



CLD101x

Write Your Own Application

- **Drivers**
- **Samples**
- **Command Reference**

Version: 1.5
Date: 12/2/2016

Contents

Foreword	0
1 Write Your Own Application	2
1.1 32 bit Version _____	3
1.2 64 bit Version _____	5
1.3 Command Reference _____	8
1.3.1 IEEE488.2 Common Commands _____	8
1.3.2 SCPI Command Reference _____	9
1.3.3 Status Reporting _____	16
1.3.4 Error Reporting _____	18

1 Write Your Own Application

In order to write your own application, you need a specific instrument driver and some tools for use in different programming environments. The driver and tools are being installed to your computer during software installation and cannot be found on the installation CD.

In this section the location of drivers and files, required for programming in different environments, are given for installation under Windows XP (32 bit), Windows VISTA (32 and 64 bit) and Windows 7 (32 and 64 bit).

In the table below you will find a summary of what files you need for particular programming environments.

Programming environment	Necessary files
C, C++, CVI	*.h (header file) *.lib (static library)
C#	.net wrapper dll
Visual Studio	*.h (header file) *.lib (static library) or .net wrapper dll
LabView	*.fp (function panel) and VXIpn Instrument Driver Beside that, LabVIEW driver vi's are provided with the *.llb container file

Note

All above environments require also the VXIpn Instrument Driver dll !

In the next sections the locations of above files are described in detail.

1.1 32 bit Version

Note

According to the VPP6 (Rev6.1) Standard the installation of the 32 bit VXIpnnp driver includes both the WINNT and GWINNT frameworks.

VXIpnnp Instrument driver:

C:\Program Files\IVI Foundation\VISA\WinNT\Bin\TL4000_32.dll

Note

This instrument driver is required for all development environments!

Source file

C:\Program Files\IVI Foundation\VISA\WinNT\TL4000\TL4000.c

Header file

C:\Program Files\IVI Foundation\VISA\WinNT\include\TL4000.h

Static Library

C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc\TL4000_32.lib

Function Panel

C:\Program Files\IVI Foundation\VISA\WinNT\TL4000\TL4000.fp

Online Help for VXIpnnp Instrument driver:

C:\Program Files\IVI Foundation\VISA\WinNT\TL4000\Manual

NI LabVIEW driver

The LabVIEW Driver is a 32 bit driver and compatible with 32bit NI-LabVIEW versions 8.5 and higher only.

C:\Program Files\National Instruments\LabVIEW xxxx\Instr.lib\TL4000...
...\TL4000.llb

(LabVIEW container file with driver vi's and an example. "LabVIEW xxxx" stands for actual LabVIEW installation folder.)

.net wrapper dll

C:\Program Files\Microsoft.NET\Primary Interop Assemblies...
...\Thorlabs.TL4000.dll

C:\Program Files\IVI Foundation\VISA\VisaCom\...
...\Primary Interop Assemblies\Thorlabs.TL4000.dll

Example for NI LabWindows/CVI (C)

Project file (NI-LabWindows™/CVI 2010):

C:\Program Files\IVI Foundation\VISA\WinNT\TL4000\Samples\CVI\...
...sample.prj

Source file:

C:\Program Files\IVI Foundation\VISA\WinNT\TL4000\Samples\CVI\...
...sample.c

Executable sample demo:

C:\Program Files\IVI Foundation\VISA\WinNT\TL4000\Samples\CVI\...
...sample.exe

Example for MS Visual Studio (C++)

Solution file:

C:\Program Files\IVI Foundation\visa\WinNT\TL4000\Samples\...
...MS_VCpp\Sample.sln

Project file:

C:\Program Files\IVI Foundation \visa\WinNT\TL4000\Samples\...
...MS_VCpp\Sample\Sample.vcproj

Executable sample demo:

C:\Program Files\IVI Foundation \visa\WinNT\TL4000\Samples\...
...MS_VCpp\Release\Sample.exe

MS Visual Studio, .NET (C#)

Solution file:

C:\Program Files\IVI Foundation\visa\WinNT\TL4000\Samples...
...\MS.NET_CS\Thorlabs.TL4000Test.sln

Project file:

C:\Program Files\IVI Foundation\visa\WinNT\TL4000\Samples...
...\MS.NET_CS\TL4000Test\Thorlabs.TL4000Test.csproj

Executable sample demo:

C:\Program Files\IVI Foundation\visa\WinNT\TL4000\Samples...
...\MS.NET_CS\TL4000Test\bin\Release\TL4000Test.exe

(Select the correct type and device mode, e.g., TMC or DFU, and enter serial number, then connect)

Example for LabVIEW

C:\Program Files\National Instruments\LabVIEW xxxx\Instr.lib\TL4000...
...\TL4000.llb

(LabVIEW container file with driver vi's and an example. "LabVIEW xxxx" stands for actual LabVIEW installation folder.)

