ID PHOTONICS GMBH

TRANSFORMING INNOVATIVE PHOTONICS IDEAS INTO REAL-WORLD SOLUTIONS REQUIRES GREAT TOOLS.

ID OSA
OPTICAL SPECTRUM ANALYZER

USER MANUAL

ID OSA

Applicable Part Numbers ID-OSA-MPD-x1, x: Wild Card

CONTENT

1 GENERAL INFORMATION

1.1 COMPLIANCE STATEMENT ELECTROMAGNETIC COMPATIBILITY AND DEVICE SAFETY

Hereby, we declare that this system has been designed and tested for compliance for the following directives.

1.1.1   INTERNATIONAL

IEC 62368-1
CISPR 11:2003 in accordance with EN 61326-1: 2006

1.1.2   UNITED STATES OF AMERICA

FCC 47 CFR Part 15, Subpart B Class A, Measurement process ANSI C63.4 (2009)

1.1.3   EUROPEAN UNION

EN 55022:2011
EN 61326-1: 2013
EN 61000-6-2: 2006
EN 61000-6-4: 2011
EN 61000-3-2: 2010
EN 61000-3-3: 2009
This conformity statement for includes EU directive 2002/95/EG (RoHS) and EU directive EG1907/2006 (REACH).

1.2 LIMITATION OF COMMUNICATION INTERFACES

Operation of all USB Ports is limited to a maximum cable length of 3m and a maximum length of 30m for all Ethernet ports present.

1.3 EUROPEAN WEEE DIRECTIVE COMPLIANCE

ID PHOTONICS has established processes in compliance with the Waste Electrical and Electronic Equipment (WEEE) Directive, 2002/96/EC. This product should not be disposed of as unsorted municipal waste and should be collected separately and disposed of according to your national regulations. In the European Union, all equipment purchased from ID PHOTONICS can be returned for disposal at the end of its useful life. ID PHOTONICS will ensure that all waste equipment returned is reused, recycled, or disposed of in an environmentally friendly manner, and in compliance with all applicable national and international waste legislation. It is the responsibility of the equipment owner to return the equipment to ID PHOTONICS for appropriate disposal. If the equipment was imported by a reseller whose name or logo is marked on the equipment, then the owner should return the equipment directly to the reseller. If you have questions concerning disposal of your equipment, contact ID PHOTONICS's at WEEE@id-photonics.com.

1.4 LINE VOLTAGE SELECTION

CoBrite DX and CoBrite MX chassis operate from any single-phase AC power source that supplies 100 ~ 240VAC at a frequency at 50/60 Hz. The input line voltage setting is done automatically by CoBrite power supply.
The CoBriteDX2 device itself does not have an integrated AC power supply but uses an external AC/DC supply that provides 12VDC to the chassis. This external supply operates from any single-phase AC power source that supplies 100 ~ 240VAC at a frequency at 50/60 Hz. The input line voltage setting is done automatically by the external power supply. Do only use the provided power supply to power the device.

1.5 SERVICE

Do not attempt to service or adjust this instrument unless an authorized person is present. Do not install substitute parts or perform any unauthorized modifications to this instrument. Contact ID Photonics or your local distributor to obtain service support.

1.6 HELP AND USER FEEDBACK

ID Photonics GmbH is dedicated to continuously improve customer experience of our products. Thus, if you have any feedback that might help us to improve our products send us an E-Mail to: feedback@id-photonics.com

1.7 SAFETY

1.7.1 GENERAL SAFETY PRECAUTIONS

The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument.

! NOTE

ID Photonics assumes no liability for the customer's failure to comply with these requirements

Before operation, review the instrument and manual for safety markings and instructions. You must follow these to ensure safe operation and to maintain the instrument in safe condition.

1.7.2 GENERAL

This product is a Safety Class 1 instrument (provided with a protective earth terminal). The protective features of this product may be impaired if it is used in a manner not specified in the operation instructions.

1.7.3 ENVIRONMENT CONDITIONS

This instrument is intended for indoor use in an installation category II, pollution degree 2 environments. It is designed to operate at a maximum relative humidity of 95% and at altitudes of up to 2000 meters.
Refer to the specification tables for the ac mains voltage requirements and ambient operating temperature range.

! NOTE

Before connecting electrical power to the unit, make sure the unit could acclimatize to ambient temperature for at least 2 hours to avoid damage by i. e. condensed humidity on electrical parts inside the unit.

1.7.4 BEFORE APPLYING POWER

Verify that all safety precautions are taken. The power cable inlet of the instrument serves as a device to disconnect from the mains in case of hazard. The instrument must be positioned so that the operator can easily access the power cable inlet. When the instrument is rack mounted the rack must be provided with an easily accessible mains switch.

1.7.5 FUSE REPLACEMENT

For continued protection against the possibility of fire, replace the fuse only with a fuse of the specified voltage, current and type ratings.

1.7.6 GROUND THE INSTRUMENT

The instrument must be connected to the ac power mains through a grounded power cable, with the ground wire firmly connected to an electrical ground (safety ground) at the power outlet. Any interruption of the protective (grounding) conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in personal injury.

1.7.7 DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE

Do not operate the instrument in the presence of flammable gases or fumes.

1.7.8 DO NOT REMOVE THE INSTRUMENT COVER

Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made only by qualified personnel.

Instruments that appear damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.

Opening the instrument will result in loss of all warranty given for the instrument and may exhibit lethal health risks.

Keep away from live circuits inside the equipment. Operating personnel must not remove equipment covers. Only factory authorized service personnel or other qualified service personnel may remove equipment covers for internal subassembly or component replacement or any internal adjustment. Do not install substitute parts or perform any unauthorized modification of the equipment or the warranty may be voided.

1.7.9 VENTILATION

Keep a space of 30 cm or more between the rear side of the device and any other objects such as walls to guarantee sufficient cooling of the device. Never block the air fan and ventilation openings.

1.7.10 CLEANING THE INSTRUMENT

To avoid personal injury, power down the device and disconnect it from line voltage before performing any of the following procedures.

To clean the exterior surface, perform the following steps:

Remove loose dust on the outside of the instrument with a lint-free cloth.

Use a soft cloth dampened with water to clean the device. Use 75% isopropyl alcohol solution as a cleaner. Do not use any abrasive or chemical cleaning agents.

1.7.11 SAFETY SYMBOLS ON INSTRUMENTS

1.7.11.1 WARNING OR CAUTION

If you see this symbol on the product, you must refer to the manuals for specific Warning or Caution information to avoid personal injury or damage to the product.

2 UNIT OVERVIEW - OPERATING PRINCIPLE AND BLOCK DIAGRAM
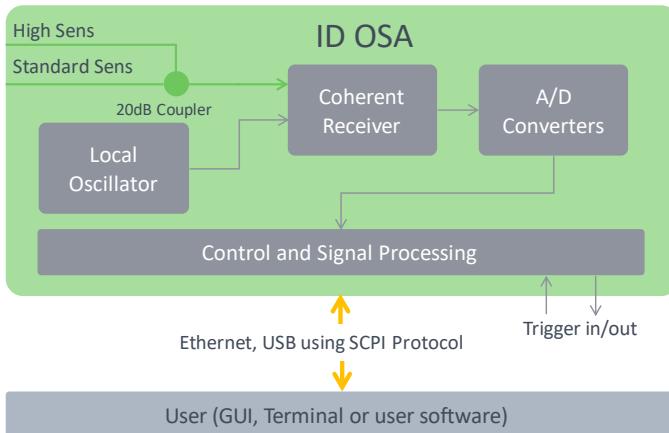
The ID OSA is an optical spectrum analyzer that provides two optical inputs measuring both higher power signal spectra and signals requiring a high sensitivity by means of an integrated coupler across the C Band. The unit automatically corrects the reported power once the user has selected the input currently in use.

The unit utilizes a coherent detection scheme to retrieve the optical spectrum of the incoming test signal. This provides a number of unique advantages over traditional grating based OSAs:

✓ **High Resolution**   - 312.5MHz/2.5pm Scan Resolution
✓ **Fast**   - 2 Hertz Scan Rate at full Resolution across C Band (15,600 data points)
✓ **Compact**   - 1HE half size 19" rack stacking
✓ **Rugged**   - No Moving Optical Parts

A local oscillator together with a coherent receiver provides the necessary mixing terms which are then converted by A/D converters followed by signal processing to extract the optical spectrum data.

User data transfer is provided via a SCPI interface utilizing USB and Ethernet interfaces. The unit supports multiple user connections and multiple user access right level administration. The user interface does not require any specific API Software or similar to be installed on host computers while providing a standardized communication protocol so that any user environment is able to communicate with unit.

The unit provides hardware triggers allowing triggering OSA sweep executions as well as indication of active sweeps.

2.1 FRONT SIDE OF UNIT
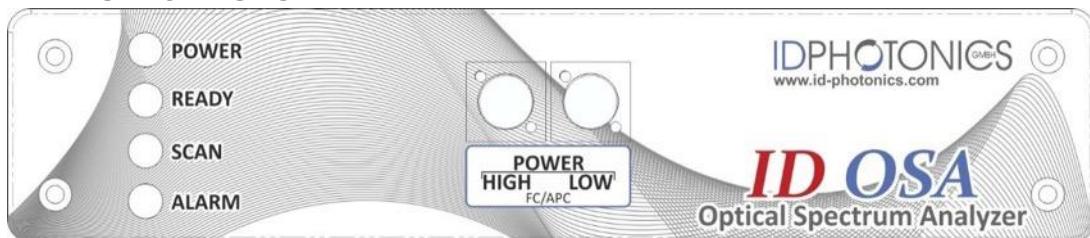
2.1.1 STATUS LEDS

Power: This green LED will light up once electrical Power has been supplied to the rear Line in and the switch located next to it is set to 1.

Ready: The green Ready LED indicates that the ID OSA Operating System was started and the Optics has reached operating condition.

Scan: The green Scan LED indicates an active spectral scan currently in progress. If repeat scan mode is active, the LED will continuously be active as well.

Alarm: The red Alarm LED indicates a pending alarm situation. Check in GUI or via remote access about the root cause of the alarm to eliminate it.

2.1.2 OPTICAL INPUT PORTS

The ID OSA provides two FC/APC type optical input ports which are coupled together before going to the detection circuit as depicted in the schematic overview of the unit. Use the GUI or remote control to set the port to be used in order to obtain calibrated measurements.

For purpose of cleaning dirty connectors, the 2 Ports are hosted a small daughter plate which are fixed via two screws which can be removed by hand. Gently pull out the assembly after the screws have been removed and disconnect the fiber heads from the FC/APC adaptors.

2.2 REAR SIDE OF UNIT
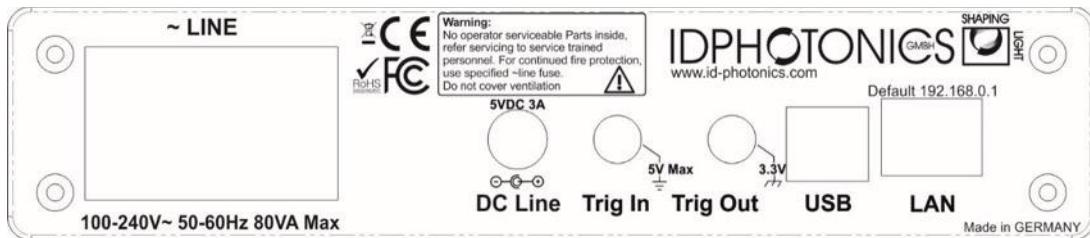
2.2.1 POWER SOCKET

! NOTE

Do not use both line ports at the same time in parallel. It may damage hardware.

2.2.1.1   AC LINE

Electrical power can be supplied using a VDE 0625, EN 60 320, C13 type electrical connector. Only use the supplied cable or an identical cable that conforms to the aforementioned standard to connect to the unit. The unit automatically adapts to local power line specifications within the range printed on the unit. There is no reconfiguration needed to switch between the AC and DC line power options.

2.2.1.2   DC LINE

Alternatively, electrical power can be supplied using 5VDC, 3 Ampere DC099 type electrical connector. Make sure to set the polarity of the VDC line on the unit which is printed on the unit near the jacket.

2.2.2 TRIGGER PORTS

The SMA type sockets can be used to achieve timewise synchronization of the ID OSA with other equipment. Both ports support 3.3V LV TTL standardized signals.

Trigger IN allows triggering OSA scans if the feature is enabled via Remote control or GUI. Polarity and user set delays can be also configured this way. For positive polarity, the scan is started with the positive flank und will be fully executed regardless of the trigger in status afterwards. The execution takes ~500ms. After ~1 second the unit is ready for the next trigger event. The exact timing can be checked using the Trigger out port. Data acquired from a preceding scan is available for reading until the next scan has been taken and was processed. For negative polarity, the scan is triggered by the negative flank on the unit.
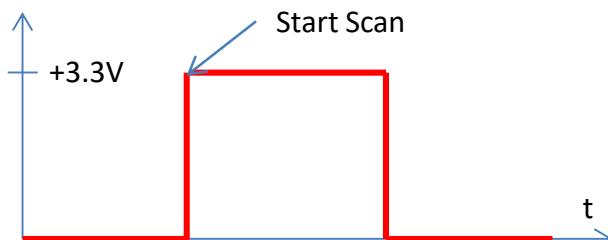
Trigger OUT provides an output signal indicating periods at which the unit acquires the optical spectrum. The polarity can be reversed via Remote control or GUI.
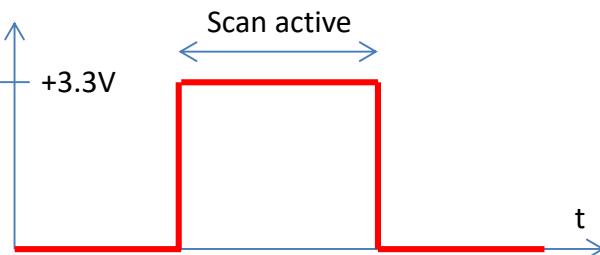
2.2.3 DATA PORTS

Ethernet – The RJ45 jacket is used to connect the unit to Ethernet networks based on the IEEE 802.3 standard. Per factory default, the unit is set to a fixed IP Address 192.168.0.1. The IP address configuration can be changed via Remote control or GUI. We recommend to initially connecting via USB to set up the Ethernet interface first.
See section 4.3 for details on how to establish an application layer connection to remote control the unit.

USB – The USB 2.0 type B jacket is used to connect the unit a host computer to be able to operate the supplied GUI or perform remote control on the unit. See section 4.4 for details on how to establish an application layer connection to remote control the unit.

2.3 OPTIONAL ACCESSORIES

2.3.1 SINGLE 19" RACKMOUNT ADAPTOR ID-OSA-ACC-RM-1
This adaptor is used to mount one ID OSA chassis in a standardized 19" Rack in 1 HE slots.

2.3.2 DUAL 19" RACKMOUNT ADAPTOR ID-OSA-ACC-RM-1
This adaptor is used to mount two ID OSA chassis side-by-side in a standardized 19" Rack in 1 HE slots.

