Digilock 110 RCI

Remote Control Interface

Manual

Manual: M-038 Version 01

TOPTICA Photonics AG
Lochhamer Schlag 19
D-82166 Graefelfing/Munich
Tel.: +49 89 85837-0
Fax: +49 89 85837-200

email: info@toptica.com
http://www.toptica.com

(June 2009 Subject to changes without notice)

Contents

1 DigiLock 110 Control Architecture

The DigiLock 110 Feedback Controlyzer is a very fast and flexible controller solution based on FPGA technology. The combination of a fast FPGA with a dedicated PC software package allows to implement the advanced locking capabilities.

An overview of the DigiLock 110 control architecture is shown in Figure 1. It consists of two levels: The DigiLock 110 software initially starts the DigiLock Module Server (DMS). The DMS automatically detects multiple DigiLock 110 modules connected to the computer via USB. It can start and manage one instance of the DigiLock User Interface (DUI) for each of the detected modules. The user can thus control the DigiLocks via independent DUIs.

NOTE !	The Remote Control Interface (RCI) and the DigiLock Module Server (DMS) are only implemented in software versions 1.5.4.70 and higher. If you are still running an earlier version please contact TOPTICA Photonics AG for a software update.

The DigiLock 110 is always controlled via the dedicated software running on a Windows PC (for a detailed software description please see the DigiLock 110 Manual). However, to allow for the implementation of application specific features like scripting and the integration into the general software environment of a larger experiment, the DigiLock Module Server (DMS) and each instance of the DUI has a build-in Remote Control Interface (RCI). Nearly all controls of the DMS and DUI can be accessed remotely by sending corresponding commands via a TCP/IP connection.

NOTE !	The DigiLock 110 RCI is a supplementary free add-on developed by TOPTICA Photonics AG on customers requests. However, due to the complexity of individual hard- and software configurations there is no support provided for the DigiLock 110 RCI.