1.2 64 bit Version

Note

According to the VPP6 (Rev6.1) Standard the installation of the 64 bit VXIppnp driver includes the WINNT, WIN64, GWINNT and GWIN64 frameworks. That means, that the 64 bit driver includes the 32 bit driver as well.

In case of a 64 bit operating system, 64bit drivers and applications are installed to

`"C:\Program Files"`

while the 32 bit files - to

`"C:\Program Files (x86)"`

Below are listed both installation locations, so far applicable.

VXIppnp Instrument driver:

`C:\Program Files (x86)\ivi foundation\visa\WinNT\Bin\TL4000_32.dll`

`C:\Program Files\IVI Foundation\VISA\Win64\Bin\TL4000_32.dll`

`C:\Program Files\IVI Foundation\VISA\Win64\Bin\TL4000_64.dll`

Note

This instrument driver is required for all development environments!

Source file

`C:\Program Files (x86)\ivi foundation\visa\WinNT\TL4000\TL4000.c`

`C:\Program Files\IVI Foundation\VISA\Win64\TL4000\TL4000.c`

Header file

`C:\Program Files\IVI Foundation\VISA\Win64\TL4000\TL4000.h`

`C:\Program Files (x86)\ivi foundation\visa\WinNT\TL4000\TL4000.h`

Static Library

`C:\Program Files (x86)\ivi foundation\visa\WinNT\lib\msc\TL4000_32.lib`

`C:\Program Files\IVI Foundation\VISA\Win64\lib\msc\TL4000_32.lib`

`C:\Program Files\IVI Foundation\VISA\Win64\Lib_x64\msc\TL4000_64.lib`

Function Panel

`C:\Program Files (x86)\ivi foundation\visa\WinNT\TL4000\TL4000.fp`

`C:\Program Files\IVI Foundation\VISA\Win64\TL4000\TL4000.fp`

Online Help for VXIppnp Instrument driver:

`C:\Program Files\IVI Foundation\VISA\WinNT\TL4000\Manual`

NI LabVIEW driver

The LabVIEW Driver supports 32bit and 64bit NI-LabVIEW2009 and higher.

`C:\Program Files\National Instruments\LabVIEW xxxx\Instr.lib\TL4000...`

`...\TL4000.llb`

(LabVIEW container file with driver vi's and an example. "LabVIEW xxxx" stands for actual LabVIEW installation folder.)

.net wrapper dll

C:\Program Files (x86)\Microsoft.NET\Primary Interop Assemblies...
...Thorlabs.TL4000.dll
C:\Program Files (x86)\ivi foundation\visa\VisaCom\...
...Primary Interop Assemblies\Thorlabs.TL4000.dll

Example for NI LabWindows/CVI (C)

Project file (NI-LabWindows™/CVI 2010):

C:\Program Files\IVI Foundation\VISA\Win64\TL4000\Samples\CVI\...
...sample.prj

Source file:

C:\Program Files\IVI Foundation\VISA\Win64\TL4000\Samples\CVI\...
...sample.c

Executable sample demo:

C:\Program Files\IVI Foundation\VISA\Win64\TL4000\Samples\CVI\...
...sample.exe

Example for MS Visual Studio (C++)

Solution file:

C:\Program Files\IVI Foundation\VISA\Win64\TL4000\Samples\...
...MS_VCpp\Sample.sln

Project file:

C:\Program Files\IVI Foundation\VISA\Win64\TL4000\Samples\...
...MS_VCpp\Sample\Sample.vcproj

Executable sample demo:

C:\Program Files\IVI Foundation\VISA\Win64\TL4000\Samples\...
...MS_VCpp\Release\Sample.exe

MS Visual Studio, .NET (C#)

Solution file:

C:\Program Files\IVI Foundation\VISA\Win64\TL4000\Samples...
...\MS.NET_CS\Thorlabs.TL4000Test.sln

Project file:

C:\Program Files\IVI Foundation\VISA\Win64\TL4000\Samples...
...\MS.NET_CS\TL4000Test\Thorlabs.TL4000Test.csproj

Executable sample demo:

```
C:\Program Files\IVI Foundation\VISA\Win64\TL4000\Samples...  
...\MS.NET_CS\TL4000Test\bin\Release\TL4000Test.exe
```

(Select the correct type and device mode, e.g., TMC or DFU, and enter serial number, then connect)

Example for LabView

```
C:\Program Files\National Instruments\LabVIEW xxxx\Instr.lib\TL4000...  
...\TL4000.llb
```

(LabVIEW container file with driver vi's and an example. "LabVIEW xxxx" stands for actual LabVIEW installation folder.)