3 QUICK START

This section contains redundant information from other chapters but is useful for a first-time usage of the instrument.
1. Power up the unit using the main power switch located at the back of the unit or provide 5VDC to the DC line jack.
2. Wait until the boot sequence is completed and the ready indicator located at the front lights up. The chassis fan will start to operate.
3. Connect the USB Port located at the rear of the unit to your Windows10 PC or connect the Ethernet Port to your LAN.

   a. **For USB** based access, all drivers will be installed automatically, Windows device manager should show now 3 devices:
   1. A storage device containing manual etc.
   2. A virtual Ethernet Interface "RNDIS". The IP Address of this interface is printed on the Product label located on the bottom side of the device.
   3. A virtual COM Port
   Both virtual COM and Ethernet Ports can be used to remote control the unit. The installed COM Port number can be retrieved from Windows device manager. To access the WebGUI, the USB virtual Ethernet IP address is needed. If the DNS configuration of your computer allows, the IP address can also be retrieved by entering the command shell "`ping osa.local`" or retrieve it from the label located on the unit.
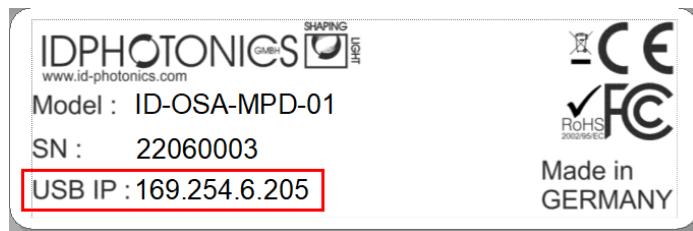
b. **For Ethernet** based access, the default IP of the unit is http://192.168.0.1 . Make sure that the host PC IP is in the same subnet as the laser unit (192.168.0.x). If this is not the case, you can change the IP settings of the unit via the USB Port of the unit. Use http://osa.local/ as an alternative to the IP address to connect to the unit if your DNS configuration of the network allows this access.

   •

4. Open your Web browser, enter http://osa.local/ or the IP Address as described on previous step in the address field and hit <enter>. The Webpage allowing to control the instrument remotely should open now.

! NOTE

Depending on the configuration of your host PC DNS structure, the http://osa.local/ name representation might not be resolved into the correct IP address. In such a case, there are 2 options to retrieve it:
1. Open a connection to the USB virtual serial interface and type the command "`USBIPADDR?;`" to retrieve the USB IP Address or "`IPADDR?;`" for the IP address of the Ethernet interface
2. Retrieve the IP address from the type label located on the unit.

4 GETTING STARTED

4.1 SOFTWARE INSTALLATION
The ID OSA can be operated without installing any software using the integrated webserver. Connect to it using the address field of your computer browser as described above.

4.2 HARDWARE INSTALLATION
- Connect the USB or Ethernet interface located at the rear of the unit to the host PC using the provided USB or Ethernet cable of the shipping package.
- Connect the power cable of the shipping package to the power outlet located at the back of the unit to a power socket and toggle the switch next to the power outlet to "1"
  - o The "Power" LED located at the front should now light up
  - o The "Alarm" LED will light up for a few seconds and then should go off again
  - o A few seconds later, the "Ready" LED should light up
- The unit is now ready to operate

4.3 ETHERNET CONNECTION
This section covers connectivity using Ethernet, skip this section if you plan on using USB.
The default IP address is 192.168.0.1, DHCP off. If you plan to connect using the Ethernet interface, make sure the host PC is within the same subnet as the ID OSA Ethernet Interface (The PC IP Address is 192.168.0.x then). The connectivity can be tested by opening a command shell (type "cmd" in windows search field and hit <enter>, a "black window with a DOS shell should open) and executing the command "ping 192.168.0.1".
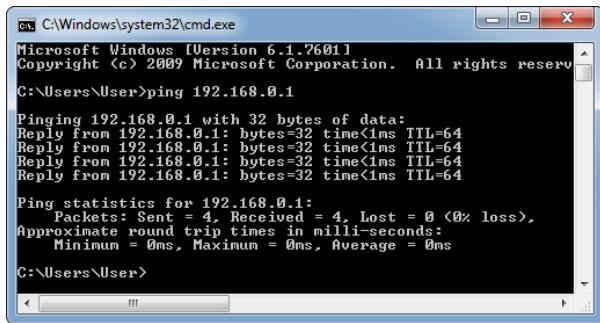
If you do wish to change the IP settings and do not have remote access to the unit follow either of the following sections depending on the type of chassis:

Note that the unit possesses two IP Addresses. One for the physical Ethernet interface and one for the virtual Ethernet Interface generated by a USB connection. Only the one of the physical Ethernet interface settings can be changed and is handled in this section.

4.3.1.1   HOST COMPUTER CAN CONNECT TO CURRENT IP ADDRESS OF ID OSA

Enter IP Address of unit in the browser address field of your host PC browser (default: 192.168.0.1). The ID OSA GUI should now open in the browser window. Go to connection tab, raise user level by entering password "IDP", now change the IP settings and restart the unit to take the new setting into effect.

4.3.1.2   HOST COMPUTER IS UNABLE TO CONNECT TO ID OSA VIA CURRENT IP ADDRESS

Use this method if the "ping test" as described above is not successful.

Connect the unit via USB and wait until driver installation is complete.

Retrieve the COM Port number of the virtual serial port using windows device. Open a terminal window to the virtual COM Port using i. e. PUTTY.

Send the command "`pass IDP`" to elevate the user level to 1 allowing changes to the IP configuration.

Use the command "`ipaddr xxx.xxx.xxx.xxx`" to set a new IP address of the physical Ethernet interface where `xxx.xxx.xxx.xxx` is the IP address you wish to use, ie. "192.168.5.12". For more settings such as DHCP and netmask, see commands defined in section remote control.

You can query the IP address setting via "`ipaddr?`".

Reboot the unit using the command "`*rst`" in order for the new IP configuration to become effective. The unit will immediately reboot after issuing this command.

4.3.2 OPENING A REMOTE CONNECTION VIA ETHERNET

The device supports a session based remote access on Port #2000.

Start the terminal program "putty.exe" supplied with the unit or use any other terminal program.

Set the following parameters in Putty and open the connection:
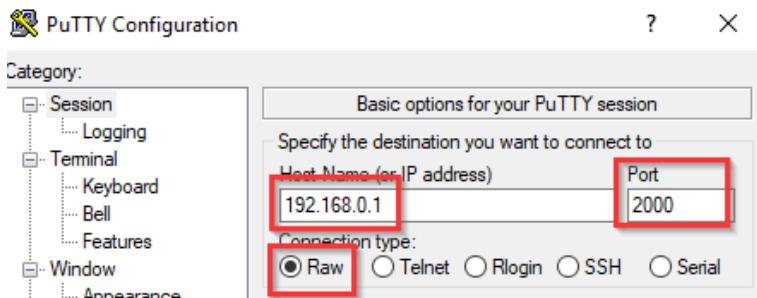

Alternatively, to the default IP Address shown above, enter http://osa.local/ or the IP address currently set to the unit. If in doubt, use the touch panel display "Device config" – "IP config" to retrieve the actual IP Address.

Note that the must be a valid route on the IP layer between the device and the host PC (i. e. the ethernet port of the host PC is set to the same subnet as the device, for example 192.168.0.2) must be established before continuing.

Use a command shell and command "`ping osa.local`" to test the route.

Once the connection is established, type in "`*idn?`" and hit <ENTER>. The unit responds with its *idn? String.

Alternatively, a request based remote control via http Port 80 is possible. To test this, open a browser window and type the following string into the address field:

> http://osa.local/scpi/*idn?

In case DNS resolution is not supported on your Host PC, use the IP Address of the instrument instead of "`osa.local`".

4.4 USB CONNECTION

This section covers connectivity using USB, skip it if you plan using Ethernet.

Once the unit is powered up and the USB cable is connected to the host computer for the first time, a new device installation should be triggered automatically within Windows.

Once installation is complete, three devices are installed on the host computer:

1. Virtual Ethernet Interface. The IP address of each unit is unique and fixed. Retrieve the IP address either by query "`USBIPADDR?`" or from the type label sticker located on the unit.
2. Virtual COM Port
3. Virtual Storage device that contains resources such as manual and programming devices as well as drivers for Windows 7.

4.4.1.1 WINDOWS 10

Windows 10 will automatically install 3 devices allowing to connect the unit

1. A virtual Ethernet interface (RNDIS)
2. A generic virtual COM port driver "Serial USB device". If you have several COM ports installed in the host PC, you may want to note the COM Port number under which the unit got installed. For further details on USB connection, see section 0.
3. A storage device similar to an USB Stick containing resources for the unit.

The virtual COM Port and the virtual Ethernet interface are concurrent ways to access the unit for remote control. The web-based control is automatically accessed via entering http://osa.local/ into your browser. If DNS resolution of your PC is not supported, enter the USB IP Address into the address field of the Browser. The IP Address is printed on the device label located on the instrument.
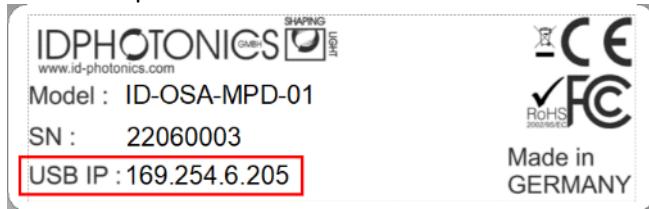
4.4.1.2 WINDOWS 7

Windows 7 will install the same structure as described in the windows 10 section above. It will automatically install a virtual Ethernet interface (RNDIS) and the USB storage device ("Linux File- Store Gadget USB Device"). However, it does not have a preconfigured driver setup for the virtual COM Port. It will try to locate a driver for the unit on the host PC first and then check online. Stop the search and select the option "Install a custom driver". Point the installation routine to the driver located on the USB storage device provided by the unit in directory "USBDriverWin7" and select file "ID OSA USBSerialDriverWindows7.inf" in this folder. If the installation has been completed already, open Windows Device Manager, locate "CDC Serial", right click on it and select "Update driver". Proceed as described before.

Continue with the installation routine. Once installed, the virtual COM port should appear in Windows Device Manager as shown below. The COM Port number will be different for your system. Note down the number for later usage in remote control applications.
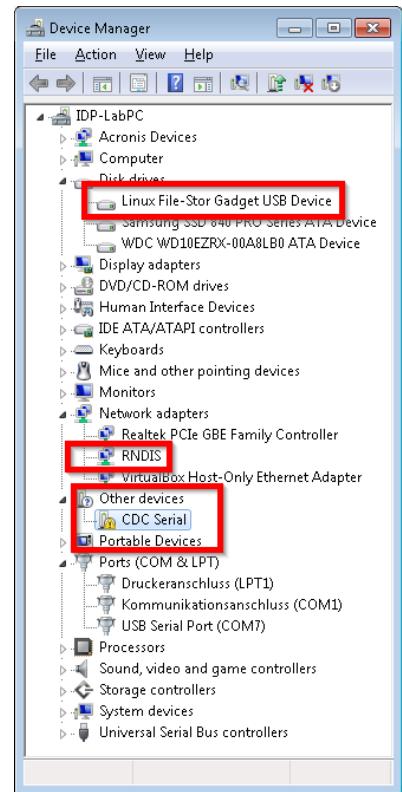
4.4.2 CHECK USB DRIVER INSTALLATION IN WINDOWS DEVICE MANAGER

Once the installation is completed, a check for successful installation can be performed by opening Windows Device Manager, i. e. press <Windows Key> + <Pause> and click on "Device Manager".

4.4.2.1 CONNECTION TO THE DEVICE VIA USB VIA HOST PC

Once installed properly, the USB connection provides a virtual COM Port and a virtual Ethernet Port to the instrument.

4.4.3 CONNECTION VIA VIRTUAL ETHERNET PORT ON USB

To access the unit via virtual Ethernet, follow the instructions as in section 4.3.2 but note that the IP Address of the unit will be different as for the physical Ethernet Port and cannot be changed as it is assigned by the Windows host PC upon installation.
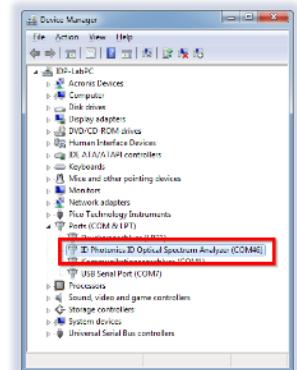
4.4.4 CONNECTION VIA VIRTUAL SERIAL PORT ON USB

Use any terminal program such as "Teraterm" or "Putty" which can be found on the supplied USB Stick to test custom remote control.
The connection parameters are:
 • Baud rate: 115.200
 • Serial settings: 8N1

Once connected, type i.e. "`*idn?`" and press <ENTER> to test the connection. This command queries info's about the OSA. A typical response is
```
IDP-OSA-MPD-01, SN 1703001, F/W Ver 1.0.0(56), HW Ver 1.10;
```

4.5 CONNECT WITH GRAPHICAL USER INTERFACE

We recommend using the WebGUI which is accessed by either entering the **IP Address** (default http://192.168.0.1 for Ethernet) http://osa.local/ into your browser address field to open it.

4.5.1 CONNECTION VIA ETHERNET

The default IP 192.168.0.1 for Ethernet based access. This Graphical User Interface (GUI) can only be accessed if the unit is connected to the same Ethernet network as the host unit running the Web Browser and the IP Address is properly set up so that it is reachable from the host PC. If the Ethernet interface is not yet configured for the network to be used, first connect to a host PC using the USB Interface to set up the Ethernet Interface, see 4.3 for details.

4.5.2 CONNECTION VIA USB

Alternatively, the USB Interface can be used to access the GUI via the virtual Ethernet Adaptor that is installed on the host PC. For details, see 4.4.

5 GRAPHICAL USER INTERFACE – WEB INTERFACE (WEBGUI)

The integrated web server of the unit allows to access and control the unit using any compatible Web Browser on any device without installing software. The Web Interface was tested on Windows 7 and Windows 10 Computers using Edge, Chrome and Firefox and iPhone/iPad using Safari browser. Note that we cannot guarantee operation in any Browser environment in general.

Hover the mouse pointer over the areas of interest. Tooltips will appear with context sensitive help.

The browser tab indicator includes the serial number of the unit currently accessed so that if multiple units are accessed at the same time can be differentiated easily.

The top right side hosts a number of status indicators that highlight system wide settings and will change color, if active.

The alarm indicator will indicate and active or latched alarms. Click on the icon to access further details.

The "shift active" indicator indicates if non zero numbers are set in the "wavelength shift" or "level shift" fields in the System tab to remind the user that the values contain an offset relative to the actual measured values.

"Remote" indicates that there are active remote sessions.

"Remote lock" indicates that the device does not accept set commands via remote connections (reading is still possible) due to a forced setting in one of the active sessions. For details, see the remote-control section of this manual.

The login button allows to elevate the user rights of the active session to access protected settings such as the IP Address configuration. The protected settings are highlighted by a lock symbol, if locked.
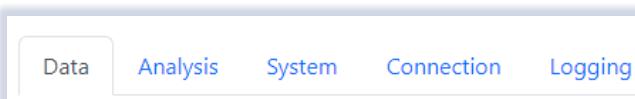
The graph window does have tabs that allow access to more features of the device. Click on the tabs to access more features of the unit. Please see the following section for a detailed description of the objects in these tabs.

All objects do have a short description displayed if the mouse pointer is hovered over the object.

The following screenshots document the tabs for your reference.

5.1 DATA TAB

This tab is designed to display the measured trace and configure most of the measurements.

