QUANTIFI PHOTONICS®
A Teradyne Company

# PCA

## 1000 SERIES

Photocurrent Amplifier

**PXIE USER MANUAL**

quantifiphotonics.com

Copyright © 2026 Quantifi Photonics Ltd All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form, be it electronically, mechanically, or by any other means such as photocopying, recording or otherwise, without the prior written permission of Quantifi Photonics Ltd (Quantifi Photonics).

Information provided by Quantifi Photonics is believed to be accurate and reliable.
However, no responsibility is assumed by Quantifi Photonics for its use nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent rights of Quantifi Photonics.

The information contained in this publication is subject to change without notice.

**Trademarks**

Quantifi Photonics' trademarks have been identified as such. However, the presence or absence of such identification does not affect the legal status of any trademark. All third party product and company names are trademarks™ or registered® trademarks of their respective holders. Neither Quantifi Photonics nor any software programs or other goods or services offered by Quantifi Photonics are affiliated with, endorsed by, or sponsored by the third parties listed below.

LabVIEW™ is a trademark of National Instruments. MATLAB® is a trademark of The MathWorks Inc. Python® is a trademark of the Python Software Foundation. Microsoft Edge®, and Microsoft Windows® are trademarks of the Microsoft Corporation. Google Chrome™ is a trademark of Google LLC.

**Units of Measurement**

Units of measurement in this publication conform to SI standards and practices.

**CE mark related enquiries**

EU Authorized Representative
Certification Company
Veluwezoom 42
1327 AH ALMERE
The Netherlands
+31 (0)36 202 40 37
info@certification-company.com

**Manufacturer information**

Quantifi Photonics Limited
12-14 Parkway Drive
Rosedale, Auckland 0632
New Zealand

**User manual version: 1.04**

# Table of contents


1 What's in this user manual? ..... 6

2 Conventions ..... 7

3 Safety information ..... 8

3.1 Optical laser radiation precautions ..... 8

3.2 Electromagnetic compatibility ..... 8

3.3 Electrostatic discharge precautions ..... 9

4 Introducing the PCA 1000 Series ..... 10

4.1 Hardware description ..... 11

4.2 Connection diagram ..... 12

4.3 Analog Output Voltage ..... 13

4.4 Status LEDs ..... 14

4.5 Environmental conditions ..... 14

5 Setting up hardware ..... 15

5.1 Install the module in a PXIe chassis ..... 16

5.2 Uninstall the module from a PXIe chassis ..... 17

6 Installing software ..... 18

6.1 Install the Cohesion Installer software package ..... 19

6.2 Cohesion Manager ..... 21

6.3 Cohesion Firmware Updater ..... 22

7 CohesionUI - Overview ..... 23

7.1 Access a module with CohesionUI ..... 24

7.2 Display modules in a chassis ..... 25

7.3 Select a module to work with ..... 26

7.4 Set values ..... 27

7.5 SET values and ACTUAL values ..... 28

7.6 Manage CohesionUI settings ..... 29

7.7 Synchronize and reinitialize CohesionUI ..... 31
7.8 SCPI CohesionUI Command Console ..... 33
7.9 View system information ..... 36
7.9.1 PXIe Chassis ..... 36
7.9.2 Module ..... 36
**8 Controlling your PCA with CohesionUI ..... 37**
**9 Controlling your product with SCPI commands ..... 39**
9.1 Overview ..... 39
9.2 Programming conventions ..... 40
9.2.1 Index addressing of modules (slot, source) and units (channel) ..... 40
9.3 Status and event registers ..... 41
9.3.1 Standard Event Status Register ..... 41
9.3.2 Standard Event Status Enable Register (Mask) ..... 41
9.3.3 Status Byte Register ..... 41
9.3.4 Service Request Enable Register (Mask) ..... 42
9.3.5 Status and event registers diagram ..... 43
9.4 PXIe Multi Chassis mode operation ..... 44
9.4.1 NI-MAX application Multi Chassis mode ..... 44
9.4.2 SCPI Multi Chassis commands ..... 45
9.5 Command summary ..... 47
9.5.1 Common commands ..... 47
9.5.2 System commands ..... 47
9.5.3 Slot commands ..... 47
9.5.4 Configuration commands ..... 48
9.6 Command descriptions ..... 49
9.6.1 Common commands ..... 49
9.6.2 System commands ..... 51

9.6.3 Slot commands ..... 52
9.6.4 Configuration Commands ..... 54
9.7 Trace acquisition ..... 64
9.8 Triggering example ..... 65
9.8.1 System commands ..... 65
9.8.2 Module commands ..... 65
9.9 Programming example: Control of the PCA 1000 Series with SCPI ..... 66
**10 Programming applications ..... 67**
10.1 Setting up NI-MAX application ..... 68
10.2 Setting up NI-VISA application ..... 69
10.3 Python® code example ..... 70
10.4 MATLAB® code example ..... 71
**11 Working with optical fibers ..... 72**
**12 System requirements ..... 74**
**13 Maintenance ..... 75**
13.1 Annual calibration schedule ..... 75
**14 Technical Support ..... 76**
14.1 Contacting the Technical Support Group ..... 76
14.2 Transportation ..... 76
**15 Warranty Information ..... 77**
15.1 General information ..... 77
15.2 Liability ..... 77
15.3 Exclusions ..... 77
15.4 Certification ..... 78
15.5 Service and repairs ..... 78

# 1 What's in this user manual?

You can find the following information in this document:

| | |
|---|---|
| **Before you begin** | Conventions<br>Safety information<br>Working with optical fibers<br>System requirements |
| **Getting started** | Introducing the PCA 1000 Series<br>Setting up hardware<br>Installing software |
| **Working with your device** | **CohesionUI GUI:**<br>CohesionUI - Overview<br>Controlling your PCA with CohesionUI<br>**SCPI commands:**<br>Controlling your product with SCPI commands<br>Programming applications |
| **Maintenance** | Cohesion Manager<br>Cohesion Firmware Updater |

## 2 Conventions

Please make yourself familiar with these conventions; we use them throughout this user manual:

**WARNING**

Indicates a potentially hazardous situation which, if not avoided, could result in **death or serious injury**.

Do not proceed unless the required conditions are met and understood.

**CAUTION**

Indicates a potentially hazardous situation which, if not avoided, may result in **minor or moderate injury** or **component damage**.

Do not proceed unless the required conditions are met and understood.

**NOTE**

Indicates relevant information that requires your attention.

# 3 Safety information

Carefully read all safety information before using your Quantifi Photonics product.

## 3.1 Optical laser radiation precautions

**WARNING**

To protect yourself from harm caused by optical radiation:

- Do not install or terminate fibers while the light source is active.
- Turn the Quantifi Photonics product OFF before inspecting the end face(s) of the product, or any optical patch cords connected to it.
- Never look directly into a live fiber; ensure that your eyes are protected at all times.
- Equip all unused ports with protective caps or blanking covers.

**CAUTION**

The use of controls, adjustments, and procedures other than those specified in this document may result in exposure to hazardous situations involving optical radiation.

## 3.2 Electromagnetic compatibility

**CAUTION**

For electromagnetic compatibility, this product is a Class A product. It is intended for use in an industrial environment. There may be potential difficulties in ensuring electromagnetic compatibility in other environments, due to conducted as well as radiated disturbances.

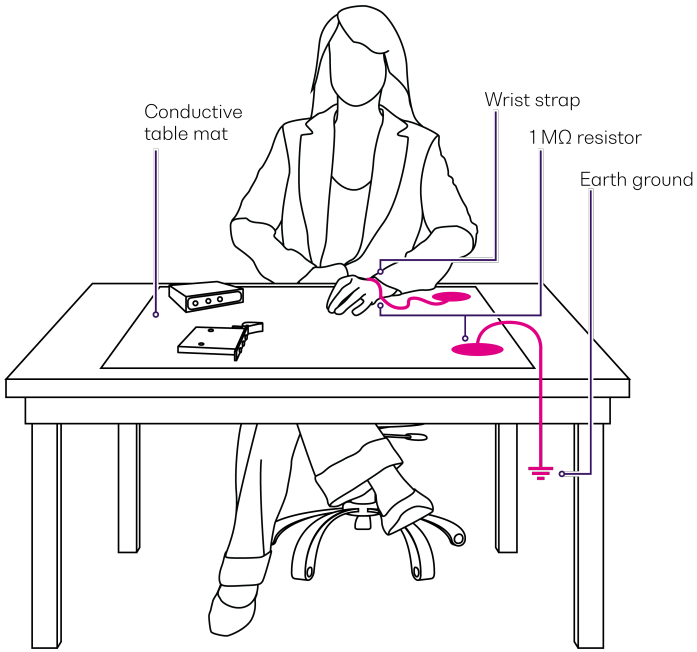
This symbol on the unit refers to documentation provided with the product for related safety information. Ensure that the required conditions are met and understood before using the product.

## 3.3 Electrostatic discharge precautions

**CAUTION**

The product is sensitive to electrostatic discharge (ESD). To ensure that you do not cause ESD damage to the product:

- Always follow proper grounding and ESD management practices.
- Store the unused product in the original protective electrostatic packaging that it was shipped in.
- Use a wrist strap and grounding table mat when unpacking or handling the product.
- Discharge RF cables before connecting them to the product, for example by shorting the center conductor to ground or by using 50Ω termination temporarily attached to the cable.



# 4 Introducing the PCA 1000 Series

The PCA-1000 Series is a compact and versatile photodiodeamplifier, designed to precisely measure photocurrent. It provides both analog and digital transimpedance amplifiers for photodiode currents ranging from 10 nA to 2 mA. Fully adjustable bias voltage control and adjustable frequency response provide flexibility. Onboard memory offers digital recording of up to 1024 current values. Its single-slot PXIe form factor enables convenient backplane triggering and it is compatible with a wide range of optical and electrical PXI test modules.

### Programming interfaces

Through its programming interfaces you can take advantage of the SCPI-compliant command language and choose from programming tools such as LabView, C++, Python, or any of the other popular programming languages used to control automatic test equipment (ATE).

### CohesionUI™

Quantifi Photonics' web-based graphical user interface CohesionUI is hosted on Microsoft Windows® and enables you to control your device from any supported web browser.

## 4.1 Hardware description

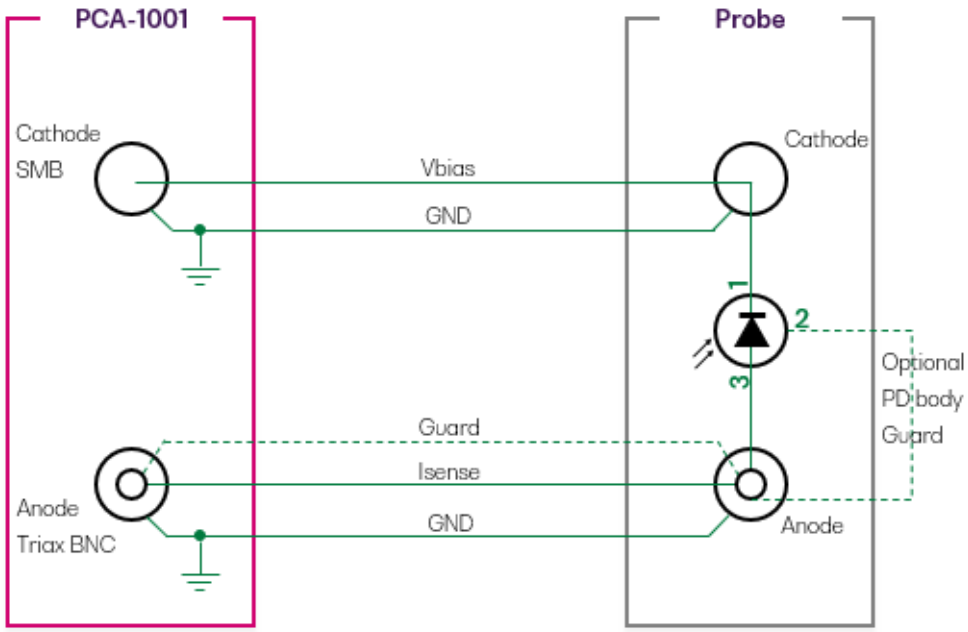
| | | | |
|---|---|---|---|
| 1a | Photodetector anode Channel 1 | 2a | Photodetector anode Channel 2 |
| 1b | Photodetector cathode Channel 1 | 2b | Photodetector cathode Channel 2 |
| 1c | Electrical analog output Channel 1 | 2c | Electrical analog output Channel 2 |
| 3 | Fastening screw | 5 | Fastening clip |
| 4 | Status LED | | |

## 4.2 Connection diagram

Refer to the following diagram when connecting your test setup with your PCA.

**NOTE**

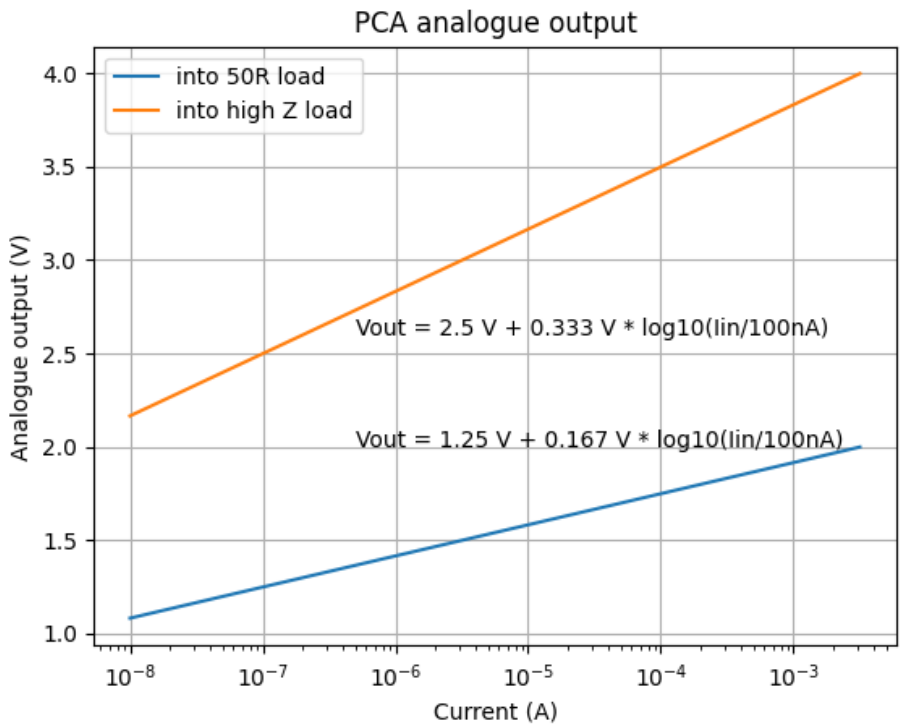
To avoid product damage, set **BIAS STATE** to **OFF** when connecting the photodector to your PCA. For details, refer to Controlling your PCA with CohesionUI or Configuration commands.



## 4.3 Analog Output Voltage

The PCA 1000 Series will output a voltage proportional to the logarithmic input current.

The maximum output voltage is 2.5 V, which will saturate at 2V @ 1mA optical input power. The typical analog output voltage response is shown below.




**NOTE**

The image above is shown for indicative purposes only. The actual response of the PCA may vary.

## 4.4 Status LEDs

The LED shows the status of the channel:

| LED | Meaning |
|---|---|
| OFF | The PCA is powered OFF |
| solid RED | Indicates that there is an error during the initialization of the PCA. |
| flashing RED | Indicates that there is an error, and the PCA is busy. |
| solid GREEN | Indicates that the PCA is operational. |
| flashing GREEN | Indicates that the product is busy processing commands such as dark-nulling or tracing. |

## 4.5 Environmental conditions

Range of environmental conditions:

| | |
|---|---|
| Indoor or outdoor use | Indoor only |
| Altitude | Max. 2000m |
| Operating temperature range | 5°C to 45°C \| 41°F to 113°F |
| Storage temperature range | -40°C to 70°C \| -40°F to 158°F |
| Relative humidity range | 20% to 80% |
| Supply voltage | 12VDC, 2A & 3.3VDC, 3A |

# 5 Setting up hardware

Quantifi Photonicsmodules are designed for easy installation in a PXIe-compatible chassis.

