empty num. array), Booleans in (empty bool. array, F), Array size in (0)

Output String array out, Values out, Booleans out, Array size out

Remarks Returns the string array and related values and boolean arrays without white space string fields.

2.8.27. Get lines and values from string.vi

Valid for All systems

Input Array size (0), Input string (empty string)

Output Numerical values, Strings

Remarks 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”.

2.8.28. Get lines from string.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, E-710, E761, Mercury

Input Array size (0), Input string (empty string)

Output Strings

Remarks 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”.

2.8.29. Get string array size without blanks.vi

Valid for All systems

Input String array (empty string array)

Output Corrected array size

Remarks This VI returns the size of a string array without counting white space strings.

2.8.30. How often does string contain regular expression.vi

Valid for	All systems
Input	Regular expression (empty string), String (empty string)
Output	Number
Remarks	This VI returns a count of the occurances of a regular expression in a string.

2.8.31. Longlasting one-axis command.vi

Valid for	C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-710, E-761, F-206, Mercury
Input	System number (1), Axis to command (empty string), Command (empty string), Axis and value? (F), Value (NaN), Error in (no error)
Output	Answer (T/F), Error out
Remarks	This VI sends a command (like REF, MNL or MPL), polls with #7 for controller-ready signal and returns original (boolean) command response.

2.8.32. Return single characters from string.vi

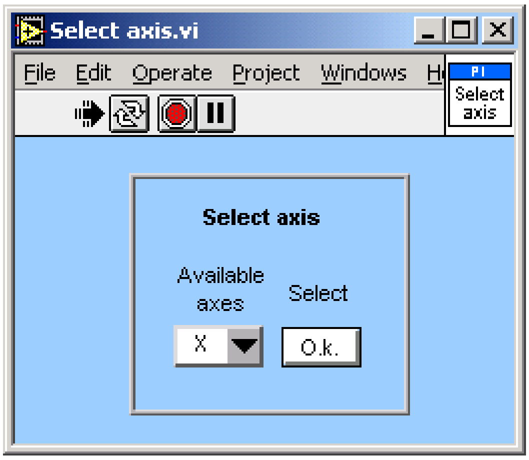
Valid for	All systems
Input	Input string (empty string), Invert order (F), Error in (no error)
Output	Character array (empty string array), Error out
Remarks	Get single characters from input string.

2.8.33. Round with options.vi

Valid for	All systems
Input	No. of digits to round to (2), Round mode selection (Round to nearest), Numeric in (0), Num array in (empty num. array)
Output	Numeric out, Num array out
Remarks	Rounds Numeric in and Num array in according to No. of digits to round to and Round mode selection.

2.8.34. Select axis.vi

Valid for	All systems
Input	System number (1)
Output	Selected axis, Index of axis in Global2
Remarks	This VI reads all connected axes from Global2 and writes them into a menu ring control for selection. The selected axis and it’s index in Global2 are returned.



2.8.35. Select values for chosen axes.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880, C-880K005, E-710, E-761, Mercury

Input Queried axes (empty string array), Values (empty num. array), Axes subset (empty string array)

Output Values subset

Remarks This VI returns only values for the given axes subset.

2.8.36. Select with boolean array input.vi

Valid for All systems

Input Size (0), T string (empty string), F string (empty string), T/F (empty boolean array)

Output String array out

Remarks 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.

2.8.37. Selection to string array.vi

Valid for All systems

Input Selection array (empty Menu Ring array, 0), String input (empty string array)

Output String array

Remarks 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).

2.8.38. Set RON and return RON status.vi

Valid for C-843, C-843.PM, C-848, C-865, C-880, Mercury

Input System number (1), All axes? (F), Affected axes (empty string array), Reference mode (empty bool. Array, F), Error in (no error)

Output Axes with RON OFF, Axes with RON ON, Error out

Remarks 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.

2.8.39. Substract axes array subset from axes array.vi

Valid for	All systems
Input	Axes to query (empty string array), Axes subset (empty string array)
Output	Axes to command
Remarks	This VI returns only these axes IDs from the Axes to query array which are **not** present in the Axes subset array. Needed by “Define axes to command from boolean array.vi”.

2.8.40. Unbundle/bundle interface clusters for PI Terminal.vi

Valid for	All systems
Input	System number (1), Interface configuration (RS232, 1000, COM1, 57600), DLL interface configuration (C-843, Board, 1), Flow control (All FALSE, x13, x11, x0)
Output	Interface, RS232 configuration system, GPIB configuration system, DLL for device, DLL interface
Remarks	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”.

2.8.41. Wait for answer 0 or 1 without polling.vi

Valid for	C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, F-206, M-8X0, Mercury
Input	System number (1), Start time (0), Timeout (s) (60), Stop refnum (F), Local stop (F), Error in (no error)
Output	Scan successful? (T/F), Error out
Remarks	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.

2.8.42. Wait for answer of longlasting command.vi

Valid for	C-843, C-843.PM, C-844, C-848, C-865, C-880, C-880K005, E-710, E-761, Mercury (but must be present for E-516, F-206, M-8X0, too)
Input	System number (1), Stop refnum (F), Local stop (F), Error in (no error)

Output Answer (T/F), Error out

Remarks 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 occured. 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.

2.8.43. Wait for axes to stop.vi

Valid for C-843, C-843.PM, C-844, C-848, C-865, C-880, Mercury (but must be present in Support.llb for all other systems also)

Input System number (1), Axes to wait for (empty string array), With status bit polling? (F), Polling cycle time, ms (400), Stop refnum (F), Local stop (F), Error in (no error)