The optical power shown by the filling slider is the integrated power across the spectrum measured by the unit.
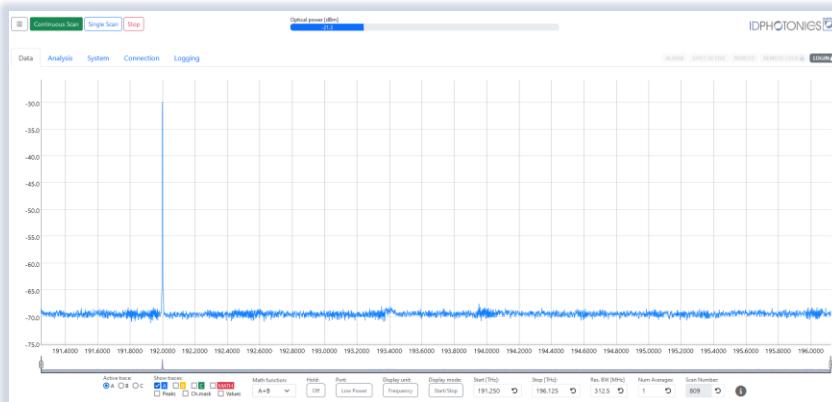
Note that parameter changes performed by one user will impact the analysis output of any other user connected to the unit in parallel.

Use the slider shown below the graph to zoom into areas of the measured data. The data outside of the zoomed area is still measured and depicted in the slider. The selected area can be moved by dragging the selected area with the cursor.
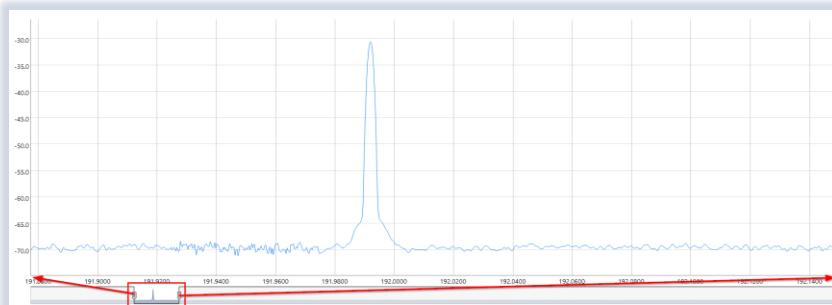
Alternatively, it is possible to zoom in via marking the area of interest using the cursor in the main graph.
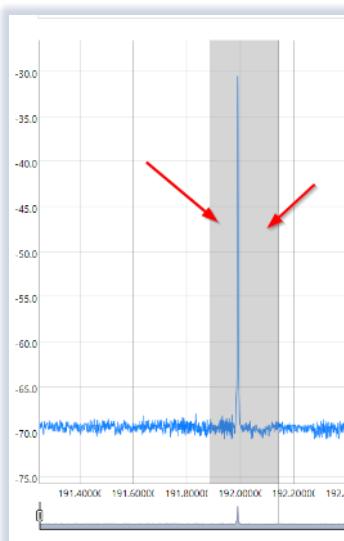
A double click anywhere in the graph restores the full view.

Use single cursor clicks to set markers. Each click adds a marker for the current values are shown in a box. Remove all markers using the box in the upper left side.
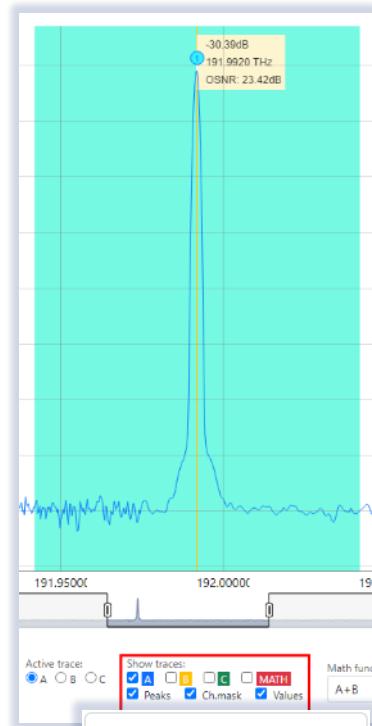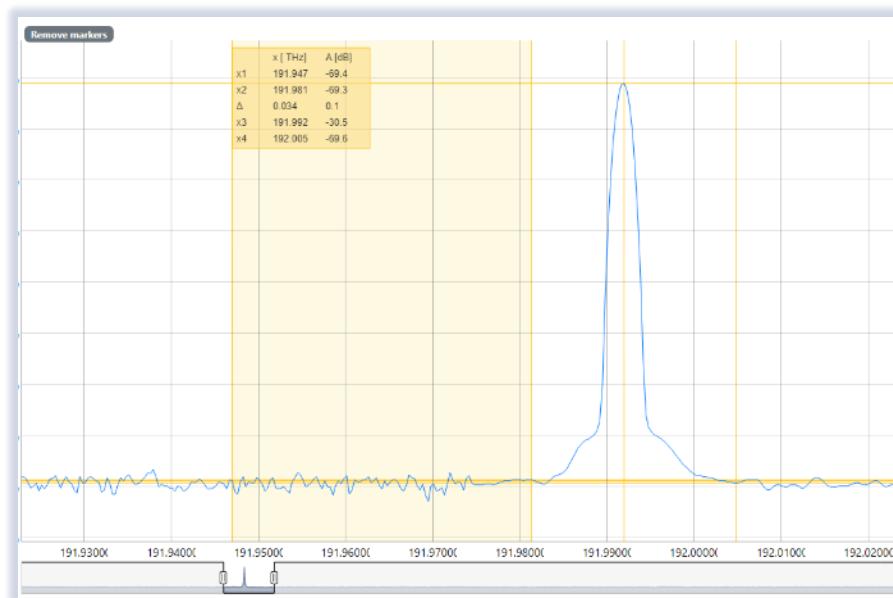
Use the checkboxes located in the lower part to make the channel mask of detected peaks visible and show analysis results.

 The menu icon allows access to various resources such as drivers, programming examples, saving data and system level features. Click on it to unfold the menu.

 This button will enable the repeating measurement mode of the instrument. Spectrum scans will be endlessly performed independently of the GUI connected or not.

 This button triggers a single spectral scan. If the previous mode was "repeat", this is aborted.

 This button aborts pending spectral scans.
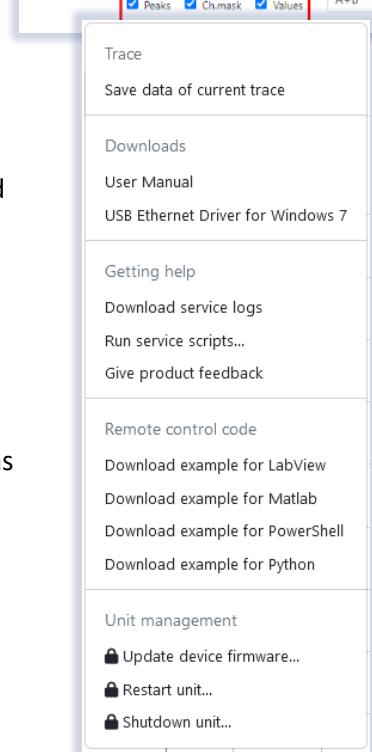
The lower pane allows to configure traces, set start and stop Frequency or Wavelength, and further parameters that are relevant for the setting up the measurement. Hover the cursor over each field to obtain further details for each one.

 Click on the return icon to reset values to its default setting.

Display Mode

Frequency This button toggles between wavelength and frequency-based display of data.

Start/Stop This button toggles between Start Stop and Center Span definition of the ID OSA scan range.

High Power The Port button toggles the input port of the unit that is currently used. See section 2.1.2 for details.

Num Averages:

5 Will change the internal moving average applied to the last n successive scans of the unit.

Hold:

Off Allows the trace data points to hold either the maximum or minimum power points measured since enabled.

5.2 ANALYSIS TAB

The analysis tab provides a table with data computed for peaks detected in the spectrum and allows changing parameters for the analysis. For Details on the settings, see 6.3. The data can be exported using the "Save analysis results" button.

WDM analysis results

Number of channels detected

1

Save analysis results

CH number	Frequency [THz]	Peak power [dBm]	OSNR [dB]
1	191.9917	-30.41	23.29

WDM analysis settings

Ch P-V Threshold [dB]

10.000

Peak mode difference [dB]

0.000

Min peak distance [GHz]

0.313

Channel mask [GHz]

100.000

Peak power mode

Peak

WDM analysis error status

Analysis OK

The left-hand side displays a table with peaks detected in the spectrum based on parameters defined on right hand side.

Note that all calculations required for the analysis are performed inside the ID OSA unit. Therefore, if a preciously trace saved to the host computer disk, the analysis cannot be performed by the GUI. Furthermore, parameter changes in the Analysis Tab performed by one user will impact the analysis output of any other user connected to the unit in parallel.

5.3 SYSTEM TAB

The System tab hosts various system wide settings. Hover the cursor over the fields to access a detailed description of them.
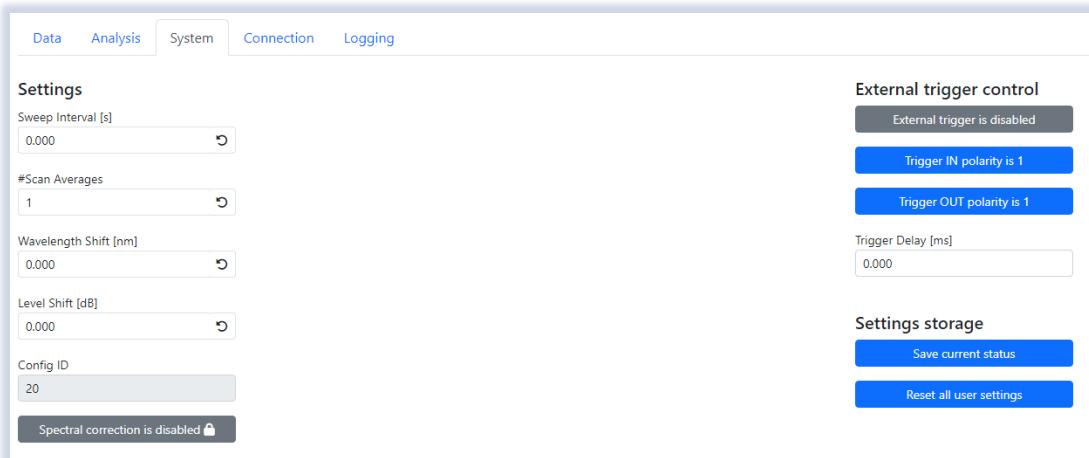
5.4 CONNECTION TAB

This tab is used to retrieve or set information on the remote-control interfaces of the unit, device info and information about current firmware and identification of the unit. Hover the cursor over the fields to access a detailed description of them.
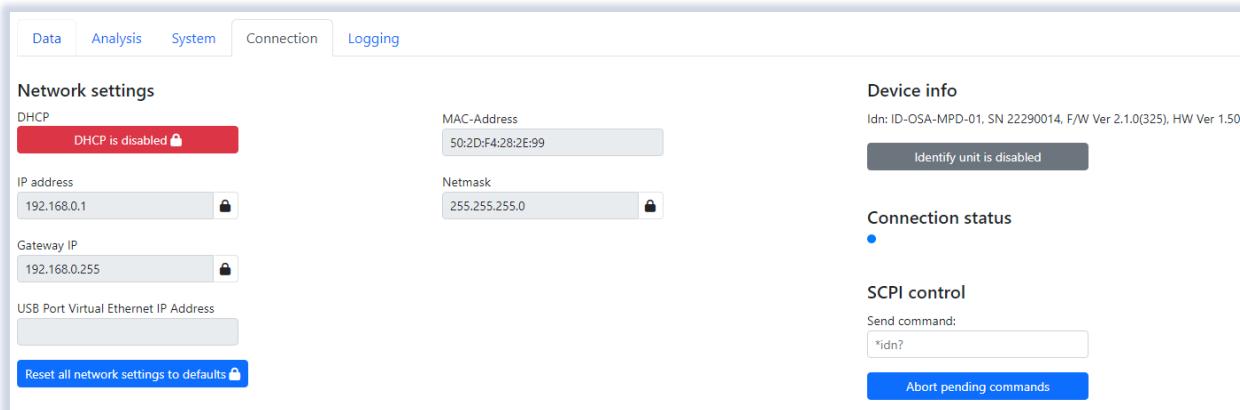
Enter any SCPI command supported by the unit. The response will be displayed in a popup window. For details on the command structure, see chapter 7.

5.5 LOGGING TAB

The logging tab allows to track the SCPI commands that are exchanged with the unit in this browser session coming from user interactions. This feature can be helpful for rapid development of remote-control scripts.
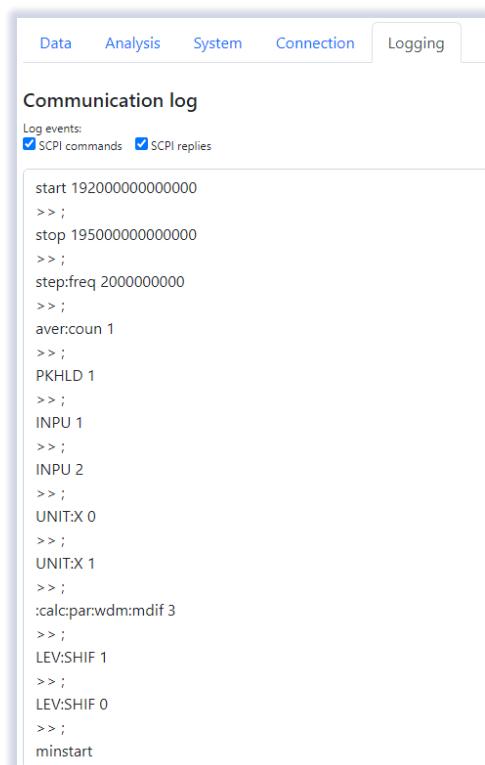
6 GENERAL CONCEPTS OF THE ID OSA

6.1 RESOLUTION BANDWIDTH AND SAMPLING INTERVAL

The resolution bandwidth of the OSA (RBW) is an independent parameter of the sampling interval (SI) of the OSA. The minimum effective filter bandwidth of the unit is limited to 1.7GHz and the sampling interval to 312.5MHz. If the sampling interval is set below RBW, an oversampling occurs and the OSA filter is acting like a moving average filter on the sampled signal. If the sampling interval significantly exceeds the OSA Filter bandwidth, the effective filter shape for each sampling point becomes close to a rectangular shape since the OSA bandwidth is small relative to the sampling interval. Please see the next paragraph for details on computation of the filter for sampling intervals > 312.5MHz.

Sampling interval of 312.5MHz (not to scale)

Sampling interval of 1.7GHz
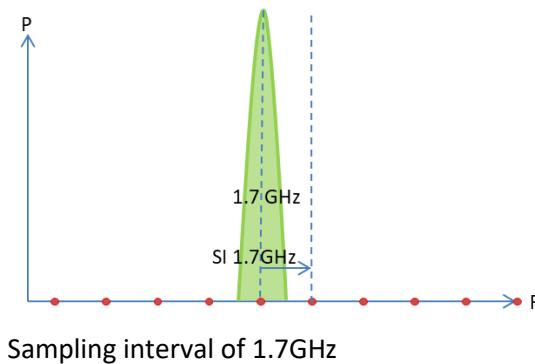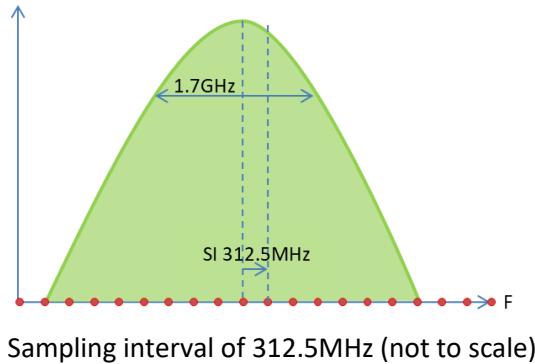
Note that internally, the power for each data point is summed up by the scheme explained in next paragraph.