Make sure to follow these instructions when installing or removing a Quantifi Photonics module from a PXIe chassis.

Ensure that the chassis being used supports PXIe (or contains PXI-hybrid compatible slots). If you are unsure if your chassis is compatible with your Quantifi Photonics product, please contact Quantifi Photonics Customer Support.

**CAUTION**

The product is sensitive to electrostatic discharge (ESD). To prevent damage from ESD:

- Do not remove the product from the antistatic packaging until required to do so.
- Wear a grounded wrist strap at all times when handling the product.

**CAUTION**

Skin contact may leave corrosive residue and damage a connector:

- Always clean optical end faces before mating.

**NOTE**

Please check for the fiber end-face type of the optical ports, such as PC or APC, and only use the same type optical connector to avoid damaging the end-face.

For advice on connector and fiber care, please refer to Working with optical fibers.

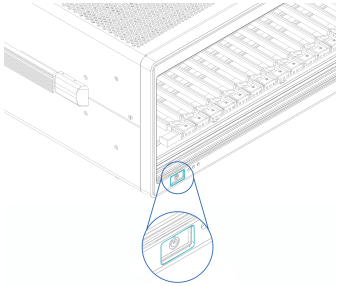
## 5.1 Install the module in a PXIe chassis

**WARNING**

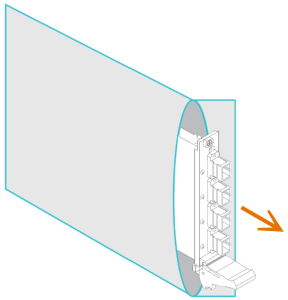
When attempting to install or remove a module or any component of the PXIe chassis:

- Power the chassis OFF.
- Follow these installation instructions.
- After powering the PXIe chassis ON, please wait at least 2 minutes before attempting to communicate with the module. This gives the chassis time to boot and initialize the communication server.

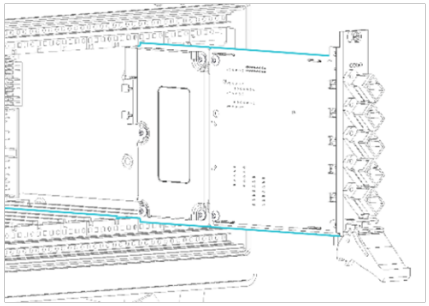
**1** Power the chassis OFF.



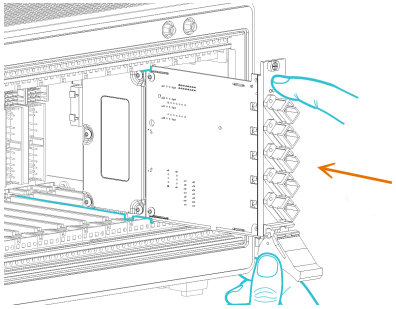
**2** Remove the module from the anti-static bag. Retain the bag.



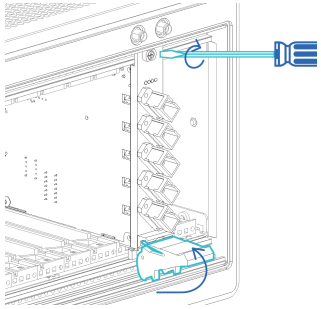
**3** Align the module with the slot guide rails.



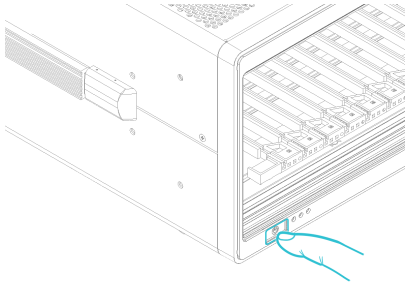
**4** Push module into slot until you feel resistance from the backplane connection.



**5** Engage the fastening clip. Secure all fastening screws.



**6** Power the chassis ON.



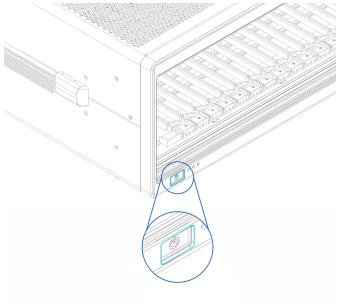
## 5.2 Uninstall the module from a PXIe chassis

**WARNING**

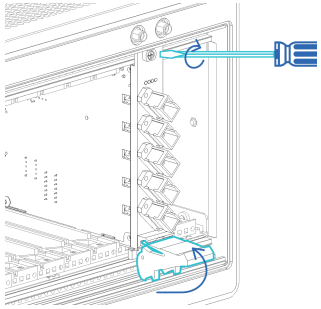
When attempting to install or remove a module or any component of the PXIe chassis:

- Power the chassis OFF.
- Follow these installation instructions.

**1** Power the chassis OFF.

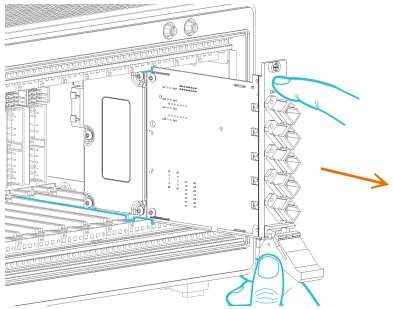


**2** Unsecure the fastening screws and fastening clip.

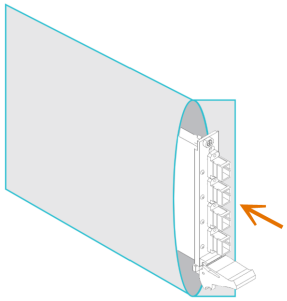


**3** Pull out the module.

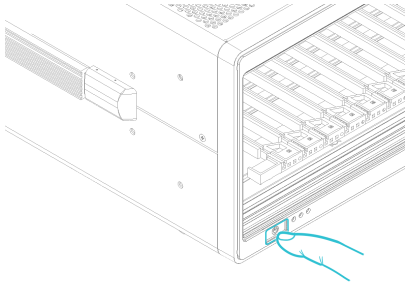
**Use the fastening clip to pull. Do NOT pull on the connectors**.



**4** Store the module in its antistatic bag.



**5** Power ON the chassis.



# 6 Installing software

The Cohesion Installer software package enables communication between the PXIe controller and Quantifi Photonics modules installed in a chassis.

The Cohesion Installer contains all required drivers and software:

| | |
|---|---|
| CohesionDriver | Driver Service for Quantifi Photonics PXIe modules |
| CohesionSCPI | VXI11 compliant server for remote SCPI communication |
| CohesionUI | Web-based Graphical User Interface |
| Cohesion Manager | Single-window utility application that shows the status of all Cohesion Software Services running on the system. |
| Cohesion Firmware Updater | Single-window utility application that shows the current firmware status of all Quantifi Photonics PXIe modules installed in the chassis. |

## 6.1 Install the Cohesion Installer software package

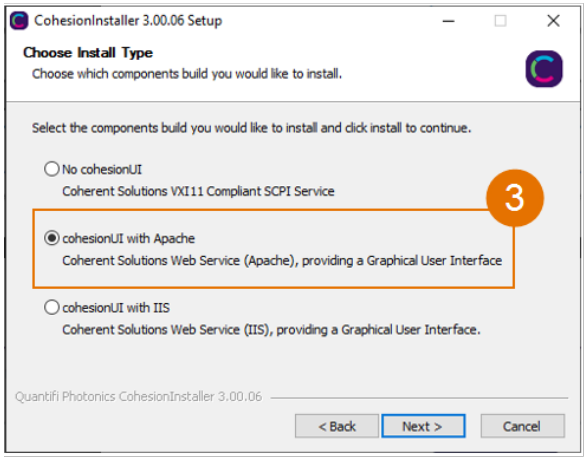
Install Cohesion Installer on:

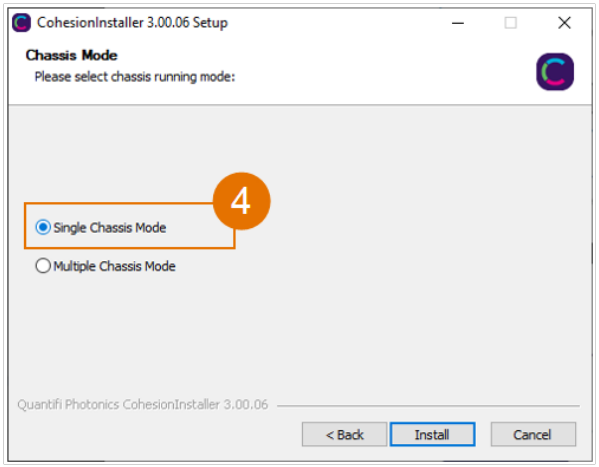
- the PXIe controller of the PXIe Chassis in which the Quantifi Photonics module(s) will be installed, or
- the controller PC (multi-chassis MXI setup)

▶ To install Cohesion Installer:

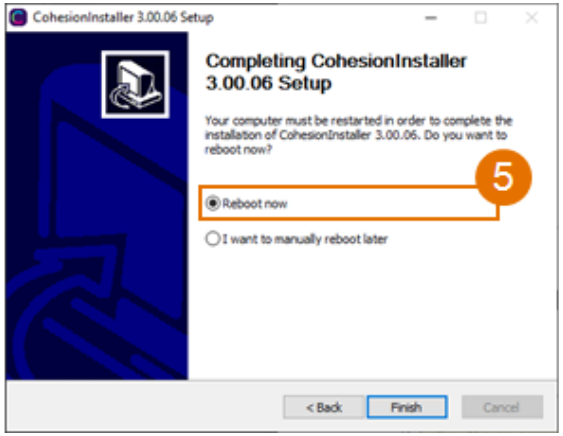
1. We recommended that you save your work and close open programs before installing Cohesion Installer.
2. Locate and run **CohesionInstaller-<version_number>.exe** from the provided USB media device (or download it from the Quantifi Photonics website) and follow the on-screen installation prompts.
3. Select the Installation Type: **CohesionUI with Apache** (this is the default setting)
4. Select the Chassis Mode: **Single Chassis Mode** (this is the default setting). If unsure, select this default setting.

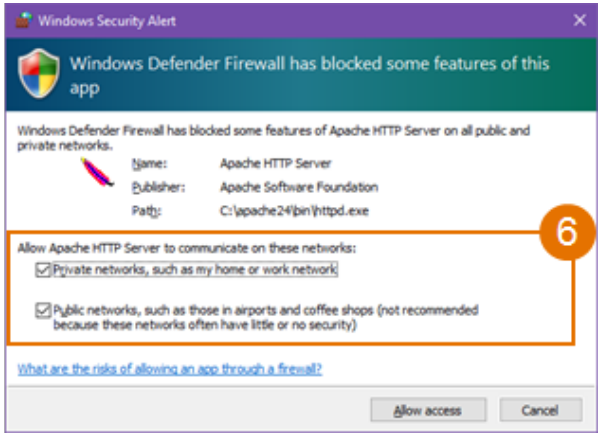
   To operate in Multiple Chassis Mode, additional hardware modules are required. As you can change the Chassis Mode later, we recommend to select **Single Mode** unless all other configuration requirements have been met.



5. At the end of the installation, we recommend you select the **Reboot now** option, and click **Finish** to complete the installation process.
6. A Windows Security Alert may prompt the user for network access. We recommend that **both options are ticked**, to allow any network configuration.




7. On startup after rebooting the system a User Account Control prompt might be displayed.
   Click **Yes** to allow running of the **Cohesion Firmware Updater Utility** and proceed with the application.

## 6.2 Cohesion Manager

Cohesion Manager is a single-window utitlity application that shows the status of all Cohesion Software Services running on the system.

By default, these Cohesion Software Services will start automatically on startup of Windows and need to be running to facilitate proper communication with the Quantifi Photonics PXIe modules.

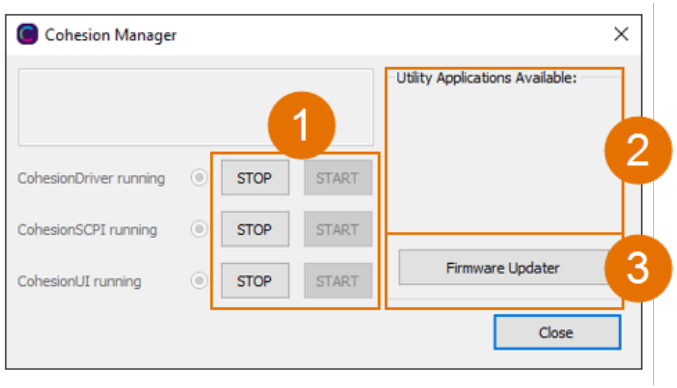
| | | |
|---|---|---|
| CohesionDriver | required | manages installed Quantifi Photonics modules |
| CohesionSCPI | required | VXI11 compliant SCPI interface for TCP communication with the installed Quantifi Photonics modules |
| CohesionUI | optional | web service providing a graphical interface for simplified operation of installed Quantifi Photonics modules |

▶ To open Cohesion Manager:

- Search for Cohesion Manager in the Windows Start Menu.

▶ From Cohesion Manager you can:

1. Start or stop the CohesionDriver service, CohesionSCPI service, or CohesionUI service independently.
2. View all installed Quantifi Photonics system utilities.
3. In this example you can open the Cohesion Firmware Updater application.




▶ If you can't detect or communicate with modules:

- Open Cohesion Manager.
- Check the status of software services, and start a service if required.

## 6.3 Cohesion Firmware Updater

Cohesion Firmware Updater launches automatically when you install a new version of Cohesion Installer on the system and reboot. Or, you can open it via the Cohesion Manager application.

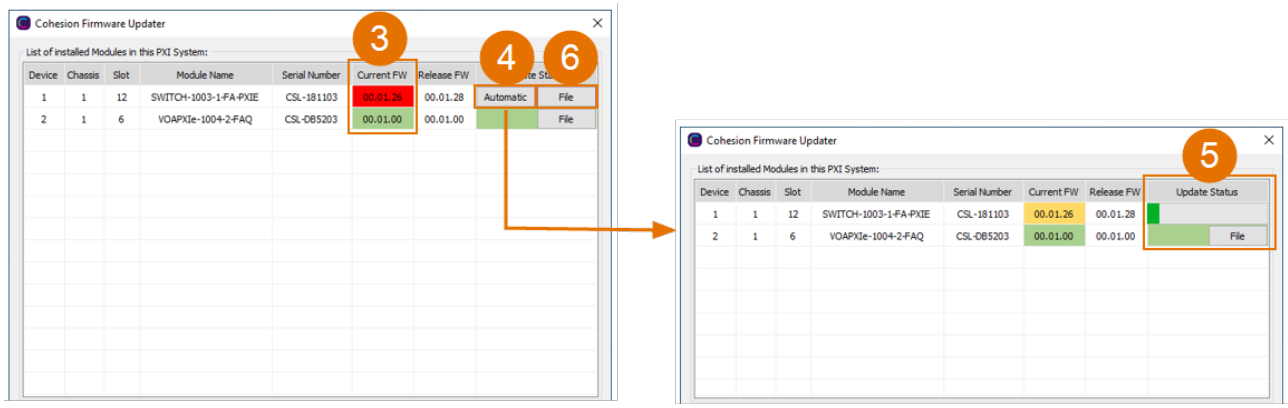
It is a single-window summary application that enables you to:

- view the current firmware status of all Quantifi Photonics PXIe modules installed in the chassis.
- update firmware to a new version if available.

> We recommended that you update firmware if a new version is available.

▶ To upgrade firmware:

1. Open **Cohesion Manager**, for example by searching for it in the Windows Start Menu.
2. In **Cohesion Manager**, click **Firmware Updater**.
3. Modules with out-of-date firmware are highlighted red.
4. Click **Automatic** to update automatically.
5. Progress will be displayed in **Update Status**.
6. Click **File** to update to a specific firmware package.