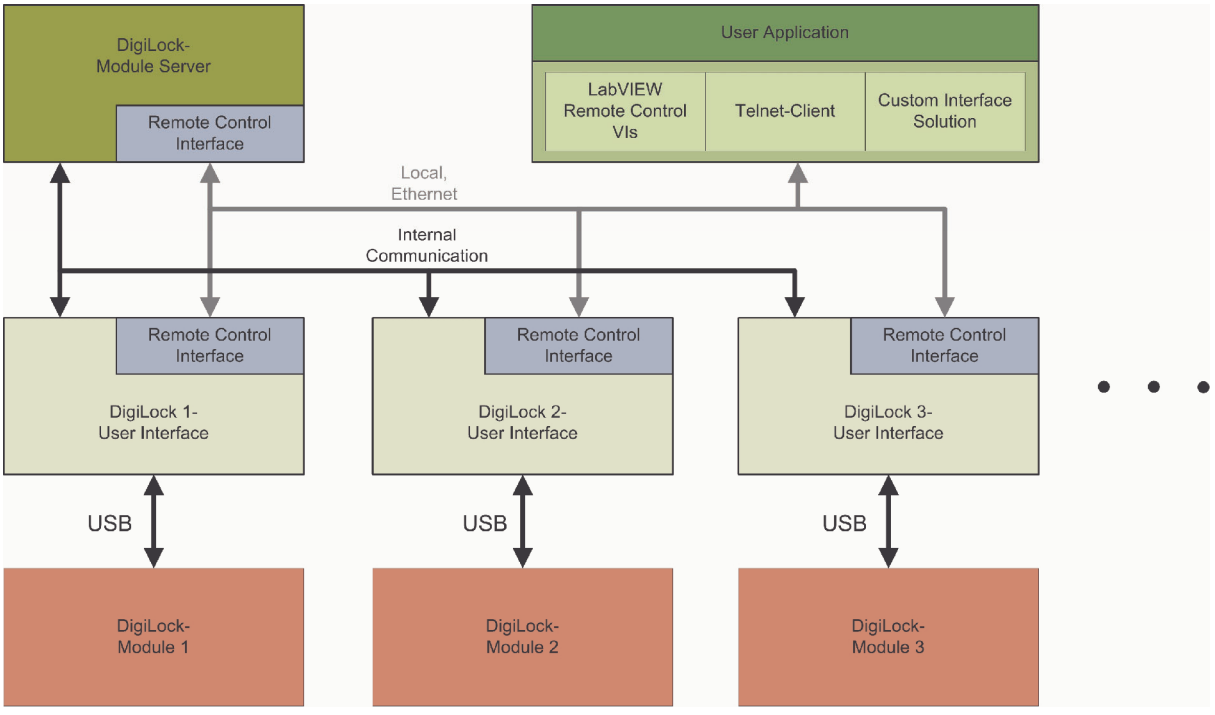

Figure 1 Overview of the DigiLock 110 Control Architecture

2 DigiLock Module Server (DMS)

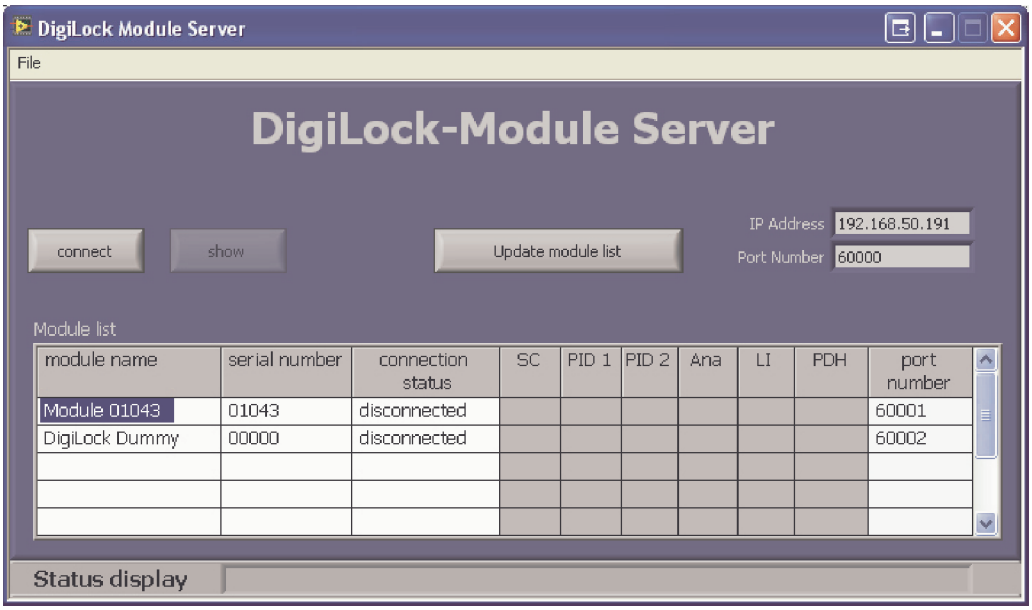

Figure 2 DigiLock Module Server Front End

Figure 2 shows the front end of the DigiLock Module Server. It automatically detects and lists all DigiLock 110 modules that are connected to the computer when the software starts up[1]. The list is updated by pressing the **Update module list** button. A specific module can be selected by clicking anywhere on the corresponding row in the list. The selected module is indicated by the highlighted module name. The modules are identified by their serial number and can be given a name by double-clicking the module name column for editing.

To connect to the selected DigiLock 110 press the **Connect** button. This will open an instance of the DigiLock User Interface (DUI). Once connected, the button changes to **Disconnect**. Pressing the button again will close the DUI and disconnect the DigiLock 110.

By clicking the **Show/Hide module** button, the DigiLock Interface can be (un)hidden from the user. Hiding the DUI is useful to clean up the desktop or to avoid excessive traffic on the USB system. When a DUI is hidden, the sampling is disabled by default. This setting can be changed in the **Settings | General** tab of the DUI.

The IP Address and Port Number fields display the corresponding values for the RCI of the DMS which are defined in the configuration file. This configuration file (DigiLock-ModuleServer-Profile.pro) is found in the profile directory (see Paragraph 4.1). The Port Numbers of the RCIs of the DUIs are listed in the last column "port number". For details on the remote control see Paragraph 3.

1. In addition to the physically present DigiLocks it also lists a dummy to allow to start a DUI with no DigiLock 110 present.

3 Remote Control Interface (RCI)

An overview of the DigiLock 110 control architecture is shown in Figure 1: The DigiLock 110 software initially starts the DigiLock Module Server (DMS). The DMS automatically detects multiple DigiLock 110 modules connected to the computer via USB. It can start and manage one instance of the DigiLock User Interface (DUI) for each of the detected modules. The user can thus control the DigiLocks via independent DUIs. For remote control the DigiLock Module Server (DMS) has a built-in remote control interface (RCI) which can be connected to the TCP/IP port 60000[2], by default. The remote commands of the DMS allow to identify the DigiLock 110 modules and start the corresponding DigiLock User Interface (DUI). Each instance of the DUI provides its own remote control interface (RCI) running on a corresponding subsequent port, by default 6000x, where x is the number of the DigiLock 110. Nearly all elements on the control panel of the DUI can be accessed remotely by sending commands from an external program. Sending the corresponding command has the same effect as operating the DUI directly (e.g. with the mouse). The command that corresponds to a given control on the DUI can be determined via *description and tip* in the context menu.

Each remote control interface (RCI) can be used with any client via TCP (Transmission Control Protocol). The functionality of the RCI can easily be tested with a telnet client like, e.g., the HyperTerminal[3] (see Paragraph 3.1). Alternatively, there are a number of free Terminal programs that provide additional functionality. For programming with LabVIEW® [4] a set of VIs is provided to access the remote commands (see Paragraph 3.3).

3.1 Telnet Client Connection

To get used to the remote control we suggest to start the DigiLock Module Server (DMS) and the DigiLock User Interface (DUI) and open connections via a telnet client, like e.g. the Hyper Terminal.