6.2 MAPPING

The ID OSA will always internally scan with the maximum resolution available regardless of the user setting yielding 15600 data points equally spread across the acquired spectrum with a distance of 312.5MHz per data point. Once the data was acquired it is internally mapped and filtered by a moving average filter. The width of this filter is defined by the user defined RBW (SCPI Command `STEP`). If the RBW is not a multiple of 312.5MHz, the unit will linearly interpolate the contribution of the data points at the edges of the filter into the result. The resulting frequency available at the SCPI interface is the center of the RBW filter. The interpolation will be done internally in linear domain regardless of the user setting by the SCPI interface of dBm (log) or mW (linear). By changing the number of Points (SCPI Command `POIN`), the step size of the moving average filter can be increased up to the width of the RBW filter.

Note that the minimum start (`MINSTAR`) and maximum stop (`MAXSTOP`) Frequency/Wavelength will change as RBW is increased to accommodate for the wider filter. The maximum number of points (`MAXPOIN`) will also be
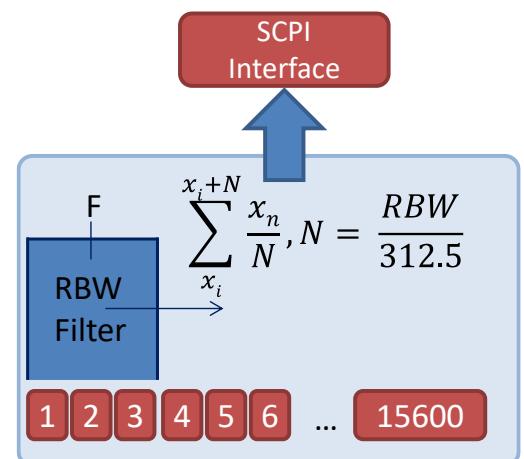
impacted by an increase of RBW (`STEP`). The following formulas are given for Frequency based X Axis but apply similarly:

`MINSTAR` = 1.9125E14 + RBW / 2

`MAXSTOP` = 1.96125E14 - RBW / 2

`MAXPOIN` = 15600 - || RBW / 3.125E8 ||

If start (`STAR`) and stop (`STOP`) of the scan are set to the minimum (`MINSTAR`) and maximum (`MAXSTOP`) possible and RBW (`STEP`) is increased, new (`STAR`) and stop (`STOP`) settings will be automatically set by the unit. The Example below illustrates the mechanism for a simple case of an RBW with of 3 * 312.5MHz = 937.5MHz is shown.

6.3 AUTOMATIC MEASUREMENTS

6.3.1 OSNR MEASUREMENT

The OSNR measurement is performed according to definitions in IEC 61280-2-9 standard.

OSNR is calculated for each peak found by the Peak detection feature which also delivers the peak power. The Noise power referenced to 0.1nm is determined by the first data points outside of box centered on the Peak defined by the OSNR mask parameter. The noise level at signal wavelength is calculated by means of linear interpolation. The OSNR is then calculated by the following formula with all input variables in [dB]:

$$OSNR\ [dB] = (P_i + N_i) - N_i$$
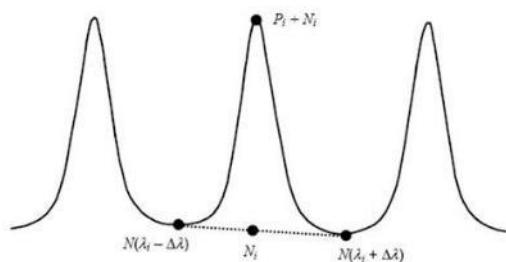
6.3.1.1   IMPACT OF CHANGING THE ID OSA RESOLUTION BANDWIDTH (RBW) ON OSNR RESULT

Independent of the RBW, the algorithm will always automatically scale the noise power N_i to 0.1nm reference so that the derived noise power will not change if the RBW is changed.

The Peak Power P_i will change as RBW of the OSA is changed as long as the signal is spectrally wider than the OSA filter which acts as moving average window across the signals as it sweeps. RBW needs to be at least as wide as the signal width to capture the signal total power. For a signal with unknown width, the minimal required RBW for a valid OSNR measurement can be found by increasing RBW until the OSA trace tends to flatten at the peak indicating that the OSA filter is wider than the signal width.

Note that at full sampling rate, the OSA oversamples the signal so that the peak power detected will be lower than the actual peak power of even a very narrow signal such as a continuous wave narrow line width laser. For peak detection, increase the RBW such that it matches the minimum filter bandwidth of the OSA. This value is about 2GHz. See section 6.1 for more details.

If the Signal Under Test is wider than current set resolution of the unit, either decrease RBW until the full signal is captured or use the integrating power setting for peak power detection. For more details, the section 6.3.2.1.
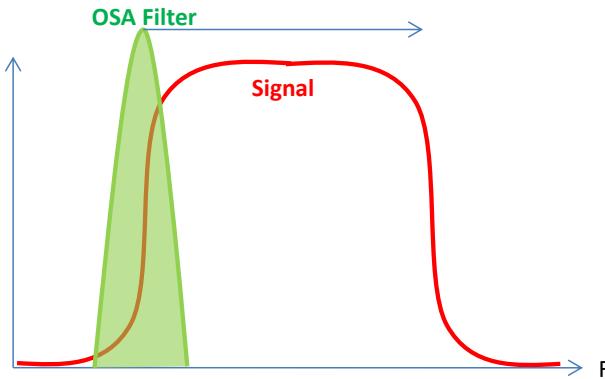
6.3.2 PARAMETER DEFINITION

Channel mask area: Parameter defines the width of a box centered on a detected peak as indicated in the diagram below. It defines the data points for which the Noise is calculated for. This parameter also set the power integration width for peak power if the mode is set to "integrate".

6.3.2.1   PEAK PARAMETER DEFINITION

These parameters define criteria which qualify a data point to be a peak detected by the peak algorithm. Each criterion must be fulfilled in order to qualify as a peak.

Channel P-V threshold: (PVT in Diagram): Defines the minimum power difference of a data point relative to the lowest power data point present in the trace. Example: Minimum Power Pmin in trace is -50dBm, PVT = 25dB. Data points with a power of > -25dBm (-50dBm + 25dB) meet this criterion.

Peak Mode difference: (PMD in Diagram): Defines the minimum power drop threshold between peaks. Example: If PMD is 10dB and the power drops by 5dB between data points meeting the PVT criteria, only the first peak meets both criteria.

Explanation based on the example diagram shown below:

- "1" qualifies as a peak –indicated by a green dot – because it exceeds the PVT and as it is the first peak, the PMD criteria is not applicable
- "2" does NOT qualify as a peak. It exceeds the PVT but does not meet the PMD criteria relative to "1"
- "3" qualifies as a peak as it exceeds the PVT and meets PMD criteria relative to the first detected peak "1" as power drops below P(1) – PMD.
- "4" does not qualify as a peak since the max power is below Pmin + PVT.
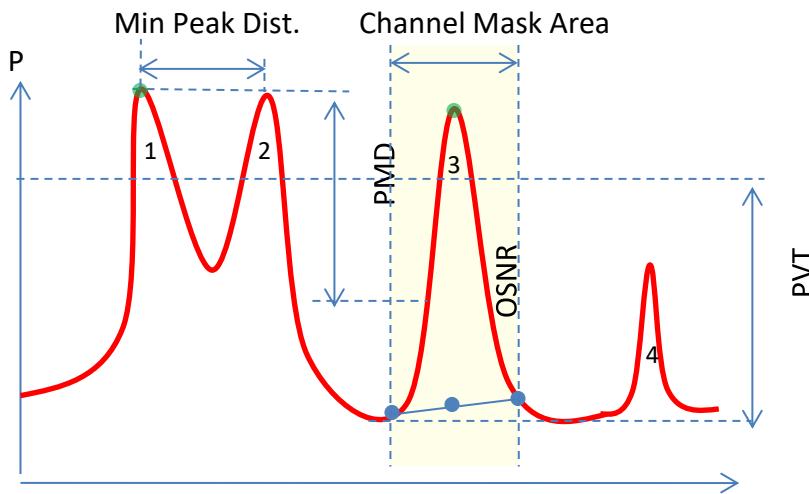
Min. Peak Distance – Allows to screen for peaks which a minimum of spectral distance given by this threshold. With a Threshold of 50GHz, each peak following a previously detected peak with a distance of less than 50GHz is ignored.

Peak Power Mode – This drop down can be set in two modes:

1. **Peak** (setting 0) – This mode will determine the power of the peak in the sampling point of the maximum power. If the Signal Under Test is wider than that, power will be distributed in neighboring samples. This power is NOT taken into account for power calculation. Increase the RBW to capture the signal power in adjacent samples as well.
2. **Integrate** (setting 1) - This mode will determine the lateral peak position by finding the sample with highest power. It will then integrate the signal power distributed in all samples that are defined within the Channel Mask parameter. By this mode, the detected Peaks power will be independent of the RBW set on the unit as long as RBW is smaller than the channel width parameter. Use this method to measure the power of spectrally wide signals that exceed the current RBW setting.

6.4 AVERAGING

ID OSA provides two averaging mechanisms which are connected in a nested loop as depicted in the flow chart below.

Scan Averages will take the number of specified scans and calculate the average out of the scans taken. Once this procedure is completed the scan counter is increased by 1 and the result is available to the user via the SCPI interface or GUI. There is no user access to intermediate scans within that procedure und unit will remain busy until it completes. Therefore, a result is only available for the SCPI Interface and GUI once the required number of scans has been taken und the refresh rate to the use will decrease linearly with the number of scan averages set to the unit.

Moving Averages (# Averages in GUI) will calculate an average of scan from the scan average input (=scan set) in a "First In First Out" (FIFO) logic. A result is immediately available at the SCPI once a first scan is in the FIFO buffer. The scan number counter will increase for each scan that is put into the buffer. The full averaging is achieved once the specified number of scans was filled into the buffer. For each scan put in thereafter, the oldest scan in the buffer is discarded. If OSA settings such as RBW, start, stop etc. are changed, the buffer is emptied and will be gradually filled as described. If the number of moving averages (NMA) is changed, data will NOT be discarded but the average calculation will be adapted accordingly. I.e. if NMA is changed from 10 to 20, the new buffer will already have 10 sets in it and will gradually add sets as they become available.

The total number of scans averaged is calculated by: Scan Averages * Moving Averages
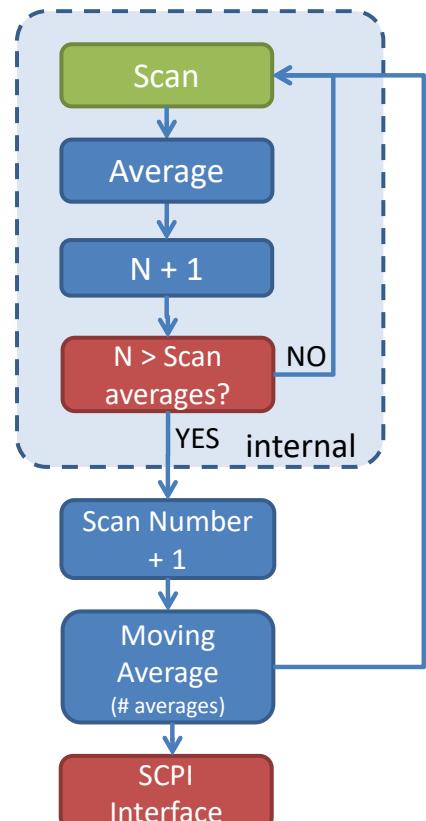
The following example illustrates the mechanism and terminology. The unit takes 3 scans (red boxes) internally which are averaged (blue) and assigned a scan number. It is then fed into the moving average buffer containing 4 sets in this example. Since scan number 5 exceeds the buffer size, scan set number 1 is discarded. The total number of scans required to fill the buffer is therefore 4 * 3 = 12. First data is available to the user upon completion of the 3rd scan which is stored in scan set 1.
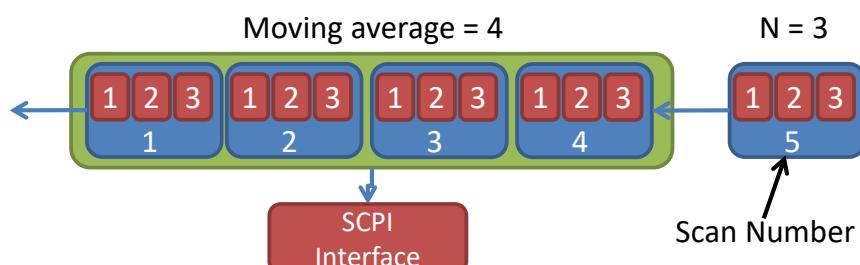
6.5 SPECTRAL CORRECTION

This feature allows applying a spectral correction vector to the data measured by the unit. The feature can be switched on and off in the system tab. By default, the correction vector elements are all set to 0. The correction applies on the maximum resolution grid provided by ID OSA which is 312.5MHz yielding to 15600 elements spread across the maximum scan range available. The correction vector defined in [dB] is stored in the ID OSA unit and internally applied to the raw data measured. If the feature is enabled, data provided by the unit will internally calculate a new power data point based on the correction:

$$P_{out,i} = P_{measured,i} + Correction_i$$

The following diagram depicts the correction visually. Blue bars represent measured raw data points, red bars represent correction values defined by the correction values. The sum of red plus blue bars is the data reported power vector by the unit.
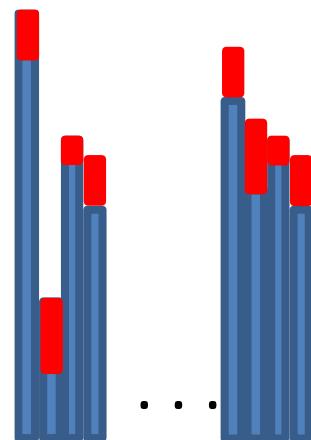
If RBW is changed, ID OSA will automatically calculate the correction vector and the result remains valid based on the vector stored in the unit.

6.6 SWEEP MODES

Overall, there are three sweep modes: **SINGLE**, **REPEAT** and **AUTO**. Although REPEAT and AUTO are defined as separate modes, they are functionally identical. This distinction exists solely for compatibility reasons.

In **SINGLE** mode, a new trigger event initiates a single sweep. The number of possible trigger events is not limited. Each sweep takes ~0.5 seconds. Any trigger event during the execution of a sweep is ignored.

In **REPEAT/AUTO** mode, an initial trigger starts a continuous loop of sweeps, with each sweep initiated at a configurable time interval. The repeat interval is set using the `INTerval` command, which accepts values between **0 and 60 seconds**. If the value is set to 0 it will run as fast as possible.

Within the repeat interval, a trigger event will initiate an immediate sweep before the interval has elapsed. After completion of the sweep, a new loop of sweeps with the set time interval is started.
Sending SMOD 1 switches to **SINGLE** mode and stops any currently running continuous scan. In contrast, sending SGL both switches to SINGLE mode and immediately starts a new single sweep.
It is possible to switch to **REPEAT** mode by sending either SMOD 2 or SMOD 3. To both switch to **REPEAT** mode and start a continuous scan, the RPT command can be used.

Trigger Events
A new sweep is executed by the following trigger events:
1. sending command INIT:IMM
2. Hardware trigger via SMA connector located at rear of unit
3. Elapsed time of interval in repeat mode

Shorthand Commands
The commands SGL, RPT, and AUTO are shorthand instructions that set the sweep mode (SMOD) and automatically trigger a sweep. They are defined as follows:
- SGL = SMOD 1; INIT:IMM
- RPT = SMOD 2; INIT:IMM
- AUTO = SMOD 3; INIT:IMM

Here, SMOD X sets the sweep mode, and INIT:IMM initiates an immediate trigger.