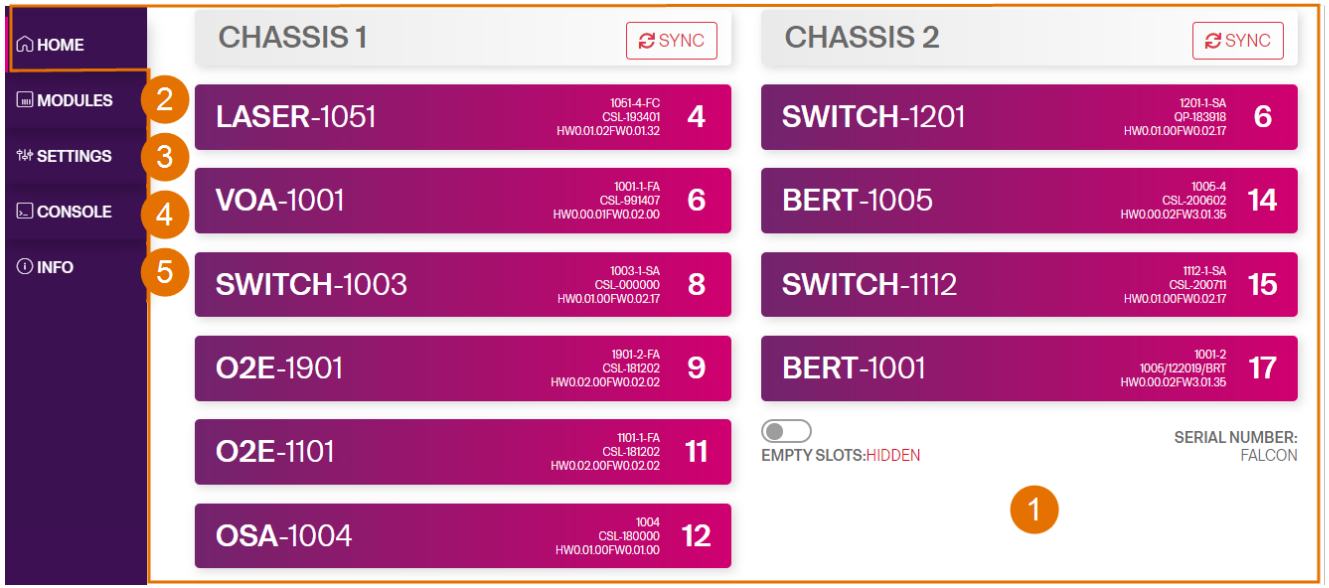


# 7 CohesionUI - Overview

CohesionUI is a web-based graphical interface that you can use to work with your Quantifi Photonics products.

> CohesionUI is part of the Cohesion Installer software package.

1. **HOME**: View all modules in the chassis
2. **MODULES**: Access a module
3. **SETTINGS**: Change CohesionUI settings
4. **CONSOLE**: Communicate with modules using SCPI commands
5. **INFO**: Display chassis information



## 7.1 Access a module with CohesionUI

You can access Quantifi Photonics modules via CohesionUI from the chassis controller, or from a controller PC.

To connect with a module, you need the IP address of the chassis the module is installed in.

▶ To obtain the IP address of the chassis:

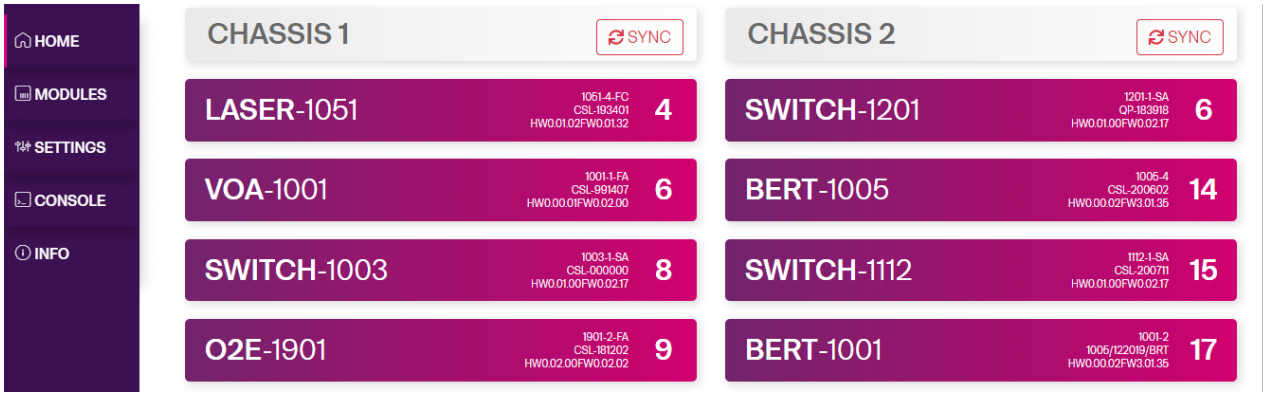
1. Open the **Command Prompt** window on the chassis controller.
2. Run the `ipconfig` command.
3. Note down the IPv4 address that is displayed.

▶ To connect with modules via CohesionUI:

1. On the controller or controller PC, open CohesionUI, for example by double-clicking the desktop icon, or open a supported browser.
2. Enter the IP address of the chassis as the URL.

   On the controller you can use `127.0.0.1` as the URL instead.

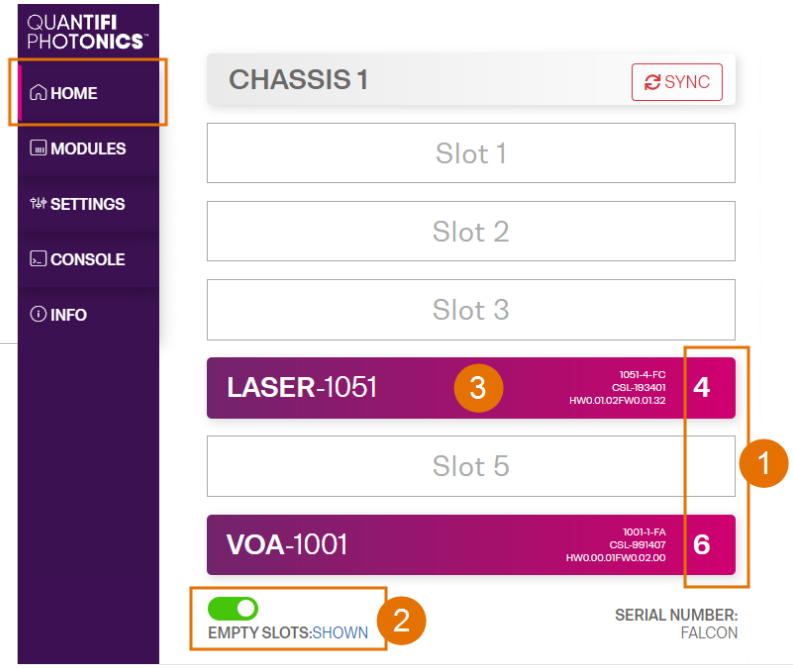
3. CohesionUI will launch in the browser, listing all available Quantifi Photonics modules installed in the chassis.



## 7.2 Display modules in a chassis

The **HOME** page is the main landing page in CohesionUI; it displays all available Quantifi Photonics modules in the PXIe chassis.

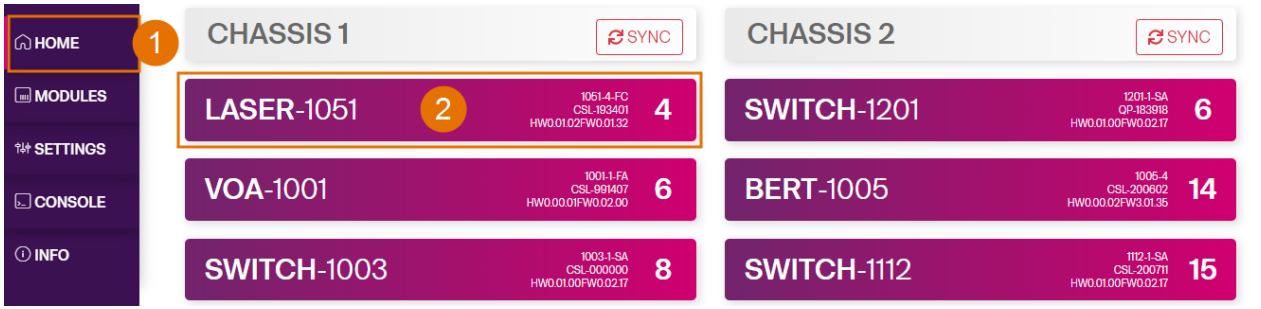
1. Numbers indicate the slots the modules are installed in.
2. You can hide (default setting) or show empty slots in the PXIe chassis by toggling the **EMPTY SLOTS** button.
3. You can select a module to work with by clicking it.



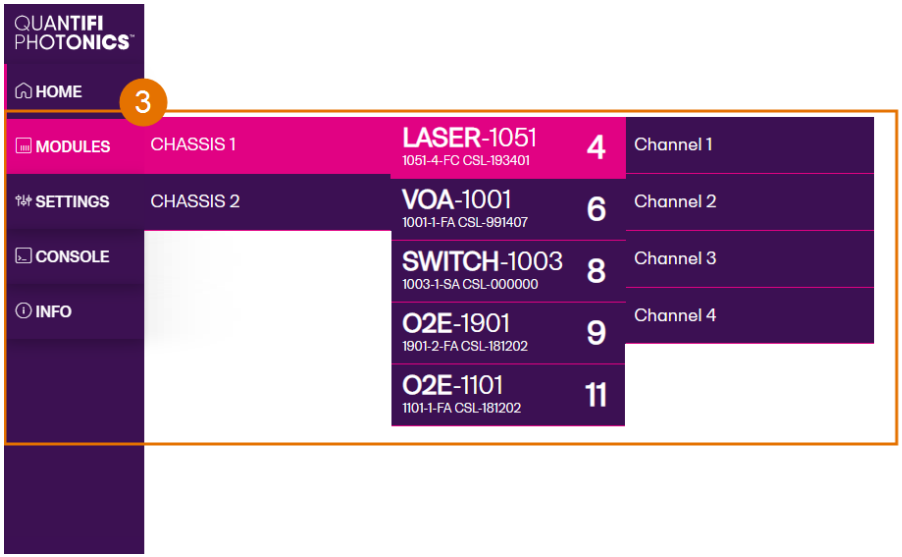
## 7.3 Select a module to work with

▶ To select a module:

1. Go to the **HOME** page.
2. Click the on the module.



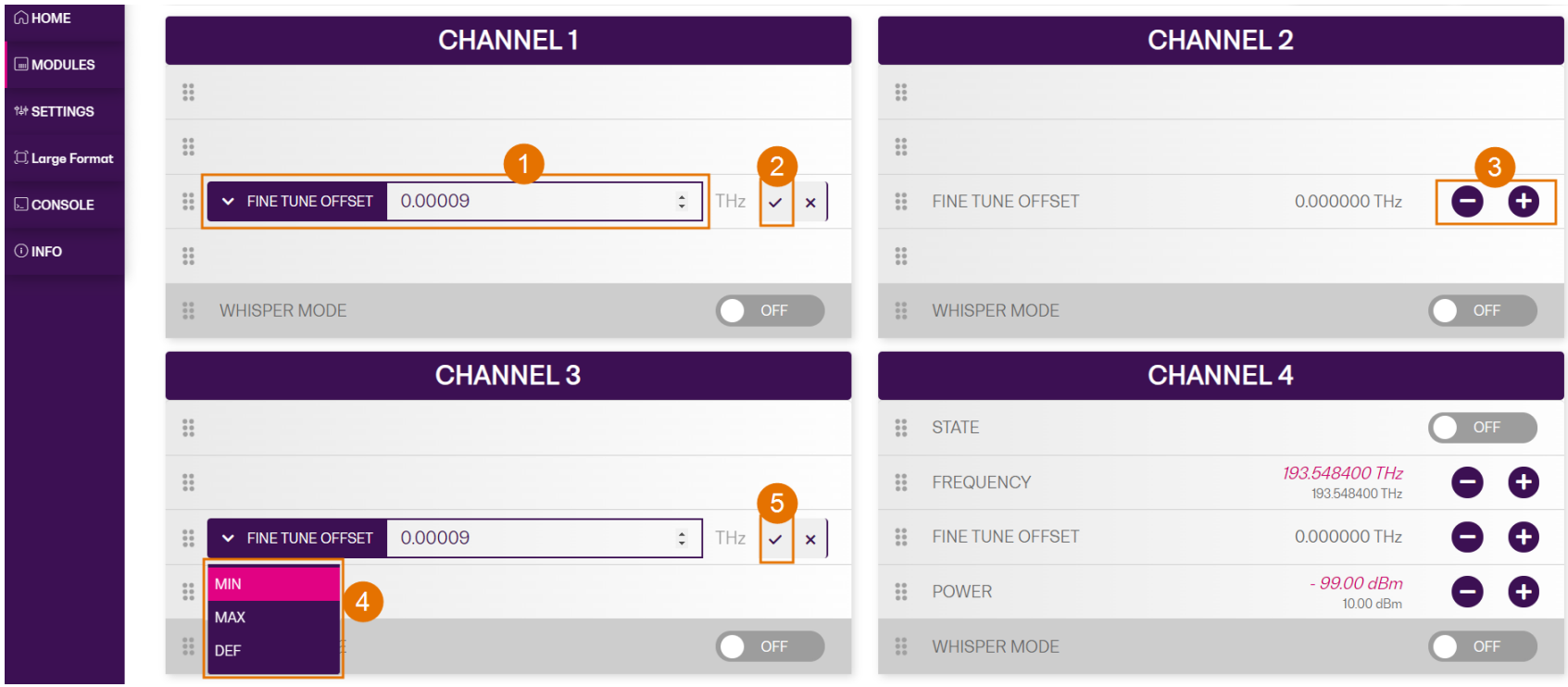

3. Or, hover over the **MODULES** menu and select a module or channel from the list.



## 7.4 Set values

In CohesionUI you can set values for parameters where applicable.

- To set a value:
  1. Click on a parameter and enter a value.
  2. Confirm the value.
  3. Alternatively, you can use + and - to increase or decrease the value. You can edit the step size in the **SETTINGS** menu.
- To set a pre-defined value, for example **MIN**, **MAX** or **DEF**:
  4. Click on a parameter and select a value from the dropdown menu.
  5. Confirm the value.

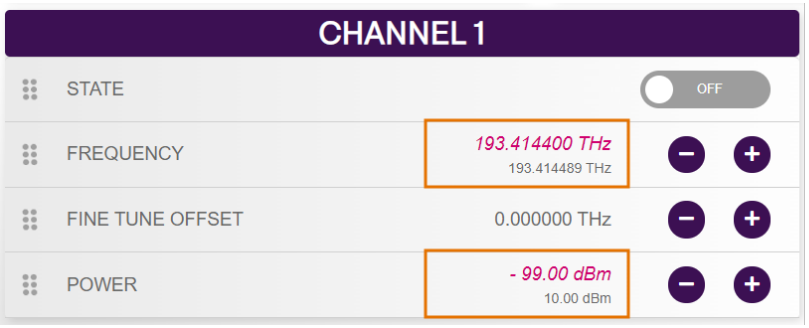



For details on how to change the step size, refer to Manage CohesionUI settings.

## 7.5 SET values and ACTUAL values

In some cases you can manually set a value that will be displayed alongside the actual value as follows:

- ***ACTUAL***: The actual value of the parameter as queried by the product.
- **SET**: The intended value of a given parameter as set by the user.



## 7.6 Manage CohesionUI settings

On the **SETTINGS** page you can configure CohesionUI settings and unit preferences.