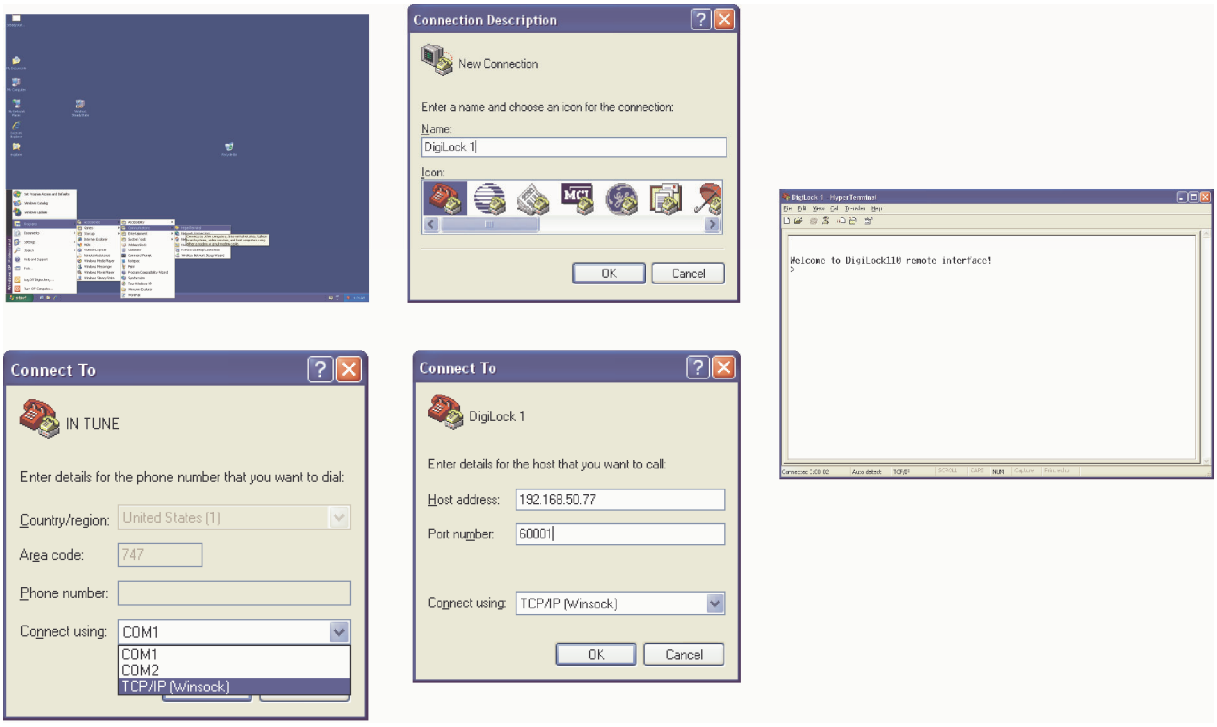

Figure 3 Setting-Up a Telnet Connection with the Hyper Terminal

2. The port number can be defined in the configuration file "DigiLock-ModuleServer-Profile.pro" of the DMS (see Paragraph 4.1).
3. HyperTerminal is normally installed together with the Windows® Operating System
4. Graphical programming environment by National Instruments

The Hyper Terminal is part of the Windows operating system. A TCP/IP telnet connection can be established in the following steps:

1. Start Hyper Terminal: Start | Programs | Accessories | Communications | HyperTerminal.

2. Assign a name to the connection.

3. Select Connect using: TCP/IP (Winsock).

4. Enter the Host Address:
   If the DigiLock 110 software and HyperTerminal run on the same computer, enter localhost; otherwise enter the IP address of the computer that is connected to the DigiLock 110 with at least the DigiLock Module Server (DMS) started[5].

5. Enter Port Number:
   60000 for the RCI of the DMS,
   6000x ($x \geq 1$) for the RCI of the DigiLock User Interfaces[6]

When a connection is established, the DMS or DUI will answer with a welcome message. Afterwards a command prompt appears (> <space>) and the connection is ready to be used. All commands are followed by a <CR><LF> sequence (ASCII 13 and 10) to finish the command entry.

NOTE !	A good starting point is "commandlist?" which will return a list of all possible commands for the RCI. To query for a value add a questionmark to the command. To find the range of possible values add ".range?". The help to a command is accessible with ".help?"

5. The IP address can, e.g., be determined by resolving the computer name using ping or on the computer where the DigiLock 110 software runs using ipconfig in the command line interface.
6. These are the default port numbers which can be changed in the configuration file (DigiLock-ModuleServer-Profile.pro, see Paragraph 4.1).

3.2 DigiLock-Application Programming Interface (API) Description

In the following the DigiLock-Application Programming Interface (API) is described. To accelerate software development in LabVIEW® a complete set of VIs provides easy access to the DigiLock 110 functions. A full list of the available commands can be found in Paragraph 4.2.

3.2.1 Command Syntax

The DMS and the DUI each have their own command set. The command to remotely operate a given control can be found in *Description and Tip* of the context menu accessible via a right-click on the control in the corresponding program window.

NOTE !	The Remote Control Interfaces (RCIs) acts in the same way as a direct user manipulation of the controls on the DMS and DUI, e.g. by a mouse action. Note that if you want to work with the graphs in the lower display area you have to activate the appropriate display. The display can be chosen by a software command. All other functions can be remote controlled without selecting the corresponding tab.