SMOD Command Details
SMOD is the short form of the [:INITiate:]SMODe command. It takes one parameter: 1, 2, or 3 and sets the mode as follows:
- **SMOD 1** → SINGLE mode
- **SMOD 2** → REPEAT mode

- **SMOD 3** → AUTO mode

Web GUI Behavior
- Clicking the **Continuous Scan** button in the Web GUI sends the RPT command.
- Clicking the **Single Scan** button sends the SGL command.

Illustration of Sweep Mode Functionality
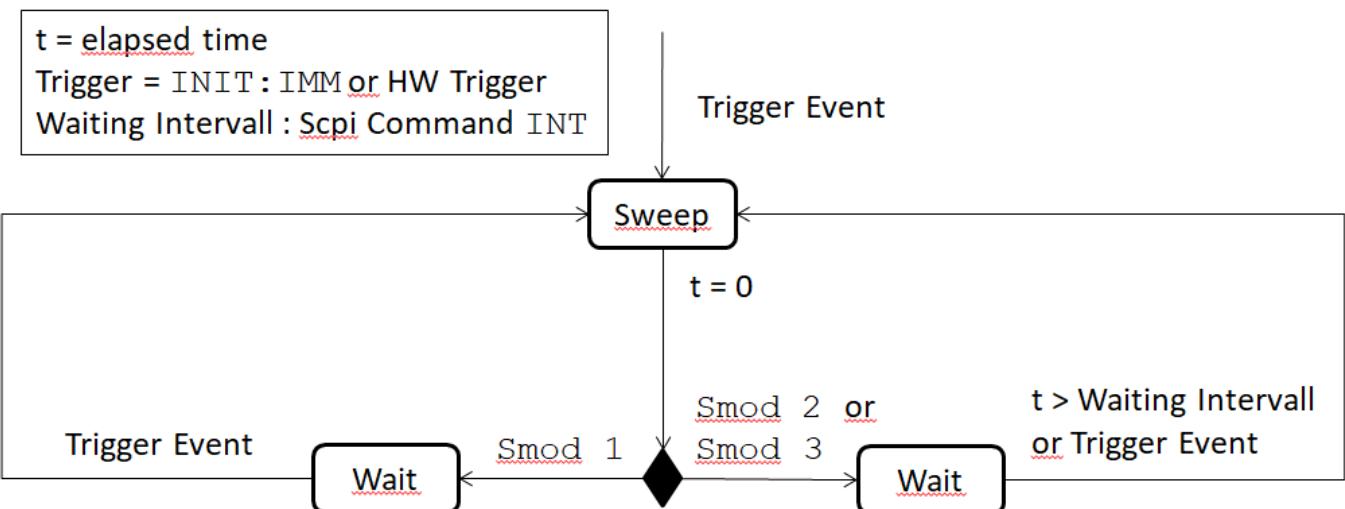
7 REMOTE CONTROL

This device operates using SCPI standard style commands which are ASCII based and allow easy communication and interpretation with the instrument. Refer to chapter "Board and Connector description" on how to establish communication to the ports available at the instrument. More detailed information on SCPI syntax can be found here: https://www.ivifoundation.org/downloads/SCPI/scpi-99.pdf
The following description is based on Firmware 2.1.0 or later.

HOST COMPUTER DRIVERS AND IMPLEMENTATIONS

7.1.1 LABVIEW
National Instruments LabView® drivers covering all SCPI commands are supplied with the USB stick delivered with the unit in directory "\RemoteControl\LabViewDrivers". The drivers are compatible to National Instruments LabView® R8.6 or later.

7.1.2 MATLAB
A Matlab script demonstrating remote operation of the ID OSA is supplied with the USB stick delivered with the unit in directory "\RemoteControl\Matlab".

7.1.3 PYTHON
A Python object script demonstrating remote operation of the ID OSA is supplied with the USB stick delivered with the unit in directory "\RemoteControl\Python".

7.1.4 POWERSHELL
PowerShell is a simple scripting language built into Windows Operating system. A demo script for remote operation of the ID OSA is supplied with the USB stick delivered with the unit in directory "\RemoteControl\PowerShell".

7.2 OPERATING MULTIPLE INTERFACE PORTS

This unit supports parallel usage of al remote-control ports available. Note that responses to commands issued are only returned to the according port from where the command was issued.

Commands are generally executed in order of time wise arrival to the controller and buffered into an Event queue. If a stack overflow occurs, an error is issued.

Note that there is no control exclusivity for a specific interface or user. Thus, parallel commands issued by different instances will be might lead to inconsistencies. It is therefore recommended to poll current parameter status to ensure integrity of set vs. actual parameters and query the operation complete register ("`*opc?`") to make sure all previous commands have been executed.

7.3 SYNTAX CONVENTIONS FOR COMMANDS

7.3.1 LONG AND SHORT FORM

The key words feature a long form and a short form. Either the short form or the long form can be entered, other abbreviations are not permissible.

Example: :SYStem:IPADDRess?= IPADDR?

❗ NOTE

The short form is marked by upper-case letters; the long form corresponds to the full expression. Upper-case and lower-case notation only serve the above purpose, the instrument itself accept both upper-case and lowercase letters.

❗ NOTE

All commands are case insensitive. Long and short form may not be mixed within a single command.

7.3.2 QUERY COMMANDS

Most commands serve a double function that allows either setting or executing a query on a parameter.

❗ NOTE

Query commands are terminated by a "?" character.

7.3.3 PARAMETER

Parameters must be separated from the header by a "white space". If several parameters are specified in a command they are separated by a comma ",".

7.3.4 COLON CHARACTER

A leading colon character `: ' instructs the instrument to interpret the command starting at the root (highest level) of the command tree. Since the Instrument also starts at the root each time you send it a new command, the leading colon is not required (although the instrument will accept it if you send it).

7.3.5 COMMAND TERMINATION CHARACTER

Each command must be terminated either by a ";" = $0x3B$ character or a line feed <LF> = $0x0A$ to signal completion of the command telegram to the controller. Depending on settings of the connection, the host buffer is only sent to the unit if a <LF> is used. Therefore, it is recommended using <LF> for sending commands. The response termination is always ";<LF>" = $0x3B0A$. In this documentation, the <LF> is not shown to enhance readability.

❗ NOTE

Sending two termination characters is a common mistake causing the unit to respond with "ERR 100;<LF>" since the first command is executed once the first termination character is received and the second termination

character causes the unit to interpret an empty command. Example:

Command: "`STEP 1e+9;<LF>`"

Response "`;<CR>ERR 100, unknown command; <LF>`"

The unit will execute the command but additionally receives the empty command which causes the error response.

7.3.6 ACKNOWLEDGEMENT OF EXECUTED COMMANDS

The controller will always acknowledge successful execution of commands by a ";`<LF>`" character. If the echo option is set (for details, see command list), the accordingly sent command is returned first.

7.3.7 COMMUNICATION EXAMPLE

Host sends: `*idn?;`
ID OSA response: `IDP-OSA-MPD-02, SN 17120007, F/W Ver 1.0.0(54), HW Ver 1.10;`

7.3.8 BASIC SCRIPT EXAMPLE

The following commands can be a minimal script to perform a basic set up of the unit and read the data

COMMAND	COMMENT
`STAR 1.9282e+14;`	Set Start of scan to 192.82THz
`STAR 1.9531e+14;`	Set Stop of scan to 195.31THz
`STEP 1e+10;`	Sets the Sampling interval to 10GHz. It is equal to the Resolution Bandwidth (RBW) for this value.
`SGL;`	Trigger a single Scan
`*WAI;`	The response to this is query is only given if *opc? status is 1 (=execution of `SGL` command is completed). Otherwise query of spectrum in next steps will be data stored in buffer from previous scan.
`FORM ASCII;`	Set the coding for traces responses to ASCII
`X?;`	Retrieve X Axis = Frequency vector (response is ASCII coded)
`Y?;`	Retrieve Y data = Power vector (response is ASCII coded)

7.4 USER ACCESS LEVEL

This feature allows execution of commands protected in standard access level to avoid accidental change of important parameters. See command description "`[:SYStem:]PASSword`" for details on how to enable enhanced access.

🛈 **NOTE**

User access level is granted session based. Each time, a new remote connection is made; the standard User access level 0 is set per default.

USER ACCESS LEVEL	PASSWORD
0	-
1	IDP

7.5 NOTATION OF SYNTAX FOR COMMAND DEFINITION

SYNTAX AND TYPE	DESCRIPTION
[ ]	An optional command level that can be omitted. For example [:SYStem:]IPADDRess? allows sending the command IPADDR?.
<P>	Denotes a parameter. The placeholder is replaced with the parameter value defined for the corresponding command. For example, the definition [:SYStem:]IPADDRess<WSP><P> allows setting parameters such as [:SYStem:]IPADDRess 192.168.0.1.
/<P>	Denotes an optional parameter.
:, *	A leading colon (':') instructs the instrument to interpret the command from the root (highest level) of the command tree. However, since the instrument starts at the root each time a new command is sent, the leading colon is optional. The instrument will accept it if included, but it is not required.
<WSP>	Denominates a white space character.
<INT>	Denominates an integer value.
<FLOAT>	Denominates a float value.
Read/Write (R/W)	Provides information on whether the command reads or writes data.
User Access Level (UAL)	Specifies the User Access Level required to execute the command. It can be either 0 or 1. The password for access level 1 is 'IDP'.
Storage Behavior (SB)	Indicates whether the setting set by the command is saved permanently. Possible values are 0, 1, and 2. 0: Setting is not saved permanently. Any user setting is discarded after reboot. The unit will start with factory defaults upon restart. 1: Setting is saved only after sending command SaveCurrSTATe. The unit will start with this saved setting upon restart. 2: If STArtDEFault is set to 0, this setting is saved immediately upon executing the command. The unit will start with this last setting set by user upon restart. If STArtDEFault is set to 1, the behavior is identical to SB = 0.

7.6 GENERAL SCPI COMMANDS

SYNTAX
`*IDN?`
Queries system type and software version. The remaining string is determined by the laser configuration of the actual device. It matches the part number that is printed on the label on the unit. **Response Type:** `STR f.e. ID-OSA-MPD-01, SN 25030013, F/W Ver 2.1.0(346), HW Ver`

SYNTAX
1.50 **Example:** `*IDN?`
`*OPC?`
Queries whether all pending commands have been executed. Note that this does NOT indicate whether the physical tuning of laser ports has been completed. **Response Type:** `INT {0;1}` **Example:** `*OPC?`
`*WAIt`
Unit waits to response until *opc? returns 1 and then acknowledges the command. This eliminates the need for a polling loop of *opc? on the remote side. **Example:** `*WAI`
`*RST`
Resets the controller, which will perform a warm start of the instrument. All connections and sessions will be closed. Requires user level 1. **Example:** `*RST`
`:SYStem:RESet`
Resets the controller, which will perform a warm start of the instrument. All connections and sessions will be closed. Requires user level 1. **Example:** `:SYS:RES`
`*CLS`
Clears all status and alarm registers of the unit. This command is used to clear latched alarm registers. **Example:** `*CLS`

7.7 COMMANDS ON SYSTEM LEVEL

SYNTAX	R/W	UAL	SB
`[:SYStem:]INFOrmation?`	R	0	0
Queries system type and software version. The remaining string is determined by the laser configuration of the actual device. It matches the part number that is printed on the label on the unit. This command is equivalent to the *idn? command. **Response Type:** `STR f.e. ID-OSA-MPD-01, SN 25030013, F/W Ver 2.1.0(346), HW Ver 1.50` **Example:** `INFO?`			
`[:SYStem:]ECHO?`	R	0	0
Queries the echo command's sent status. This setting applies to the current session only.			

SYNTAX	R/W	UAL	SB
Response Type: `INT {0;1}` **Example:** `ECHO?`			
`[:SYStem:]ECHO<WSP><P>`	W	0	0
Sets the echo command's sent status. This setting applies to the current session only. **Parameter Type:** `INT {0;1}` **Example:** `ECHO 0`			
`[:SYStem:]IPConfigDEFault`	W	1	0
Resets all remote interface settings to factory defaults. This change becomes effective only after a reboot of the unit. **Example:** `IPCDEF`			
`[:SYStem:]DHCP?`	R	0	0
Queries the DHCP setting for the Ethernet interface. **Response Type:** `STR {off;on}` **Example:** `DHCP?`			
`[:SYStem:]DHCP<WSP><P>`	W	1	2
Configures the DHCP setting for the Ethernet interface. **Parameter Type:** `STR {off;on}` **Example:** `DHCP off`			
`[:SYStem:]IPADDRess?`	R	0	0
Queries the IP address setting of the unit's Ethernet interface. This command is only applicable to DX and DX2 devices. **Response Type:** `xxx.xxx.xxx.xxx ; xxx = INT {0;...;255}` **Example:** `IPADDR?`			
`[:SYStem:]IPADDRess<WSP><P>`	W	1	2
Configures the IP address setting of the unit's Ethernet interface. This command is only applicable to DX and DX2 devices. **Parameter Type:** `xxx.xxx.xxx.xxx ; xxx = INT {0;...;255}` **Example:** `IPADDR 192.168.0.1`			
`[:SYStem:]NETMASK?`	R	0	0
Queries the netmask setting of the unit's Ethernet interface. This command is only applicable to DX and DX2 devices. **Response Type:** `xxx.xxx.xxx.xxx ; xxx = INT {0;...;255}` **Example:** `NETMASK?`			
`[:SYStem:]NETMASK<WSP><P>`	W	1	2
Configures the netmask setting of the unit's Ethernet interface. This command is only applicable to DX and DX2			

SYNTAX	R/W	UAL	SB
devices. **Parameter Type:** `xxx.xxx.xxx.xxx ; xxx = INT {0;...;255}` **Example:** `NETMASK 255.255.255.0`			
`[:SYStem:]GATEWAYIP?`	R	0	0
Queries the gateway IP address of the unit's Ethernet interface. This command is only applicable to DX and DX2 devices. **Response Type:** `xxx.xxx.xxx.xxx ; xxx = INT {0;...;255}` **Example:** `GATEWAYIP?`			
`[:SYStem:]GATEWAYIP<WSP><P>`	W	1	2
Configures the gateway IP address of the unit's Ethernet interface. This command is only applicable to DX and DX2 devices. **Parameter Type:** `xxx.xxx.xxx.xxx ; xxx = INT {0;...;255}` **Example:** `GATEWAYIP 192.168.0.255`			
`[:SYStem:]DNSIP?`	R	0	0
Queries the IP address of the primary DNS server for the Ethernet interface. **Response Type:** `xxx.xxx.xxx.xxx ; xxx = INT {0;...;255}` **Example:** `DNSIP?`			
`[:SYStem:]DNSIP<WSP><P>`	W	1	0
Configures the IP address of the primary DNS server for the Ethernet interface. **Parameter Type:** `xxx.xxx.xxx.xxx ; xxx = INT {0;...;255}` **Example:** `DNSIP 192.168.0.1`			
`[:SYStem:]MACADDRESS?`	R	0	0
Queries the MAC Address of the physical Ethernet interface. This command is only applicable to DX and DX2 devices. **Response Type:** `XX:XX:XX:XX:XX:XX ; XX = HEX {00;...;FF}` **Example:** `MACADDRESS?`			
`[:SYStem:]IPConfigCHanged?`	R	0	0
Check if IP configuration of the Ethernet Interface was changed but no reboot was triggered yet to activate the new settings. This can be used to determine if the actual settings differ from the settings that will be active upon restart. **Example:** `IPCCH?`			
`[:SYStem:]USBIPADDRess?`	R	0	0
Queries the IP address setting of the virtual Ethernet interface over USB. **Response Type:** `xxx.xxx.xxx.xxx ; xxx = INT {0;...;255}` **Example:** `USBIPADDR?`			
`[:SYStem:]USBNETMASK?`	R	0	0

SYNTAX	R/W	UAL	SB
Queries the netmask setting of the virtual Ethernet interface over USB. **Response Type:** `xxx.xxx.xxx.xxx ; xxx = INT {0;...;255}` **Example:** `USBNETMASK?`			
`[:SYStem:]REMOte?`	R	0	0
Queries the remote status of the device and checks for any open remote sessions via Ethernet. **Response Type:** `INT {0;1}` **Example:** `REMO?`			
`[:SYStem:]PASSword?`	R	0	0
Queries the current user level status. **Response Type:** `INT {0;1;9}` **Example:** `PASS?`			
`[:SYStem:]PASSword<WSP><P>`	W	0	0
Sets a new user level status for this session by sending a password. **Parameter Type:** `STR` **Example:** `PASS IDP`			
`[:SYStem:]SetPASSword<WSP><P>`	W	1	2
Sets a password to access the current user level. The current user level must match the level for which the password is being set. The parameter is a string defining the password. **Parameter Type:** `STR` **Example:** `SPASS IDP`			
`[:SYStem:]INTerfaceInit`	W	0	0
Resets session parameters to their defaults. Call this after opening the remote port. This command resets ECHO, PASS, FORMAT, unit:X, LINLOG and EVENT. **Example:** `INTI`			
`[:SYStem:]TIME?`	R	0	0
Queries the system time. Note that the time is stored in volatile memory only and must be set after each cold start. **Response Type:** `INT {0;1;...;2147483647}` **Example:** `TIME?`			
`[:SYStem:]TIME<WSP><P>`	W	0	0
Sets the system time. Note that the time is stored in volatile memory only and must be set after each cold start. **Parameter Type:** `INT {0;1;...;2147483647}` **Example:** `TIME 946685651`			
`[:SYStem:]ALARm?`	R	0	0
Queries the alarm status of device. See the Alarm Code Definition section for details.			