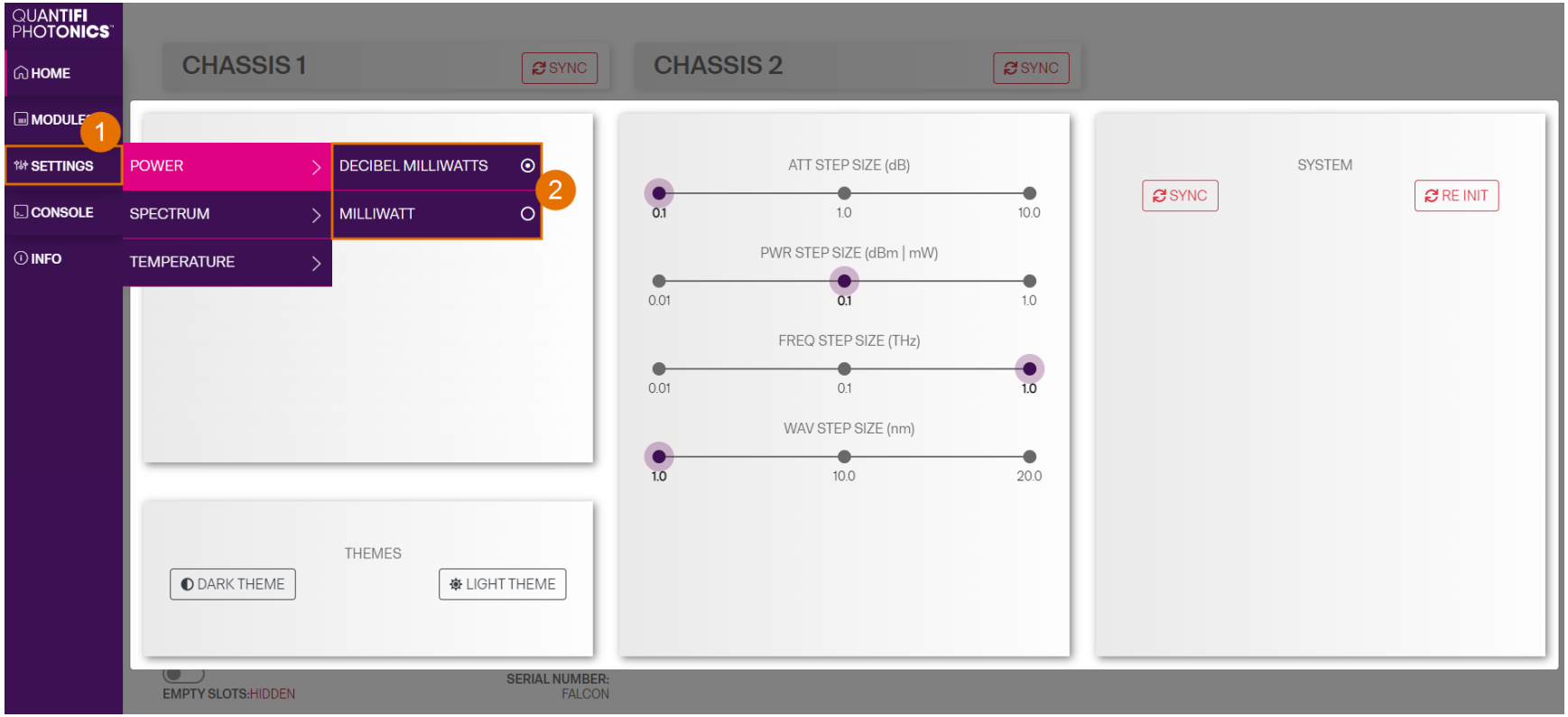
> **NOTE**
>
> CohesionUI reverts to default settings when power-cycling the chassis.

▶ To view all settings and unit preferences and adjust as required:

1. Click **SETTINGS**.
2. Change settings or unit preferences as required, for example temperature units.
   Please note that the units displayed on this page are not always relevant for each product.
3. **Step size** refers to the amount by which a value is increased or decreased when clicking the **+** or **-** button.

To adjust unit preferences one at a time:

1. Hover over **SETTINGS**.
2. Select a unit from the dropdown, for example the power unit.



## 7.7 Synchronize and reinitialize CohesionUI

You can update CohesionUI with the latest information from your Quantifi Photonics modules by synchronizing or reinitializing.

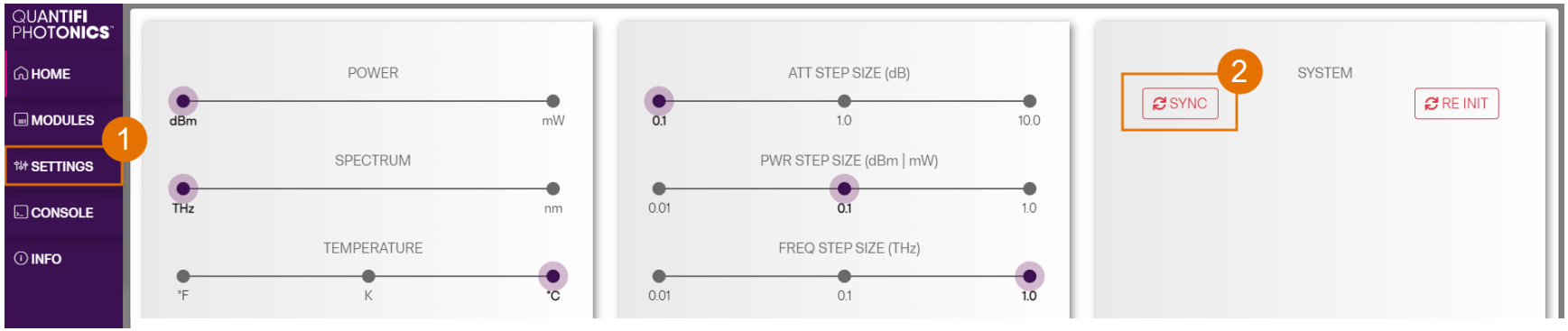
| | |
|---|---|
| Synchronizing | Updates **CohesionUI** with the latest information from the **CohesionSCPI service** |
| Reinitializing | Updates **CohesionUI and the CohesionSCPI service** with the latest information from the **CohesionDriver service** |

This can be particularly useful when operating a multi-chassis MXI setup and enables you to:

- Re-discover modules that CohesionUI does not display as expected.
- Discover modules that have been installed after the initial startup.

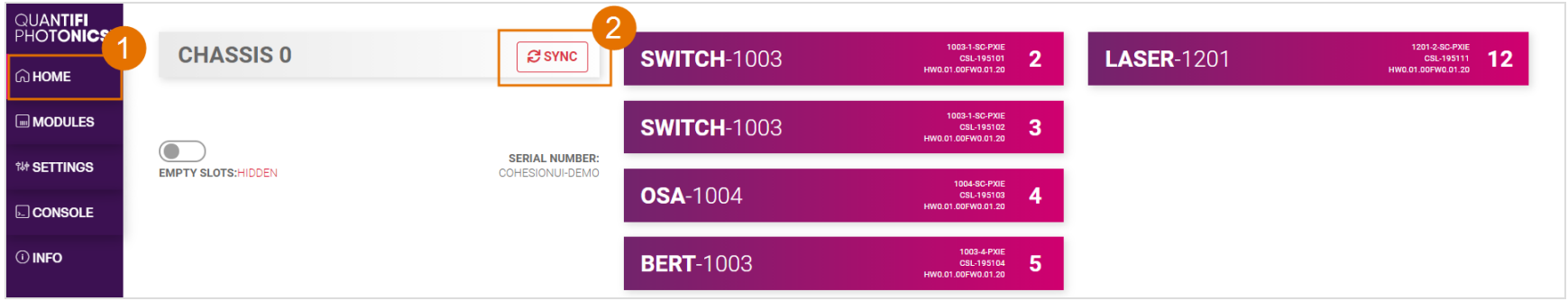
▶ To **synchronize** CohesionUI across **all modules in all chassis**:

1. Click **SETTINGS**.
2. Click **SYNC**.
3. The page will be disabled while synchronizing.



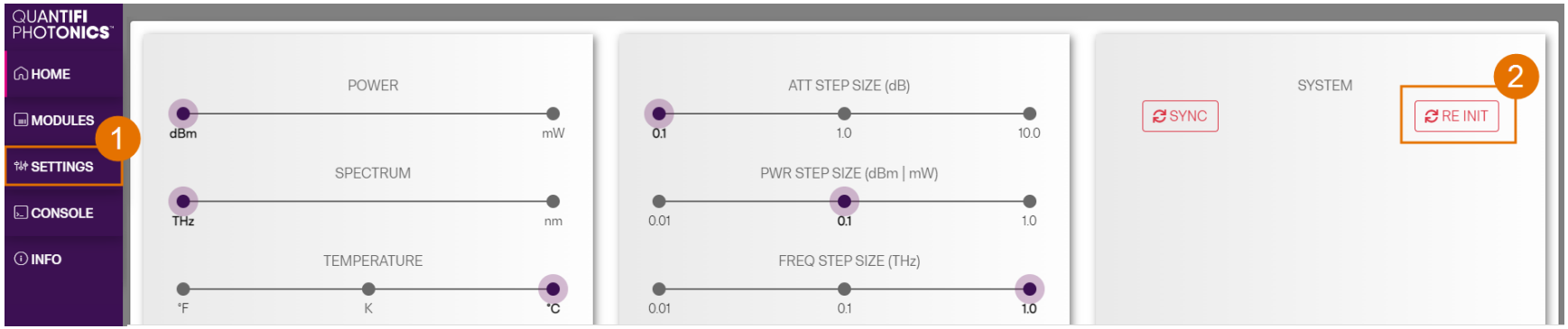
▶ To **synchronize** CohesionUI across **all modules in a selected chassis** only:

1. Click **HOME**.
2. Click **SYNC** for a selected chassis.
3. The page will be disabled while synchronizing.




▶ To **reinitialize** CohesionUI across **all modules in all chassis**:

1. Click **SETTINGS**.
2. Click **RE-INIT**.
3. All modules will be disabled and temporarily disconnected while reinitializing.



## 7.8 SCPI CohesionUI Command Console

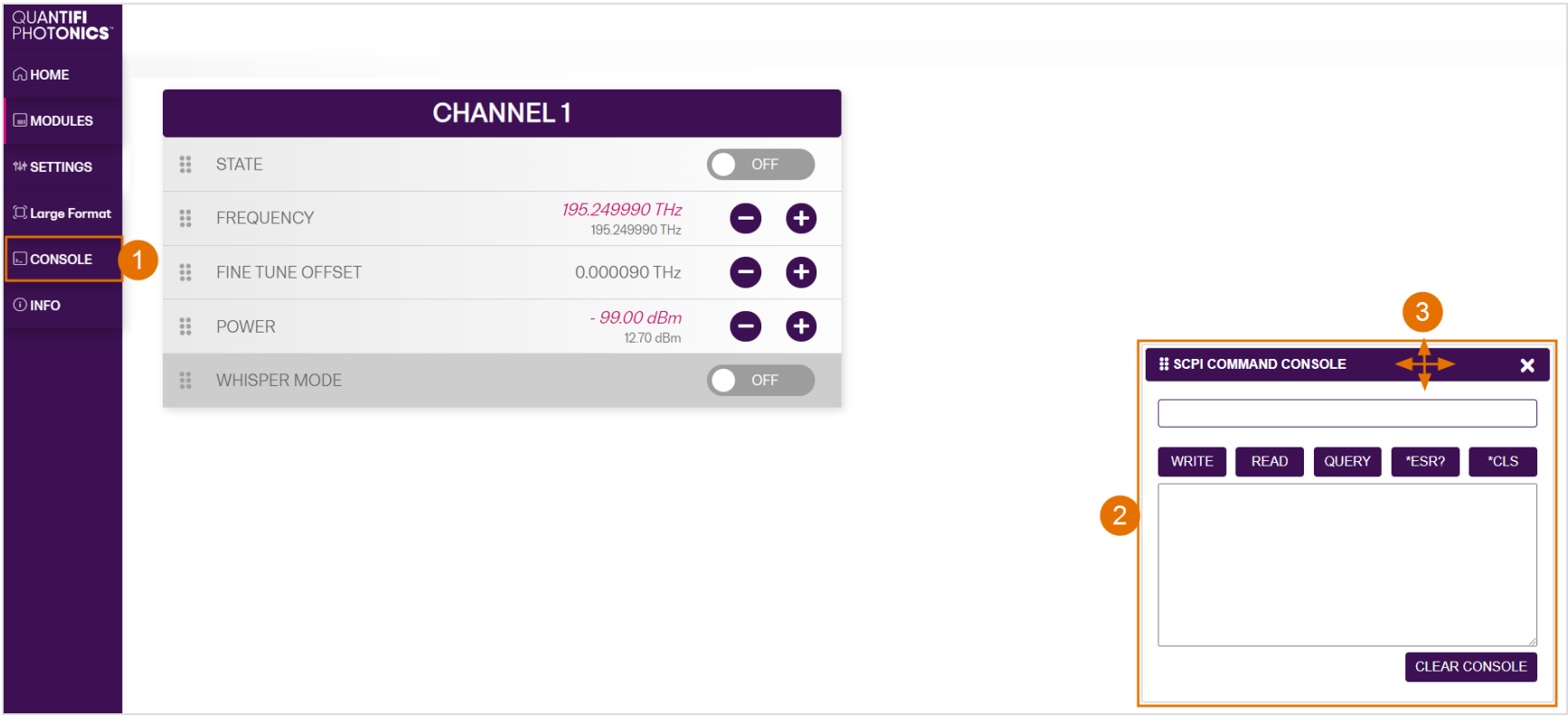
The CohesionUI SCPI Command Console enables you to communicate with Quantifi Photonics PXIE modules via SCPI commands. It enables you to test commands and verify their syntax.

For details on available SCPI commands, refer to the SCPI command section.

▶ To open the SCPI Command Console:

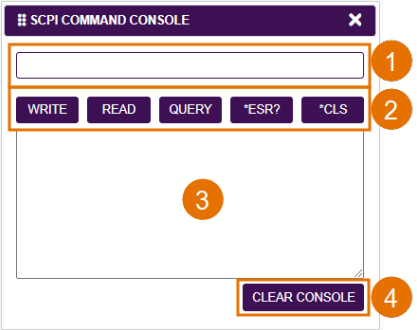
1. On the CohesionUI menu, click **CONSOLE**.
2. The console will appear in the bottom right corner of the screen.
3. You can move the console by clicking on the title bar and dragging it to any position on the screen. On closing and re-opening, the console will re-appear at its last position.

The console remains open when navigating between different modules. It floats on top of the UI so that you can observe the effect of SCPI commands on a module in real-time.



To communicate with a module via the SCPI Command Console:

1. Enter a command.
2. Select action(s).
3. Review the action response in the output area.
4. (optional) Clear the output area.




You can choose from these SCPI command actions:

| Action | Meaning | FAILED response |
|---|---|---|
| WRITE | Send the command to the instrument | The command is invalid. Please check the command and syntax. |
| READ | (after WRITE) Request the response from the instrument | Response buffer is empty. |
| QUERY | WRITE and READ | |
| *ESR? | Query the status event status register (ESR) – this will give you more details and specific information about command failures.<br><br>For details on error codes, please refer to the programming guide in this manual. | |
| *CLS | Clear the response buffer and start fresh – useful when getting out of sync with WRITE and READ actions | |

Example 1: Send instrument identification query `*idn?`

1. Enter the command: `*idn?`
2. Click **QUERY**.
3. The module returns the requested information.

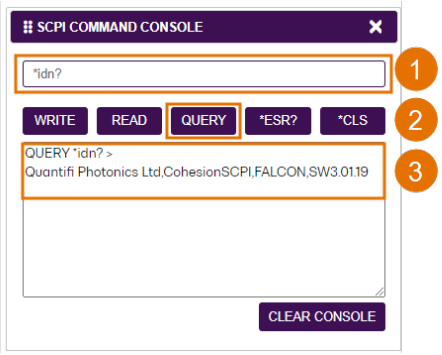
Example 2: What happens when I send an incorrect command?

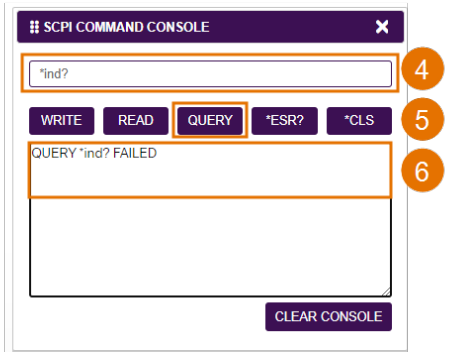
4. Enter an incorrect command, for example: ***ind?**
5. Click **QUERY**.
6. The module returns **FAILED**.

