# VHFLI
# User Manual

## 200 MHz Lock-in Amplifier

Zurich
Instruments

# VHFLI User Manual

Zurich Instruments AG

Revision 26.01

The contents of this document are provided by Zurich Instruments AG (ZI), "as is". ZI makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice.

# Table of Contents

# CE Declaration of Conformity

The manufacturer

Zurich Instruments
Technoparkstrasse 1
8005 Zurich
Switzerland

declares that the product

VHFLI Very High Frequency Lock-in Amplifier

is in conformity with the provisions of the relevant Directives and Regulations of the Council of the European Union:

| Directive / Regulation | Conformity proven by compliance with the standards |
|---|---|
| 2014/30/EU<br>(Electromagnetic compatibility [EMC]) | EN 61326-1:2020,<br>EN 55011:2016,<br>EN 55011:2016/A1:2017,<br>EN 55011:2016/A11:2020<br>(Group 1, Class A and B equipment) |
| 2014/35/EU<br>(Low voltage equipment [LVD]) | EN 61010-1:2010,<br>EN 61010-1:2010/A1:2019,<br>EN 61010-1:2010/A1:2019/AC:2019-04 |
| 2011/65/EU, as amended by 2015/863 and 2017/2102<br>(Restriction of the use of certain hazardous substances [RoHS]) | EN IEC 63000:2018 |
| (EC) 1907/2006<br>(Registration, Evaluation, Authorisation, and Restrictions of Chemicals [REACH]) | - |

Zurich, October 20th, 2022

Flavio Heer

Flavio Heer, CTO

# UKCA Declaration of Conformity

The manufacturer

Zurich Instruments
Technoparkstrasse 1
8005 Zurich
Switzerland

declares that the product

VHFLI Very High Frequency Lock-in Amplifier

is in conformity with the provisions of the relevant UK Statutory Instruments:

| Statutory Instruments | Conformity proven by compliance with the standards |
|---|---|
| S.I. 2016/1091 (Electromagnetic Compatibility Regulations) | EN 61326-1:2020,<br>EN 55011:2016,<br>EN 55011:2016/A1:2017,<br>EN 55011:2016/A11:2020<br>(Group 1, Class A and B equipment) |
| S.I. 2016/1101 (Electrical Equipment (Safety) Regulations) | EN 61010-1:2010,<br>EN 61010-1:2010/A1:2019,<br>EN 61010-1:2010/A1:2019/AC:2019-04 |
| S.I. 2012/3032 (Restriction of the Use of Certain Hazardous Substances Regulations) | EN IEC 63000:2018 |

Zurich, October 20th, 2022

Flavio Heer

Flavio Heer, CTO

# 1. Change Log

### Info

A complete summary of all changes can be found in the LabOne Release Notes. This page only lists changes not present in the LabOne Release Notes.

## 1.1. Release 26.01

**Release date:** 31-January-2026

See Release Notes 26.01 for a detailed list of all changes.

# 2. Getting Started

This first chapter guides you through the initial set-up of your VHFLI Instrument in order to make your first measurements.

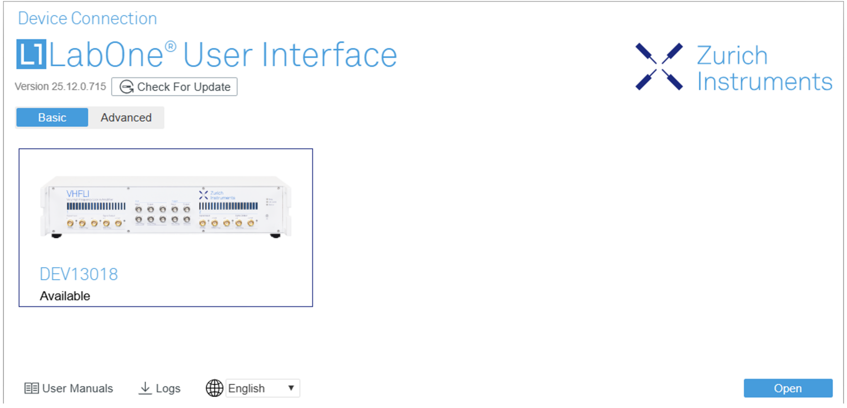
Please refer to:

- Section 2.1, "Quick Start Guide", on page 1 for a Quick Start Guide for the impatient.
- Section 2.2, "Inspect the Package Contents", on page 2 for inspecting the package content and accessories.
- Section 2.3, "Handling and Safety Instructions", on page 3 for a list of essential handling and safety instructions.
- Section 2.4, "Software Installation", on page 5 - Section 2.6, "Software Update", on page 28 for help connecting to the VHFLI Instrument with the LabOne software.
- Section 2.7, "Troubleshooting", on page 30 for a handy list of troubleshooting guidelines.