SYNTAX	R/W	UAL	SB
Response Type: `INT {0;1;...;65535}` **Example:** `ALAR?`			
`[:SYStem:]ERRor[:NEXT]?`	R	0	0
Queries data from the error queue and deletes it. **Example:** `ERR?`			
`[:SYStem:]FAN?`	R	0	0
Queries the chassis fan level as a percentage of the maximum level, if a fan is present in the chassis. **Example:** `FAN?`			
`[:SYStem:COMMunicate:]LOCKout?`	R	0	0
Checks if other sessions are allowed to execute write commands on the unit. **Response Type:** `INT {0;1}` **Example:** `LOCK?`			
`[:SYStem:COMMunicate:]LOCKout<WSP><P>`	W	1	0
Locks other sessions from performing write commands on the unit. The lock is automatically released if the active session closes. **Parameter Type:** `INT {0;1}` **Example:** `LOCK 0`			
`[:SYStem:COMMunicate:]ParameterREFresh?`	R	0	0
This query detects any changes made to the unit configuration. Each time the counter increases, it indicates a configuration change. This is useful in multi-user environments to determine if a parallel session has modified the unit's settings. **Response Type:** `INT` **Example:** `PREF?`			
`[:]ABORt`	W	0	0
Aborts all currently executing pending commands as quickly as possible. Query *OPC? to determine the status once all pending commands have been aborted. **Example:** `ABOR`			
`[:SYStem:]IDENTify<WSP><P>`	W	0	0
Enables or disables blinking on the unit, allowing identification of the unit controlled by this remote session. This is helpful for installations with multiple OSA units. **Parameter Type:** `INT {0;1}` **Example:** `IDENT 0`			
`[:SCRIPTing:]WAITMilliSeconds<WSP><P>`	W	0	0
This command causes the unit to wait for specified time until the next command in buffer is executed. This is helpful if a batch of commands is uploaded to the unit for execution. Requires Firmware Version 1.2.1 or later			

SYNTAX	R/W	UAL	SB
Parameter Type: `INT {0;1;...;60000}` **Example:** `WAITMS 100`			
`[:SYStem:]SaveCurrSTATe`	W	0	0
Saves the current settings permanently. Note: This command is queued by the device and may be executed after a response is returned. Therefore, it is recommended to verify that the command has been completed and the command queue is empty by using *OPC? or *WAI before sending any further configuration changes. **Example:** `SCSTAT`			
`[:SYStem:]DEFAULT`	W	0	0
Resets user settings, such as laser settings, to the factory default. This affects all settings, including trigger configuration, except for remote interface settings. **Example:** `DEFAULT`			

7.8 OSA SPECIFIC COMMANDS

7.8.1 OSA SENSE COMMANDS

SYNTAX	R/W	UAL	SB
`[:SENSE:SWEEP:]SGL`	W	0	0
Triggers a single sweep by combining the commands SMOD 1 and INIT:IMM. **Example:** `SGL`			
`[:SENSE:SWEEP:]RPT`	W	0	0
Triggers repeated sweeps. The last completed sweep is stored in the buffer. Use the command [:SENSe:SWEep:TIME:]INTerval/? to configure fixed-time scans. **Example:** `RPT`			
`[:SENSE:SWEEP:]AUTO`	W	0	0
Triggers a repeat sweep, equivalent to RPT. This combines the commands SMOD 3 and INIT:IMM. **Example:** `AUTO`			
`[:SENSE:SWEEP:]NUMBer?`	R	0	0
Reads the OSA scan number. Each OSA scan is assigned a consecutive identification number starting from 0 (indicating no scan has been performed yet) since the last reboot, instrument reset, or counter reset. This can be used to query if a scan has been completed by comparing the previous scan number to current. The current scan number is additionally returned when trace data is queried, allowing for synchronization. **Response Type:** `INT {0;1;...;1000000}` **Example:** `NUMB?`			
`[:SENSE:SWEEP:]NUMBer`	W	0	0

SYNTAX	R/W	UAL	SB
Resets the OSA scan number. Each OSA scan is assigned a consecutive identification number starting from 0 (indicating no scan has been performed yet) since the last reboot, instrument reset, or counter reset. This command will reset the counter to 0 or to the number sent as parameter. **Parameter Type:** `INT {0;1;...;1000000}` **Example:** `NUMB 6`			
`[:SENSe:CORRection:]LEVel:SHIFt?`	R	0	0
Queries the offset value for the power level detected in dB. This value will be internally subtracted from the measured value. Positive and negative values are possible. **Response Type:** `FLOAT {-100;...;100} [dB]` **Example:** `LEV:SHIF?`			
`[:SENSe:CORRection:]LEVel:SHIFt<WSP><P>`	W	0	0
Sets the offset value for the power level detected in dB. This value will be internally subtracted from the measured value. Positive and negative values are possible. **Parameter Type:** `FLOAT {-100;...;100} [dB]` **Example:** `LEV:SHIF 30.9`			
`[:SENSe:CORRection:]LEVel:SPECtralCALibration:Y?`	R	0	0
Queries the offset value in dB applied to the power level detected at each spectral data point with maximum resolution. This value is internally subtracted from the measured value. Both positive and negative values are allowed. A vector of 15600 values is returned. The format is determined by the [:]FORMat[:DATA]/? setting. At instrument default, all values are 0. See the spectral correction section for details. **Response Type:** `Vector of 15600 FLOAT values [dB]` **Example:** `LEV:SPECCAL:Y?`			
`[:SENSe:CORRection:]LEVel:SPECtralCALibration:Y<WSP><P>,<P>,<P>,...`	W	1	0
Sets the offset value in dB applied to the power level detected at each spectral data point with maximum resolution. This value is internally subtracted from the measured value. Both positive and negative values are allowed. Send the index of the first integer value starting at 0 followed by comma separated values in dB. For example lev:speccal:y 0,5 sets the first element of the vector to 5 and lev:speccal:y 0,5,1.5,3 sets the first vector element to 5, the second to 1.5 and the third to 3. Send a maximum of 500 values per write command. If you want to write all 15600 values, execute several write commands to stitch the full vector together. Value is only permanently saved if LEVel:SPECtralCALibration:SAVe command is executed. See the spectral correction section for details. **Parameter Type:** `INT {0;1;...;15600 - number of parameters to be set}` , up to 500 `FLOAT values [dB]` **Example:** `LEV:SPECCAL:Y 0,-4.56,-2.76`			
`[:SENSe:CORRection:]LEVel:SPECtralCALibration:ENAble?`	R	0	0
Queries whether the application of the stored spectral correction vector defined by the LEV:SPECCAL command is			

SYNTAX	R/W	UAL	SB
enabled (1) or disabled (0). See section "Spectral Correction" for details. **Response Type:** `INT {0;1}` **Example:** `LEV:SPECCAL:ENA?`			
`[:SENSe:CORRection:]LEVel:SPECtralCALibration:ENAble<WSP><P>`	W	1	0
Enables (1) or disables (0) the application of the stored spectral correction vector defined by the LEV:SPECCAL command. See section "Spectral Correction" for details. **Parameter Type:** `INT {0;1}` **Example:** `LEV:SPECCAL:ENA 0`			
`[:SENSe:CORRection:]LEVel:SPECtralCALibration:SAVe`	W	1	0
Saves the current spectral correction vector to the ID OSA for permanent usage after power cycling. See section spectral correction for details. **Example:** `LEV:SPECCAL:SAV`			
`[:SENSe:CORRection:]WAVelength:SHIFt?`	R	0	0
Queries the offset value for the wavelength in m. **Response Type:** `FLOAT [m]` **Example:** `WAV:SHIF?`			
`[:SENSe:CORRection:]WAVelength:SHIFt<WSP><P>`	W	0	0
Sets the offset value for the wavelength in m. **Parameter Type:** `FLOAT {-1e-8;...;1e-8} [m]` **Example:** `WAV:SHIF 0.9e-8`			
`[:SENSe:SWEep:]POINts?`	R	0	0
Queries the number of samples measured per sweep. **Response Type:** `INT {1;2;...;15600}` **Example:** `POIN?`			
`[:SENSe:SWEep:]POINts<WSP><P>`	W	0	0
Sets the number of samples measured per sweep. **Parameter Type:** `INT {1;2;...;15600}` **Example:** `POIN 1`			
`[:SENSe:SWEep:]POINts:AUTO?`	R	0	0
Queries whether the number of samples per sweep is determined automatically. 0 means that it isn't determined automatically. **Response Type:** `INT {0;1}` **Example:** `POIN:AUTO?`			

SYNTAX	R/W	UAL	SB
`[:SENSe:SWEep:]POINts:AUTO<WSP><P>`	W	0	0

Sets whether the number of samples per sweep is determined automatically. 0 means that it isn't determined automatically.
Parameter Type: `INT {0;1}`
Example: `POIN:AUTO 0`

SYNTAX	R/W	UAL	SB
`[:SENSe:SWEep:]MAXPOINts?`	R	0	0

Queries whether the number of samples per sweep is determined automatically. If Maxpoint is enabled, the maximum number of points (i.e., maximum resolution) is automatically set for the given START/STOP window. See the mapping section for details.
Response Type: `INT {1;2;...;15600}`
Example: `MAXPOIN?`

SYNTAX	R/W	UAL	SB
`[:SENSe:SWEep:]MAXPOINts`	W	0	0

Sets whether the number of samples per sweep is determined automatically. If Maxpoint is enabled, the maximum number of points (i.e., maximum resolution) is automatically set for the given START/STOP window. See the mapping section for details.
Example: `MAXPOIN`

SYNTAX	R/W	UAL	SB
`[:SENSe:SWEep:TIME:]INTerval?`	R	0	0

Queries the time interval in seconds between the start of one sweep and the start of the next during repeated sweeps. If the interval is shorter than the sweep duration, sweeps are repeated continuously. Set to 0 to disable this feature.
Response Type: `FLOAT {0;...;60} [s]`
Example: `INT?`

SYNTAX	R/W	UAL	SB
`[:SENSe:SWEep:TIME:]INTerval<WSP><P>`	W	0	0

Sets the time interval in seconds between the start of one sweep and the start of the next during repeated sweeps. If the interval is shorter than the sweep duration, sweeps are repeated continuously. Set to 0 to disable this feature.
Parameter Type: `FLOAT {0;...;60} [s]`
Example: `INT 8.5`

SYNTAX	R/W	UAL	SB
`[:SENSe:WAVelength:]CENTer?`	R	0	0

Queries the measurement condition center wavelength or frequency. The response is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). The sweep span will be automatically clipped if it exceeds the OSA limits.
Response Type: `FLOAT {{1.52857849840663e-06;...;1.56754226405229e-06};{1.9125e14;...;1.96125e14}}{[m];[Hz]}`
Example: `CENT?`

SYNTAX	R/W	UAL	SB
`[:SENSe:WAVelength:]CENTer<WSP><P>`	W	0	0

Sets the measurement condition center wavelength or frequency. The parameter is given in either meters (m) or

SYNTAX	R/W	UAL	SB
Hertz (Hz), depending on the base setting (wavelength or frequency). The sweep span will be automatically clipped if it exceeds the OSA limits. **Parameter Type:** `FLOAT {{1.52857849840663e-06;...;1.56754226405229e-06};{1.9125e14;...;1.96125e14}}{[m];[Hz]}` **Example:** `CENT 1.936875e14;`			
`[:SENSe:WAVelength:]SPAN?`	R	0	0
Queries the measurement condition span. The response is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). **Response Type:** `FLOAT {1e-12;...;1e13}{[m];[Hz]}` **Example:** `SPAN?`			
`[:SENSe:WAVelength:]SPAN<WSP><P>`	W	0	0
Sets the measurement condition span. The parameter is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). **Parameter Type:** `FLOAT {1e-12;...;1e13}{[m];[Hz]}` **Example:** `SPAN 4.875e12`			
`[:SENSe:WAVelength:]STARt?`	R	0	0
Queries the measurement condition start wavelength or frequency. The response is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). **Response Type:** `FLOAT {{1.52857849840663e-06;...;1.56754226405229e-06};{1.9125e14;...;1.96125e14}}{[m];[Hz]}` **Example:** `STAR?`			
`[:SENSe:WAVelength:]STARt<WSP><P>`	W	0	0
Sets the measurement condition start wavelength or frequency. The parameter is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). **Parameter Type:** `FLOAT {{1.52857849840663e-06;...;1.56754226405229e-06};{1.9125e14;...;1.96125e14}}{[m];[Hz]}` **Example:** `STAR 1.9125e14`			
`[:SENSe:WAVelength:]STOP?`	R	0	0
Queries the measurement condition stop wavelength or frequency. The response is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). **Response Type:** `FLOAT {{1.52857849840663e-06;...;1.56754226405229e-06};{1.9125e14;...;1.96125e14}}{[m];[Hz]}` **Example:** `STOP?`			
`[:SENSe:WAVelength:]STOP<WSP><P>`	W	0	0
Sets the measurement condition stop wavelength or frequency. The parameter is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). **Parameter Type:** `FLOAT {{1.52857849840663e-06;...;1.56754226405229e-06};{1.9125e14;...;1.96125e14}}{[m];[Hz]}`			