3.2.2 Command Types

There are five different types of commands as illustrated by the examples below (with the corresponding reply of the RCI where applicable). For explanation the lines are commented after the two backslashes. After sending the command and display of the possible reply the terminal will return to a command prompt for further input.

- **Query commands**
  Query commands are used to retrieve the current value of a control are given by the command string followed directly by a questionmark:
  > pid2: proportional // query command followed by <CR><LF>
  pid2: proportional=10000 // answer
  > // command prompt

- **Set commands**
  Set commands are used to set the value of a control are given by the command string followed by an equality sign and the value of the corresponding data type (see below):
  > pid2: proportional=100 // <command name>=<value> followed by <CR><LF>

NOTE !	Not all control values can be set, some are read-only. Please see the remote command list in Paragraph 4.2 for details.

- **Query control range**
  The control range query gets the allowed parameter range which also indicates the data type. It is formed by the command followed by .range?
  > scan module: frequency.range? // query command followed by <CR><LF>
  scan module: frequency.range=0.1 ... 10000// answer with the allowed range

- **Query control help**
  The control help query gets the help description. It is formed by the command followed by .help?
  > pid2:integral.help? // query command followed by <CR><LF>
  pid2:integral.help=integral gain of PID 2

- **Special query commands**
  are mostly related to the interfaces and the communication:
  > messages waiting? // query command followed by <CR><LF>
  messages waiting=0 // answer

3.2.3 Data Types

There are three different data types for controls: Booleans, Numeric Values and Enumerations. The range of values for the data types can be found by the range command, i.e. <command name>.range?.

Examples:

- Boolean: <pid1:enable=true> // {true, false}
- Numeric: <pid1:gain=25> // separator according to setting in the operating system
- Enumeration: <pid1:input=main in>

3.2.4 Limitations

The concept of remotely controlling the user interface allows to access nearly all functionality that is provided by the dedicated DUI. However, the reaction time is limited by the performance of the DUI and can be affected by the general performance of the control computer as well as the network connection. Currently only a few functions can not be controlled remotely, e.g. setting the module name in the DMS or detailed handling of profiles in the DUI.

3.2.5 Error Handling

To ensure a error free operation the remote software provides feedback if an error occurs. There are several kinds of error messages starting with *%% Error*:<space> .

Error	**Description**
bad command-<expression entered>	command does not exist
bad parameter-<command>	command exists but the parameter is invalid
value out of range-<command>	parameter is out of the allowed range
read only command-<command>	parameter is read only

3.3 LabVIEW® Driver VIs

LabVIEW® is a graphical programming language often used in the laboratory environment. In order to accelerate the software development of experiments written in LabVIEW®, TOPTICA Photonics AG provides a full set of driver VIs which enables the user to implement the DigiLock 110 in an easy manner. These VIs are developed with LabVIEW® version 7.1.1 Therefore you only can use the driver with versions ≥ 7.1.1

Before the drivers can be used, they have to be installed in the LabVIEW® palette of VIs. To do so follow the steps below (consult the LabVIEW® documentation for further details):

1. Open the folder "Remote Control \LabVIEW" on the Digilock 110 Software Installation CD.

2. Copy the folder "DigiLock110" found in the above folder to the following destination: <LabVIEW installation directory>\instr.lib\.

3. When LabVIEW® is started the next time, a new subpalette is available. It can be found in Instrument I/O | Instrument Drivers | DigiLock110 (see Figure 4).

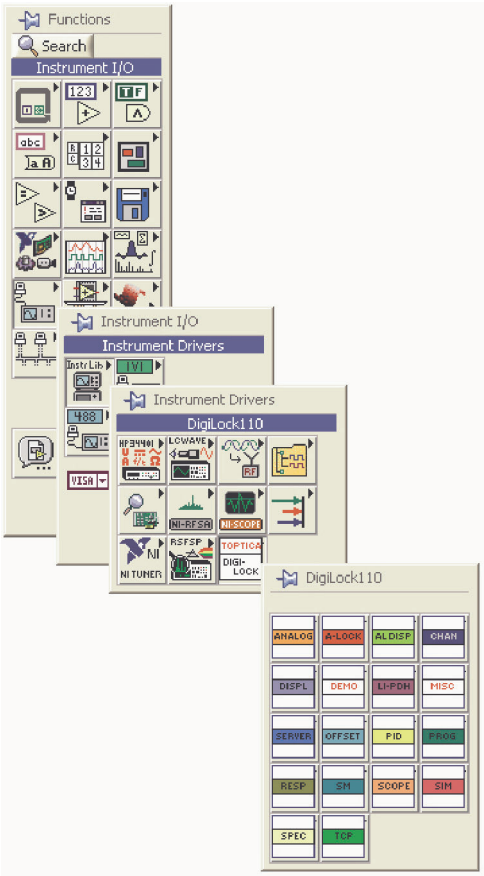

Figure 4 Set of LabVIEW® VIs to control the DigiLock 110

4 Appendix

4.1 Directories of the DigiLock 110 Software

NOTE !	%SystemDrive% refers to the harddrive partition where Windows and the user specific application data are installed, typically C:\. <UserProfile> is the currently active Windows user account.