1.3 Command Reference

1.3.1 IEEE488.2 Common Commands

Common commands are device commands that are common to all devices according to the IEEE488.2 standard. These commands are designed and defined by this standard. Most of the commands are described in detail in this section. The following common commands associated with the status structure are covered in the “Status Structure” section: *CLS, *ESE, *ESE?, *ESR?, *SRE,

Command summary

Mnemonic	Name	Description
*CLS	Clear status	Clears all event registers and Error Queue
*ESE <NRf>	Event enable command	Sets the Standard Event Enable Register
*ESE?	Event enable query	Returns the Standard Event Enable Register
*ESR?	Event status register query	Returns and clear the Standard Event Register
*IDN?	Identification query	Returns the unit's identification string
*OPC	Operation complete command	Sets the Operation Complete bit in the Standard Event Register
*OPC?	Operation complete query	Places a “1” into the output queue when all device operations have been completed
*RST	Reset command	Returns the unit to the *RST default condition
*SRE <NRf>	Service request enable command	Sets the Service Request Enable Register
*SRE?	Service request enable query	Returns the Service Request Enable Register
*STB?	Status byte query	Returns the Status Byte Register
*TST?	Self-test query	Performs the unit's self-test and returns the result.
*WAI	Wait-to-continue command	Waits until all previous commands are executed

Command reference

1. *IDN? – identification query - read identification code

The identification code includes the manufacturer, model code, serial number, and firmware revision levels and is sent in the following format: Thorlabs,MMM,SSS,X.X.X, where

MMM is the model code
 SSS is the serial number
 X.X.X is the instrument firmware revision level

2. *OPC – operation complete - set OPC bit

3. *OPC? – operation complete query – places a “1” in output queue

When *OPC is sent, the OPC bit in the Standard Event Register will set after all pending command operations are complete. When *OPC? is sent, an ASCII “1” is placed in the Output Queue after all pending command operations are complete.

Typically, either one of these commands is sent after the INITiate command. The INITiate command is used to take the instrument out of idle in order to perform measurements. While operating within the trigger model layers, many sent commands will not execute. After all programmed operations are completed, the instrument returns to the idle state at which time all pending commands (including *OPC and/or *OPC?) are executed. After the last pending command is executed, the OPC bit and/or an ASCII “1” is placed in the Output Queue.

4. *RST – reset – return instrument to defaults

When the *RST command is sent, the instrument performs the following operations:

- Returns the instrument to default conditions
- Cancels all pending commands.
- Cancels response to any previously received *OPC and *OPC? commands.

5. *TST? – self-test query – run self test and read result

Use this query command to perform the instrument self-test routine. The command places the coded result in the Output Queue. A returned value of zero (0) indicates that the test passed, other values indicate that the test failed.

6. *WAI – wait-to-continue – wait until previous commands are completed

The *WAI command is a no operation command for the instrument and thus, does not need to be used. It is there for conformance to IEEE488.2.

1.3.2 SCPI Command Reference

SYSTEM subsystem commands

Command	Description	SCPI
SYSTEM	Path to SYSTEM subsystem.	<input checked="" type="checkbox"/>
:BEEPer		<input checked="" type="checkbox"/>
[:IMMediate]	Issues an audible signal	<input checked="" type="checkbox"/>
:STAtE {ON 1 OFF 0}	Activates/deactivates the beeper	<input checked="" type="checkbox"/>
:STAtE?	Returns the state of the beeper	<input checked="" type="checkbox"/>
:VOLume <value>	Sets the beeper volume	<input checked="" type="checkbox"/>
:VOLume?	Returns the beeper volume	<input checked="" type="checkbox"/>
:ERRor		<input checked="" type="checkbox"/>
[:NEXT]?	Returns the latest error code and message	<input checked="" type="checkbox"/>
:MOUNt		
[:TYPE]?	Returns the mount type (<NR1>,description)	
:VERSion?	Returns level of SCPI standard (1999.0)	<input checked="" type="checkbox"/>

DISPlay subsystem commands

Command	Description	SCPI
DISPlay	Path to DISPlay subsystem.	<input checked="" type="checkbox"/>
:BRIGHtness <value>	Sets the display brightness	<input checked="" type="checkbox"/>
:BRIGHtness?	Returns the display brightness value	<input checked="" type="checkbox"/>
:CALibratIon[:TOUCH][:INITiate]	Initiates Touchscreen calibration	
:FADeout		
[:STAtE] {ON 1 OFF 0}	Activates/deactivates automatic dimming	
[:STAtE]?	Returns the state of automatic dimming	