## 2.1. Quick Start Guide

This page addresses all the people who have been impatiently awaiting their new gem to arrive and want to see it up and running quickly. Please proceed with the following steps:

1. Section 2.2, "Inspect the package contents", on page 2 . Besides the Instrument there should be a country-specific power cable, a USB cable and an Ethernet cable.
2. Check Section 2.3, "Handling and Safety Instructions", on page 3 for the Handling and Safety Instructions.
3. Download and install the latest LabOne software from the Zurich Instruments Download Center.
4. Choose the download file that suits your computer (e.g. Windows with 64-bit addressing). For more detailed information see Section 2.4, "Software Installation", on page 5 .
5. Connect the instrument to the power outlet and turn it on.
6. There are two options to connect to the instrument:
   - Connect the instrument to a switch in the local area network using the port labeled "LAN 1GbE" and an Ethernet cable. To properly setup the connection follow the instructions on Subsection 2.5.13, "Connecting to the instrument - 1GbE", on page 24 .
   - Connect the instrument to the computer through USB using the port labeled "USB-C" and a USB 3.0 cable. To properly setup the connection follow the instructions on Subsection 2.5.12, "Connecting to the instrument - USB", on page 23 .
7. Start the LabOne User Interface from the Windows Start Menu. The default web browser will open and display your instrument in a start screen as shown below. Use Chrome, Edge, Firefox, or Opera for best user experience.



Version 25.12.0.715

DEV13018

Available

8. The LabOne User Interface start-up screen will appear. Click the **Open** button on the lower right of the page. The default configuration will be loaded and the first signals can be generated. If the user interface does not start up successfully, please refer to Section 2.5, "Connecting to the Instrument", on page 14 .

If any problems occur while setting up the instrument and software, please see Section 2.7, "Troubleshooting", on page 30 at the end of this chapter for troubleshooting.

When connecting cables to the instrument's SMA ports on the back panel, use a torque wrench specified for brass core SMA (4 in-lbs, 0.5 Nm). Using a standard SMA torque wrench (8 in-lbs) or a wrench without torque limit can damage the connectors.

After you have finished using the instrument, it is recommended to shut it down using the soft power button on the front panel of the instrument instrument or by clicking on the ⏻ button at the bottom left of the user interface screen before turning off the power switch on the back panel of the instrument.

Once the Instrument is up and running we recommend going through some of the tutorials given in Chapter 4, "Tutorials", on page 38 .

For specific application know-how, the blog section of the Zurich Instruments website will serve as a valuable resource that is constantly updated and expanded.

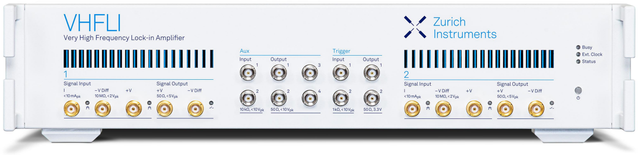
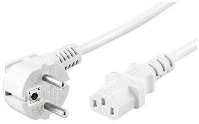
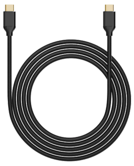
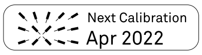
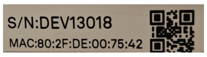
## 2.2. Inspect the Package Contents

If the shipping container appears to be damaged, keep the container until you have inspected the contents of the shipment and have performed basic functional tests.

Please verify the following:

- You have received 1 Zurich Instruments VHFLI Instrument
- You have received 1 power cord with a power plug suited to your country
- You have received 1 USB 3.0 cable and/or 1 LAN cable (category 5/6 required)
- The "Next Calibration" sticker on the rear panel of the instrument indicates a date approximately 2 years in the future → Zurich Instruments recommends calibration intervals of 2 years
- The MAC address of the instrument is displayed on a sticker on the back panel

Table 2.1: Package contents for the VHFLI

| | |
|---|---|
|  VHFLI<br>Very High Frequency Lock-in Amplifier<br>Zurich Instruments | VHFLI instrument |
|  | the power cord (e.g. EU norm) |
|  | the USB-C cable |
| | the power inlet, with power switch |
| | the LAN / Ethernet cable (category 5/6 required) |
| <br>Next Calibration<br>Apr 2022 | the "Next Calibration" sticker on the back panel of the instrument |
| <br>S/N:DEV13018<br>MAC:80:2F:DE:00:75:42 | the MAC address sticker on the back panel of the instrument |

| | |
|---|---|
| | 2x BNC-BNC cables |
| | short cap |
| | Quick start guide (A4) |
| | Certificate of Calibration (A4) |
| | Packing list (A4) |
| | Test report (A4) |
| | CE and UKCA Conformity Declarations (A4) |
| | Handling and safety instructions (A4) |

The VHFLI Instrument is equipped with a multi-mains switched power supply, and therefore can be connected to most power systems in the world. The fuse holder is integrated with the power inlet and can be extracted by grabbing the holder with two small screwdrivers at the top and at the bottom at the same time. A spare fuse is contained in the fuse holder. The fuse description is found in the specifications chapter.