C-843: With status bit polling? = FALSE

C-843.PM: With status bit polling? = FALSE

C-844: With status bit polling? = FALSE

C-880: With status bit polling? = TRUE

Mercury: With status bit polling? = FALSE

Output Error out

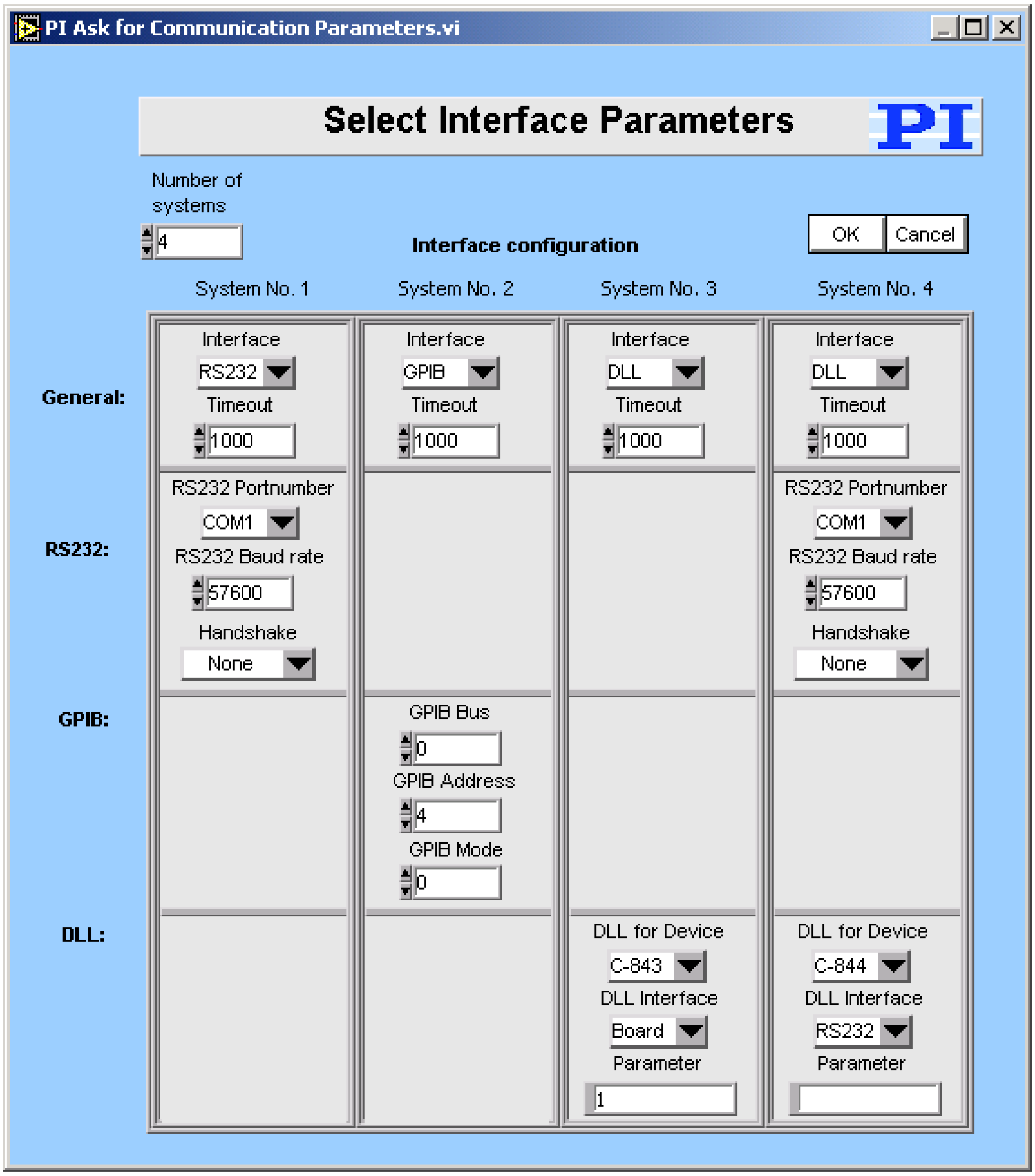
Remarks This VI waits for the specified axes to stop using #5 polling. It also stops if a communication error occured, Stop refnum or Local stop is TRUE. Requires “STA?.vi” to be present. Required by “General wait for movement to stop.vi”. When using as a sub-VI, use Stop refnum to stop VI from caller.

3. High Level VIs

3.1. PI Terminal.vi

The terminal VI is a stand-alone application. It first asks the user to specify the full configuration (number of controlled systems, RS-232, GPIB or DLL communication, communications parameters), then it establishes a connection with a selected system. This will work for all PI devices which support the PI General Command Set, or at least follow the same syntax rules and support the *IDN? and ERR? commands.

After starting the VI, the interface parameters of the systems with which to communicate must be selected. For this reason, “PITerminal.vi” calls “PI Ask for Communication Parameters.vi”. Select here the number of connected PI systems that you want to communicate with. For each system, select the appropriate interface parameters.



PI Ask for Communication Parameters.vi

Select Interface Parameters

Number of systems: 4

Interface configuration

	System No. 1	System No. 2	System No. 3	System No. 4
General:	Interface: RS232 Timeout: 1000	Interface: GPIB Timeout: 1000	Interface: DLL Timeout: 1000	Interface: DLL Timeout: 1000
RS232:	RS232 Portnumber: COM1 RS232 Baud rate: 57600 Handshake: None			RS232 Portnumber: COM1 RS232 Baud rate: 57600 Handshake: None
GPIB:		GPIB Bus: 0 GPIB Address: 4 GPIB Mode: 0		
DLL:			DLL for Device: C-843 DLL Interface: Board Parameter: 1	DLL for Device: C-844 DLL Interface: RS232 Parameter:

C-843: Interface = DLL, DLL for Device = C-843, DLL Interface = Board, Parameter = Board number (1 for first C-843 board).

C-843.PM: Interface = DLL, DLL for Device = C-843.PM, DLL Interface = Board, Parameter = Board number (1 for first C-843 board).

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

C-880: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE.

C-848: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE.

C-880K005: Interface = RS232, Input and output HW handshake must be FALSE.

E-516: Interface = RS232 or GPIB, RS232: Input and output HW handshake must be TRUE.

E-710: Interface = DLL, DLL for Device = E-710, DLL Interface = RS232 or GPIB, Parameter = empty string

E-761: Interface = DLL, DLL for Device = E-761, DLL Interface = Board, Parameter = Board number (1 for first E-761 board).

E-816: Interface = RS232 (supports only RS-232 communication), Input and output HW handshake must be TRUE.

F-206: Interface = RS232 or GPIB, 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.

M-8X0: Interface = RS232 or GPIB. RS232: Input and output handshake settings must be FALSE.

Mercury: Interface = DLL, DLL for Device = Mercury, DLL Interface = RS232 (even if using USB), Parameter = empty string, RS232 baud rate = same as set on controller hardware

If the chosen timeout value is greater than 300 ms, it will automatically be set to 300 ms for a fluid program operation.

In the upper window ("Send") the user can enter commands which will be transmitted to the chosen device one line at a time when the ENTER key is pressed.

All controller responses are displayed in the Receive response window, which can be cleared by pressing the Clear Receive Window button or F2.