	English version	**German version**
Standard installation directory	%SystemDrive%\Programs\ Toptica\DigiLock	%SystemDrive%\ Programme\Toptica\DigiLock
Default configuration files	%SystemDrive%\Documents and Settings\ <UserProfile>\Application Data\Toptica\DigiLock\ Program Files\Profiles	%SystemDrive%\Dokumente und Einstellungen\ <UserProfile>\Anwendungs-daten\Toptica\DigiLock\ Program Files\Profiles
Module dependent calibration files	%SystemDrive%\Documents and Settings\ <UserProfile>\Application Data\Toptica\DigiLock\ Program Files\Calibration	%SystemDrive%\Dokumente und Einstellungen\ <UserProfile>\Anwendungs-daten\Toptica\DigiLock\ Program Files\Calibration

Table 1

4.2 Application Programming Interface (API) Commands

The following tables list all available commands of the DigiLock remote API. In addition to the access type and the data type a short description is given. More details on the function of the commands can be found in the explanations of the corresponding control in the DigiLock 110 manual.

4.2.1 Digilock Module Server (DMS) RCI

Command	Access Type	Data Type	Description
access control	S	enum	access control of the commands
commandlist	Q	array	displays all available commands
echo	S	boolean	echoes the sent characters
messages waiting	Q	numeric	messages waiting in the command queue
module:connect	Q,S	boolean	establish a connection to the selected DigiLock module
module:show	Q,S	boolean	show user interface of active DigiLock module
modules:connection status	Q	array	list of connection statuses of modules
modules:names	Q	array	list of module names
modules:port numbers	Q	array	list of port numbers of modules
modules:serial numbers	Q	array	list of serial numbers of modules
number connected modules	Q	numeric	number of DigiLock modules with a connection established
number of modules	Q	numeric	number of DigiLock modules connected to the computer
program:exit	Q,S	boolean	stops the application
program:ip address	Q,S	string	ip address of this computer
program:port number	Q,S	numeric	port number of the module server
program:update module list	Q,S	boolean	search for connected DigiLock modules
selected module	Q,S	numeric	index of the currently active module

4.2.2 Digilock User Interface (DUI) RCI

Command	Access Type	Data Type	Description
access control	S	enum	access control of the commands
analog:lock:enable	Q,S	boolean	lock the analog controller
analog:proportional	Q,S	numeric	proportional gain of analog controller
analog:sign	Q,S	boolean	sign of analog controller
analog:slope	Q,S	boolean	slope of analog controller
analog:tab	Q,S	enum	
autolock:controller: analog	Q,S	boolean	activate the analog controller for autolock
autolock:controller: pid1	Q,S	boolean	activate PID 1 for autolock
autolock:controller: pid2	Q,S	boolean	activate PID 2 for autolock
autolock:cursor:snap	Q,S	boolean	cursor snaps to the setpoint
autolock:cursor:track	Q,S	boolean	cursor tracks the trace location
autolock:display: active trace	Q,S	enum	trace in the autolock display to which the lock cursor snaps
autolock:display:ch1: channel	Q,S	enum	channel 1 of the autolock graph
autolock:display:ch1: mean	Q	numeric	mean value of CH 1 data
autolock:display:ch1: overload	Q	boolean	CH 1 overload
autolock:display:ch1: rms	Q	numeric	root mean square value of CH 1 data
autolock:display:ch1: show	Q,S	boolean	show/hide CH 1 trace
autolock:display:ch2: channel	Q,S	enum	channel 2 of the autolock graph
autolock:display:ch2: mean	Q	numeric	mean value of CH 2 data
autolock:display:ch2: overload	Q	boolean	CH 2 overload
autolock:display:ch2: rms	Q	numeric	root mean square value of CH 2 data
autolock:display:ch2: show	Q,S	boolean	show/hide captured scan trace
autolock:display:chx: channel	Q,S	enum	x-channel of the autolock graph
autolock:display: cursor index	Q,S	numeric	index of the cursor (1...1000) for selection of the lock-point
autolock:display:graph	Q	2D array	autolock data
autolock:display:hold	Q,S	boolean	freeze traces
autolock:enable	Q,S	boolean	activation of the autolock module
autolock:input	Q,S	enum	input channel of the autolock module
autolock:lock:enable	Q,S	boolean	lock the controllers configured for autolock