Carefully inspect your instrument. If there is mechanical damage or the instrument does not pass the basic tests, then you should immediately notify the Zurich Instruments support team through email.

## 2.3. Handling and Safety Instructions

The VHFLI is an instrument designed to measure small electrical signals, intended for use in a scientific laboratory. It may be used as benchtop equipment or installed in a suitable rack enclosure. A host PC is required to connect to the instrument, for instructions on how to properly connect please see the chapter Section 2.5, "Connecting to the Instrument", on page 14 . Details on how to operate the instrument can be found in the tutorial section of this manual and on the graphical user interface of the LabOne control software.

The VHFLI Instrument is a sensitive piece of electronic equipment, and under no circumstances should its casing be opened, as there are high-voltage parts inside which may be harmful to human beings. There are no serviceable parts inside the instrument. Do not install substitute parts or perform any unauthorized modification to the product. Opening the instrument immediately voids the warranty provided by Zurich Instruments. The instrument may be used and operated within the environmental specifications listed in the Chapter 5, "Specifications", on page 42 .

Do not use this product in any manner not specified by the manufacturer. The protective features of this product may be affected if it is used in a way not specified in the operating instructions.

The following general safety instructions must be observed during all phases of operation, service, and handling of the instrument. The disregard of these precautions and all specific warnings elsewhere in this manual may negatively affect the operation of the equipment and its lifetime.

Zurich Instruments assumes no liability for the user’s failure to observe and comply with the instructions in this user manual.

## Caution

The BNC connectors on the front panel are made for transmitting radio frequencies and can be damaged if handled inappropriately. Take care when attaching or detaching cables or when moving the instrument.

Table 2.2: Safety Instructions

| | |
|---|---|
| Ground the instrument | The instrument chassis must be correctly connected to earth ground by means of the supplied power cord. The ground pin of the power cord set plug must be firmly connected to the electrical ground (safety ground) terminal at the mains power outlet. Interruption of the protective earth conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in personal injury and potential damage to the instrument. |
| Ground loops | The BNC connectors are not floating. For sensitive operations and in order to avoid ground loops, consider adding dc-blocks at the Inputs and Outputs of the device. |
| Electromagnetic environment | This equipment has been certified to conform with industrial electromagnetic environment as defined in EN 61326-1.<br>Emissions, that exceed the levels required by the document referenced above, can occur when connected to a test object. |
| Measurement category | This equipment is of measurement category I (CAT I). Do not use it for CAT II, III, or IV. Do not connect the measurement terminals to mains sockets. |
| Maximum ratings | The specified electrical ratings for the connectors of the instrument should not be exceeded at any time during operation. Please refer to the Chapter 5, "Specifications", on page 42 for a comprehensive list of ratings. |
| Do not service or adjust anything yourself | There are no serviceable parts inside the instrument. |
| Software updates | Frequent software updates provide the user with many important improvements as well as new features. Only the last released software version is supported by Zurich Instruments. |
| Warnings | Instructions contained in any warning issued by the instrument, either by the software, the graphical user interface, the notes on the instrument or mentioned in this manual, must be followed. |
| Notes | Instructions contained in the notes of this user manual are of essential importance for correctly interpreting the acquired measurement data. |
| High voltage transients due to inductive loads | When measuring devices with high inductance, take adequate measures to protect the Signal Input connectors against the high voltages of inductive load switching transients. These voltages can exceed the maximum voltage ratings of the Signal Inputs and lead to damage. |
| Location and ventilation | This instrument or system is intended for indoor use in an installation category II and pollution degree 2 environment as per IEC 61010-1. Do not operate or store the instrument outside the ambient conditions specified in the Chapter 5, "Specifications", on page 42 section. Do not block the ventilator opening on the back or the air intake on the chassis side and front, and allow a reasonable space for the air to flow. |
| Cleaning | To prevent electrical shock, disconnect the instrument from AC mains power and disconnect all test leads before cleaning. Clean the outside of the instrument using a soft, lint-free cloth slightly dampened with water. Do not use detergent or solvents. Do not attempt to clean internally. |
| AC power connection and mains line fuse | For continued protection against fire, replace the line fuse only with a fuse of the specified type and rating. Use only the power cord specified for this product and certified for the country of use. Always position the device so that its power switch and the power cord are easily accessible during operation. |
| Main power disconnect | Unplug product from wall outlet and remove power cord before servicing. Only qualified, service-trained personnel should remove the cover from the instrument. |

| | |
|---|---|
| RJ45 sockets labeled ZSync | The RJ45 socket on the back panel labeled "ZSync" is not intended for Ethernet LAN connection. Connecting an Ethernet device to this socket may damage the instrument and/or the Ethernet device. |
| Operation and storage | Do not operate or store the instrument outside the ambient conditions specified in the Chapter 5, "Specifications", on page 42 section. |
| Handling | Handle with care. Do not drop the instrument. Do not store liquids on the device, as there is a chance of spillage resulting in damage. |
| Safety critical systems | Do not use this equipment in systems whose failure could result in loss of life, significant property damage or damage to the environment. |

If you notice any of the situations listed below, immediately stop the operation of the instrument, disconnect the power cord, and contact the support team at Zurich Instruments, either through the website form or through email.