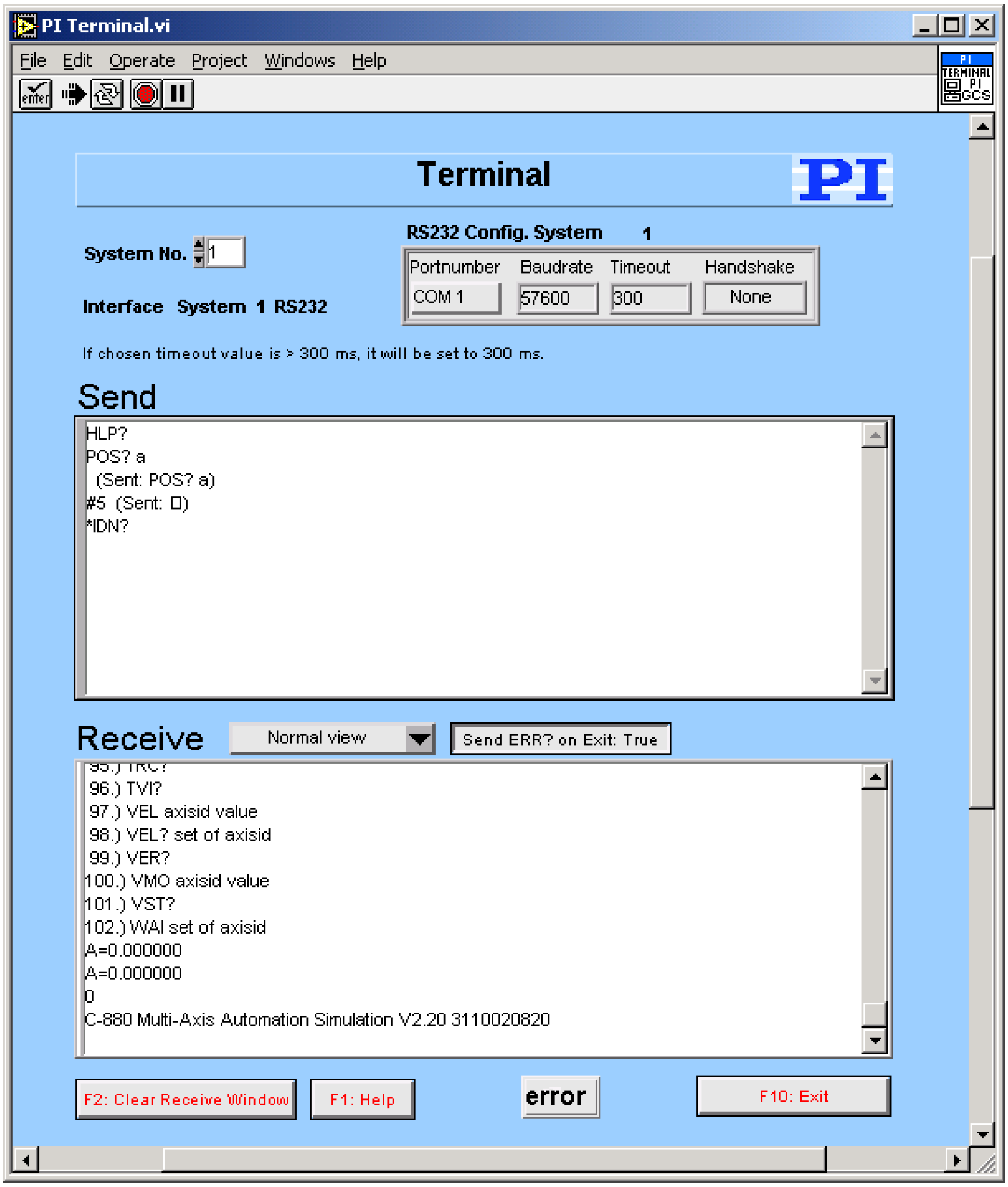
The view style of the Receive window can be changed to Show all characters or Hex View using the menu ring above the Receive window.

Exit or F10 will terminate the terminal application.

To send the last command again, just press the ENTER key again. The next line will then show the following entry: "(Send: *cmd*)" with *cmd* being the command from the line before, which was resent.

When the terminal application has just been started, pressing ENTER without entering a command will send "*IDN?" to the chosen system.

New commands can only be inserted into the last line of the Send window. The user can scroll through the history of the Send window using the scroll bar or the cursor up/down keys, but cannot change the history or resend commands by pressing ENTER unless in the last line. Pressing ENTER will always resend the last command, no matter where the cursor is positioned. Selecting text and using copy and paste (Ctrl+C, Ctrl+V) works for single lines, if only the contents of one single line (the command text) is selected and copied, not the full line (including the LineFeed) or multiple lines.



System No. 1

Interface System 1 RS232

RS232 Config. System 1

Portnumber	Baudrate	Timeout	Handshake
COM 1	57600	300	None

If chosen timeout value is > 300 ms, it will be set to 300 ms.

Send ERR? on Exit: True

```
HLP?
POS? a
 (Sent: POS? a)
#5  (Sent: □)
*IDN?
```

```
 95.) TRC?
 96.) TVI?
 97.) VEL axisid value
 98.) VEL? set of axisid
 99.) VER?
100.) VMO axisid value
101.) VST?
102.) WAI set of axisid
A=0.000000
A=0.000000
0
C-880 Multi-Axis Automation Simulation V2.20 3110020820
```

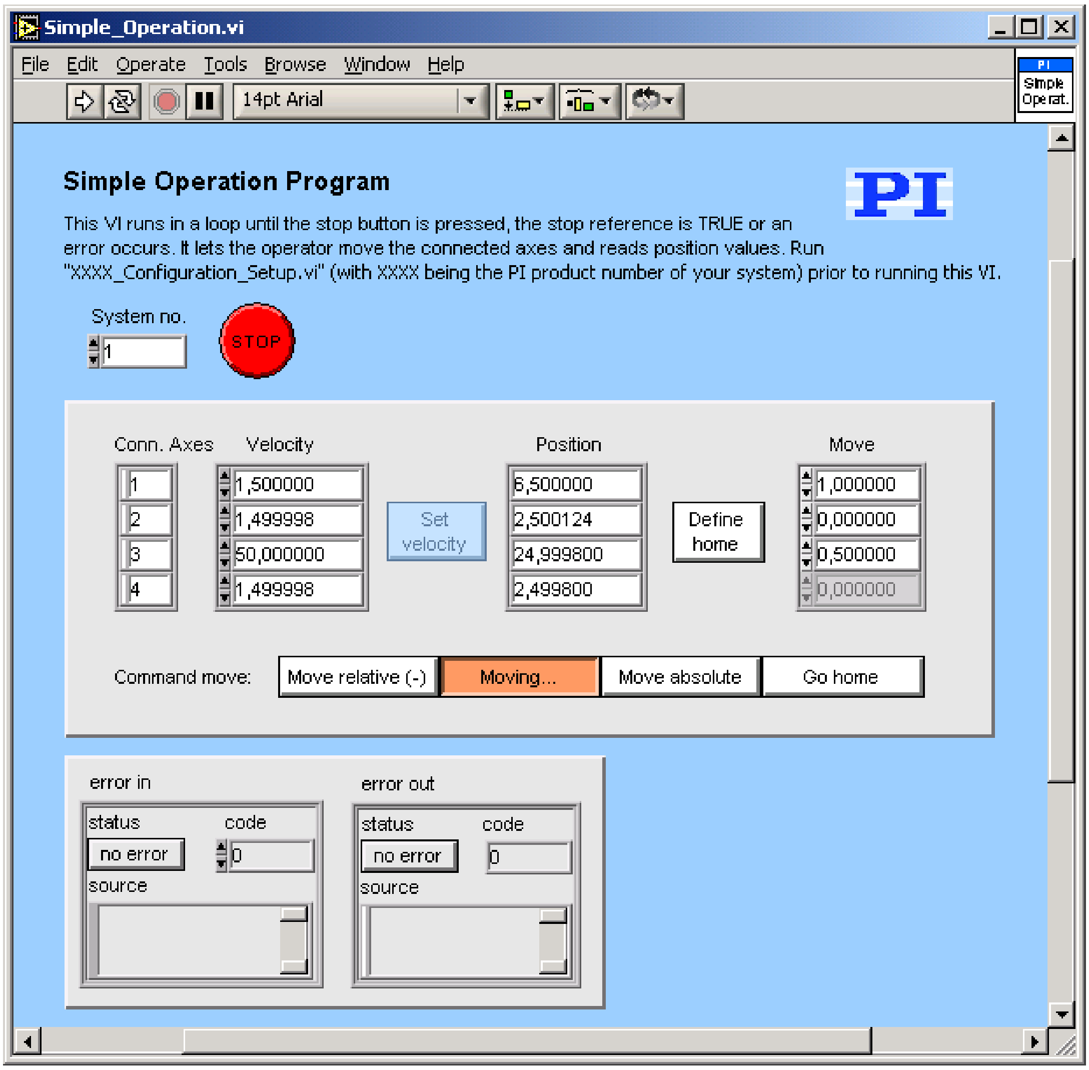
Many of PI's General Command Set compatible devices support single-byte commands. For example, the user can stop a fast scan of a C-880 or F-206 by sending an ASCII 24 (decimal). To enter this command into the Send window simply type a "#" followed by the decimal value of the byte to be sent, e.g. enter "#24" and presses ENTER to stop a fast scan. An entry "(Send: *)" will be added to the original command with * being the corresponding ASCII character of the single byte sent.