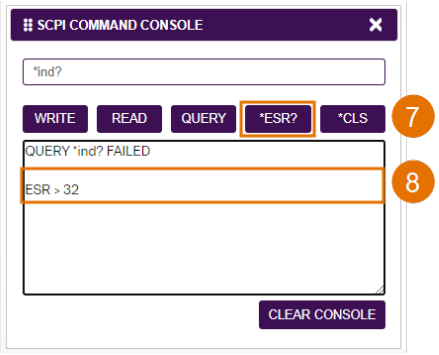
Example 3: Investigate a command failure:

7. Click `*ESR?` to query the event status register and request information about the command failure.
8. The instrument returns the error code, for example 32.

   For details on error codes, please refer to the *ESR? command in the programming guide.



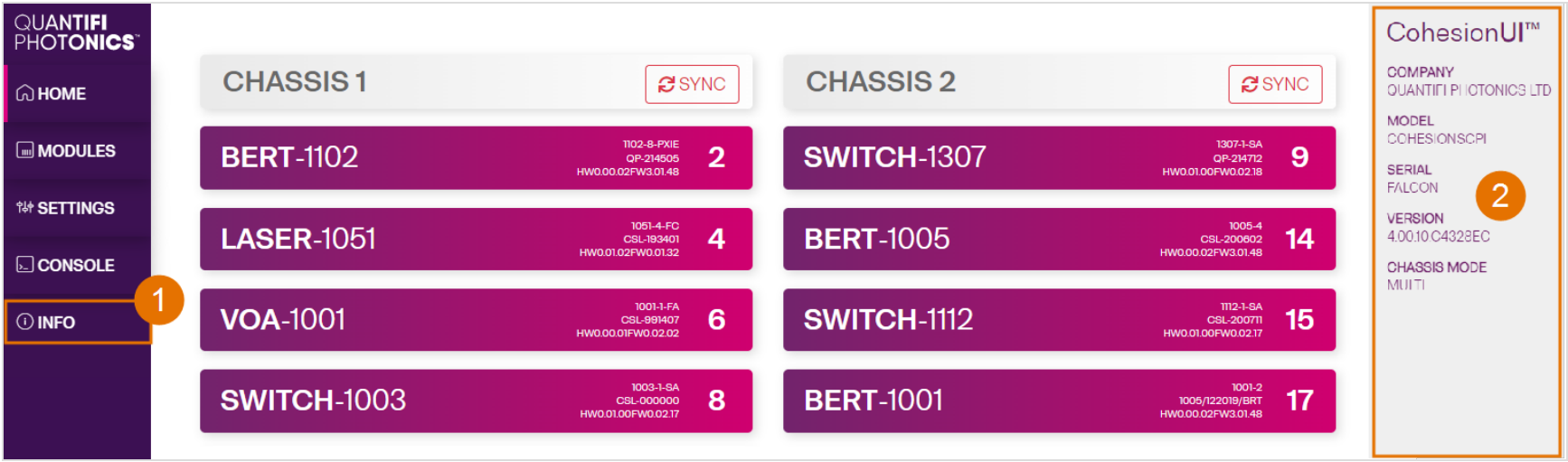



## 7.9 View system information

### 7.9.1 PXIe Chassis

▶ To display chassis information:

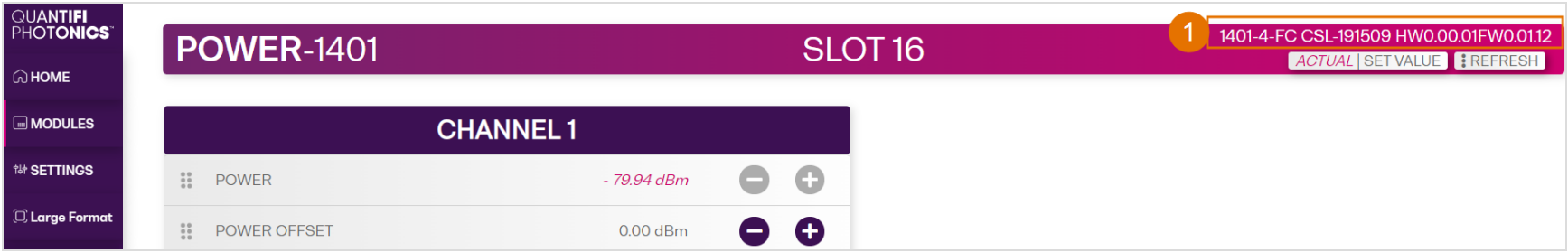
1. Click **INFO**.
2. The information panel will display operation mode, manufacturer, model, and serial number of the chassis, and the version of CohesionUI and CohesionSCPI service running on the chassis.




### 7.9.2 Module

▶ To view module information when working with a module in CohesionUI:

1. Model number, serial number and firmware versions are displayed in the top right corner.



# 8 Controlling your PCA with CohesionUI

You can use Quantifi Photonics' graphical user interface CohesionUI to work with your module. For details on how to get started with CohesionUI, refer to CohesionUI - Overview.

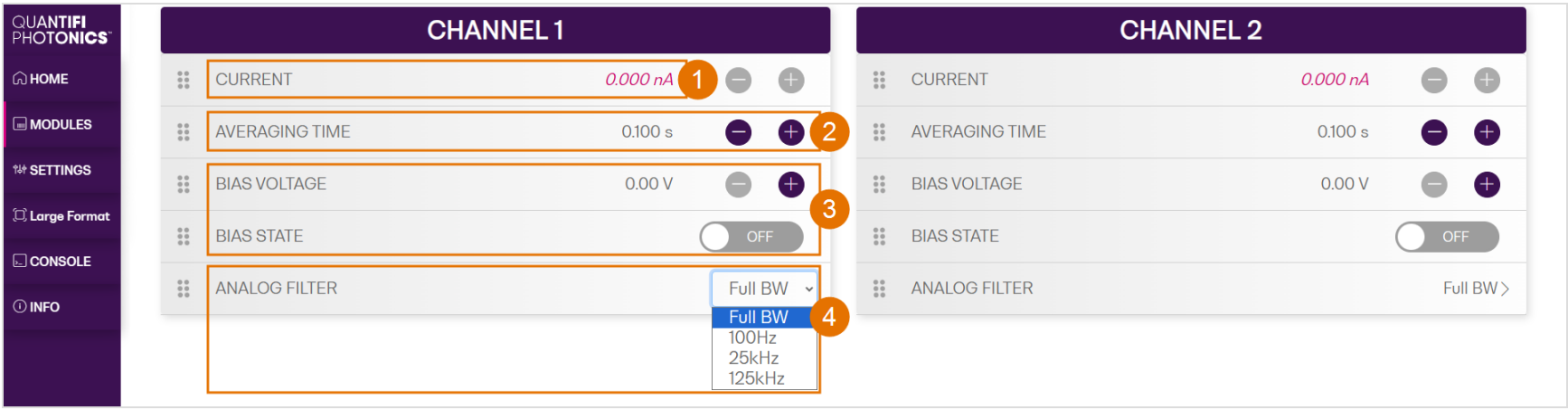
With a photodector connected to your PCA (refer to Connection diagram for details), you can work with CohesionUI to:

1. View the current photocurrent measurement value.
2. Set averaging time.
3. Select and apply bias voltage.
4. Set an analog filter if required for your test setup.

| Parameter | Edit values | Description |
|---|---|---|
| **CURRENT** | read only | The measured photocurrent value as read by the PCA. |
| **AVERAGING TIME** | ✓ | The amount of time the current is logged for before the PCA calculates the **CURRENT** value. |
| **BIAS VOLTAGE** | ✓ | Allows you to adjust reverse diode bias between 0 and 10 Volts to suit your photodiode/test setup. |
| **BIAS STATE** | ✓ | ON: Applies the selected BIAS VOLTAGE value.<br>OFF: BIAS VOLTAGE = 0V |
| **ANALOG FILTER** | ✓ | Allows you to adjust analog output bandwidth for use with your test setup. |

**NOTE**

To avoid product damage, set **BIAS STATE** to **OFF** when connecting a photodector to your PCA. Once connected, you can safely set **BIAS STATE** to **ON**.



**NOTE**

You can perform a trace by using SCPI commands. For details, refer to Configuration commands.

# 9 Controlling your product with SCPI commands

Remote communication with the CohesionSCPI service is achieved through the Standard Commands for Programmable Instruments (SCPI).

Support for VISA I/O API over TCP/IP is provided by the VXI-11 compliant CohesionSCPI service. With VISA communication drivers installed on the client, the implementation of VISA programming within environments such as MATLAB becomes available.

This section details the programming and measurement conventions to follow while executing the commands for the CohesionSCPI service.

**NOTE**

In NI-MAX a RIO interface will show up, however there are no communication methods available or implemented on this interface. Quantifi Photonics products are **ONLY** accessible through the **VISA TCPIP INSTR** interface provided by the CohesionSCPI service installed on the system.

## 9.1 Overview

You can operate your Quantifi Photonics module using SCPI commands.

You can:

- Query the current photocurrent measurement value.
- Set and query the averaging time and bias voltage.
- Define, run and query traces.

For details on available SCPI commands, refer to:

- Command summary
- Command descriptions

## 9.2 Programming conventions

This section details the programming and measurement conventions to follow while executing the commands for the CohesionSCPI service.

| Parameter | Default Unit | Alternative Units |
|---|---|---|
| Photocurrent | A | N/A |
| Time | S | N/A |
| Voltage | V | N/A |
| Rate | HZ | N/A |

| Argument | Data Format |
|---|---|
| `<wsp>` | Specifies whitespace character ($01_{16}$ – $09_{16}$, $0B_{16}$ – $20_{16}$) |
| `<value>` | Is numerical data, an integer, a decimal, exponential (10e-9 or 5.8e6) or string |
| `[VALUE1 \| VALUE2]` | A parameter choice. The '\|' separates the unique parameters available, only one of the choices can be used. In the example, either the input parameter `[VALUE1]` or `[VALUE2]` can be used, but not both. Some commands may have more than two choices available. This parameter can be omitted where the command has a default defined in the command description. |

### 9.2.1 Index addressing of modules (slot, source) and units (channel)

When executing commands, it is almost always necessary to provide the index of a specific module or an index of a specific installed unit.

For the commands that require index values:

| Index | Description | Value |
|---|---|---|
| `<c>` | the chassis index in which the specific blade module is installed | integer, inclusive of `0` |
| `<n>` | the slot (or source) index of the specific blade module | integer `<1` to `18>` |
| `<m>` | the channel index of a specific unit in the module | integer `<1`to `2>` |

**Message queues**

Information is exchanged in the form of messages. These messages are held in input and output queues.

The output queue stores responses to query commands. The CohesionSCPI service transmits any data in the output queue when a read request is received. Unless specified, all output response data is transmitted in ASCII format.

## 9.3 Status and event registers

### 9.3.1 Standard Event Status Register

The Standard Event Status Register (SESR) is modified by the Quantifi Photonics product with the results of the command operations.

| Bit | Description |
|---|---|
| 7 (MSB), 6 | Not used |
| 5 | Is set when a Command Error event has been detected |
| 4 | Is set when a command Execution Error has been detected |
| 3 | Is set when a Device Dependent Error event has been detected |
| 2 | Is set when there a Query Error event has been detected |
| 1 | Not used |
| 0 (LSB) | Is set when an Operation Complete event has been generated |

### 9.3.2 Standard Event Status Enable Register (Mask)

The Standard Event Status Enable Register (SESR Mask) is used to build the Event Status Bit (ESB) within the Status Byte Register (STB). To ignore any of the events detected and set in the SESR, set the corresponding bit within the SESR Mask to 0. The STB can then be queried and the value of the ESB can be used to determine service request requirements based on the SESR Mask applied.

**NOTE**

The 0 (LSB) value within the SESR Mask is 0.

### 9.3.3 Status Byte Register

The Status Byte Register (STB) is built from all other status registers and masks. This register can be used in queries to determine if an event has been detected and where that event has been detected.

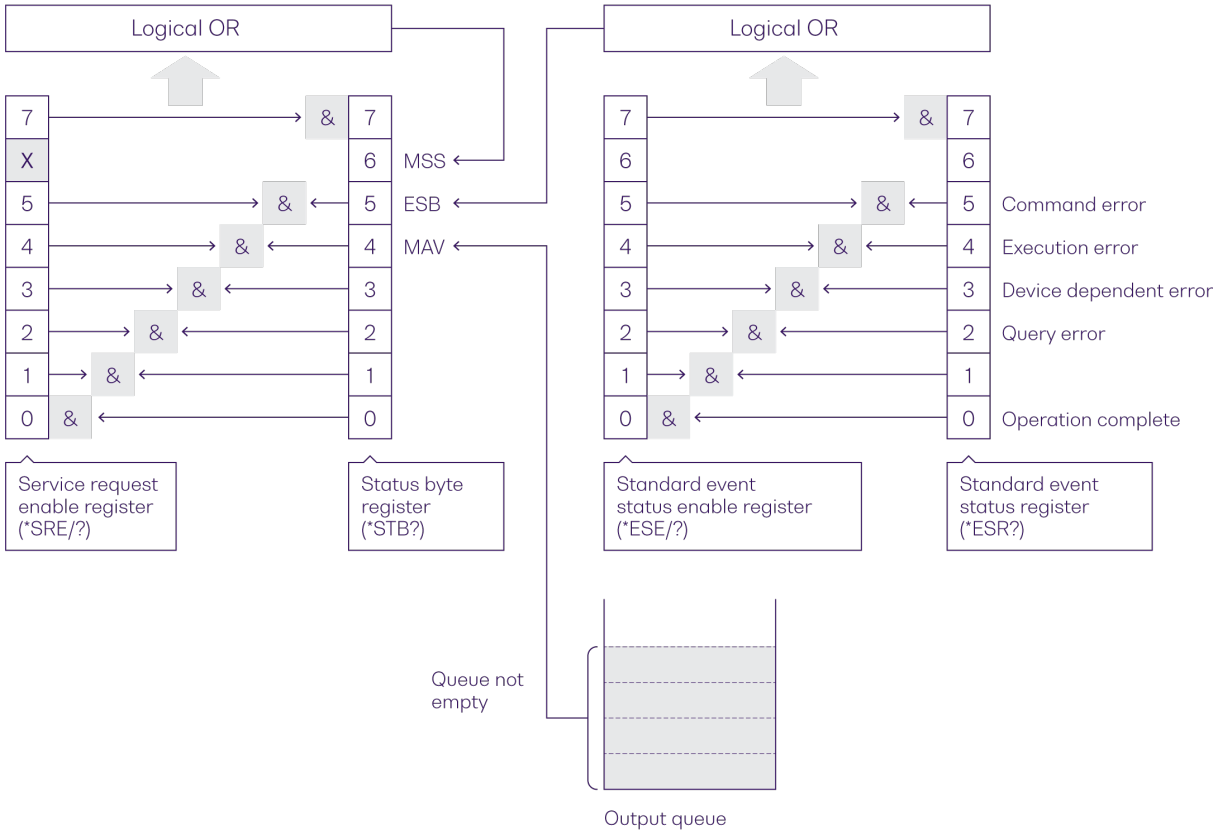
| Bit | Description |
|---|---|
| 7 (MSB) | Not used |
| 6 | The Master Summary Status (MSS) bit is set from the STB and SRE Mask |
| 5 | The Event Status Bit (ESB) is set from the SESR and the SESR Mask |
| 4 | Message Available (MAV) is set when there is data in the output queue |
| 3, 2, 1, 0 (LSB) | Not used |

### 9.3.4 Service Request Enable Register (Mask)

The Standard Request Enable Register (SRE Mask) is used to build the Master Summary Status Bit (MSS) within the Status Byte Register (STB). To ignore any of the events detected and set in the STB register itself, set the corresponding bit within the SRE Mask to 0. The STB can then be queried and the value of the MSS can be used to determine the type of service request required based on the SRE Mask applied.

| Bit | Description |
|---|---|
| 7 (MSB) | Not used |
| 6 | The Master Summary Status (MSS) bit is set from the STB and SRE Mask |
| 5 | The Event Status Bit (ESB) is set from the SESR and the SESR Mask |
| 4 | Message Available (MAV) is set when there is data in the output queue |
| 3, 2, 1, 0 (LSB) | Not used |

### 9.3.5 Status and event registers diagram



## 9.4 PXIe Multi Chassis mode operation

Multiple chassis can be connected to operate in **Multi Chassis Mode**.