STATus subsystem commands

Command	Description	SCPI
STATus		<input checked="" type="checkbox"/>
:MEASurement	Path to control measurement event registers	
[:EVENT]?	Returns the event register	<input checked="" type="checkbox"/>
:CONDition?	Returns the condition register	<input checked="" type="checkbox"/>
:PTRansition <value>	Sets the positive transition filter	<input checked="" type="checkbox"/>
:PTRansition?	Returns the positive transition filter	<input checked="" type="checkbox"/>
:NTRansition <value>	Sets the negative transition filter	<input checked="" type="checkbox"/>
:NTRansition?	Returns the negative transition filter	<input checked="" type="checkbox"/>
:ENABle <value>	Sets the enable register	<input checked="" type="checkbox"/>
:ENABle?	Returns the enable register	<input checked="" type="checkbox"/>
:OPERation	Path to control operation event registers	<input checked="" type="checkbox"/>
[:EVENT]?	Returns the event register	<input checked="" type="checkbox"/>
:CONDition?	Returns the condition register	<input checked="" type="checkbox"/>
:PTRansition <value>	Sets the positive transition filter	<input checked="" type="checkbox"/>
:PTRansition?	Returns the positive transition filter	<input checked="" type="checkbox"/>
:NTRansition <value>	Sets the negative transition filter	
:NTRansition?	Returns the negative transition filter	<input checked="" type="checkbox"/>
:ENABle <value>	Sets the enable register	<input checked="" type="checkbox"/>
:ENABle?	Returns the enable register	<input checked="" type="checkbox"/>
:QUEStionable	Path to control questionable event registers	<input checked="" type="checkbox"/>
[:EVENT]?	Returns the event register	<input checked="" type="checkbox"/>
:CONDition?	Returns the condition register	<input checked="" type="checkbox"/>
:PTRansition <value>	Sets the positive transition filter	<input checked="" type="checkbox"/>
:PTRansition?	Returns the positive transition filter	<input checked="" type="checkbox"/>
:NTRansition <value>	Sets the negative transition filter	<input checked="" type="checkbox"/>
:NTRansition?	Returns the negative transition filter	<input checked="" type="checkbox"/>
:ENABle <value>	Sets the enable register	<input checked="" type="checkbox"/>
:ENABle?	Returns the enable register	<input checked="" type="checkbox"/>
:AUXiliary	Path to control auxiliary event registers	
[:EVENT]?	Returns the event register	<input checked="" type="checkbox"/>
:CONDition?	Returns the condition register	<input checked="" type="checkbox"/>
:PTRansition <value>	Sets the positive transition filter	<input checked="" type="checkbox"/>
:PTRansition?	Returns the positive transition filter	<input checked="" type="checkbox"/>
:NTRansition <value>	Sets the negative transition filter	<input checked="" type="checkbox"/>
:NTRansition?	Returns the negative transition filter	<input checked="" type="checkbox"/>
:ENABle <value>	Sets the enable register	<input checked="" type="checkbox"/>
:ENABle?	Returns the enable register	<input checked="" type="checkbox"/>
:PRESet	Return status registers to default states.	<input checked="" type="checkbox"/>

LD output subsystem commands

Command	Description	SCPI
OUTPut[1]	Path to LD output	✓
[:STATe] {ON 1 OFF 0}	Enables (ON) or disables (OFF) LD output	✓
[:STATe]?	Returns output state	✓
:PON		
[:STATe] {ON 1 OFF 0}	Sets the power-on LD output state	✓
[:STATe]?	Returns power-on LD output state setting	✓
:CONDition?	Returns the output condition (query only, 1 0)	
:FILTer[:LPASs]		✓
[:STATe] {ON 1 OFF 0}	Enables/disables LD output low pass filter	✓
[:STATe]?	Returns output filter state	✓
:PROTection	Path to LD output protection	✓
:INTLock[:TRIPped]?	Returns interlock circuit protection tripped	
:KEYLock[:TRIPped]?	Returns key lock protection tripped	
:OTEMperature[:TRIPped]?	Returns over temperature protection tripped	
:CONNection[:TRIPped]?	Returns connection failure protection tripped	
:TEMPerature		
:MODE {OFF PROTection ENABle}	Sets temperature protection mode	
:MODE?	Returns temperature protection mode	
{[:TRIPped]?	Returns protection tripped	
:POLarity?	Returns the LD polarity (AG/CG) (FW>1.8)	

PD input subsystem commands

Command	Description	SCPI
INPut[1]	Path to photodiode input	✓
:POLarity {AG CG}	Sets PD polarity (FW>1.8, CLD1011LP only)	
:POLarity?	Returns the PD polarity (AG/CG) (FW>1.8)	
:BIAS	Path to PD BIASing	✓
[:STATe] {ON 1 OFF 0}	Enables (ON) or disables (OFF) BIAS	✓
[:STATe]?	Returns BIAS state	✓
:VOLTagE	Path to PD BIAS voltage	✓
[:DC] {MIN MAX DEF <volts>}	Sets BIAS voltage value	✓
[:DC]? [{MIN MAX DEF}]	Returns BIAS voltage value	✓