Pressing F1 or the Help button will pop up a help window. To return to the terminal application, press Esc. If Send ERR? on Exit? is TRUE, an “ERR?” query is sent to the device when Exit is pressed to prevent the controller from keeping an error condition produced during the use of the terminal application.

3.2. Simple_Operation.vi

This VI is runs in a loop until the stop button is pressed, the stop reference is TRUE or an error occures. It lets the operator set the velocity, move the connected axes to relative, absolute or to the home position, and reads the current position continuosly. The diagram shows how to combine the driver and support VIs for these tasks. 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.

Run “ XXXX_Configuration_Setup.vi” (with XXXX being the PI product number of your system) prior to running this VI.



Simple Operation Program

This VI runs in a loop until the stop button is pressed, the stop reference is TRUE or an error occurs. It lets the operator move the connected axes and reads position values. Run "XXXX_Configuration_Setup.vi" (with XXXX being the PI product number of your system) prior to running this VI.

System no. 1

Conn. Axes	Velocity	Position	Move
1	1,500000	6,500000	1,000000
2	1,499998	2,500124	0,000000
3	50,000000	24,999800	0,500000
4	1,499998	2,499800	0,000000

error in: status no error, code 0, source

error out: status no error, code 0, source

Valid for C-843, C-843.PM, C-865, Mercury

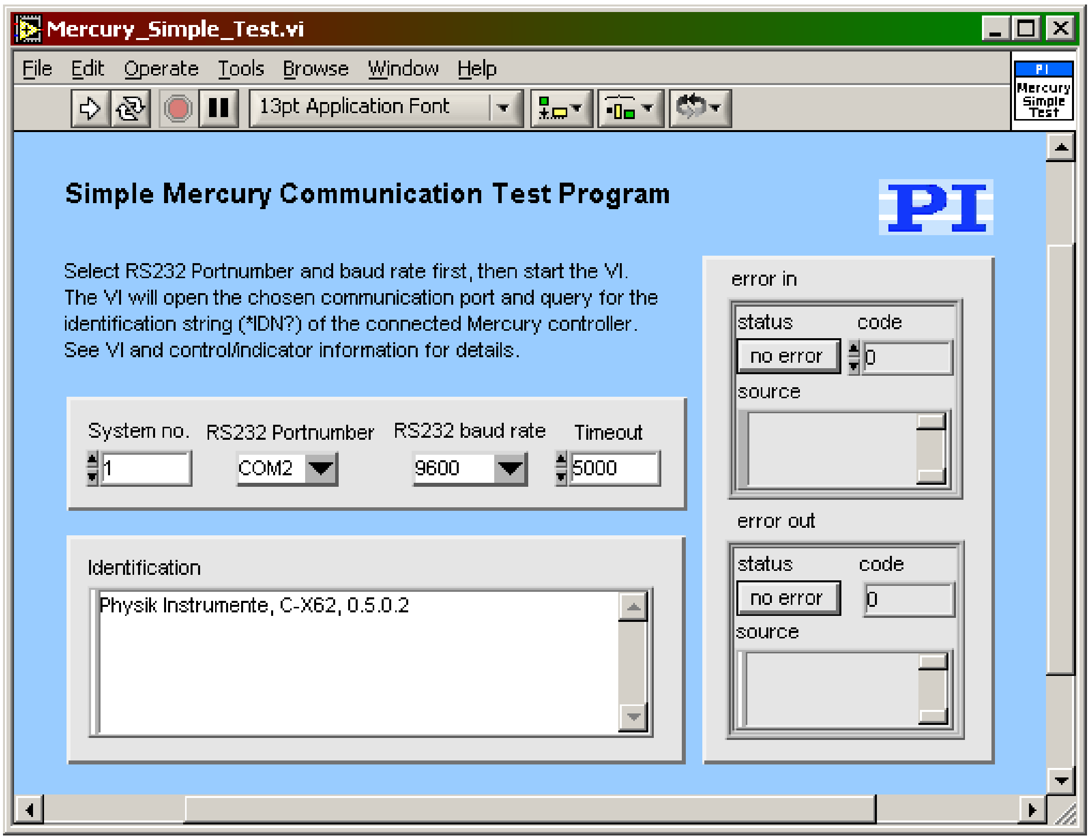
3.3. Mercury_Simple_Test.vi

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),
- RS232 port number,
- RS232 baud rate and
- Timeout value (in milliseconds).

Then start the VI. The VI will open the chosen communication port and query for the identification string of the connected Mercury controller. The diagram shows how to combine the driver and support VIs for these tasks.



Simple Mercury Communication Test Program

Select RS232 Portnumber and baud rate first, then start the VI.
The VI will open the chosen communication port and query for the identification string (*IDN?) of the connected Mercury controller.
See VI and control/indicator information for details.

System no.: 1
RS232 Portnumber: COM2
RS232 baud rate: 9600
Timeout: 5000

Identification: Physik Instrumente, C-X62, 0.5.0.2

error in
status: no error
code: 0
source:

error out
status: no error
code: 0
source:

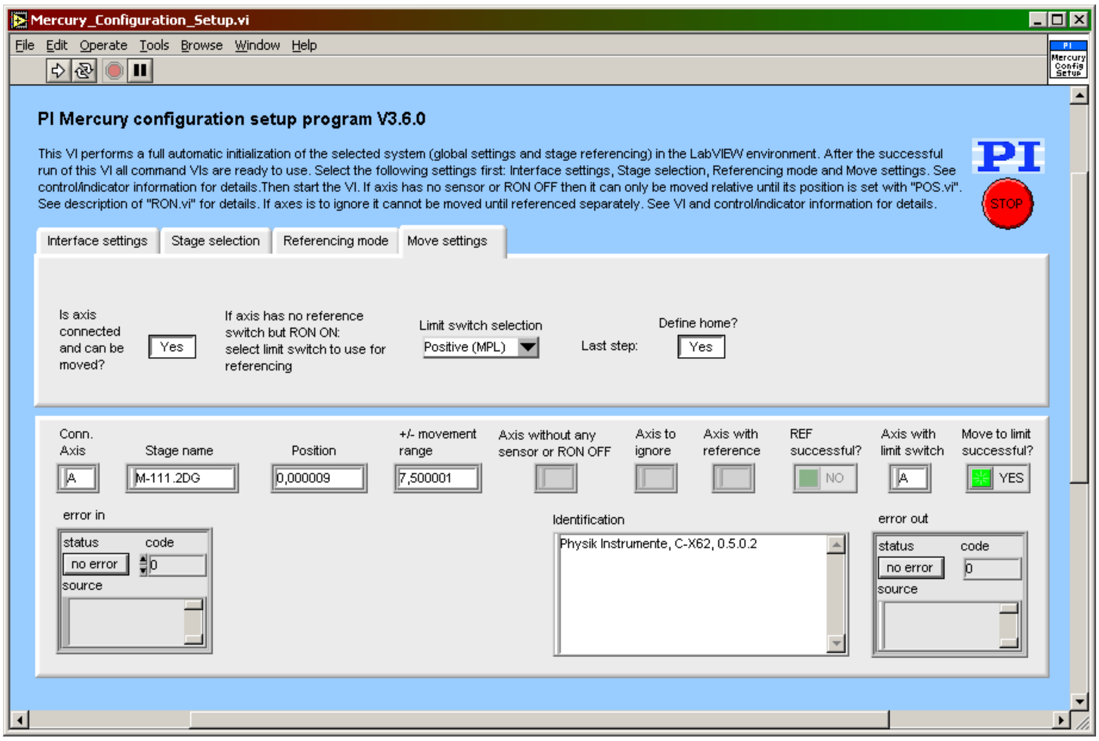
3.4. Mercury_Configuration_Setup.vi

This VI performs a fully automatic initialization of the selected system (global settings and stage referencing) in the LabVIEW environment. 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.

After the successful run of this VI, all command VIs are ready to use. Specify the correct parameters first:

- System No.: 1 in a one-system-only configuration.
- RS-232 settings: Choose the port number the Mercury is connected to and the correct baud rate.
- Timeout: choose an appropriate timeout value here (default is 5000 ms).
- Use dialog to define connected stage: if Yes, you can select the connected stage after starting the VI. If No, the dialog allowing stage selection will not appear and you must type the correct stage name into Stage name input.
- Reference mode (only if needed, see description of RON.vi for details and warnings)
- Whether or not axis is connected and can be moved (depends on mechanical setup).
- Which limit switch to use for referencing an axis that is to be referenced but has no reference switch.
- If all axes are to be moved to the middle position of their travel range.
- Whether the final position, half way between the limit switches (if any), should be set as home position.

Then start the VI.



PI Mercury configuration setup program V3.6.0

This VI performs a full automatic initialization of the selected system (global settings and stage referencing) in the LabVIEW environment. After the successful run 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. If axis has no sensor or RON OFF then it can only be moved relative until its position is set with "POS.vi". See description of "RON.vi" for details. If axes is to ignore it cannot be moved until referenced separately. See VI and control/indicator information for details.

Is axis connected and can be moved? Yes

If axis has no reference switch but RON ON: select limit switch to use for referencing

Limit switch selection: Positive (MPL)

Last step: Define home? Yes

Conn. Axis	Stage name	Position	+/- movement range	Axis without any sensor or RON OFF	Axis to ignore	Axis with reference	REF successful?	Axis with limit switch	Move to limit successful?
A	M-111.2DG	0,000009	7,500001				NO	A	YES

error in: status: no error; code: 0; source:

Identification: Physik Instrumente, C-X62, 0.5.0.2

error out: status: no error; code: 0; source:

“Mercury_Configuration_Setup.vi” performs the following initialization tasks:

1. Runs “PI Open Interface of one system.vi” to open a connection to the Mercury controller.
2. Runs “*IDN?.vi” to query for the controller identification string.
3. Defines the selected system to be "Mercury".
4. Runs “Define connected axes.vi” with Read from controller = FALSE and Connected axes = 1.
5. Runs “CST.vi” to determine which stage is connected to the axis of the Mercury controller, according to your selection.
6. Runs “Define connected axes.vi” with Read from controller = TRUE. If stage name “NOSTAGE” was selected, axis will not appear as connected axis.
7. Runs “CST?.vi” to query for the connected stage (a NOSTAGE axis will not appear as a connected stage).
8. Runs “INI.vi”
9. Sets the reference mode for the axis, if specified (Reference mode tab).
10. Reads whether axis has reference mode OFF.
11. If not, determines if axis has a reference switch (REF?).
12. If yes, moves axis to the reference position (REF).
13. If axis has no reference switch but has reference mode ON, reads if axis has a limit switch.
14. If yes, moves axis to the positive or negative limit position (MPL/MNL) as specified.
15. Reads the position range (TMN?, TMX?).
16. If Move to middle? is TRUE, moves axis if it has been referenced or moved to a limit switch to the middle position of its range (MOV).
17. Waits for motion to stop
18. Defines home (if so specified).
19. Runs “POS?.vi” to query for the position.
20. Runs “ERR?.vi” to query the controller for its error status.
21. Runs “GCSTranslateError.vi” to append the error message which corresponds with a GCS error number returned by “ERR?.vi” to Source from Error out.

If Use dialog to define connected stage is TRUE, the VI reads the available stages from PIStages.dat (by querying with “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.** See chapter 1.1 for information about methods for proper registration of PIStages.dat. In Microsoft Explorer, right-click the file PIStages.dat and select "Properties". Make sure that the “read-only“ attribute is not checked.

If axis has referencing mode OFF, referencing was not successful, or axis was designated not to be moved, it will report NaN as position value. An Axis without any sensor or with RON OFF can only be moved relative until its position is set with "POS.vi". See description of "RON.vi" for details. If Axis was designated not to be moved during runtime of Mercury_Configuration_Setup it cannot be moved until referenced separately.

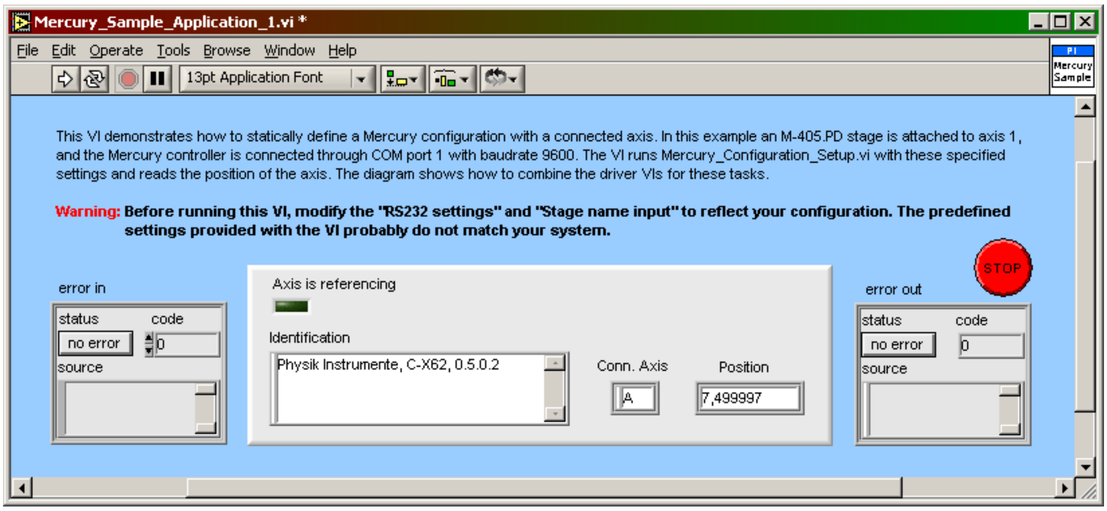
Use this VI as the initialization VI for the Mercury controller in your application.