SYNTAX	R/W	UAL	SB
Example: `STOP 1.96125e14`			
`[:SENSe:WAVelength:]MINSTARt?`	R	0	0
Queries the minimal start wavelength or frequency supported by the hardware. The response is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). **Response Type:** `FLOAT{[m];[Hz]}` **Example:** `MINSTAR?`			
`[:SENSe:WAVelength:]MINSTARt`	W	0	0
Sets the minimal start wavelength or frequency supported by the hardware. It is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). **Example:** `MINSTAR`			
`[:SENSe:WAVelength:]MAXSTOP?`	R	0	0
Queries the maximal start wavelength or frequency supported by the hardware. The response is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). **Response Type:** `FLOAT{[m];[Hz]}` **Example:** `MAXSTOP?`			
`[:SENSe:WAVelength:]MAXSTOP`	W	0	0
Sets the maximal start wavelength or frequency supported by the hardware. It is given in either meters (m) or Hertz (Hz), depending on the base setting (wavelength or frequency). **Example:** `MAXSTOP`			
`[:SENSe:]INPUt?`	R	0	0
Queries which of the two optical inputs is being used. Power correction is applied automatically. 1: Standard Sensitivity 2: High Sensitivity **Response Type:** `INT {1;2}` **Example:** `INPU?`			
`[:SENSe:]INPUt<WSP><P>`	W	0	0
Sets which of the two optical inputs is being used. Power correction is applied automatically. 1: Standard Sensitivity 2: High Sensitivity Note that any write operation is always permanently saved automatically. **Parameter Type:** `INT {1;2}` **Example:** `INPU 1`			
`[:SENSe:SWEep:]STEP?`	R	0	0
Queries the sampling interval for measurements. The sampling interval (SI) is equal to the Resolution Bandwidth (RBW) if the SI is larger than RBW. If the SI is set to be smaller to RBW, the unit will oversample the available resolution. 1) Automatically setting the sampling interval for measurement (SENSe:SWEep:POINts:AUTO command) is ON, the sampling number to be measured that has been set can be queried. 2) When the function of automatically setting the sampling number to be measured (SENSe:SWEep:POINts:AUTO command) is ON, this command will be automatically set to OFF. 3) When the sampling interval for measurement is set using this command, the sampling intervals for measurements (SENSe:SWEep:POINts) will be automatically set. STEP-Value			

SYNTAX	R/W	UAL	SB
can only be set as a frequency (not wavelength)! **Response Type:** `FLOAT {3.125e8;...;4.8746875e12} [Hz]` **Example:** `STEP?`			
`[:SENSe:SWEep:]STEP<WSP><P>`	W	0	0
Sets the sampling interval for measurements. The sampling interval (SI) is equal to the Resolution Bandwidth (RBW) if the SI is larger than RBW. If the SI is set to be smaller to RBW, the unit will oversample the available resolution. 1) Automatically setting the sampling interval for measurement (SENSe:SWEep:POINts:AUTO command) is ON, the sampling number to be measured that has been set can be queried. 2) When the function of automatically setting the sampling number to be measured (SENSe:SWEep:POINts:AUTO command) is ON, this command will be automatically set to OFF. 3) When the sampling interval for measurement is set using this command, the sampling intervals for measurements (SENSe:SWEep:POINts) will be automatically set. STEP-Value can only be set as a frequency (not wavelength)! **Parameter Type:** `FLOAT {3.125e8;...;4.8746875e12} [Hz]` **Example:** `STEP 3.125e8`			
`[:SENSe:SWEep:]STEP:FREQuency?`	R	0	0
Same as STEP command, except that this command will always query frequency values (regardless of UNIT:X mode). **Response Type:** `FLOAT {3.125e8;...;4.8746875e12} [Hz]` **Example:** `STEP:FREQ?`			
`[:SENSe:SWEep:]STEP:FREQuency<WSP><P>`	W	0	0
Same as STEP command, except that this command will always set frequency values (regardless of UNIT:X mode). The user can use this mode to set the STEP value when wavelength mode is selected through UNIT:X. **Parameter Type:** `FLOAT {3.125e8;...;4.8746875e12} [Hz]` **Example:** `STEP:FREQ 3.125e8`			
`[:SENSe:SWEep:]MINSTEP?`	R	0	0
Queries the minimum sampling interval in wavelength (m) or frequency (Hz) supported by the hardware. **Response Type:** `FLOAT{[m];[Hz]}` **Example:** `MINSTEP?`			
`[:SENSe:SWEep:]MINSTEP`	W	0	0
Sets the minimum sampling interval in wavelength (m) or frequency (Hz) supported by the hardware. The minimal possible step size is set to the unit. **Example:** `MINSTEP`			
`[:SENSe:SWEep:]MAXSTEP?`	R	0	0
Queries the maximum sampling interval in wavelength (m) or frequency (Hz) supported by the hardware. **Response Type:** `FLOAT{[m];[Hz]}` **Example:** `MAXSTEP?`			
`[:SENSe:SWEep:]MAXSTEP`	W	0	0

SYNTAX	R/W	UAL	SB
Sets the maximum sampling interval in wavelength (m) or frequency (Hz) supported by the hardware. The maximal possible step size is set to the unit. **Example:** `MAXSTEP`			
`[:SENSe:SWEep:]TRIggerPOLarity?`	R	0	0
Queries the trigger signal polarity of input or output. Input (IN): 1 is the default setting. It induces a high-level signal to trigger a new scan if enabled. 0 will cause a low-level signal to trigger a new measurement trigger. Output (OUT): 1 is the default setting. It leads to the generation of a high-level signal if a scan is complete and available to read. It will return low once the next scan is started. The signal is low if no scan has been performed yet. The 0 setting will result in the output being low if a scan is completed and high again once a new scan is triggered. In this case the signal is high if no scan has been performed yet. **Response Type:** `{IN;OUT},{0;1}` **Example:** `TRIPOL?`			
`[:SENSe:SWEep:]TRIggerPOLarity<WSP><P>` `<P>`	W	0	0
Sets the trigger signal polarity of input or output. Input (IN): 1 is the default setting. It induces a high-level signal to trigger a new scan if enabled. 0 will cause a low-level signal to trigger a new measurement trigger. Output (OUT): 1 is the default setting. It leads to the generation of a high-level signal if a scan is complete and available to read. It will return low once the next scan is started. The signal is low if no scan has been performed yet. The 0 setting will result in the output being low if a scan is completed and high again once a new scan is triggered. In this case the signal is high if no scan has been performed yet. **Parameter Type:** `{IN;OUT},{0;1}` **Example:** `TRIPOL IN,0`			
`[:SENSe:CORRection:]AVERage:COUNt?`	R	0	0
Queries the number of moving averages to be taken. Averaging begins with the first scan and accumulates over subsequent scans, up to the specified limit. A counter tracks how many scans have been included in the current average. Each added scan increments the scan number variable. In single scan mode, one scan is performed and included in the moving average calculation. The maximum moving average window is 100 scans. **Response Type:** `INT {1;2;...;100}` **Example:** `AVER:COUN?`			
`[:SENSe:CORRection:]AVERage:COUNt<WSP>` `<P>`	W	0	0
Sets the number of moving averages to be taken. Averaging begins with the first scan and accumulates over subsequent scans, up to the specified limit. A counter tracks how many scans have been included in the current average. Each added scan increments the scan number variable. In single scan mode, one scan is performed and included in the moving average calculation. The maximum moving average window is 100 scans. **Parameter Type:** `INT {1;2;...;100}` **Example:** `AVER:COUN 20`			
`[:SENSe:CORRection:]AVERage:CURrentCOUNt?`	R	0	0
Queries the number of moving average scans currently taken into moving average calculation.			

SYNTAX	R/W	UAL	SB
Response Type: INT **Example:** AVER:CURCOUN?			
[:SENSe:CORRection:]AVERage:SCANCOUNt?	R	0	0
Queries the number of averages taken per scan. The maximum number of scans is 1000. Internally, the unit takes the specified number of scans, calculates the average and reports a single new scan to the user. The scan time per scan will then be a multiplier of the value specified here. A separate command is available to track the progress of the scan acquisition. **Response Type:** INT {1;2;...;1000} **Example:** AVER:SCANCOUN?			
[:SENSe:CORRection:]AVERage:SCANCOUNt< WSP><P>	W	0	0
Sets the number of averages taken per scan. The maximum number of scans is 1000. Internally, the unit takes the specified number of scans, calculates the average and reports a single new scan to the user. The scan time per scan will then be a multiplier of the value specified here. A separate command is available to track the progress of the scan acquisition. **Parameter Type:** INT {1;2;...;1000} **Example:** AVER:SCANCOUN 100			
[:SENSe:CORRection:]AVERage:CURrentSCA NCOUNt?	R	0	0
Queries the number of scans taken in the unit to derive the next scan to be provided to the user. **Response Type:** INT **Example:** AVER:CURSCANCOUN?			
[:SENSe:SWEep:]TRIggerDELay?	R	0	0
Queries the trigger delay for Scan Trigger IN. Setting this delay causes the OSA to postpone the start of the scan relative to the trigger event by the specified number of milliseconds. A value of 0 will start the scan immediately, without any delay. **Response Type:** INT [ms] **Example:** TRIDEL?			
[:SENSe:SWEep:]TRIggerDELay<WSP><P>	W	0	0
Queries the trigger delay for Scan Trigger IN. Setting this delay causes the OSA to postpone the start of the scan relative to the trigger event by the specified number of milliseconds. A value of 0 will start the scan immediately, without any delay. **Parameter Type:** INT [ms] **Example:** TRIDEL 5			

7.8.2 OSA TRACE COMMANDS

SYNTAX	R/W	UAL	SB
`:TRACe[:DATA]:SNUMber?`	R	0	0
Queries the number of data points sampled. **Response Type:** `INT {1;2;...;15600}` **Example:** `:TRAC:SNUM?`			
`:TRACe[:DATA]:SNUMber<WSP><P>`	W	0	0
Sets the number of data points sampled. **Parameter Type:** `INT {1;2;...;15600}` **Example:** `:TRAC:SNUM 1`			
`[:]TRACe[:DATA]:X?`	R	0	0
Queries the wavelength axis data. The data is output in units of wavelength [m], regardless of whether the current display mode is set to wavelength or frequency. The output is equivalent to what is shown in the GUI. If any adjustments—such as offset parameters—are configured, these are reflected in the data retrieved by this command. Please note that the format of the data is controlled by the [:FORMat[:DATA]]? command. We recommend using binary mode for efficient data transfer or using the XY? command, which provides both X and Y values in the most efficient manner. The number of output data points can be determined by executing :TRACe[:DATA]:SNUMber?, then adding 1 to account for the scan number. The data format is defined by the :FORMat[:DATA]? command. **Response Type:** `Vector of FLOAT values [m]` **Example:** `TRAC:X?`			
`X?`	R	0	0
Queries the wavelength axis data. The data is output in units of wavelength [m], regardless of whether the current display mode is set to wavelength or frequency. The output is equivalent to what is shown in the GUI. If any adjustments—such as offset parameters—are configured, these are reflected in the data retrieved by this command. Please note that the format of the data is controlled by the [:FORMat[:DATA]]? command. We recommend using binary mode for efficient data transfer or using the XY? command, which provides both X and Y values in the most efficient manner. The number of output data points can be determined by executing :TRACe[:DATA]:SNUMber?, then adding 1 to account for the scan number. The data format is defined by the :FORMat[:DATA]? command. **Response Type:** `Vector of FLOAT values [m]` **Example:** `X?`			
`[:]TRACe[:DATA]:XAUTO?`	R	0	0
Same as TRAC:X?, but output unit depends on UNIT:X setting (wavelength or frequency). **Response Type:** `Vector of FLOAT values{[m];[Hz]}` **Example:** `TRAC:XAUTO?`			
`XAUTO?`	R	0	0
Same as TRAC:X?, but output unit depends on UNIT:X setting (wavelength or frequency). **Response Type:** `Vector of FLOAT values{[m];[Hz]}`			

SYNTAX	R/W	UAL	SB
Example: `XAUTO?`			
`[:]TRACe[:DATA]:Y?`	R	0	0
Queries the power level data. The data is output in order of wavelength, from shortest to longest, regardless of whether the current mode is set to wavelength or frequency. The output corresponds to what is displayed in the GUI. If averaging or other adjustments—such as offset parameters—are applied, these are reflected in the data retrieved by this command. Please note that the format of the data is determined by the `[:FORMat[:DATA]]? <P>` command. For optimal performance, we recommend using binary mode to retrieve the data or using the XY? command, which efficiently provides both X (wavelength) and Y (power level) values. When the level scale is set to LOG, the data is returned in logarithmic values. Refer to the :TRACe[:DATA]:LINLog? command for more information. **Response Type:** `Vector of FLOAT values` **Example:** `TRAC:Y?`			
`Y?`	R	0	0
Queries the power level data. The data is output in order of wavelength, from shortest to longest, regardless of whether the current mode is set to wavelength or frequency. The output corresponds to what is displayed in the GUI. If averaging or other adjustments—such as offset parameters—are applied, these are reflected in the data retrieved by this command. Please note that the format of the data is determined by the `[:FORMat[:DATA]]? <P>` command. For optimal performance, we recommend using binary mode to retrieve the data or using the XY? command, which efficiently provides both X (wavelength) and Y (power level) values. When the level scale is set to LOG, the data is returned in logarithmic values. Refer to the :TRACe[:DATA]:LINLog? command for more information. **Response Type:** `Vector of FLOAT values` **Example:** `Y?`			
`[:]TRACe[:DATA]:LINLog?`	R	0	0
Queries the output format of the power level data trace. If set to LOG, the output is in [dBm]. If set to LIN, the output is in [mW]. **Response Type:** `{LIN;LOG}{[mW];[dBm]}` **Example:** `TRAC:LINL?`			
`[:]TRACe[:DATA]:LINLog<WSP><P>`	W	0	0
Sets the output format of the power level data trace. If set to LOG, the output is in [dBm]. If set to LIN, the output is in [mW]. **Parameter Type:** `{LIN;LOG}{[mW];[dBm]}` **Example:** `TRAC:LINL LIN`			
`[:]TRACe[:DATA]:XY?`	R	0	0
Queries both the wavelength/frequency axis and the corresponding power level data in a single command. The output is always sorted in ascending order values of the X vector. The order in the data is X1,Y1,X2,Y2, …, Xn,Yn. Output is always in REAL32 binary form. Independent of the LINLog setting, the output is always in LOG format (dB values). The command UNIT:X can be used to switch between wavelength and frequency mode. **Example:** `TRAC:XY?`			

SYNTAX	R/W	UAL	SB
XY?	R	0	0

Queries both the wavelength/frequency axis and the corresponding power level data in a single command. The output is always sorted in ascending order values of the X vector. The order in the data is X1,Y1,X2,Y2, …, Xn,Yn. Output is always in REAL32 binary form. Independent of the LINLog setting, the output is always in LOG format (dB values). The command UNIT:X can be used to switch between wavelength and frequency mode.
Example: `XY?`

SYNTAX	R/W	UAL	SB
[:TRACe:DATA:]POWer?	R	0	0

Reads the integrated power of the scan.
Response Type: `FLOAT`
Example: `POW?`

SYNTAX	R/W	UAL	SB
[:TRACe:DATA:]PeaKHoLD?	R	0	0

Queries the Peak Hold measurement. 0: feature is disabled ; 1: Max Peak Hold is enabled ; 2: Min Peak Hold is enabled
Response Type: `{0;1;2}`
Example: `PKHLD?`

SYNTAX	R/W	UAL	SB
[:TRACe:DATA:]PeaKHoLD<WSP><P>	W	0	0

Sets the Peak Hold measurement. 0: Disables feature ; 1: Enables Max Peak Hold ; 2: Enables Min Peak Hold
Parameter Type: `{0;1;2}`
Example: `PKHLD 0`

7.8.3 OSA MISCELLANEOUS COMMANDS

SYNTAX	R/W	UAL	SB
[:INITiate:]ReaDY?	R	0	0

Reports whether the module is ready (1) for scanning or not (0).
Response Type: `INT {0;1}`
Example: `RDY?`