Command	Access Type	Data Type	Description
autolock:lock:mode	Q,S	enum	operating mode of the autolock module
autolock:lock:strategy	Q,S	enum	strategy the autolock module uses in full automatic locking mode
autolock:relock: amplitude	Q,S	numeric	relock scan amplitude
autolock:relock:enable	Q,S	boolean	turn the relock option on/off
autolock:relock: frequency	Q,S	numeric	relock scan frequency
autolock:relock:output	Q,S	enum	PID channel whose output channel starts to scan
autolock:setpoint	Q,S	numeric	setpoint of the autolock module
autolock:smart: engage	Q,S	boolean	intelligent detection of a slope or extremum to engage a proper lock
autolock:smart: setpoint	Q,S	boolean	automatic detection of the setpoint
autolock:spectrum	Q,S	enum	display channel for the autolock graph
autolock:spectrum: enable	Q,S	boolean	automatic setting of the spectrum channel
autolock:tab	Q,S	enum	
autolock:window: channel	Q,S	enum	detection channel for supervision limits
autolock:window: enable	Q,S	boolean	turn the lock window option on/off
autolock:window:max	Q,S	numeric	maximum allowed signal level
autolock:window:min	Q,S	numeric	minimum allowed signal level
autolock:window:reset: delay	Q,S	numeric	time to wait from triggering the window to reset/ relock
aux in:invert	Q,S	boolean	inversion of the aux in signal
aux in:low pass:bypass	Q,S	boolean	bypass the low-pass filter
aux in:low pass: frequency	Q,S	numeric	cut-off frequency of the low-pass filter
aux in:low pass:order	Q,S	numeric	order of the low-pass filter
commandlist	Q	array	displays all available commands
controller:tab	Q,S	enum	
dio out:function	Q,S	enum	function of the DIO output
dio out:manual state	Q,S	boolean	manual state of the DIO output
display:sampling	Q,S	boolean	turn the sampling on/off
display:update rate	Q,S	numeric	display update rate
display:view	Q,S	enum	select the corresponding display view
echo	S	boolean	echos the sent characters
function:view	Q,S	enum	select the corresponding function view
li/pdh:modulation: signal type	Q,S	enum	signal type of the modulation signal for LI and PDH
li/pdh:tab	Q,S	enum	

Command	Access Type	Data Type	Description
li:first filter notch	Q,S	enum	frequency of the first filter notch in fractions of the modulation frequency
li:input	Q,S	enum	input channel of LI-module
li:modulation: amplitude	Q,S	numeric	modulation amplitude
li:modulation:enable	Q,S	boolean	turn the modulation on/off
li:modulation: frequency act	Q	numeric	actual modulation frequency
li:modulation: frequency set	Q,S	numeric	set modulation frequency
li:modulation:output	Q,S	enum	modulation output channel
li:offset	Q,S	numeric	offset value that is subtracted from the original LI-signal
li:phase adjust	Q,S	boolean	automatic adjustment of phase shift
li:phase shift	Q,S	numeric	phase shift between modulation and reference signal
main in:gain	Q,S	enum	analog gain value
main in: high pass:bypass	Q,S	boolean	bypass the high-pass filter
main in:high pass: frequency	Q,S	numeric	cut-off frequency of the high-pass filter
main in:high pass:order	Q,S	numeric	order of the high-pass filter
main in:input offset	Q,S	numeric	input offset subtraced from <Main in>
main in:invert	Q,S	boolean	inversion of the main in signal
main in:low pass: bypass	Q,S	boolean	bypass the low-pass filter
main in:low pass: frequency	Q,S	numeric	cut-off frequency of the low-pass filter
main in:low pass:order	Q,S	numeric	order of the low-pass filter
messages waiting	Q	numeric	messages waiting in the command queue
offset:output	Q,S	enum	output destination for DC-offset
offset:value	Q,S	numeric	DC-offset voltage of the selected output channel
pdh:input	Q,S	enum	input channel of PDH-module
pdh:modulation: amplitude	Q,S	numeric	modulation amplitude
pdh:modulation: enable	Q,S	boolean	turn the modulation on/off
pdh:modulation: frequency set	Q,S	enum	set modulation frequency
pdh:modulation: output	Q,S	enum	modulation output channel
pdh:offset	Q,S	numeric	offset value that is subtracted from the original PDH-signal
pdh:phase adjust	Q,S	boolean	automatic adjustment of phase shift
pdh:phase shift	Q,S	numeric	phase shift between modulation and reference signal