To operate in Multi Chassis Mode, **CohesionSCPI service must be version 1.02.06 or later**.

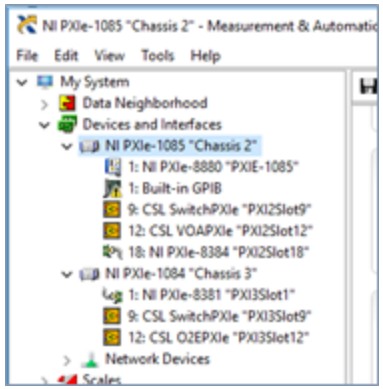
### 9.4.1 NI-MAX application Multi Chassis mode

**NOTE**

The CohesionSCPI service does not manage the chassis numbers. These are controlled by the NI Platform Services (and through NI-MAX).

Even if the CohesionSCPI service is in Multi Chassis mode, if a chassis is connected but has no installed modules, it will not show up when `*OPT?` is run.

In the example shown below, there are two chassis connected via the PXIe-8384 to PXIe-8381 connection. Chassis #2 has the controller running CohesionSCPI service, and Chassis #3 is the 'extended' chassis.



### 9.4.2 SCPI Multi Chassis commands

**NOTE**

Changing the CohesionSCPI service Chassis Mode will rediscover all Chassis and installed modules.

| Command | `:SYSTEM:CHASSIS?` |
|---|---|
| Syntax | `:SYSTEM:CHASSIS?<wsp>[LIST\|MODE]` |
| Description | Query the Chassis Mode configuration |
| Parameters | No parameters |
| Response | **`List`**: Returns a comma separated list of valid chassis index numbers discovered by the CohesionSCPI service. These are chassis that have modules installed. |
| | **`MODE`**: Returns the current Chassis Mode the CohesionSCPI service is operating in (SINGLE or MULTI). |
| | **`None`**: Returns the number of chassis managed by the CohesionSCPI service. If this is greater than 1, then the system should be set to MULTI mode for correct operation. |
| Example | In Single chassis mode:<br>`:SYSTEM:CHASSIS? -> 1`<br>`:SYSTEM:CHASSIS? LIST -> 0`<br>`:SYSTEM:CHASSIS? MODE -> SINGLE`<br><br>In Multi chassis mode:<br>`:SYSTEM:CHASSIS? -> 2`<br>`:SYSTEM:CHASSIS? LIST -> 2,3`<br>`:SYSTEM:CHASSIS? MODE -> MULTI` |

| Command | `:SYSTEM:CHASSIS` |
|---|---|
| Syntax | `:SYSTEM:CHASSIS<wsp>[SINGLE\|MULTI]` |
| Description | Set the Chassis Mode configuration |
| Parameters | **`SINGLE`**: Set CohesionSCPI service to operate in SINGLE Chassis Mode |
| | **`MULTI`**: Set CohesionSCPI service to operate in MULTI Chassis Mode |
| Response | No response |
| Example | `:SYSTEM:CHASSIS SINGLE` |

In Multi chassis mode, all commands listed in the command summary section will still work, but they must be prefixed with `:CHASSIS<c>`.

Common command example:

| | |
|---|---|
| Single Chassis Mode | `:SLOT2:IDN?` |
| Multi Chassis Mode | `:CHASSIS1:SLOT2:IDN?` |

Specific command example:

| | |
|---|---|
| Single Chassis Mode | `:SOUR2:CHAN2:POW? MAX` |
| Multi Chassis Mode | `:CHASSIS1:SOUR2:CHAN2:POW? MAX` |

## 9.5 Command summary

### 9.5.1 Common commands

| Command | Description |
|---|---|
| `*CLS` | Clear session message queues >> |
| `*IDN?` | Query the CohesionSCPI service identification >> |
| `*OPC?` | Query the Operation Complete Status >> |
| `*OPT?` | Query the modules managed by the CohesionSCPI service >> |
| `*ESR?` | Query the Standard Event Status Register >> |
| `*ESE?` | Query the Standard Event Status Enable Register (Mask) >> |
| `*ESE` | Set the Standard Event Status Enable Register >> |
| `*TST?` | Query the self-test status of all modules >> |
| `*RST` | Reset modules to default power-on settings >> |

### 9.5.2 System commands

| Command | Description |
|---|---|
| `:SYSTEM` | |
| `:PRESet` | Restore module settings of all modules to power-on default values >> |

### 9.5.3 Slot commands

| Slot commands | Description |
|---|---|
| `:SLOT<n>` | |
| `:OPC?` | Query the Operation Complete Status of the module >> |
| `:OPTions?` | Query the modules managed by the CohesionSCPI service >> |
| `:IDN?` | Query the slot identification >> |
| `:TeST?` | Query the module self-test status >> |
| `:ReSeT` | Reset the module to default power-on settings >> |
| `:PRESet` | Restore module settings to power-on default values >> |

### 9.5.4 Configuration commands

| Configuration commands | Description |
|---|---|
| :SENSe<n> | |
| :CHANnel<m> | |
| :CURRent? | Query the measured current >> |
| :AVERagingtime? | Query the measurement averaging time >> |
| :AVERagingtime | Set the measurement averaging time >> |
| :BIAS? | Query the bias voltage >> |
| :BIAS | Set the bias voltage >> |
| :STATE? | Query the bias state >> |
| :STATE | Set the bias state >> |
| :AnalogFILter? | Query the set analog filter >> |
| :AnalogFILter | Set an analog filter >> |
| :NULLing | Perform dark current nulling >> |
| :TIMEnulling | Query the amount of time remaining for dark current nulling to complete >> |
| :TRACE<m>? | Query the trace buffer >> |
| :TRACE<m> | |
| :PoinTS? | Query the number of trace buffer data elements >> |
| :PoinTS | Set the number of trace buffer data elements >> |
| :RATE? | Query the trace buffer sampling rate >> |
| :RATE | Set the trace buffer sampling rate >> |
| :TRIGger? | Set the trace trigger mode >> |
| :TRIGger | Set the trace trigger mode >> |
| :CoMPlete? | Query the trace completion status >> |

## 9.6 Command descriptions

### 9.6.1 Common commands

| Command | ***CLS** | Summary >> |
|---|---|---|
| Syntax | `*CLS` | |
| Description | Clear session message queues | |
| Parameters | N/A | |
| Response | N/A | |
| Example | `*CLS` | |

| Command | ***IDN?** | Summary >> |
|---|---|---|
| Syntax | `*IDN?` | |
| Description | Query the CohesionSCPI service identification | |
| Parameters | N/A | |
| Response | Comma separated string with the <manufacturer>,<server name>,<chassis controller name>,<server version> | |
| Example | `*IDN? -> Quantifi Photonics, CohesionSCPIService,PXIE-8133,FW2.0.15` | |

| Command | ***OPC?** | Summary >> |
|---|---|---|
| Syntax | `*OPC?` | |
| Description | Query the Operation Complete Status | |
| Parameters | | |
| Response | **1**: all modules installed in the chassis are ready to execute commands | |
| | **0**: modules installed in the chassis still have commands to execute in the input queue | |
| | NOTE: Any commands sent to the module when `:MODUle:OPC?` is NOT equal `1`, may not execute or return an error. | |
| Example | `*OPC? -> 1` | |

| Command | ***OPT?** | Summary >> |
|---|---|---|
| Syntax | `*OPT?` | |
| Description | Query the modules managed by the CohesionSCPI service | |
| Parameters | N/A | |
| Response | Comma separated string of the installed modules in the chassis | |
| Example | `*OPT? -> ,LASER-2001-1-FA-PXIE,SWITCH-1003-1-FC-PXIE,,VOA-1001-2-FA-PXIE,,,,O2E-1001-1-FC-PXIE,,,,,,,,` | |

| Command | **`*ESR?`** | | | Summary >> |
|---|---|---|---|---|
| Syntax | `*ESR?` | | | |
| Description | Query the Standard Event Status Register | | | |
| Parameters | N/A | | | |
| Response | Unsigned integer 8 bit value for the register <0 to 255>, as a string. | | | |
| | Bit | Description | Decimal Value | |
| | 7 (MSB) | Not used | 0 | |
| | 6 | Not used | 0 | |
| | 5 | Command error | 32 | |
| | 4 | Command Execution Error | 16 | |
| | 3 | Device Dependent Error | 8 | |
| | 2 | Not used | 0 | |
| | 1 | Not used | 0 | |
| | 0 (LSB) | Operation Complete | 1 | |
| Example | `*ESR? -> 8`<br>`*ESR? -> 32` | | | |

**NOTE**

It is recommended to use the `*ESR?` command query after every command that is sent to the device. The `*ESR?` query will be able to catch:

- **Device dependent Error** – the device is reporting an error in operation.
- **Execution Error** – SCPI was unable to execute the given command.
- **Command Error** – SCPI was unable to parse the given command, likely due to an incorrect command.

| Command | **`*ESE?`** | Summary >> |
|---|---|---|
| Syntax | `*ESE?` | |
| Description | Query the Standard Event Status Enable Register (Mask) | |
| Parameters | N/A | |
| Response | Unsigned integer 8 bit value for the register <0 to 255>, as a string. | |
| Example | `*ESE? -> 254` | |

| Command | **`*ESE`** | Summary >> |
|---|---|---|
| Syntax | `*ESE<wsp><value>` | |
| Description | Set the Standard Event Status Enable Register | |
| Parameters | N/A | |
| Response | N/A | |
| Example | `*ESE 254` | |

| Command | **`*TST?`** | Summary >> |
|---|---|---|
| Syntax | `*TST?` | |
| Description | Query the self-test status of all modules | |
| Parameters | N/A | |
| Response | **1**: error | |
| | **0**: no error | |
| Command | `*TST? -> 0` | |

| Command | **`*RST`** | Summary >> |
|---|---|---|
| Syntax | `*RST` | |
| Description | Reset modules to default power-on settings | |
| Parameters | N/A | |
| Response | N/A | |
| Example | `*RST` | |

### 9.6.2 System commands

| Command | **`:SYSTEM:PRESet`** | Summary >> |
|---|---|---|
| Syntax | `:SYSTEM:PRESet` | |
| Description | Restore module settings of all modules to power-on default values<br>Disables triggering by setting the trace trigger mode to `STOPped`.<br>Disables Trigger ARM by setting the Trigger ARMed state to `DISABLE`. | |
| Parameters | N/A | |
| Response | N/A | |
| Example | `:SYSTEM:PRES` | |

### 9.6.3 Slot commands

| Command | **:SLOT<n>:OPC?** | Summary >> |
|---|---|---|
| Syntax | :SLOT<n>:OPC? | |
| Description | Query the Operation Complete Status of the module | |
| Parameters | N/A | |
| Response | **1**: the module is ready to accept a new command | |
| | **0**: the module is busy performing a previous operation | |
| | NOTE: Any commands sent to the module when :SLOT:OPC? is NOT 1, may not execute or return an error. | |
| Description | :SLOT1:OPC? -> 1 | |

| Command | **:SLOT<n>:OPTions?** | Summary >> |
|---|---|---|
| Syntax | :SLOT<n>:OPTions? | |
| Description | Query the modules managed by the CohesionSCPI service | |
| Parameters | N/A | |
| Response | A comma separated array, or a single integer value based on the arguments given. If a module is not installed in a channel, it will not return any identification string. | |
| Example | :SLOT1:OPT? -> 1,1,, | |

| Command | **:SLOT<n>:IDN?** | Summary >> |
|---|---|---|
| Syntax | :SLOT<n>:IDN? | |
| Description | Query the slot identification | |
| Parameters | N/A | |
| Response | A comma-separated string containing "<manufacturer>,<part number>,<serial number>,<hardware version><firmware version>".<br><br>Note that the hardware and firmware versions are not comma separated. | |
| Example | :SLOT1:IDN? -><br>Quantifi Photonics, POWER-1500-2-FC-PXIE, QP-192001, HW1.0FW1.02 | |

| Command | **:SLOT<n>:TeST?** | Summary >> |
|---|---|---|
| Syntax | :SLOT<n>:TeST? | |
| Description | Query the module self-test status | |
| Parameters | N/A | |
| Response | Functional readiness status of the module. A non-zero response reports an error. | |
| Example | :SLOT1:TST? -> 0 | |

| Command | **:SLOT<n>:ReSeT** | Summary >> |
|---|---|---|
| Syntax | :SLOT<n>:ReSeT | |
| Description | Reset the module to default power-on settings | |
| Parameters | N/A | |
| Response | N/A | |
| Example | :SLOT1:RST | |

| Command | **:SLOT<n>:PRESet** | Summary >> |
|---|---|---|
| Syntax | :SLOT<n>:PRESet | |
| Description | Restore module settings to power-on default values<br><br>Disables triggering by setting the trace trigger mode to STOPped.<br><br>Disables Trigger ARM by setting the Trigger ARMed state to DISABLE. | |
| Parameters | N/A | |
| Response | N/A | |
| Example | :SLOT1:PRES | |

### 9.6.4 Configuration Commands

| Command | **:SENSe<n>:CHANnel<m>:CURRent?** Summary >> |
|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent?[<wsp><MIN\|MAX\|ACT\|ALL\|UNIT>] |
| Description | Query the measured current |
| Parameters | **MIN**: Returns the minimum value that can be measured in the default unit |
| | **MAX**: Returns the maximum value that can be measured in the default unit |
| | **ACT**: Returns the actual measured value in the default unit |
| | **ALL**: Returns all the above parameters in a comma separated string: MIN, MAX, ACT |
| | **UNIT**: Returns the default unit |
| | None: Returns the actual measured value in the default unit |
| Units | **Default unit**: A |
| | **Alternative units**: N/A |
| Response | A single value |
| Example | :SENS3:CHAN1:CURR? ACT -> 0.0000 |

| Command | **:SENSe<n>:CHANnel<m>:CURRent:AVERagingtime?** Summary >> |
|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent:AVERagingtime?[<wsp><MIN\|MAX\|DEF\|SET\|ALL\|UNIT>] |
| Description | Query the measurement averaging time |
| Parameters | **MIN**: Returns the minimum value in the default unit |
| | **MAX**: Returns the maximum value in the default unit |
| | **DEF**: Returns the default value |
| | **SET**: Returns the currently set value in the default unit |
| | **ALL**: Returns all the above values in a comma separated string: MIN, MAX, DEF, SET |
| | **UNIT**: Returns the default unit |
| | None: Returns the currently set value in the default unit |
| Units | **Default unit**: S |
| | **Alternative units**: N/A |
| Response | A single value, or a comma-separated array of values |
| Example | :SENS3:CHAN1:CURR:AVER? ALL -> 0.000000,10.000000,0.100000,5.210000<br><br>:SENS3:CHAN1:CURR:AVER? UNIT -> S |

| Command | **:SENSe<n>:CHANnel<m>:CURRent:AVERagingtime** Summary >> |
|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent:AVERagingtime<wsp><value\|MIN\|MAX\|DEF> |
| Description | Set the measurement averaging time |
| Parameters | **<value>**: Sets this value in the default unit. Specify non-default units if used.<br>To determine the valid range of values, query **MIN** and **MAX** values. |
| | **MIN**: Sets the minimum value |
| | **MAX**: Sets the maximum value |
| | **DEF**: Sets the default value |
| Units | **Default unit**: S |
| | **Alternative units**: N/A |
| Response | N/A |
| Example | :SENS3:CHAN1:CURR:AVER 5.21<br><br>:SENS3:CHAN1:CURR:AVER MIN |