PD current sensing subsystem commands

Command	Description	SCPI
SENSe[1]	Path to photodiode current sensing	✓
[:CURRent] [:DC]		✓
:RANGe[:UPPer] {MIN MAX <amps>}	Sets the photodiode range	✓
:RANGe[:UPPer]? [{MIN MAX}]	Returns the photodiode range	✓
:CORRection:POWer		
[:PDIode] [:RESPonse]	Sets the photodiode response value	
{MIN MAX DEF <amps>}		
[:PDIode] [:RESPonse]?	Returns the photodiode response value	
{MIN MAX DEF}]		
[:DATA]? [{MIN MAX}]	Returns the measured PD current	
:POWer[:DC]		
[:DATA]? [{MIN MAX}]	Returns the measured opt. LD power	

LD current sensing subsystem commands

Command	Description	SCPI
SENSE3 [:CURRent] [:DC] [:DATA]? [{MIN MAX}]	Path to laser diode current sensing Returns the measured LD current	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

LD voltage sensing subsystem commands

Command	Description	SCPI
SENSE4 [:VOLTage] [:DC] [:DATA]? [{MIN MAX}]	Path to laser diode voltage sensing Returns the measured LD voltage	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

LD source subsystem commands

Command	Description	SCPI
SOURCE[1] :FUNCTION [:MODE] {CURRENT POWER} [:MODE]? [:CURRENT] :LIMIT [:AMPLitude] {MIN MAX <amps>} [:AMPLitude]? [{MIN MAX}] :TRIPped? [:LEVEL] [:IMMEDIATE] [:AMPLitude] {MIN MAX <amps>} [:AMPLitude]? [{MIN MAX}] :POWER [:LEVEL] [:IMMEDIATE] [:AMPLitude] {MIN MAX <watts>} [:AMPLitude]? [{MIN MAX}] :DIODE[:CURRENT] [:IMMEDIATE] [:AMPLitude] {MIN MAX <amps>} [:AMPLitude]? [{MIN MAX}] :AM [:STATE] {ON 1 OFF 0} [:STATE]?	Sets LD driver source function Returns LD driver source function Sets limit current value Returns limit current value Returns limit detection tripped Sets LD current setpoint value Returns LD current setpoint value Sets LD power setpoint value Returns power setpoint value Sets LD power via photodiode current Sets photodiode current setpoint Returns the photodiode current setpoint Enables (ON), disables (OFF) modulation Returns modulation state	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

TEC output subsystem commands

Command	Description	SCPI
OUTPut2 [:STATE] {ON 1 OFF 0} [:STATE]? :PON [:STATE] {ON 1 OFF 0} [:STATE]? :PROTEction :TRANSDUCER[:TRIPped]? :OTEMperature[:TRIPped]? :CONNECTION[:TRIPped]? :COMPLiance :MODE {OFF PROTEction} :MODE? {:TRIPped}?	Path to TEC output Enable (ON) or disable (OFF) TEC output Returns output state Sets the power-on TEC output state Returns power-on TEC output state setting Path to TEC output protection Returns temperature transducer failure tripped Returns over temperature protection tripped Returns connection failure protection tripped (for firmware level 1.1.2 and higher) Sets the 'compliance voltage protection' mode Returns the 'compliance voltage protection' mode Returns 'compliance voltage protection' tripped	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

TEC driver source subsystem commands

Command	Description	SCPI
SOURce2		✓
:FUNCTION		✓
[:MODE] {TEMPerature CURRENT}	Sets TEC driver source function	✓
[:MODE]?	Returns TEC driver source function	✓
:CURRENT		✓
:LIMIT		
[:AMPLitude] {MIN MAX <amps>}	Sets limit current value	✓
[:AMPLitude]? [{MIN MAX}]	Returns limit current value	✓
[:LEVel] [:IMMediate]		
[:AMPLitude] {MIN MAX <amps>}	Sets current setpoint value	✓
[:AMPLitude]? [{MIN MAX}]	Returns current setpoint value	✓
:DATA? [{MIN MAX}]	Returns the TEC current value	
[:TEMPerature]		✓
[:SPOint] {MIN MAX DEF <temp>}	Sets temperature setpoint	✓
[:SPOint]? [{MIN MAX DEF}]	Returns temperature setpoint	✓
:LIMIT		✓
[:UPPer] {MIN MAX <temp>}	Sets settable temperature high limit	✓
[:UPPer]? [{MIN MAX}]	Returns settable temperature high limit	✓
:LOWer {MIN MAX <temp>}	Sets settable temperature low limit	✓
:LOWer? [{MIN MAX}]	Returns settable temperature low limit	✓
:LCONstants		✓
[:GAIN] {MIN MAX DEF <value>}	Sets PID loop P value	✓
[:GAIN]? [{MIN MAX DEF}]	Returns PID loop P value	✓
:INTegral {MIN MAX DEF <value>}	Sets PID loop I value	✓
:INTegral? [{MIN MAX DEF}]	Returns PID loop I value	✓
:DERivative {MIN MAX DEF <value>}	Sets PID loop D value	✓
:DERivative? [{MIN MAX DEF}]	Returns PID loop D value	✓
:PERiod {MIN MAX DEF <sec>}	Sets thermal load oscillation period	
:PERiod? [{MIN MAX DEF}]	Returns thermal load oscillation period	