Made in Switzerland — Zurich Instruments AG, Technoparkstrasse 1, 8005 Zürich, Switzerland.

Table 2.3: Unusual Conditions

| | |
|---|---|
| Fan is not working properly or not at all | Switch off the instrument immediately to prevent overheating of sensitive electronic components. |
| Power cord or power plug on instrument is damaged | Switch off the instrument immediately to prevent overheating, electric shock, or fire. Please exchange the power cord only with one for this product and certified for the country of use. |
| Instrument emits abnormal noise, smell, or sparks | Switch off the instrument immediately to prevent further damage. |
| Instrument is damaged | Switch off the instrument immediately and ensure it is not used again until it has been repaired. |

Table 2.4: Symbols

| | |
|---|---|
| ⏚ | Earth ground |
| ⊥ | Chassis ground |
| ⚠ | Caution. Refer to accompanying documentation |
| ⎓ | DC (direct current) |

# 2.4. Software Installation

The VHFLI Instrument is operated from a host computer with the LabOne software. To install the LabOne software on a computer, administrator rights may be required. In order to simply run the software later, a regular user account is sufficient. Instructions for downloading the correct version of the software packages from the Zurich Instruments website are described below in the platform-dependent sections. It is recommended to regularly update to the latest software version provided by Zurich Instruments. Thanks to the Automatic Update check feature, the update can be initiated with a single click from within the user interface, as shown in Section 2.6, "Software Update", on page 28 .

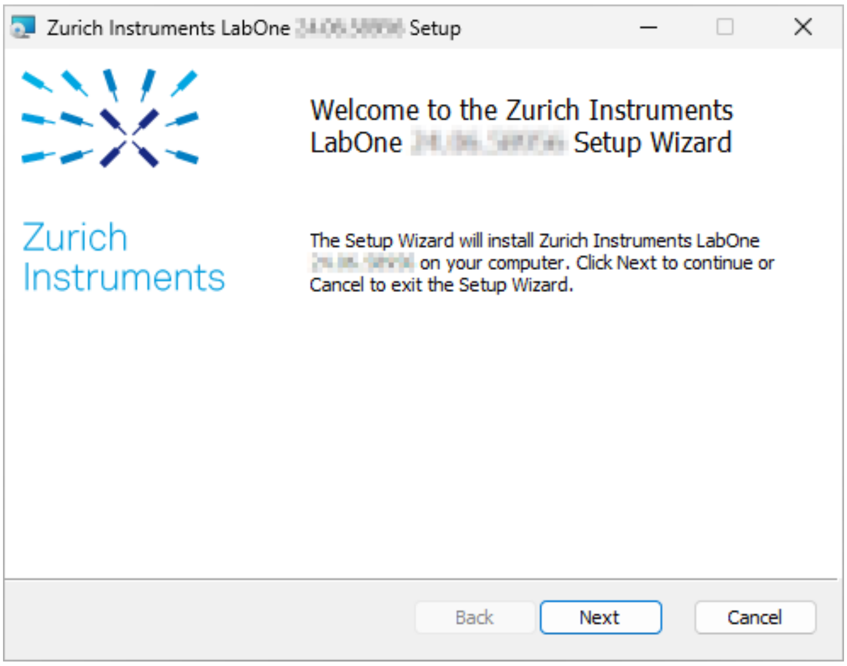
## 2.4.1. Installing LabOne on Windows

The installation packages for the Zurich Instruments LabOne software are available as Windows installer .msi packages. The software is available on the Zurich Instruments Download Center. Please ensure that you have administrator rights for the PC on which the software is to be installed. See LabOne compatibility for a comprehensive list of supported Windows systems.

## 2.4.2. Windows LabOne Installation

1. The VHFLI Instrument should not be connected to your computer during the LabOne software installation process.

2. Start the LabOne installer program with a name of the form `LabOne64-XX.XX.XXXXX.msi` by a double click and follow the instructions. Windows Administrator rights are required for installation. The installation proceeds as follows:
   - On the welcome screen click the **Next** button.