SYNTAX	R/W	UAL	SB
[:]FORMat[:DATA]?	R	0	0

Queries the format used for data transfer; When the format is set to REAL (binary) using this command, the output data of the following commands are produced in the REAL format: :TRACe[:DATA]:X? ; :TRACe[:DATA]:Y? The factory default is ASCII mode. However, we do not recommend to use this mode for remote scripts using "continuous" mode as this mode produces up to 31200 data points per second (15600 data points per scan at 2 scans per second) which may cause timing issues due to high processing load and glitches in assigning the trace to the according scan. Additionally, parallel sessions such as a Web Interface connection, may experience a significant slowdown of response time. We recommend to use either the binary mode or to use the [:]TRACe[:DATA]:XY? command. The default mode can be changed using the SaveCurrSTATe command. When the *RST command is executed, the format is reset to the default mode. To change the

SYNTAX	R/W	UAL	SB
default mode, set the desired mode and execute SaveCurrSTATe. Be aware that all settings stored by SaveCurrSTATe are saved as well. The ASCII format outputs a list of numerics, each of which is delimited by a comma (,). Example: 12345,12345,.... By default, the REAL format outputs data in fixed length blocks of 64 bits, floating-point binary numerics. If REAL,32 is specified in the parameter, data is output in the 32-bit, floating-point binary form. The fixed length block is defined by IEEE 488.2 and consists of "#" (ASCII), one numeric (ASCII) indicating the number of bytes that specifies the length after #, length designation (ASCII), and binary data of a specified length in this order. Binary data consists of a floating-point data string of 8 bytes (64 bits) or 4 bytes (32 bits). Floating-point data consists of lower order bytes to higher-order bytes. E.g.: #18 [eight <byte data>] #280[80 <byte data>] #48008[8008 <byte data>] For data output in the 32-bit floating-point binary form, cancellation of significant digits is more likely to occur in comparison with transfer of data in the 64-bit, floating-point binary form. **Response Type:** `{REAL,32;REAL,64;ASCII}` **Example:** `FORM?`			
`[:]FORMat[:DATA]<WSP><P>/<P>`	W	0	0
Sets the format used for data transfer; When the format is set to REAL (binary) using this command, the output data of the following commands are produced in the REAL format: :TRACe[:DATA]:X? ; :TRACe[:DATA]:Y? The factory default is ASCII mode. However, we do not recommend to use this mode for remote scripts using "continuous" mode as this mode produces up to 31200 data points per second (15600 data points per scan at 2 scans per second) which may cause timing issues due to high processing load and glitches in assigning the trace to the according scan. Additionally, parallel sessions such as a Web Interface connection, may experience a significant slowdown of response time. We recommend to use either the binary mode or to use the [:]TRACe[:DATA]:XY? command. The default mode can be changed using the SaveCurrSTATe command. When the *RST command is executed, the format is reset to the default mode. To change the default mode, set the desired mode and execute SaveCurrSTATe. Be aware that all settings stored by SaveCurrSTATe are saved as well. The ASCII format outputs a list of numerics, each of which is delimited by a comma (,). Example: 12345,12345,.... By default, the REAL format outputs data in fixed length blocks of 64 bits, floating-point binary numerics. If REAL,32 is specified in the parameter, data is output in the 32-bit, floating-point binary form. The fixed length block is defined by IEEE 488.2 and consists of "#" (ASCII), one numeric (ASCII) indicating the number of bytes that specifies the length after #, length designation (ASCII), and binary data of a specified length in this order. Binary data consists of a floating-point data string of 8 bytes (64 bits) or 4 bytes (32 bits). Floating-point data consists of lower order bytes to higher-order bytes. E.g.: #18 [eight <byte data>] #280[80 <byte data>] #48008[8008 <byte data>] For data output in the 32-bit floating-point binary form, cancellation of significant digits is more likely to occur in comparison with transfer of data in the 64-bit, floating-point binary form. **Parameter Type:** `{REAL,32;REAL,64;ASCII}` **Example:** `FORM REAL,32`			
`[:INITiate:]SMODe?`	R	0	0
Queries the sweep mode. (SINGle = SINGLE sweep mode ; REPeat = REPEAT sweep mode ; AUTO = AUTO sweep mode) Note that the response to command is given immediately. In Single mode, query *OPC? to wait for scan to finish and update data readout buffer before reading result to avoid reading data from previous scan. In repeat mode, use numb? query to monitor updates of scan before reading result from buffer. AUTO is implemented only due to compatibility reasons. It is equivalent to the REPEAT mode. For more information about Sweep Modes, see the Sweep Modes section in this manual. **Response Type:** `{{1;2;3};{SINGle;REPeat;AUTO}}` **Example:** `SMOD?`			

SYNTAX	R/W	UAL	SB
`[:INITiate:]SMODe<WSP><P>`	W	0	0
Sets the sweep mode. (SINGle = SINGLE sweep mode ; REPeat = REPEAT sweep mode; AUTO = AUTO sweep mode) Note that the response to command is given immediately. In Single mode, query *OPC? to wait for scan to finish and update data readout buffer before reading result to avoid reading data from previous scan. In repeat mode, use numb? query to monitor updates of scan before reading result from buffer. AUTO is implemented only due to compatibility reasons. It is equivalent to the REPEAT mode. For more information about Sweep Modes, see the Sweep Modes section in this manual. **Parameter Type:** `{{1;2;3};{SINGle;REPeat;AUTO}}` **Example:** `SMOD 1`			
`[:]INITiate[:IMMediate]`	W	0	0
Starts a sweep. The sweep mode (AUTO, SINGLE, or REPEAT) is configured using the :INITiate:SMODe command. In SINGLE mode, use the *OPC? query to wait for the scan to complete and ensure the data readout buffer is updated before reading the result. This prevents retrieving data from a previous scan. In REPEAT mode, use the NUMB? query to monitor scan updates before reading data from the buffer. **Example:** `INIT`			
`*TRG`	W	0	0
Starts a sweep. The sweep mode (AUTO, SINGLE, or REPEAT) is configured using the :INITiate:SMODe command. In SINGLE mode, use the *OPC? query to wait for the scan to complete and ensure the data readout buffer is updated before reading the result. This prevents retrieving data from a previous scan. In REPEAT mode, use the NUMB? query to monitor scan updates before reading data from the buffer. **Example:** `*TRG`			
`[:]UNIT:X?`	R	0	0
Queries the units for the X-axis. The unit can be specified using either numeric values {0, 1} or text values {WAV, FREQ}. Response options: 0 = Wavelength ; 1 = Frequency **Response Type:** `{{0;1};{WAV;FREQ}}` **Example:** `UNIT:X?`			
`[:]UNIT:X<WSP><P>`	W	0	0
Sets the units for the X-axis. The unit can be specified using either numeric values {0, 1} or text values {WAV, FREQ}. Response options: 0 = Wavelength ; 1 = Frequency **Parameter Type:** `{{0;1};{WAV;FREQ}}` **Example:** `UNIT:X 0`			

7.8.4 OSA CALCULATION COMMANDS

SYNTAX	R/W	UAL	SB
`:CALCulate:CATegory?`	R	0	0
Queries the analysis category. This command was introduced due to compatibility reasons.			

SYNTAX	R/W	UAL	SB
Response Type: `STR` **Example:** `:CALC:CAT?`			
`:CALCulate:CATegory<WSP><P>`	W	0	0
Sets the analysis category. This command was introduced due to compatibility reasons. **Parameter Type:** `STR` **Example:** `:CALC:CAT WDM`			
`:CALCulate:DATA?`	R	0	0
Queries the result of the analysis. **Response Type:** `<channel number>,<center wavelength>,<peak level>,<snr>,<channel number> ... [-,Hz,dBm,dB]` **Example:** `:CALC:DATA?`			
`:CALCulate:DATA:CPOWers?`	R	0	0
Queries the power level data of the WDM measurement. **Response Type:** `CSV Vector containing the channel powers. [dBm]` **Example:** `:CALC:DATA:CPOW?`			
`:CALCulate:DATA:CSNR?`	R	0	0
Queries the OSNR power level data of the WDM measurement. **Response Type:** `CSV Vector containing the OSNR channel powers [dB]` **Example:** `:CALC:DATA:CSNR?`			
`:CALCulate:DATA:NCHannels?`	R	0	0
Queries the number of channels detected by the WDM peak function. **Response Type:** `INT` **Example:** `:CALC:DATA:NCH?`			
`:CALCulate:DATA:CWAVelengths?`	R	0	0
Queries the wavelength of peaks found in the WDM peak analysis. The response is either in frequency or wavelength depending on the UNIT:X setting. **Response Type:** `FLOAT{[m];[Hz]}` **Example:** `:CALC:DATA:CWAV?`			
`:CALCulate:PARameter[:CATegory]:WDM:TH ?`	R	0	0
Queries the threshold at which a peak is detected from the lowest measured value in [dB]. **Response Type:** `FLOAT {0;...;80} [dB]` **Example:** `:CALC:PAR:WDM:TH?`			
`:CALCulate:PARameter[:CATegory]:WDM:TH <WSP><P>`	W	0	0

SYNTAX	R/W	UAL	SB
Sets the threshold at which a peak is detected from the lowest measured value in [dB]. **Parameter Type:** `FLOAT {0;...;80} [dB]` **Example:** `:CALC:PAR:WDM:TH 0.8`			
`:CALCulate:PARameter[:CATegory]:WDM:MD` `IFf?`	R	0	0
Queries the Mode difference between peaks in [dB]. A drop in power between adjacent peaks is required. **Response Type:** `FLOAT {0;...;80} [dB]` **Example:** `:CALC:PAR:WDM:MDIF?`			
`:CALCulate:PARameter[:CATegory]:WDM:MD` `IFf<WSP><P>`	W	0	0
Sets the Mode difference between peaks in [dB]. A drop in power between adjacent peaks is required. **Parameter Type:** `FLOAT {0;...;80} [dB]` **Example:** `:CALC:PAR:WDM:MDIF 10`			
`:CALCulate:PARameter[:CATegory]:WDM:MA` `Rea?`	R	0	0
Queries the Channel Mask Area around each detected peak, next to which the noise level is measured or the mask peak power is calculated, in case the peak power integration method is used. Note that the unit must be specified using the UNIT:X setting. **Response Type:** `FLOAT {{0;3.89823354659521e-8};{0;2.4375001088e+12}}{[m];[Hz]}` **Example:** `:CALC:PAR:WDM:MAR?`			
`:CALCulate:PARameter[:CATegory]:WDM:MA` `Rea<WSP><P>`	W	0	0
Sets the Channel Mask Area around each detected peak, next to which the noise level is measured or the mask peak power is calculated, in case the peak power integration method is used. Note that the unit must be specified using the UNIT:X setting. **Parameter Type:** `FLOAT {{0;3.89823354659521e-8};{0;2.4375001088e+12}}{[m];[Hz]}` **Example:** `:CALC:PAR:WDM:MAR 1e11`			
`:CALCulate:PARameter[:CATegory]:WDM:Mi` `nDIST?`	R	0	0
Queries the minimum horizontal distance between a detected peak and the next peak. Note that the unit needs to be given by unit UNIT:X setting. **Response Type:** `FLOAT{[m];[Hz]}` **Example:** `:CALC:PAR:WDM:MDIST?`			
`:CALCulate:PARameter[:CATegory]:WDM:Mi` `nDIST<WSP><P>`	W	0	0
Sets the minimum horizontal distance between a detected peak and the next peak. Note that the unit needs to be given by unit UNIT:X setting. **Parameter Type:** `FLOAT{[m];[Hz]}`			

SYNTAX	R/W	UAL	SB
Example: `:CALC:PAR:WDM:MDIST 1e-6`			
`:CALCulate:PARameter[:CATegory]:WDM:POWerINTegrate?`	R	0	0
Queries whether peak power detection integrates power across the window (independent of the RBW setting), or reports peak power based on the RBW setting. Allowed values: 0 = peak, 1 = integrate. **Response Type:** `{0;1}` **Example:** `:CALC:PAR:WDM:POWINT?`			
`:CALCulate:PARameter[:CATegory]:WDM:POWerINTegrate<WSP><P>`	W	0	0
Sets whether peak power detection integrates power across the window (independent of the RBW setting), or reports peak power based on the RBW setting. Allowed values: 0 = peak, 1 = integrate. **Parameter Type:** `{0;1}` **Example:** `:CALC:PAR:WDM:POWINT 0`			

7.9 SCPI CODE ERROR DEFINITION

ERROR #	DESCRIPTION
100	Invalid SCPI Command: i.e. wrong parameter, parameter out of range or device is incompatible
102	Invalid SCPI Command due to an illegal parameter.
104	Occurs if a command can't be executed because the laser cards haven't been powered up yet.
201	Occurs if the SCPI authentication level (= user access level / UAL) is insufficient for the command (i.e., the required 'pass xxx' is missing).
202	Occurs when the internal command queue overflows.
207	Occurs if 'Lock' is set using the LOCK command.
250	Occurs when no scan has been performed yet, but data is being requested.

7.10   ALARM CODE DEFINITION

The following table details potential alarms raised by the unit. The alarm status can be queried using the "alarm?" command. The response is an ASCII coded unsigned 16-bit integer number, little endian. Convert this number into binary format to retrieve the alarm status. Each "1" bit represents an active alarm.

The most common responses to "`alarm?`" are:

0: no alarm

1: Optics not ready

2: Optical Power too high

Use the SCPI Command "`*cls`" to clear all latched alarms.

BIT #	ALARM	CONDITION/DESCRIPTION
0	Init; Optics not ready	After cold start, the optics requires about 45 seconds to become operational. The alarm is automatically cleared once operational.
1	Optical Input Power too high	The optical power applied to the unit input is too high. The measurement may be still ok but no validity is given.
2	Optics failure	Pending failure in the detection optics. Contact ID Photonics for guidance.
3	Optics failure (latched)	A failure in the detection optics was detected.
4	System failure	Pending failure of the unit. Contact ID Photonics for guidance.
5	System failure (latched)	A failure of the unit was detected.
6	Reserved	
7	Internal	
8	Reserved	
9	Reserved	
10	Internal	
11	Internal	
12	Internal	
13	Internal	
14	Internal	
15	Internal	

IDPHOTONICS GMBH

SHAPING LIGHT.

HELPING ENGINEERS AND SCIENTISTS IN ADVANCING HOW THE WORLD COMMUNICATES, SENSES AND CONNECTS

TUNABLE LASER

TRANSMITTER SOLUTIONS
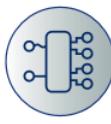
RECEIVER SOLUTIONS

SPECTRAL ANALYSIS
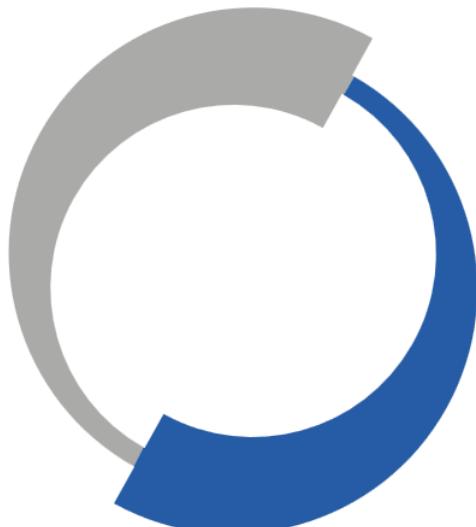
Anton-Bruckner-Strasse 6
85579 Neubiberg
GERMANY
Tel: +49-89-201 899 16
info@id-photonics.com
id-photonics.com