| Command | **:SENSe<n>:CHANnel<m>:CURRent:BIAS?** Summary >> |
|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent:BIAS?[<wsp><MIN\|MAX\|DEF\|SET\|ALL\|UNIT>] |
| Description | Query the bias voltage |
| Parameters | **MIN**: Returns the minimum value in the default unit |
| | **MAX**: Returns the maximum value in the default unit |
| | **DEF**: Returns the default value in the default unit |
| | **SET**: Returns the set value in the default unit |
| | **ALL**: Returns all the above parameters in a comma separated string: MIN, MAX, DEF, SET |
| | **UNIT**: Returns the default unit |
| | None: Returns the set value in the default unit |
| Units | **Default unit**: V |
| | **Alternative units**: N/A |
| Response | A single value, or a comma-separated array of values |
| Example | :SENS3:CHAN1:CURR:BIAS? ALL -> 0.00,10.00,0.00,9.21<br><br>:SENS3:CHAN1:CURR:BIAS? UNIT -> V |

| Command | **:SENSe<n>:CHANnel<m>:CURRent:BIAS** Summary >> |
|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent:BIAS[<wsp><value\|MIN\|MAX\|DEF>] |
| Description | Set the bias voltage |
| Parameters | **<value>**: Sets this value in the default unit. Specify non-default units if used.<br>To determine the valid range of values, query **MIN** and **MAX** values. |
| | **MIN**: Sets the minimum value |
| | **MAX**: Sets the maximum value |
| | **DEF**: Sets the default value |
| | None: Sets the default value |
| Units | **Default unit**: V |
| | **Alternative units**: N/A |
| Response | N/A |
| Example | :SENS3:CHAN1:CURR:BIAS 9.21<br><br>:SENS3:CHAN1:CURR:BIAS MAX |

| Command | **:SENSe<n>:CHANnel<m>:CURRent:BIAS:STATe?** Summary >> |
|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent:BIAS:STATe?[<wsp><SET>] |
| Description | Query the bias state |
| Parameters | **SET**: Returns the set value |
| | None: Returns the set value |
| Units | **Default unit**: N/A |
| | **Alternative units**: N/A |
| Response | **ON**: Bias state is ON , the set bias voltage is applied |
| | **OFF**: Bias state is OFF, bias voltage = 0V |
| Example | :SENS3:CHAN1:CURR:BIAS:STATe? -> OFF |

| Command | **:SENSe<n>:CHANnel<m>:CURRent:BIAS:STATe** Summary >> |
|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent:BIAS:STATe<wsp><ON\|OFF> |
| Description | Set the bias state |
| Parameters | **ON**: Turns the bias state ON and applies the set bias voltage |
| | **OFF**: Turns the bias state OFF, bias voltage = 0V |
| Units | **Default unit**: N/A |
| | **Alternative units**: N/A |
| Response | N/A |
| Example | :SENS3:CHAN1:CURR:BIAS:STAT ON |

| Command | **:SENSe<n>:CHANnel<m>:CURRent:BIAS:AnalogFILter?** Summary >> |
|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent:AnalogFILter?[<wsp><DEF\|LIST\|ALL\|INFO>] |
| Description | Query the set analog filter |
| Parameters | **DEF**: Returns the default value |
| | **LIST**: Returns all supported values as numerals |
| | **ALL**: Returns all the above parameters in a comma separated string: <DEF>, None, <LIST> |
| | **INFO**: Returns all supported values, mapping numerals and text format |
| | None: Returns the set value |
| Units | **Default unit**: N/A |
| | **Alternative units**: N/A |
| Response | A single value, a comma-separated array of values, or the mapping between numerals and text format |
| Example | :SENS3:CHAN1:CURR:BIAS:AFIL? -> 3<br><br>:SENS3:CHAN1:CURR:BIAS:AFIL? LIST -> 0,1,2,3<br><br>:SENS3:CHAN1:CURR:BIAS:AFIL? INFO -><br>0:Full BW<br>1:100Hz<br>2:25kHz<br>3:125kHz |

| Command | **:SENSe<n>:CHANnel<m>:CURRent:BIAS:AnalogFILter** Summary >> |
|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent:AnalogFILter<wsp><0\|1\|2\|3\|DEF> |
| Description | Set an analog filter |
| Parameters | **0**: Sets the filter to Full BW |
| | **1**: Sets the filter to 100Hz |
| | **2**: Sets the filter to 25kHz |
| | **3**: Sets the filter to 125kHz |
| | **DEF**: Sets the default value |
| Units | **Default unit**: N/A |
| | **Alternative units**: N/A |
| Response | N/A |
| Example | :SENS3:CHAN1:CURR:BIAS:AFIL 1 |

| Command | **:SENSe<n>:CHANnel<m>:CURRent:NULLing** | Summary >> |
|---|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent:NULLing | |
| Description | Perform dark current nulling on the channel | |
| Parameters | N/A | |
| Response | N/A | |
| Example | :SENS3:CHAN2:CURR:NULL | |

| Command | **:SENSe<n>:CHANnel<m>:CURRent:TIMEnulling?** | Summary >> |
|---|---|---|
| Syntax | :SENSe<n>:CHANnel<m>:CURRent:TIMEnulling? | |
| Description | Query the amount of time remaining for dark current nulling to complete | |
| Parameters | N/A | |
| Response | Averaging time in seconds | |
| Example | :SENS3:CHAN2:CURR:TIME? -> 0.000000 | |

| Command | **:SENSe<n>:TRACE<m>?** | Summary >> |
|---|---|---|
| Syntax | :SENSe<n>:TRACE<m>? | |
| Description | Query the trace buffer<br><br>NOTE: Before querying this, wait for :SENS<n>:TRACE:CMP? to complete.<br><br>NOTE: This command will error out if no sample points have been triggered. | |
| Parameters | N/A | |
| Units | **Default unit**: A | |
| | **Alternative units**: N/A | |
| Response | A comma-separated array of values listing measured photocurrent values over time in the default unit. Each line contains the channel 1 value on the left, and the channel 2 value on the right. | |
| Example | :SENS3:TRACE1? -><br>1.009e-03,1.069e-14<br>1.008e-03,1.071e-14<br>1.008e-03,1.069e-14<br>...<br>1.008e-03,1.071e-14<br>1.008e-03,1.071e-14<br>1.009e-03,1.070e-14 | |

| Command | **:SENSe<n>:TRACE:PoinTS?** Summary >> |
|---|---|
| Syntax | :SENSe<n>:TRACE:PoinTS?[<wsp><MIN\|MAX\|DEF\|SET\|ALL>] |
| Description | Query the number of trace buffer data elements |
| Parameters | **MIN**: Returns the minimum value |
| | **MAX**: Returns the maximum value |
| | **DEF**: Returns the default value as set at power on |
| | **SET**: Returns the currently set value |
| | **ALL**: Returns all the above parameters in a comma separated string |
| | None: Returns the currently set value |
| Units | **Default unit**: N/A |
| | **Alternative units**: N/A |
| Response | A single value, or a comma-separated array of values |
| Example | :SENS3:TRACE:PTS? ALL -> 1,1024,1024,1024 |

| Command | **:SENSe<n>:TRACE:PoinTS** Summary >> |
|---|---|
| Syntax | :SENSe<n>:TRACE:PoinTS<wsp><value\|MIN\|MAX\|DEF> |
| Description | Set the number of trace buffer data elements |
| Parameters | **value**: Sets this value. To determine the valid range of values, query **MIN** and **MAX** values. |
| | **MIN**: Sets the minimum value |
| | **MAX**: Sets the maximum value |
| | **DEF**: Sets the default value |
| Units | **Default unit**: N/A |
| | **Alternative units**: N/A |
| Response | N/A |
| Example | :SENS3:TRACE:PTS 512<br><br>:SENS3:TRACE:PTS MAX |

| Command | **:SENSe<n>:TRACE:RATE?** Summary >> |
|---|---|
| Syntax | :SENSe<n>:TRACE:RATE?[<wsp><MIN\|MAX\|DEF\|SET\|ALL>] |
| Description | Query the trace buffer sampling rate in samples per second |
| Parameters | **MIN**: Returns the minimum value in the default unit |
| | **MAX**: Returns the maximum value in the default unit |
| | **DEF**: Returns the default value in the default unit |
| | **SET**: Returns the currently set value in the default unit |
| | **ALL**: Returns all the above parameters in a comma separated string |
| | None: Returns the currently set value in the default unit |
| Units | **Default unit**: HZ |
| | **Alternative units**: N/A |
| Response | A single value, or a comma-separated array of values |
| Example | :SENS3:TRACE:RATE? ALL -> 0.183,12000.000,12000.000,12000.000 |

| Command | **:SENSe<n>:TRACE:RATE** Summary >> |
|---|---|
| Syntax | :SENSe<n>:TRACE:RATE<wsp><value\|MIN\|MAX\|DEF> |
| Description | Set the trace buffer sampling rate in samples per second |
| Parameters | **value**: Sets this value in the default unit. Specify non-default units if used. To determine the valid range of values, query **MIN** and **MAX** values. |
| | **MIN**: Sets the minimum value in the default unit |
| | **MAX**: Sets the maximum value in the default unit |
| | **DEF**: Sets the default value in the default unit |
| Units | **Default unit**: HZ |
| | **Alternative units**: N/A |
| Response | N/A |
| Example | :SENS3:TRACE:RATE 100 |

| Command | **:SENSe<n>:TRACE:TRIGger?** Summary >> |
|---|---|
| Syntax | :SENSe<n>:TRACE:TRIGer?[<wsp><SET\|DEF\|INFO>] |
| Description | Query the trace trigger mode |
| Parameters | **SET**: Returns the currently set value |
| | **DEF**: Returns the default value |
| | **INFO**: Returns the mapping between numerals and text format of supported values |
| | None: Returns the default value |
| Units | **Default unit**: N/A |
| | **Alternative units**: N/A |
| Response | A single value, or the mapping between numerals and text format |
| Example | :SENS3:TRACE:TRIG? INFO -><br>0:STOPPED<br>1:IMMEDIATE<br>2:SWEXTERNAL<br>3:HWINTERNAL<br>4:HWEXTERNAL<br>5:HWCLK |

| Command | **:SENSe<n>:TRACE:TRIGger** Summary >> |
|---|---|
| Syntax | :SENSe<n>:TRACE:TRIGer<wsp><0\|STOPped\|1\|IMMEDIATE\|FORCE\|2\|SWEXTERNAL\|SWEXT\|3\|HWINTERNAL<br>\|HWINIT\|4\|HWEXTERNAL\|HWEXT\|5\|HWCLOCK\|HWCLK> |
| Description | Set the trace trigger mode |
| Parameters | **0\|STOPped**: Triggering is disabled |
| | **1\|IMMEDIATE\|FORCE**: Software trigger event, start sampling immediately |
| | **2\|SWEXTERNAL\|SWEXT**: Software trigger event, sampling aligned to PXIe clock |
| | **3\|HWINTERNAL\|HWINT**: PXIe line trigger event, sampling aligned to internal timer |
| | **4\|HWEXTERNAL\|HWEXT**: PXIe line trigger event, sampling aligned to PXIe clock |
| | **5\|HWCLOCK\|HWCLK**: PXIe line trigger event, sampling aligned to PXIe line edge |
| Units | **Default unit**: N/A |
| | **Alternative units**: N/A |
| Response | N/A |
| Example | :SENS3:TRACE:TRIG IMMEDIATE |

| Command | **`:SENSe<n>:TRACE:CoMPlete?`** Summary >> |
|---|---|
| Syntax | `:SENSe<n>:TRACE:CoMPlete?` |
| Description | Query the trace completion status |
| Parameters | N/A |
| Units | **Default unit**: N/A |
| | **Alternative units**: N/A |
| Response | **`1`**: the trace has been acquired and the buffer is ready to be read out |
| | **`0`**: the trace acquisition is still in progress, or the buffer is not ready to be read out |
| Example | `:SENS3:TRACE:CMP? -> 1` |

## 9.7 Trace acquisition

The PCA 1000 Series has the ability to acquire traces. The following is an example for the sampling to be started immediately upon command execution and aligned to the internal timer.

Trace acquisition can be realized with the following types of trigger events:

| Name | Trigger Event | Sampling Event | Description |
|---|---|---|---|
| `STOP` | None | None | triggering is disabled (or cancelled), no events will occur |
| `FORCED \| IMMEDIATE` | Software | Internal Clock | triggered upon command execution and sampling aligned to internal timer |
| `SWEXT \| SWEXTERNAL` | Software | PXIe Clock | triggered upon command execution and sampling aligned to PXIe Clock |
| `HWINT \| HWINTERNAL` | PXIe Line | Internal Clock | triggered upon PXIe Line edge and sampling aligned to internal timer |
| `HWEXT \| HWEXTERNAL` | PXIe Line | PXIe Clock | triggered upon PXIe Line edge and sampling aligned to PXIe Clock |
| `HWCLK` | PXIe Line | PXIe Line | triggered upon command execution and sampling aligned to PXIe Line edge |

```
#Set the module to the desired Trigger event mode
:SENS1:TRACE1:TRIGGER IMMEDIATE

#Monitor the Trace completion status
:SENS1:TRACE1:COMPLETE? -> 1

#Query (Read) the Trace buffer once populated (data elements depend on trace configuration)
:SENS1:TRACE1? -> “x1,x2,x3,….,xn”
```

## 9.8 Triggering example

This is an example of the commands and procedure for configuration and use of the system (or chassis) triggering features of the Quantifi Photonics products.

### 9.8.1 System commands

```
#Set the Triggering delay for the slot.
#Delay is from 0 to 10 seconds, with a resolution of 0.001 seconds.
#Default is 0 seconds.
:TRIGger<slot>:DELay 0

#Set the desired Triggering Line detection Mode to AND or OR.
#This represents triggering upon events on single, any or specific lines/signals.
:TRIGger<slot>:MODE OR

#Set the desired Triggering source(s).
#The sources represent the PXI trigger lines (integer values from 0 to 7) that the module can be configured for events.
:TRIGger<slot>:SOURce 0

#Arm the triggering in preparation for the next event.
:TRIGger<slot>:ARM ENABLE
```

### 9.8.2 Module commands

The following is the example of a PCA PXIe installed in Slot 3:

```
#Once the above system commands are configured, set the module to the desired Trigger event mode.
#The following is for the sampling to be started upon a PXIe trigger line event and samples aligned to the PXIe Clock.
:SENS3:TRACE1:TRIGGER SWEXT

#Monitor the Trace completion status.
:SENS3:TRACE1:COMPLETE? -> 1

#Query (Read) the Trace buffer once populated (data elements depend on trace configuration).
:SENS3:TRACE1? -> “x1,x2,x3,….,xn”
```

## 9.9 Programming example: Control of the PCA 1000 Series with SCPI

The following is a simple example of how to control the PCA 1000 Series using SCPI commands. See the previous section for specific details and extra parameters that the listed commands accept.

We recommend that you use the `*ESR?` query after every command that is sent to the device. This enables you to debug unreceived or incorrect commands sent to the product.

```
#Identifying the product
:*IDN?                                      #Query to confirm the correct instrument/PXIe is setup
:*OPT?                                      #Query the available instrument module configuration
:SLOT3:IDN?                                 #Query the identification information for a specific module

#Configuring the product
:SENSe3:CHANnel1:NULLing                    #Run dark current nulling
                                            #With no optical input connected and the dust cap still fastened
:SENSe3:CHANnel1:CURRent:TIMEnulling?       #Query the time remaining for nulling to be completed
:SENSe3:CHANnel1:CURRent:AVERagingtime 0.1 S  #Set the averaging time to 0.1 s

#Querying the product configuration values
:SENSe3:CHANnel1:CURRent:AVERagingtime?     #Query the averaging time

#Querying the optical power
:SENSe3:CHANnel1:CURRent?                   #Query the measured current with an optical input connected to the PCA

#Configuring and initiating the power tracing
:SENSe3:TRACE:PTS 1000                      #Set the number of trace points to 1000
:SENSe3:TRACE:RATE MAX                      #Set the trace sampling rate to the MAX value
:SENSe3:TRACE:TRIG FORCE                    #Set the trace trigger mode - this command will initiate a trace as soon as
                                            #the conditions for the set trigger mode are met

#Querying the acquired power trace
:SENSe3:TRACE:CMP?                          #Query the status of the trace completion
:SENSe3:TRACE1?                             #If the trace complete status is = 1, query the acquired trace on Channel 1
```

# 10 Programming applications

Remote communication with the CohesionSCPI service is achieved through the Standard Commands for Programmable Instruments (SCPI).