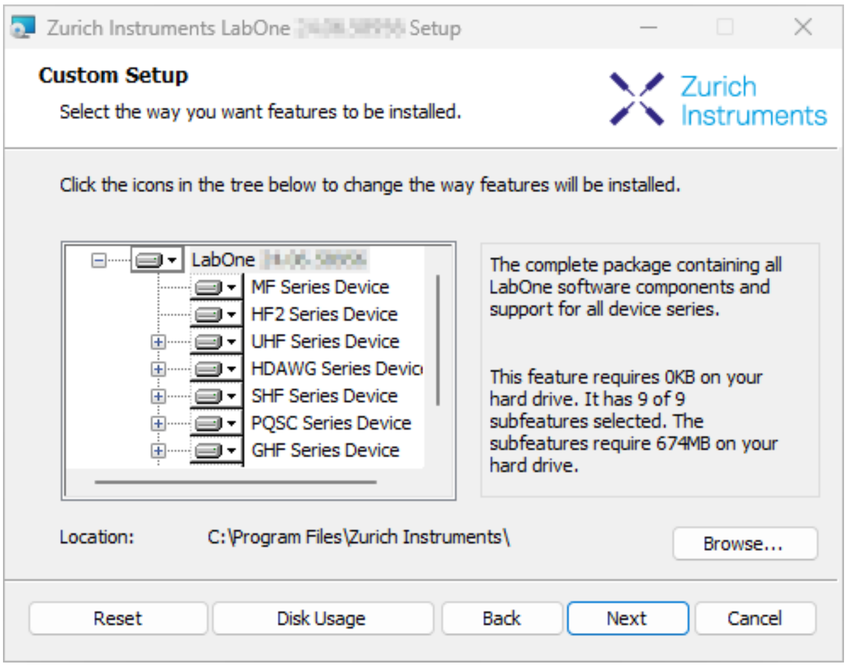
Zurich Instruments LabOne [redacted] Setup

Welcome to the Zurich Instruments LabOne [redacted] Setup Wizard

The Setup Wizard will install Zurich Instruments LabOne [redacted] on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Figure 2.1: Installation welcome screen

   - After reading through the Zurich Instruments license agreement, check the "I accept the terms in the License Agreement" check box and click the **Next** button.
   - Review the features you want to have installed. For the VHFLI Instrument the "VHFLI Series Device", "LabOne User Interface" and "LabOne APIs" features are required. Please install the features for other device classes as well, if required. To proceed click the **Next** button.



Zurich Instruments LabOne [redacted] Setup

**Custom Setup**

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

- LabOne [redacted]
  - MF Series Device
  - HF2 Series Device
  - UHF Series Device
  - HDAWG Series Devic
  - SHF Series Device
  - PQSC Series Device
  - GHF Series Device

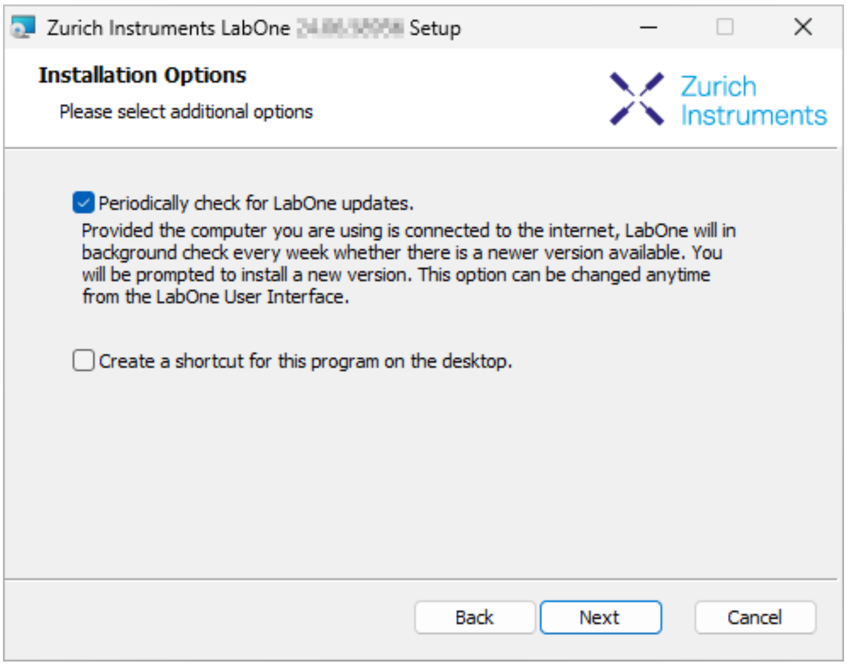
The complete package containing all LabOne software components and support for all device series.

This feature requires 0KB on your hard drive. It has 9 of 9 subfeatures selected. The subfeatures require 674MB on your hard drive.

Location: C:\Program Files\Zurich Instruments\

Figure 2.2: Custom setup screen

   - Select whether the software should periodically check for updates. Note, the software will still not update automatically. This setting can later be changed in the user interface. If you would like to install shortcuts on your desktop area, select "Create a shortcut for this program on the desktop". To proceed click the **Next** button.