As the initialization is a complex procedure which uses a large number of sub-VIs, Mercury_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. Advanced users who have a need to do so, should please contact their local sales representative to receive the password.

3.5. Mercury_Sample_Application_1.vi

This VI demonstrates how to statically define a Mercury configuration with a connected axis. In this example an M-405.PD stage is attached to axis 1, and the Mercury controller is connected through COM port 1 with baudrate 9600. The VI runs Mercury_Configuration_Setup.vi with these specified settings and reads the position of all axes. The diagram shows how to combine the driver VIs for these tasks.

Warning: Before running this VI, modify RS232 settings and Stage name input to reflect your configuration. The predefined settings provided with the VI probably do not match your system.



Mercury_Sample_Application_1.vi *

This VI demonstrates how to statically define a Mercury configuration with a connected axis. In this example an M-405.PD stage is attached to axis 1, and the Mercury controller is connected through COM port 1 with baudrate 9600. The VI runs Mercury_Configuration_Setup.vi with these specified settings and reads the position of the axis. The diagram shows how to combine the driver VIs for these tasks.

Warning: Before running this VI, modify the "RS232 settings" and "Stage name input" to reflect your configuration. The predefined settings provided with the VI probably do not match your system.

error in: status: no error; code: 0; source:

Axis is referencing

Identification: Physik Instrumente, C-X62, 0.5.0.2

Conn. Axis: A

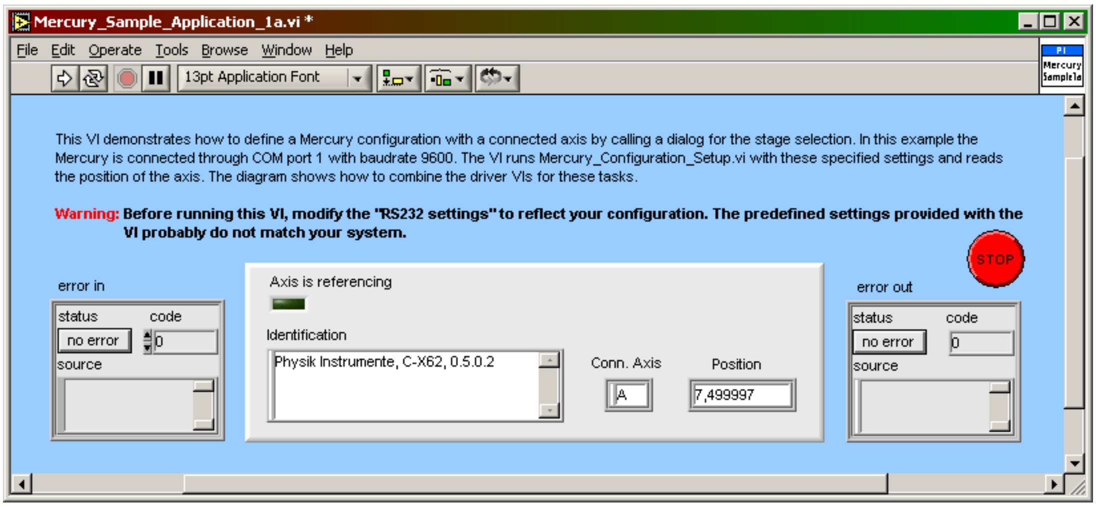
Position: 7,499997

error out: status: no error; code: 0; source:

3.6. Mercury_Sample_Application_1a.vi

This VI demonstrates how to define a Mercury configuration with a connected axis by calling a dialog for the stage selection. In this example the Mercury controller is connected through COM port 1 with baudrate 9600. The VI runs Mercury_Configuration_Setup.vi with these specified settings and reads the position of all axes. The diagram shows how to combine the driver VIs for these tasks.

Warning: Before running this VI, modify RS232 settings to reflect your configuration. The predefined settings provided with the VI probably do not match your system.


Mercury_Sample_Application_1a.vi *
File Edit Operate Tools Browse Window Help
13pt Application Font
PI Mercury Sample1a
STOP

This VI demonstrates how to define a Mercury configuration with a connected axis by calling a dialog for the stage selection. In this example the Mercury is connected through COM port 1 with baudrate 9600. The VI runs Mercury_Configuration_Setup.vi with these specified settings and reads the position of the axis. The diagram shows how to combine the driver VIs for these tasks.

Warning: **Before running this VI, modify the "RS232 settings" to reflect your configuration. The predefined settings provided with the VI probably do not match your system.**

error in

status: no error

code: 0

source:

Axis is referencing

Identification: Physik Instrumente, C-X62, 0.5.0.2

Conn. Axis: A

Position: 7,499997

error out

status: no error

code: 0

source:

4. PI Systems Currently Supported by This Driver Set

Product	works with LabVIEW driver version	if product firmware version is equal to or newer than
C-843	2.01 – 2.02 2.05 – 2.06 3.1.2., 3.1.2a 3.4.3	MC-DLL 1.0.2.2 MC-DLL 1.0.2.3 MC-DLL 1.0.2.3 MC-DLL 1.0.2.8
C-843.PM	3.1.0 3.4.3a	MC-DLL 1.0.2.5 MC-DLL 1.0.2.5
C-848	3.0.2	1.0
C-865	3.3.0	MC_C865.dll 1.0
C-880	1.1 1.2 2.04 2.05 – 2.06 3.2.0	2.00 2.10 2.20 2.21 2.40
C-880K005	2.06	1.0
C-880K006	2.06	1.0
C-880K007	2.06	1.0
E-516	1.0 – 2.02 2.05 – 2.06 3.4.2	DSP V3.01, MCU V5 DSP V3.11, MCU V5 DSP V3.30, MCU V5
E-710 3- & 4-channel versions	3.4.0 3.4.4	5.027 5.0.33, 6.0.33
E-710 6-channel	3.4.4	2.13
E-761	3.5.0	1.0.0
E-816	2.01 – 2.06	2.02
F-206	1.1 – 2.06	Fhx0035
M-840	2.03 – 2.06 2.2.0 3.0.1 3.1.1	Hex0037 Hex0045 Hex0050 Hex0051
M-850	2.03 – 2.06 3.0.1 3.1.1	Hex0040 Hex0050 Hex0051
Mercury	3.6.0	1.0.6 PI_MERCURY_GCS_DLL.dll V 1.0.0.17

5. Appendix A

Error codes are not unambiguous, but can result from a PI error message or LabVIEW internal error code. In addition to the list below see National Instruments error codes.