Command	Access Type	Data Type	Description
pid1:differential	Q,S	numeric	differential gain of PID 1
pid1:gain	Q,S	numeric	overall gain of PID 1
pid1:input	Q,S	enum	input channel of PID 1
pid1:integral	Q,S	numeric	integral gain of PID 1
pid1:integral:cutoff: enable	Q,S	boolean	activation of I cut-off
pid1:integral:cutoff: frequency	Q,S	numeric	cut-off frequency for integral gain
pid1:limit enable	Q,S	boolean	activation of output limits
pid1:limit:max	Q,S	numeric	maximum relative output level of PID 1
pid1:limit:min	Q,S	numeric	minimum relative output level of PID 1
pid1:lock:enable	Q,S	boolean	lock the PID1 controller
pid1:lock:state	Q	boolean	lock status of the PID 1 controller
pid1:output	Q,S	enum	output channel of PID 1
pid1:proportional	Q,S	numeric	proportional gain of PID 1
pid1:relock:amplitude	Q,S	numeric	relock scan amplitude
pid1:relock:enable	Q,S	boolean	turn the relock option on/off
pid1:relock:frequency	Q,S	numeric	relock scan frequency
pid1:setpoint	Q,S	numeric	setpoint of PID 1
pid1:sign	Q,S	boolean	sign of PID 1
pid1:slope	Q,S	boolean	slope to lock to for PID 1
pid1:tab	Q,S	enum	
pid1:turnoff reset: autolock	Q,S	boolean	when turning off PID 1 in autolock mode a reset may be configured
pid1:turnoff reset: manual	Q,S	boolean	when turning off PID 1 in manual mode a reset may be configured
pid1:window:channel	Q,S	enum	detection channel for supervision limits
pid1:window:enable	Q,S	boolean	turn the lock window option on/off
pid1:window:max	Q,S	numeric	maximum allowed signal level
pid1:window:min	Q,S	numeric	minimum allowed signal level
pid1:window:reset: delay	Q,S	numeric	time to wait from triggering the window to reset/ relock
pid1:window:reset: enable	Q,S	boolean	reset the PID if the window is triggered
pid1:window:reset:rate	Q,S	numeric	rate at which the PID output is reset
pid2:differential	Q,S	numeric	differential gain of PID 2
pid2:gain	Q,S	numeric	overall gain of PID 2
pid2:input	Q,S	enum	input channel of PID 2
pid2:integral	Q,S	numeric	integral gain of PID 2
pid2:limit:enable	Q,S	boolean	activation of output limits
pid2:limit:max	Q,S	numeric	maximum relative output level of PID 2
pid2:limit:min	Q,S	numeric	minimum relative output level of PID 2
pid2:lock:enable	Q,S	boolean	lock the PID2 controller

Command	Access Type	Data Type	Description
pid2:lock:state	Q	boolean	lock status of the PID 2 controller
pid2:low pass:bypass	Q,S	boolean	bypass the low-pass filter
pid2:low pass: frequency	Q,S	numeric	cut-off frequency of the low-pass filter
pid2:low pass:order	Q,S	numeric	order of the low-pass filter
pid2:output	Q,S	enum	output channel of PID 2
pid2:proportional	Q,S	numeric	proportional gain of PID 2
pid2:relock:amplitude	Q,S	numeric	relock scan amplitude
pid2:relock:enable	Q,S	boolean	turn the relock option on/off
pid2:relock:frequency	Q,S	numeric	relock scan frequency
pid2:setpoint	Q,S	numeric	setpoint of PID 2
pid2:sign	Q,S	boolean	sign of PID 2
pid2:slope	Q,S	boolean	slope to lock to for PID 2
pid2:tab	Q,S	enum	
pid2:turnoff reset: autolock	Q,S	boolean	when turning off PID 2 in autolock mode a reset may be configured
pid2:turnoff reset: manual	Q,S	boolean	when turning off PID 2 in manual mode a reset may be configured
pid2:window:channel	Q,S	enum	detection channel for supervision limits
pid2:window:enable	Q,S	boolean	turn the lock window option on/off
pid2:window:max	Q,S	numeric	maximum allowed signal level
pid2:window:min	Q,S	numeric	minimum allowed signal level
pid2:window:reset: delay	Q,S	numeric	time to wait from triggering the window to reset/ relock
pid2:window:reset: enable	Q,S	boolean	reset the PID if the window is triggered
pid2:window:reset:rate	Q,S	numeric	rate at which the PID output is reset
program:disable popup windows	Q,S	boolean	disable all popup windows during runtime
program:exit	Q,S	boolean	
program: sample when hidden	Q,S	boolean	set the sampling option when the frontend is hidden
program:status bar	Q	numeric	display of the progress of a procedure
program: system message	Q,S	string	display of system messages
program: upload firmware	Q,S	enum	configuration of firmware upload procedure at start-up
response:graph	Q,S	2D array	response data
response:input	Q,S	enum	response signal input
response:modulation: amplitude	Q,S	numeric	modulation amplitude
response:modulation: output	Q,S	enum	stimulus signal output

Command	Access Type	Data Type	Description
response:show: reference	Q,S	boolean	show/hide the reference tracks in the graph
response:show: time signal	Q,S	boolean	show/hide the response signal on the scope
response:start	Q,S	boolean	start the measurement
response:sweep: averaging	Q,S	numeric	# of averaging points per measurement frequency
response:sweep: samples	Q,S	numeric	# of samples within the frequency sweep range
response:sweep: scaling	Q,S	enum	mode of spreading the measurement points across the frequency sweep range
response:sweep:start	Q,S	numeric	start frequency of the sweep
response:sweep:stop	Q,S	numeric	stop frequency of the sweep
scan:amplitude	Q,S	numeric	scan amplitude
scan:enable	Q,S	boolean	turn scan control on / off
scan:frequency	Q,S	numeric	scan frequency
scan:output	Q,S	enum	output destination of scan signal
scan:signal type	Q,S	enum	waveform type of output signal
scope:average: number	Q,S	numeric	# of averaged consecutive traces of one scope channel
scope:ch1:average: enable	Q,S	boolean	activate trace averaging for CH 1
scope:ch1:channel	Q,S	enum	input signal to CH 1 of the scope
scope:ch1:mean	Q	numeric	mean value of CH 1 data
scope:ch1:overload	Q	boolean	CH 1 overload
scope:ch1:rms	Q	numeric	root mean square value of CH 1 data
scope:ch1:show	Q,S	boolean	show/hide CH 1 trace
scope:ch1:smooth: enable	Q,S	boolean	activate smoothing for CH 1 trace
scope:ch2:average: enable	Q,S	boolean	activate trace averaging for CH 2
scope:ch2:channel	Q,S	enum	input signal to CH 2 of the scope
scope:ch2:mean	Q	numeric	mean value of CH 2 data
scope:ch2:overload	Q	boolean	CH 2 overload
scope:ch2:rms	Q	numeric	root mean square value of CH 2 data
scope:ch2:show	Q,S	boolean	show/hide CH 2 trace
scope:ch2:smooth: enable	Q,S	boolean	activate smoothing for CH 2 trace
scope:chx:channel	Q,S	enum	input signal to the x-channel in the xy-mode
scope:graph	Q	2D array	scope data
scope:smooth:number	Q,S	numeric	# of averaged consecutive data points in one scope trace
scope:timescale	Q,S	enum	time span
scope:xymode	Q,S	boolean	activate the xy-mode