Zurich Instruments LabOne Setup

Installation Options

Please select additional options

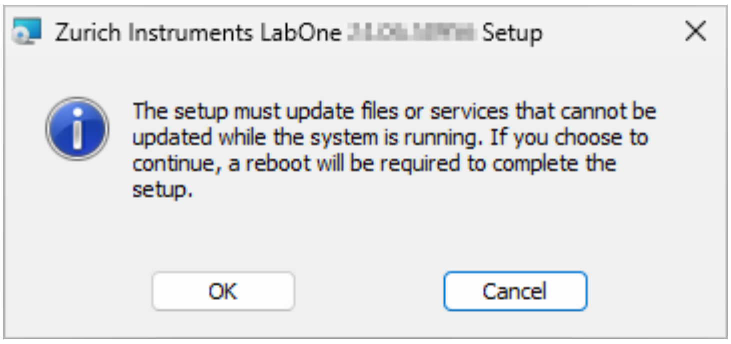
☑ Periodically check for LabOne updates.

Provided the computer you are using is connected to the internet, LabOne will in background check every week whether there is a newer version available. You will be prompted to install a new version. This option can be changed anytime from the LabOne User Interface.

□ Create a shortcut for this program on the desktop.

Figure 2.3: Automatic update check

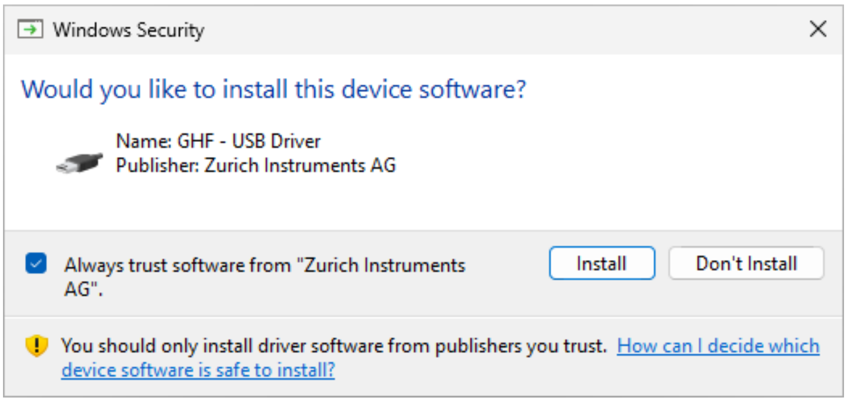
- Click the **Install** button to start the installation process.
- Windows may ask up to two times to reboot the computer if you are upgrading. Make sure you have no unsaved work on your computer.


Zurich Instruments LabOne Setup

The setup must update files or services that cannot be updated while the system is running. If you choose to continue, a reboot will be required to complete the setup.

Figure 2.4: Installation reboot request

- During the first installation of LabOne, it is required to confirm the installation of some drivers from the trusted publisher Zurich Instruments. Click on **Install**.


Windows Security

Would you like to install this device software?

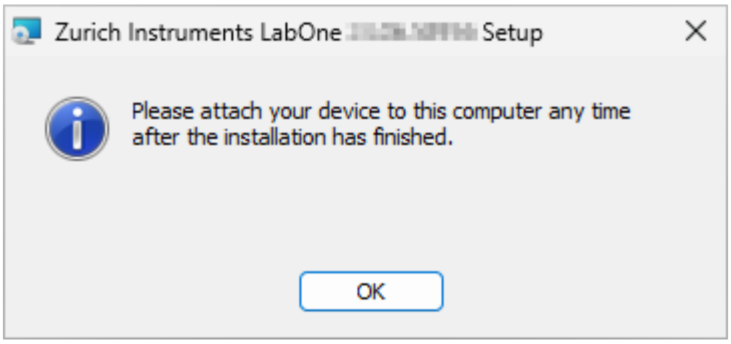
Name: GHF - USB Driver
Publisher: Zurich Instruments AG

☑ Always trust software from "Zurich Instruments AG".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Figure 2.5: Installation driver acceptance

- Click **OK** on the following notification dialog.



Zurich Instruments LabOne [redacted] Setup

Please attach your device to this computer any time after the installation has finished.

Figure 2.6: Installation completion screen

3. Click **Finish** to close the Zurich Instruments LabOne installer.
4. You can now start the LabOne User Interface as described in Subsection 2.5.5, "LabOne Software Start-up", on page 16 and choose an instrument to connect to via the Device Connection dialog shown in Figure 2.12, "Device Connection dialog", on page 17 .

### Warning

Do not install drivers from another source other than Zurich Instruments.

## 2.4.3. Running LabOne manually from the Command Line

After installing the LabOne software, the Web Server and Data Server can be started manually using the command-line. The more common way to start LabOne under Windows is described in Subsection 2.5.5, "LabOne Software Start-up", on page 16 . The advantage of using the command line is being able to observe and change the behavior of the Web and Data Servers.

### Running the Web Server from the Command Line

Before running the Web Server from the terminal, the user needs to ensure there is no other instance of the Web Server running in the background, since only one instance of the Web Server can run on a computer at a time. This can be checked using the Tray Icon as shown below.