Note In local operation mode (front panel operation) it is not possible to operate the TEC-Source (SOURce2) in current mode. When switching from remote mode to local operation mode the source function automatically switches to temperature mode.

Temperature sensing subsystem commands

Command	Description	SCPI
SENSe2		<input checked="" type="checkbox"/>
[:TEMPerature]		
:DATA? [{MIN MAX}]	Returns the temperature value	
[:THERmistor]		
:METHod {EXPonential SHH}	Sets thermistor calculation method	
:METHod?	Returns thermistor calculation method	
[:SHH]		
:A {MIN MAX DEF <value>}	Sets Steinhart-Hart parameter A	
:A? [{MIN MAX DEF}]	Returns Steinhart-Hart parameter A	
:B {MIN MAX DEF <value>}	Sets Steinhart-Hart parameter B	
:B? [{MIN MAX DEF}]	Returns Steinhart-Hart parameter B	
:C {MIN MAX DEF <value>}	Sets Steinhart-Hart parameter C	
:C? [{MIN MAX DEF}]	Returns Steinhart-Hart parameter C	
:EXPonential		
:R0 {MIN MAX DEF <ohms>}	Sets exponential parameter R_0	
:R0? [{MIN MAX DEF}]	Returns exponential parameter R_0	
:T0 {MIN MAX DEF <temp>}	Sets exponential parameter T_0	
:T0? [{MIN MAX DEF}]	Returns exponential parameter T_0	
:BETA {MIN MAX DEF <value>}	Sets exponential parameter beta	
:BETA? [{MIN MAX DEF}]	Returns exponential parameter beta	
:PROtection		<input checked="" type="checkbox"/>
:DELay {MIN MAX DEF <sec>}	Sets protection delay	
:DELay? [{MIN MAX DEF}]	Returns protection delay	
:WINDow[:AMPLitude]	Sets temperature window amplitude	
{MIN MAX DEF <temp>}		
:WINDow[:AMPLitude]?	Returns temperature window amplitude	
[{MIN MAX DEF}]		
[:TRIPped]?	Returns protection tripped	
:RESistance		
[:DATA]? [{MIN MAX}]	Returns the resistance value	

UNIT subsystem commands

Command	Description	SCPI
UNIT		<input checked="" type="checkbox"/>
:TEMPerature	Sets the temperature unit	<input checked="" type="checkbox"/>
{C CEL CELSius F FAR FAHReinheit K KELVin}		
:TEMPerature?	Returns the temperature unit	<input checked="" type="checkbox"/>