Support for VISA I/O API over TCP/IP is provided by the VXI-11 compliant CohesionSCPI service. With VISA communication drivers installed on the client, the implementation of VISA programming within environments such as MATLAB becomes available.

This section details the programming and measurement conventions to follow while executing the commands for the CohesionSCPI service.

**NOTE**

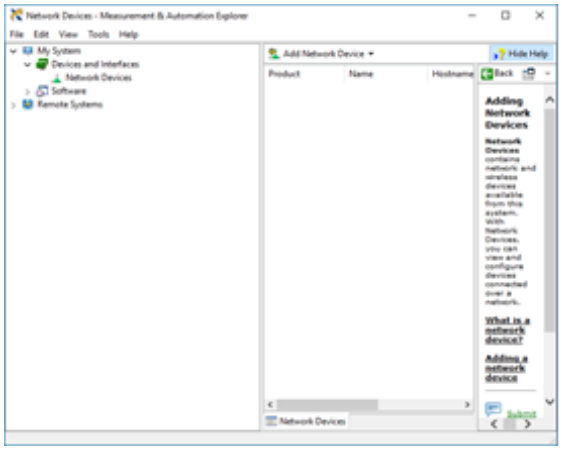
In NI-MAX a RIO interface will show up, however there are no communication methods available or implemented on this interface. Quantifi Photonics products are **ONLY** accessible through the **VISA TCPIP INSTR** interface provided by the CohesionSCPI service installed on the system.

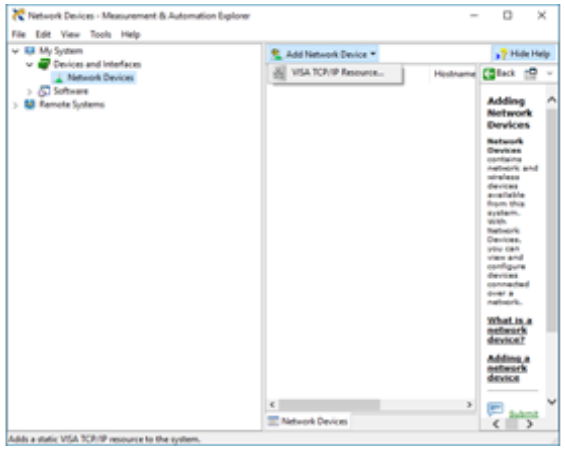
## 10.1 Setting up NI-MAX application

To communicate with any Quantifi Photonics product, the chassis / benchtop product must first be setup as a TCP/IP instrument.

1. After installing NI-MAX, launch the application. In the left side panel of the window, click the **Devices and Interfaces** option. A drop down of available instruments detected will show up.

2. Click on **Network Devices**, then click **Add Network Devices** and select **VISA TCP/IP Resource**.



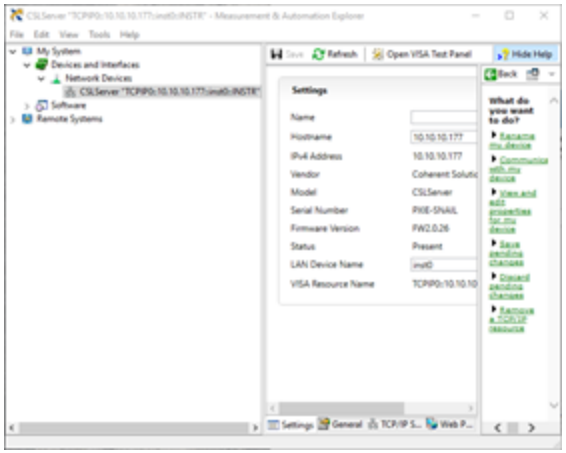




3. Select **Manual Entry of LAN Instrument**. Enter in the Hostname or IP Address.
   Note when operating locally, enter in the localhost IP address of **127.0.0.1**. Click **Finish** to end the setup process.

## 10.2 Setting up NI-VISA application

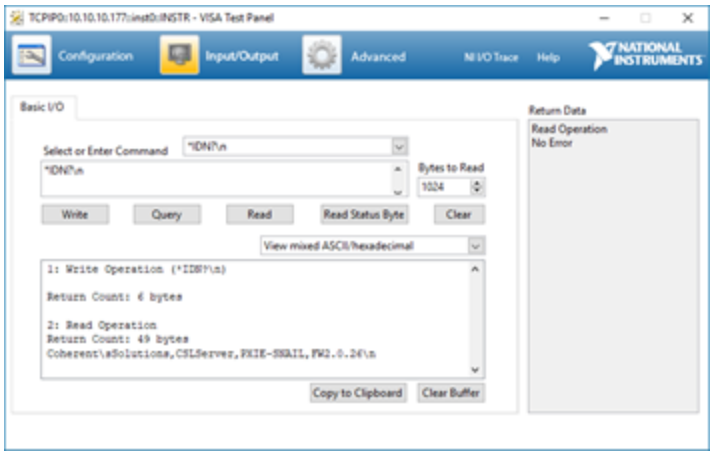
NI-VISA is used to communicate with the PXIe chassis or installed modules / instruments. The above steps must be completed before attempting to communicate using NI-VISA.

1. Launch NI-MAX. In the left-hand side menu, select an instrument from the **Network Devices** list.



2. On the right-hand side panel, select **Open VISA Test Panel**. A new window will popup. Click the **Input / Output** button from the window menu.

   Valid chassis and module commands can be entered in, and their returns queried



## 10.3 Python® code example

The following example shows how to communicate with the Quantifi Photonics product using Python code.

```python
# You can get VXI11 from pip:
# pip install python-vxi11==0.9
import vxi11
from vxi11.vxi11 import Vxi11Exception
# replace this with the IP of your device
ip = "127.0.0.1"
try:
    print("connecting to " + ip + " ... ")
    instrument = vxi11.Instrument(ip)
    print("connected")
    print("checking IDN...")
    command = "*IDN?"
    data = instrument.ask(command)
    print("IDN: " + data)
    print("checking OPT...")
    command = "*OPT?"
    data = instrument.ask(command)
    print("OPT: " + data)
    # replace this with a valid command for your device (read # the programming guide section for examples)
    command = ""
    print("writing a specific command")
    instrument.write(command)
    print("checking ESR")
    command = "*ESR?"
    data = instrument.ask(command)
    print("*ESR?: " + data)
except Vxi11Exception as e:
    # pass
    print("ERROR" + str(e) + ", command: " + str(command))
```

## 10.4 MATLAB® code example

To communicate with the Quantifi Photonics product in MATLAB® the installation of a VISA IO driver is required. These drivers enable the creation of the Interface Object for instrument communication.

If developing locally on the PXIE Platform, then these will already be installed. However, if development is on a remotely connected system the VISA Libraries, e.g. National Instruments NI-VISA will have to be installed.

**NOTE**

MATLAB 2010x or later with the Instrument Control Toolbox is required to execute the code detailed in this section.

The following example shows how to communicate with a Quantifi Photonics product using MATLAB code.

```
% Find a VISA-TCPIP object. This is if the VISA object has already been
% created with tmtool or has been removed from the workspace without
% first being closed (cleanly disconnected).
PXIE_Chassis = instrfind('Type', 'visa-tcpip', ...
'RsrcName', 'TCPIP0::10.10.10.89::inst0::INSTR', 'Tag', '');
% Create the ‘agilent’ VISA-TCPIP object if it does not exist
% otherwise use the object that was found.
if isempty(PXIE_Chassis)
    PXIE_Chassis = visa('agilent', 'TCPIP0::10.10.10.89::inst0::INSTR');
else
    fclose(PXIE_Chassis);
    PXIE_Chassis = PXIE_Chassis (1);
end
% Open the connection to the VISA object.
fopen(PXIE_Chassis);
% Query the PXIE_Chassis.
response = query(PXIE_Chassis, '*IDN?');
disp('The *IDN query response:');
disp(response);
response = query(PXIE_Chassis, '*OPT?');
disp('The *OPT query response:');
disp(response);
% Replace this with a valid command for your device (read the programming
% guide section for examples)
command = ''
% Close the connection to the object.
```

# 11 Working with optical fibers

Quantifi Photonics products are equipped with high quality optical connectors in compliance with EIA-455-21A standards.

**CAUTION**

Keep connectors clean and in good condition to ensure maximum power and to avoid erroneous readings. Quantifi Photonics is not responsible for damage or errors caused by bad fiber cleaning or handling.

- Always inspect fiber end faces for cleanliness using a fiber inspection probe before inserting them into a port..
- If required, clean fibers and faces as detailed below.

**NOTE**

- To avoid damaging ferrules or fiber faces due to mismatched connectors, always check ports and connector type information before inserting a connector. All Quantifi Photonics units are labeled with connector type information.
- Failing to align and/or connect fiber-optic cables properly will result in significant signal loss and reflection.

▶ When connecting a fiber-optic cable to a port:

1. Visually inspect the fiber end face using a fiber inspection microscope.
2. If a **connector end face** is dirty:
    - Wipe the connector end face using a reel-type cleaner and inspect again.
    - For stubborn hard to clean connectors:
        - Use lint-free fiber-cleaning wipes soaked in a fiber optic cleaning solution.
        - Wipe the connector on the soaked part.
        - Dry the connector by wiping on the dry part of the wipe, or by using a reel-type cleaner.
    - Repeat the process until connector inspection shows a clean fiber face.
3. If a **bulkhead inner connector face** is dirty:
    - Use a pen-type dry cleaner, align the cleaning tip with the port and push the cleaner until you hear the characteristic click. Inspect again.
    - For stubborn hard to clean bulkhead connectors:
        - Use a stick-type cleaner dipped in a fiber optic cleaning solution.
        - Carefully align and insert the stick into the connector and gently rotate the stick for several seconds applying light pressure.
        - Use a pen-type cleaner to dry the connector.
    - Repeat the process until connector inspection shows a clean fiber face.
4. If the fiber end face is clean:
    - Carefully align the connector and port to prevent the fiber end from touching the outside of the port or other surfaces. If the connector features a key, mate it correctly into the corresponding notch of the port bulkhead.

- Push the connector in so that the fiber-optic cable is firmly in place with adequate contact. If your connector features a screw sleeve, tighten the connector to firmly maintain the fiber in place. Do not over-tighten, as this will dam-age the fiber and the port bulkhead.

# 12 System requirements

### Quantifi Photonics PXIe modules

| | |
|---|---|
| Supported browsers for working with CohesionUI | Google Chrome™<br>Microsoft Edge® |
| Chassis | PXIe-compatible chassis that<br>• supports PXIe, or<br>• contains PXI hybrid compatible slots |
| Recommended PXIe controller operating system | Microsoft Windows® 10 (64-bit) |

### Quantifi Photonics MATRIQ / EPIQ instruments

| | |
|---|---|
| Supported browsers for working with CohesionUI | Google Chrome™<br>Microsoft Edge® |
| Recommended client computer operating system | Microsoft Windows® 10 (64-bit) |

# 13 Maintenance

To help ensure long, trouble-free operation:

- Always inspect fiber-optic connectors before using them and clean them if necessary.
- Keep the unit free of dust.
- Store the unit at room temperature in a clean and dry area. Keep the unit out of direct sunlight.
- Avoid high humidity or significant temperature fluctuations.
- Avoid unnecessary shocks and vibrations.
- If any liquids are spilled on or into the unit, power off the chassis immediately. Remove the unit and allow to dry completely.

**WARNING**

The use of controls, adjustments, and procedures other than those specified herein may result in exposure to hazardous situations or impair the protection provided by this unit.

## 13.1 Annual calibration schedule

To ensure that the unit is performing within specification, we recommend it is re-calibrated every 12 months.

All Quantifi Photonics products are calibrated during manufacture, and each product is shipped to the customer with a Calibration Certificate. On this certificate, the calibration date, as well as the next calibration due date are mentioned.

We recommend your product is returned for re-calibration before the listed due date, to ensure continued performance of the product. For re-calibration service information, or to send in a product for re-calibration service, email support@quantifiphotonics.com.

If the Calibration Certificate has been misplaced, or the calibration due date is not known, email support@quantifiphotonics.com.

# 14 Technical Support

## 14.1 Contacting the Technical Support Group

To obtain after-sales service or technical support for this product, contact Quantifi Photonics:

support@quantifiphotonics.com

To accelerate the process, please provide information such as the name and the serial number of the product (see the product identification label), as well as a description of your problem.

## 14.2 Transportation

Maintain a temperature range within specifications when transporting the unit.

**Transportation damage can occur from improper handling.**

The following steps are recommended to minimize the possibility of damage:

- Pack the product in its original packing material when shipping. If the original packaging is unavailable, use appropriate foam packaging to provide shock absorption and avoid displacement of the product inside the shipping box. Please keep all input connectors covered with the supplied anti-static plastic covers during transport and avoid any shipping material making contact with the sensitive connectors of the product.
- Avoid high humidity or large temperature fluctuations.
- Keep the product out of direct sunlight.
- Avoid unnecessary shocks and vibrations.

# 15 Warranty Information

## 15.1 General information

Quantifi Photonics Ltd (Quantifi Photonics) warrants from the date of the original shipment (the Warranty Period) that this product will conform to specifications and will be free from defects in material and workmanship for the applicable Warranty Period. Quantifi Photonics also warrants that the equipment will meet applicable specifications under normal use.

**NOTE**

The warranty can become null and void if:

- The unit has been tampered with, repaired, or worked upon by unauthorized individuals or non-Quantifi Photonics personnel.
- The warranty sticker has been removed.
- The unit has been opened, other than as explained in this guide.
- The unit serial number has been altered, erased, or removed.
- The unit has been misused, neglected, or damaged by accident.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL QUANTIFI PHOTONICS BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

For full warranty terms and conditions, please visit quantifiphotonics.com.

## 15.2 Liability

Quantifi Photonics shall not be liable for damages resulting from the use of the product, nor shall be responsible for any failure in the performance of other items to which the product is connected or the operation of any system of which the product may be a part.

Quantifi Photonics shall not be liable for damages resulting from improper usage, transportation or unauthorized modification of the product, its accompanying accessories and software.

The external power supply that has been supplied by Quantifi Photonics with the unit can only be used with that unit, do not use it with any other product.

## 15.3 Exclusions

Quantifi Photonics reserves the right to make changes in the design or construction of any of its products at any time without incurring obligation to make any changes whatsoever on units purchased. Accessories, including but not limited to fuses, pilot lamps, batteries and universal interfaces (EUI)

used with Quantifi Photonics products are not covered by this warranty.

This warranty excludes failure resulting from: Improper use or installation, normal wear and tear, accident, abuse, neglect, fire, water, lightning or other acts of nature, causes external to the product or other factors beyond the control of Quantifi Photonics.

## 15.4 Certification

Quantifi Photonics certifies that this equipment met its published specifications at the time of shipment from the factory.

## 15.5 Service and repairs

To send any equipment for service, repair or calibration please contact the Technical Support Group: support@quantifiphotonics.com.

QUANTIFI PHOTONICS®
A Teradyne Company

# Test. Measure. Solve.™

Quantifi Photonics is transforming the world of photonics test and measurement. Our portfolio of optical and electrical test instruments is rapidly expanding to meet the needs of engineers and scientists around the globe. From enabling ground-breaking experiments to driving highly efficient production testing, you'll find us working with customers to solve complex problems with optimal solutions.

## To find out more, get in touch with us today.


| | |
|---|---|
| **General Enquiries** | sales@quantifiphotonics.com |
| **Technical Support** | support@quantifiphotonics.com |
| **Phone** | +64 9 478 4849 |
| **North America** | +1-800-803-8872 |


**quantifiphotonics.com**