Command	Access Type	Data Type	Description
settings:tab	Q,S	enum	
simulation:frequency: start	Q,S	numeric	minimum display frequency
simulation:frequency: stop	Q,S	numeric	maximum display frequency
simulation:frequency: unit	Q,S	enum	units used on the frequency axis
simulation:graph	Q	2D array	simulation data
simulation:pid: differential	Q,S	numeric	derivative gain
simulation:pid:gain	Q,S	numeric	overall gain
simulation:pid: get parameters	S	boolean	get parameters of the selected PID controller
simulation:pid:input	Q,S	enum	input channel for the selected PID
simulation:pid:integral	Q,S	numeric	integral gain
simulation:pid:integral: cutoff:enable	Q,S	boolean	activate the I cut-off
simulation:pid:integral: cutoff:frequency	Q,S	numeric	cut-off frequency for integral gain
simulation:pid:output	Q,S	enum	output channel of the selected controller
simulation:pid: proportional	Q,S	numeric	proportional gain
simulation:pid: send parameters	S	boolean	send the chosen parameters to the selected PID controller
simulation:selected pid	Q,S	enum	selected PID controller
spectrum:average: number	Q,S	numeric	# of averaged consecutive traces of one spectrum analysis channel
spectrum:ch1: average:enable	Q,S	boolean	activate trace averaging for CH 1
spectrum:ch1:channel	Q,S	enum	input signal to CH 1 of the spectrum analyzer
spectrum:ch1:hold	Q,S	boolean	freeze CH 1 trace
spectrum:ch1:hold	Q,S	boolean	freeze CH 1 trace
spectrum:ch1:mean	Q	numeric	mean value of CH 1 data
spectrum:ch1:overload	Q	boolean	CH 1 overload
spectrum:ch1:rms	Q	numeric	root mean square value of CH 1 data
spectrum:ch1:show	Q,S	boolean	show/hide CH 1 trace
spectrum:ch1:show dc	Q,S	boolean	show the DC-component of the frequency spectrum for CH 1
spectrum:ch1:smooth: enable	Q,S	boolean	activate smoothing for CH 1 trace
spectrum:ch2: average:enable	Q,S	boolean	activate trace averaging for CH 2
spectrum:ch2:channel	Q,S	enum	input signal to CH 2 of the spectrum analyzer
spectrum:ch2:hold	Q,S	boolean	freeze CH 2 trace
spectrum:ch2:hold	Q,S	boolean	freeze CH 2 trace

Command	Access Type	Data Type	Description
spectrum:ch2:mean	Q	numeric	mean value of CH 2 data
spectrum:ch2:overload	Q	boolean	CH 2 overload
spectrum:ch2:rms	Q	numeric	root mean square value of CH 2 data
spectrum:ch2:show	Q,S	boolean	show/hide CH 2 trace
spectrum:ch2:show dc	Q,S	boolean	show the DC-component of the frequency spectrum for CH 2
spectrum:ch2:smooth: enable	Q,S	boolean	activate smoothing for CH 2 trace
spectrum:frequency scale	Q,S	enum	frequency span
spectrum:graph	Q	2D array	spectrum data
spectrum:smooth: number	Q,S	numeric	# of averaged consecutive data points in one spectrum analysis trace
visibility:li/pdh	Q,S	boolean	show/hide lock-in and PDH module
visibility:lock	Q,S	boolean	show/hide lock navigation button
visibility:lock	Q,S	boolean	show/hide lock navigation button
visibility:offset	Q,S	boolean	show/hide offset module
visibility:pid	Q,S	boolean	show/hide autolock and PID module
visibility:response	Q,S	boolean	show/hide response navigation button
visibility:scan	Q,S	boolean	show/hide scan module
visibility:settings	Q,S	boolean	show/hide settings navigation button
visibility:simulation	Q,S	boolean	show/hide simulation navigation button
visibility:system	Q,S	boolean	show/hide system navigation button