100	PI LabVIEW driver reports error. See source control for details.
0	No error
1	Parameter syntax error
2	Unknown command
3	Command length out of limits or command buffer overrun
4	Error while scanning
5	Unallowable move attempted on unreferenced axis, or move attempted with servo off
6	Parameter for SGA not valid
7	Position out of limits
8	Velocity out of limits
9	Attempt to set pivot point while U,V and W not all 0
10	Controller was stopped by command
11	Parameter for SST or for one of the embedded scan algorithms out of range
12	Invalid axis combination for fast scan
13	Parameter for NAV out of range
14	Invalid analog channel
15	Invalid axis identifier
16	Unknown stage name
17	Parameter out of range
18	Invalid macro name
19	Error while recording macro
20	Macro not found
21	Axis has no brake
22	Axis identifier specified more than once
23	Illegal axis
24	Incorrect number of parameters
25	Invalid floating point number
26	Parameter missing
27	Soft limit out of range
28	No manual pad found

29	No more step-response values
30	No step-response values recorded
31	Axis has no reference sensor
32	Axis has no limit switch
33	No relay card installed
34	Command not allowed for selected stage(s)
35	No digital input installed
36	No digital output configured
37	No more MCM responses
38	No MCM values recorded
39	Controller number invalid
40	No joystick configured
41	Invalid axis for electronic gearing, axis can not be slave
42	Position of slave axis is out of range
43	Slave axis cannot be commanded directly when electronic gearing is enabled
44	Calibration of joystick failed
45	Referencing failed
46	OPM (Optical Power Meter) missing
47	OPM (Optical Power Meter) not initialized or cannot be initialized
48	OPM (Optical Power Meter) Communication Error
49	Move to limit switch failed
50	Attempt to reference axis with referencing disabled
51	Selected axis is controlled by joystick
52	Controller detected communication error
53	MOV! motion still in progress
54	Unknown parameter
55	No commands were recorded with REP
56	WPA and SEP password invalid
57	Data Record Table does not exist
58	Source does not exist; number too low or too high
59	Source Record Table number too low or too high
60	Protected Param: current Command Level (CCL) too low
61	Befehlsausführung nicht möglich, so lange Autozero läuft
62	Autozero allowed only on linear axes
100	PI LabVIEW driver reports error. See source control for details.
200	No stage connected to axis
201	File with axis parameters not found
202	Invalid axis parameter file

203	Backup file with axis parameters not found
204	PI internal error code 204
205	SMO with servo on
206	uudecode: incomplete header
207	uudecode: nothing to decode
208	uudecode: illegal UUE format
209	CRC32 error
210	Illegal file name (must be 8-0 format)
211	File not found on controller
212	Error writing file on controller
213	VEL command not allowed in DTR Command Mode
214	Position calculations failed
301	Send buffer overflow
302	Voltage out of limits
303	Attempt to set voltage when servo on
304	Received command is too long
305	Error while reading/writing EEPROM
306	Error on I2C bus
307	Timeout while receiving command
308	A lengthy operation has not finished in the expected time
309	Insufficient space to store macro
310	Configuration data has old version number
311	Invalid configuration data
333	Internal hardware error
555	BasMac: unknown controller error
601	not enough memory
602	hardware voltage error
603	hardware temperature out of range
1000	Too many nested macros
1001	Macro already defined
1002	Macro recording not activated
1003	Invalid parameter for MAC
1004	PI internal error code 1004
2000	Controller already has a serial number
4000	Sector erase failed
4001	Flash program failed
4002	Flash read failed
4003	HW match code missing/invalid

4004	FW match code missing/invalid
4005	HW version missing/invalid
4006	FW version missing/invalid
0	No error occurred during function call
-1	Error during com operation (could not be specified)
-2	Error while sending data
-3	Error while receiving data
-4	Not connected (no port with given ID open)
-5	Buffer overflow
-6	Error while opening port
-7	Timeout error
-8	There are more lines waiting in buffer
-9	There is no interface or DLL handle with the given ID
-10	Event/message for notification could not be opened
-11	Function not supported by this interface type
-12	Error while sending "echoed" data
-13	IEEE488: System error
-14	IEEE488: Function requires GPIB board to be CIC
-15	IEEE488: Write function detected no listeners
-16	IEEE488: Interface board not addressed correctly
-17	IEEE488: Invalid argument to function call
-18	IEEE488: Function requires GPIB board to be SAC
-19	IEEE488: I/O operation aborted
-20	IEEE488: Interface board not found
-21	IEEE488: Error performing DMA
-22	IEEE488: I/O operation started before previous operation completed
-23	IEEE488: No capability for intended operation
-24	IEEE488: File system operation error
-25	IEEE488: Command error during device call
-26	IEEE488: Serial poll-status byte lost
-27	IEEE488: SRQ remains asserted
-28	IEEE488: Return buffer full
-29	IEEE488: Address or board locked
-30	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	RS-232: Error configuring the COM port
-32	Error dealing with internal system resources (events, threads, ...)
-33	A DLL or one of the required functions could not be loaded

-34	FTDIUSB: invalid handle
-35	FTDIUSB: device not found
-36	FTDIUSB: device not opened
-37	FTDIUSB: IO error
-38	FTDIUSB: insufficient resources
-39	FTDIUSB: invalid parameter
-40	FTDIUSB: invalid baud rate
-41	FTDIUSB: device not opened for erase
-42	FTDIUSB: device not opened for write
-43	FTDIUSB: failed to write device
-44	FTDIUSB: EEPROM read failed
-45	FTDIUSB: EEPROM write failed
-46	FTDIUSB: EEPROM erase failed
-47	FTDIUSB: EEPROM not present
-48	FTDIUSB: EEPROM not programmed
-49	FTDIUSB: invalid arguments
-50	FTDIUSB: not supported
-51	FTDIUSB: other error
-1001	Unknown axis identifier
-1002	Number for NAV out of range--must be in [1,10000]
-1003	Invalid value for SGA--must be one of 1, 10, 100, 1000
-1004	Controller sent unexpected response
-1005	No manual control pad installed, calls to SMA and related commands are not allowed
-1006	Invalid number for manual control pad knob
-1007	Axis not currently controlled by a manual control pad
-1008	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
-1009	Internal error--could not start thread
-1010	Controller is (already) in macro mode--command not valid in macro mode
-1011	Controller not in macro mode--command not valid unless macro mode active
-1012	Could not open file to write or read macro
-1013	No macro with given name on controller, or macro is empty
-1014	Internal error in macro editor
-1015	One or more arguments given to function is invalid (empty string, index out of range, ...)
-1016	Axis identifier is already in use by a connected stage
-1017	Invalid axis identifier
-1018	Could not access array data in COM server