CALibration subsystem commands

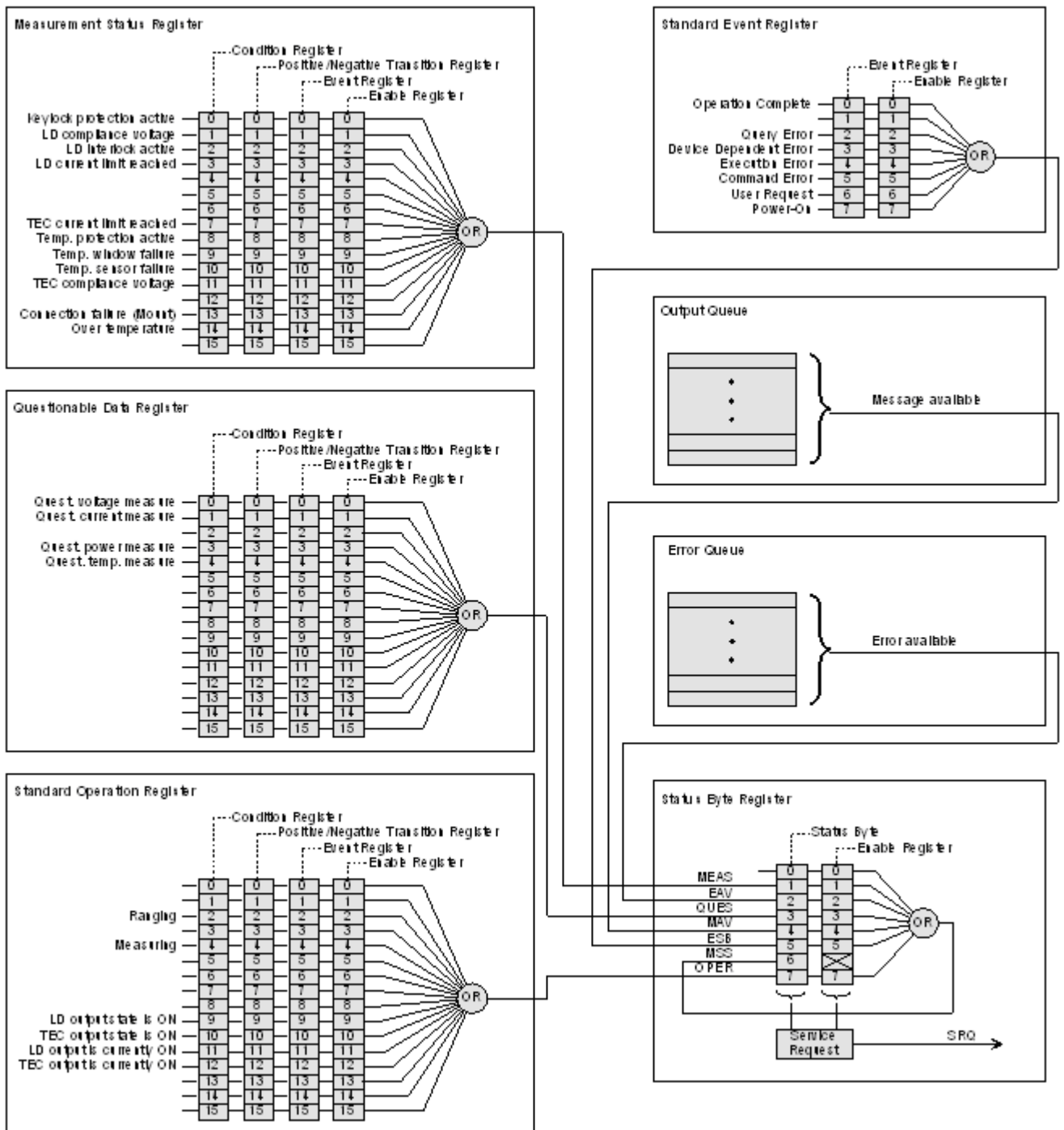
Command	Description	SCPI
CALibration		<input checked="" type="checkbox"/>
:STRing?	Returns the calibration string	

Measurement commands

Command	Description	SCPI
ABORt	Aborts current measurement	<input checked="" type="checkbox"/>
CONFigure [:SCALar]		<input checked="" type="checkbox"/>
:CURRent[1][:DC]	Configures instrument LD current measurement	<input checked="" type="checkbox"/>
:VOLTage[1][:DC]	Configures instrument LD voltage measurement	<input checked="" type="checkbox"/>
:CURRent2[:DC]	Configures instrument PD current measurement	
:POWer[1]	Configure for opt. power measurement via PD current	
:TEMPerature	Configures instrument for temperature measurement	<input checked="" type="checkbox"/>
:RESistance	Configures instrument for NTC resistance measurement	<input checked="" type="checkbox"/>
CONFigure?	Query configuration	<input checked="" type="checkbox"/>
INITiate[:IMMediate]	Starts measurement	<input checked="" type="checkbox"/>
FETCh?	Returns last measurement data	<input checked="" type="checkbox"/>
FETCh		
:CURRent[1][:DC]?	Return last LD current measurement	
:VOLTage[1][:DC]?	Return last LD voltage measurement	
:CURRent2[:DC]?	Return last PD current measurement	
:POWer[1]?	Return last opt. power measurement via PD current	
:TEMPerature?	Return last temperature measurement	
:RESistance?	Return last NTC resistance measurement	
READ?	Starts new measurement (as configured) and read data	<input checked="" type="checkbox"/>
MEASure[:SCALar]		<input checked="" type="checkbox"/>
:CURRent[1][:DC]?	Perform LD current measurement	<input checked="" type="checkbox"/>
:VOLTage[1][:DC]?	Perform LD voltage measurement	<input checked="" type="checkbox"/>
:CURRent2[:DC]?	Perform PD current measurement	
:POWer[1]?	Perform opt. power measurement via PD current	
:TEMPerature?	Perform temperature measurement	<input checked="" type="checkbox"/>
:RESistance?	Perform NTC resistance measurement	<input checked="" type="checkbox"/>

1.3.3 Status Reporting

The figure below gives an overview of the device's status reporting structure. See also section [STATUS subsystem commands](#) for a detailed description of the related commands and their syntax.