-1019	Range of array does not fit the number of parameters
-1020	Invalid parameter ID given to SPA or SPA?
-1021	Number for AVG out of range--must be >0
-1022	Incorrect number of samples given to WAV
-1023	Generation of wave failed
-1024	Motion error while axis in motion
-1025	Controller is (already) running a macro
-1026	Configuration of PZT stage or amplifier failed
-1027	Current settings are not valid for desired configuration
-1028	Unknown channel identifier
-1029	Error while reading/writing wave generator parameter file
-1030	Could not find description of wave form. Maybe WG.INI is missing?
-1031	The WGWaveEditor DLL function was not found at startup
-1032	The user cancelled a dialog
-1033	Error from C-844 Controller
-1034	DLL necessary to call function not loaded, or function not found in DLL
-1035	The open parameter file is protected and cannot be edited
-1036	There is no parameter file open
-1037	Selected stage does not exist
-1038	There is already a parameter file open. Close it before opening a new file
-1039	Could not open parameter file
-1040	The version of the connected controller is invalid
-1041	Parameter could not be set with SPA--parameter not defined for this controller!
-1042	The maximum number of wave definitions has been exceeded
-1043	The maximum number of wave generators has been exceeded
-1044	No wave defined for specified axis
-1045	Wave output to axis already stopped/started
-1046	Not all axes could be referenced
-1047	Could not find parameter set required by frequency relation
-1048	Command ID given to SPP or SPP? is not valid
-1049	A stage name given to CST is not unique
-1050	A uuencoded file transfered did not start with "begin" followed by the proper filename
-1051	Could not create/read file on host PC
-1052	Checksum error when transfering a file to/from the controller
-1053	The PiStages.dat database could not be found. This file is required to connect a stage with the CST command
-1054	No wave being output to specified axis
-1055	Invalid password

-1056	Error during communication with OPM (Optical Power Meter), maybe no OPM connected
-1057	WaveEditor: Error during wave creation, incorrect number of parameters
-1058	WaveEditor: Frequency out of range
-1059	WaveEditor: Error during wave creation, incorrect index for integer parameter
-1060	WaveEditor: Error during wave creation, incorrect index for floating point parameter
-1061	WaveEditor: Error during wave creation, could not calculate value
-1062	WaveEditor: Graph display component not installed
-1063	User Profile Mode: Command is not allowed, check for required preparatory commands
-1064	User Profile Mode: First target position in User Profile is too far from current position
-1065	Controller is (already) in User Profile Mode
-1066	User Profile Mode: Block or Data Set index out of allowed range
-1067	ProfileGenerator: No profile has been created yet
-1068	ProfileGenerator: Generated profile exceeds limits of one or both axes
-1069	ProfileGenerator: Unknown parameter ID in Set/Get Parameter command
-1070	ProfileGenerator: Parameter out of allowed range
-1071	User Profile Mode: Out of memory
-1072	User Profile Mode: Cluster is not assigned to this axis

6. Index

#

#24.vi 36
#5.vi 36
#5_old.vi 44
#7.vi 36

*

*IDN?.vi 24

A

Analyse input string for terminal.vi 53
Assign booleans from string to axes.vi 53
Assign NaN for chosen axes.vi 53
Assign SPA values from string to axes.vi 53
Assign SPA_Hex values from string to axes.vi 54
Assign values from string to axes.vi 54

B

BRA.vi 36
BRA?.vi 36
Build command substring.vi 54, 55
Build DIO? query command substring.vi 54
Build query command substring.vi 55
Build SPA command substring.vi 55
Build SPA query command substring.v 55
Build SPA_Hex command substring.vi 56
Build SPA_Hex query command substring.v 56
Build SPA_Hex_String command substring.vi 56
Build SPA_String command substring.vi 56
Build stringplusnum substring.vi 56

C

C-843 First Steps 7
C-843.PM First Steps 8
C-848 First Steps 9
C-865 First Steps 10
C-880 First Steps 11
C-880K005 First Steps 12
Close connection if open.vi 20
CLR.vi 37
Commanded axes connected?.vi 57
Commanded stage name available?.vi 57
Controller names.ctl 24
Convert num array to string.vi 57
Count occurrences in string.vi 57
CST.vi 37
CST?.vi 37

D

Define axes to command from boolean array.vi 57
Define connected axes.vi 24
Define connected stages with dialog.vi 58
Define connected systems.vi 25
Define macro contents.vi 50
DEL.vi 38
DFF.vi 38
DFF?.vi 38
DFH.vi 45
DFH?.vi 45
DIO.vi 39
DIO?.vi 39

E

E-516 First Steps 13
E-710 First Steps 14
E-761 First Steps 15
E-816 First Steps 16
ERR?.vi 25

F

F-206 First Steps 17

G

GCSTranslateError.vi 58
GCSTranslator DLL Functions.vi 20
General wait for movement to stop.vi 58
Get arrays without blanks.vi 59
Get lines and values from string.vi 59
Get lines from string.vi 59
Get string array size without blanks.vi 59
Global1.vi 20
Global2.vi 26
GOH.vi 46

H

HLP?.vi 26
HLT.vi 26
How often does string contain regular expression.vi 60

I

INI.vi 40

L

LIM?.vi 46
Longlasting one-axis command.vi 60

M

M-840 / M-850 First Steps 18
MAC BEG.vi 50
MAC DEL.vi 50
MAC END.vi 50
MAC NSTART.vi 51
MAC START.vi 51
MAC?.vi 51
Mercury First Steps 19
Mercury_Configuration_Setup.vi 69
Mercury_Sample_Application_1.vi 72
Mercury_Sample_Application_1a.vi 73
Mercury_Simple_Test.vi 68
MNL.vi 46
MOV and TAV?.vi 52
MOV.vi 26
MOV?.vi 27
MPL.vi 46
MVR.vi 27

O

ONT?.vi 27

P

PI Ask for Communication Parameters.vi 20
PI Open Interface of one system.vi 22
PI Open Interface.vi 21
PI Receive String.vi 23
PI ReceiveNCharacters RS232.vi 23
PI ReceiveString GPIB.vi 23
PI Send String.vi 23
PI Terminal.vi 64
POS.vi 40
POS?.vi 28

R

REF.vi 47
REF?.vi 47
Return single characters from string.vi 60
RON.vi 47
RON?.vi 48
Round with options.vi 60

S

SAI.vi 41
SAI?.vi 29
Select axis.vi 60
Select values for chosen axes.vi 61
Select with boolean array input.vi 61
Selection to string array.vi 61
Set RON and return RON status.vi 61
SetQMC_SetQMCA.vi 24
Simple_Operation.vi 67
SPA.vi 29
SPA?.vi 31
SRG?.vi 41
STA?.vi 41
STP.vi 32
Substract axes array subset from axes array.vi 62
SVO.vi 33
SVO?.vi 33

T

TAC?.vi 52
TAV?.vi 52
TIO?.vi 43
TMN?.vi 48
TMX?.vi 49
TVI?.vi 43

U

Unbundle/bundle interface clusters for PI Terminal.vi 62

V

VEL.vi 34
VEL?.vi 34
VER?.vi 35
VST?.vi 43

W

Wait for answer 0 or 1 without polling.vi 62
Wait for answer of longlasting command.vi 62
Wait for axes to stop.vi 63
Wait for hexapod system axes to stop.vi 44

End of document