Status Byte Register

The Status Byte Register gives a summary of all underlying status structures. See also IEEE488.2-1992-§11.2.

Bit #	Mnemonic	Description
7	OPER	Standard Operation Status Structure Summary Bit
6	RQS/MSS	Request Service / Master Summary Status
5	ESB	Standard Event Status Bit
4	MAV	Message Available. There is response data available for readout
3	QUES	Questionable Status Structure Summary Bit
2	EAV	Error Available. There is at least one error in the error queue.
1	MEAS	Measurement Status Structure Summary Bit
0		reserved, read as 0

Standard Event Status Structure

The Standard Event Status Structure is described in IEEE488.2-1992-§11.5.

Standard Operation Register

The Standard Operation Status Structure is described in SCPI1999.0-Vol1-§9.3. In addition bit 8..12 are used as output state/on indicators.

Bit #	Mnemonic	Description
15..13		See SCPI1999.0-Vol1-§9.3
12	TECON	TEC output is currently ON
11	LDON	LD output is currently ON
10	TECST	TEC output state is ON
9	LDST	LD output state is ON
8		reserved, read as 0
7..0		See SCPI1999.0-Vol1-§9.3

Questionable Data Register

The Questionable Data Status Structure is described in SCPI1999.0 Vol1 §9.4.

Measurement Status Register

The Measurement Status Register Status Byte Register reports device operation and measurement states.

Bit #	Description
15	reserved, read as 0
14	Over temperature (Instrument is too hot)
13	Socket connection failure (Socket missing or unknown socket detected)

Bit #	Description
12	reserved, read as 0
11	TEC output compliance voltage reached
10	Temperature sensor failure.
9	Temperature window failure.
8	Temperature protection is active.
7	TEC current limit reached
6	reserved, read as 0
5	reserved, read as 0
4	reserved, read as 0
3	LD current limit reached
2	LD interlock is active
1	LD output compliance voltage reached
0	Keylock protection is active

1.3.4 Error Reporting

The device stores errors in a queue containing up to 10 entries. The error queue may be read out by the ``SYSTEM:ERROR[:NEXT]?'` command. The following table lists all error numbers and the according descriptive messages. Note: negative numbers are defined by SCPI while positive error numbers are device dependent.

Error	Description
0	No error
1	The error couldn't be specified more precisely
2	Floating point domain error
3	Device temperature too high
4	General GUI error
5	Authentication required for operation
6	Authentication process failed
7	Operation is not allowed in service mode
8	Operation is allowed in service mode only
9	A measurement is currently in process
20	Operation not allowed while LD output is on
21	Wrong operating mode for this operation
22	INTERLOCK circuit is open
23	KEYLOCK is active
24	Operation not allowed because of a 'OPEN CIRCUIT' condition
26	Temperature protection is active

Error	Description
30	Operation not allowed while TEC output is on
31	Wrong operating mode for this Operation
32	Operation not allowed while a procedure is running
34	Operation not allowed because of a 'SENSOR FAILURE' condition
40	The device's setup is not valid
120	I ² C wires stuck
121	Illegal START/STOP condition
122	Slave address not acknowledged (Not a valid bus address?)
123	Incomplete write operation (Slave rejected to receive all data in the buffer)
124	Arbitration lost
130	EEPROM Timeout
131	Checksum error
140	General keyboard bootloader error
141	Keyboard bootloader write error
142	Keyboard bootloader read error
143	Keyboard bootloader verify error
144	Keyboard data link error
145	Keyboard command error
146	Keyboard nonvolatile memory error
150	General DTC bootloader error
151	DTC bootloader write error
152	DTC bootloader verify error
153	DTC communication error
154	unknown DTC error
155	DTC adjustment is invalid
-100	General command error
-101	Invalid character
-102	Syntax error
-103	Invalid separator
-104	Data type error
-105	GET not allowed
-108	Parameter not allowed
-109	Missing parameter
-110	Command header error

Error	Description
-111	Header separator error
-112	Program mnemonic too long
-113	Undefined header (Unknown command)
-114	Header suffix out of range
-115	Unexpected number of parameters
-120	Numeric data error
-151	Invalid string data
-200	General execution error
-210	General trigger error
-211	Trigger ignored
-212	ARM ignored
-213	Init ignored
-220	Parameter error
-221	Settings conflict
-222	Data out of range
-223	Too much data
-230	Data corrupt or stale
-240	Hardware error
-310	System error
-313	Calibration memory lost
-314	Save/recall memory lost
-315	Configuration memory lost
-321	Out of memory
-330	Self-test failed
-340	Calibration failed
-350	Queue overflow
-363	Input buffer overrun
-410	Query INTERRUPTED