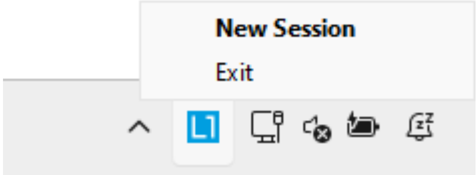


Figure 2.7: LabOne Tray Icon in Windows 11

To start the Web Servers manually, open a command-line terminal (Command Prompt, PowerShell (Windows) or Bash (Linux)). The current working directory needs to be the installation directory of the Web Server, usually `C:\Program Files\Zurich Instruments\LabOne\WebServer`. The behavior of the Web Server can be changed by providing command line arguments. For a detailed list of all arguments see the command line help text:

```
$ ziWebServer --help
```

One useful application of running the Webserver manually from a terminal window is to change the data directory from its default path in the user home directory. The data directory is a folder in which the LabOne Webserver saves all the measured data in the format specified by the user.

The corresponding command line argument to specify the data path is `--data-path` and the command to start the LabOne Webserver with a non-default directory path, e.g., `C:\data` is

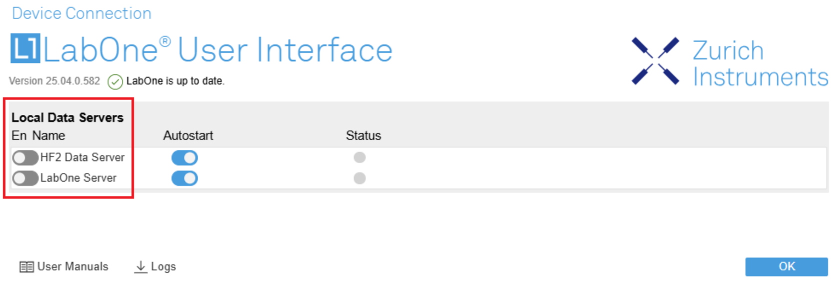
```
C:\Program Files\Zurich Instruments\LabOne\WebServer> ziWebServer --data-path "C:
\data"
```

## Running the Data Server from the Command Line

By default, the Data Server runs on Windows as a background service. To avoid conflicts with TCP port assignment, before running the Data Server from the terminal the user needs to ensure that the Data Server running in the background is stopped.

There are two ways to enable/disable the data servers, one from the LabOne user interface and one from the Windows services application.

In the "Advanced" mode of LabOne Session Manager, press the "Configure" button to open the following window for switching on/off the data servers.



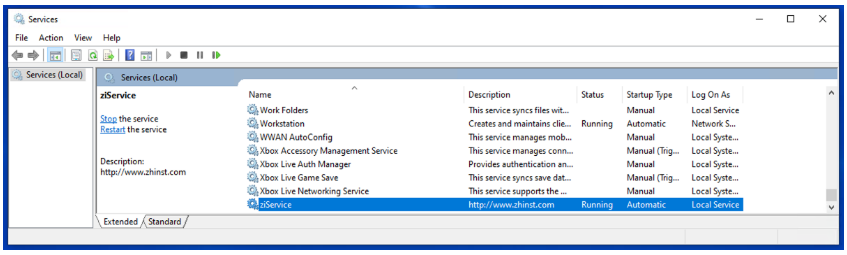
Device Connection

LabOne® User Interface

Version 25.04.0.582 LabOne is up to date.

| Local Data Servers | | |
|---|---|---|
| En Name | Autostart | Status |
| ○ HF2 Data Server | ● | ○ |
| ○ LabOne Server | ● | ○ |

Alternatively, open the Windows "Services" app as shown below, look for the ziService, right click on it and click "Stop".



ziService

Stop the service
Restart the service

Description:
http://www.zhinst.com

| Name | Description | Status | Startup Type | Log On As |
|---|---|---|---|---|
| Work Folders | This service syncs files wit... | | Manual | Local Service |
| Workstation | Creates and maintains clie... | Running | Automatic | Network S... |
| WWAN AutoConfig | This service manages mob... | | Manual | Local Syste... |
| Xbox Accessory Management Service | This service manages conn... | | Manual (Trig... | Local Syste... |
| Xbox Live Auth Manager | Provides authentication an... | | Manual | Local Syste... |
| Xbox Live Game Save | This service syncs save dat... | | Manual (Trig... | Local Syste... |
| Xbox Live Networking Service | This service supports the ... | | Manual | Local Syste... |
| ziService | http://www.zhinst.com | Running | Automatic | Local Service |

Now that the Data Server is not running anymore in the background, it can be started manually. Open a command-line terminal (Command Prompt, PowerShell (Windows) or Bash (Linux)) and run:

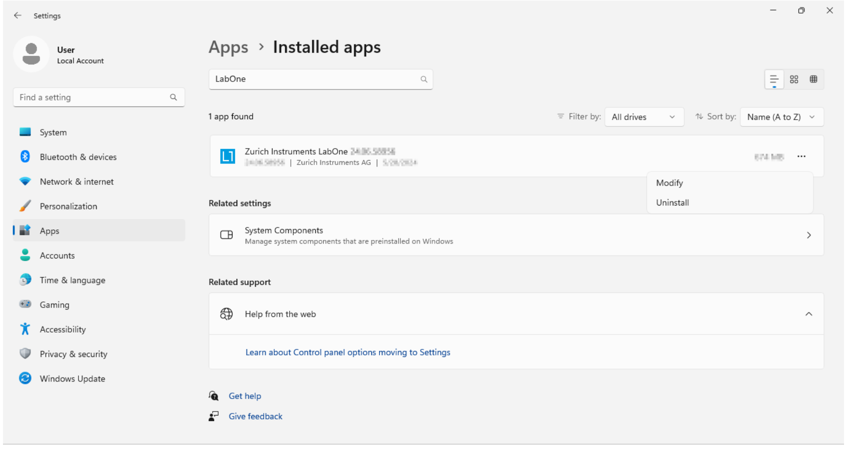
```
PS C:\Users\user> & 'C:\Program Files\Zurich
Instruments\LabOne\DataServer\ziDataServer.exe'
```

To show logs with higher verbosity, the `--debug 1` flag can be used:

```
PS C:\Users\user> & 'C:\Program Files\Zurich
Instruments\LabOne\DataServer\ziDataServer.exe' --debug 1
```

## 2.4.4. Windows LabOne Uninstallation

To uninstall the LabOne software package from a Windows computer, one can open the "Installed apps" page from the Windows start menu and search for LabOne. By selecting the LabOne item in the list of apps, the user has the option to "Uninstall" or "Modify" the software package as shown in Figure 2.8 on page 10 .



Apps › Installed apps

LabOne

1 app found

Zurich Instruments LabOne [illegible]
[illegible] | Zurich Instruments AG | [illegible]

Related settings

System Components
Manage system components that are preinstalled on Windows

Related support

Help from the web

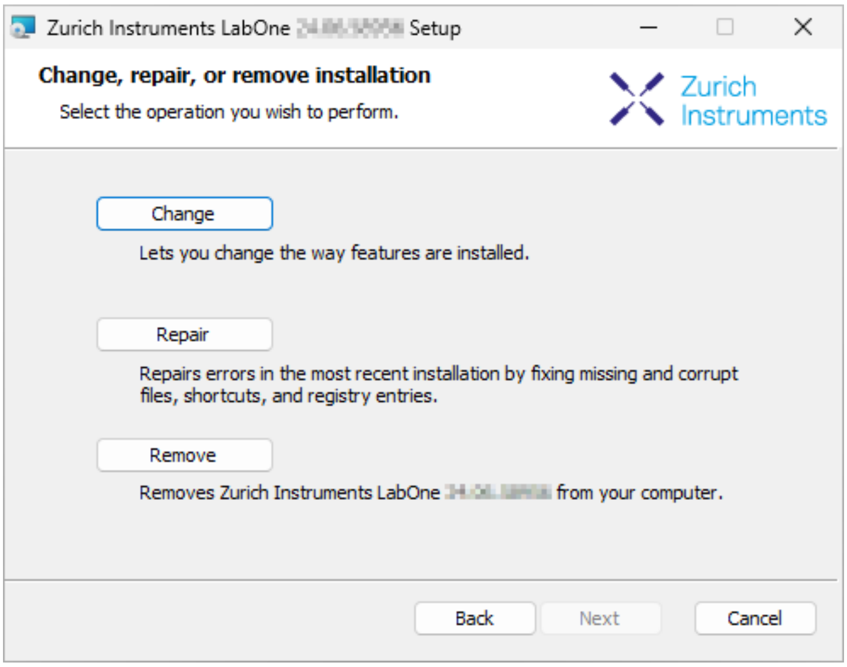
Learn about Control panel options moving to Settings

Figure 2.8: Uninstallation of LabOne on Windows computers

## Warning

Although it is possible to install a new version of LabOne on a currently-installed version, it is highly recommended to first uninstall the older version of LabOne from the computer and then, install the new version. Otherwise, if the installation process fails, the current installation is damaged and cannot be uninstalled directly. The user will need to first repair the installation and then, uninstall it.

In case a current installation of LabOne is corrupted, one can simply repair it by selecting the option "Modify" in Figure 2.8 on page 10 . This will open the LabOne installation wizard with the option "Repair" as shown in Figure 2.9 on page 10 .



Zurich Instruments LabOne [illegible] Setup

**Change, repair, or remove installation**

Select the operation you wish to perform.

Lets you change the way features are installed.

Repairs errors in the most recent installation by fixing missing and corrupt files, shortcuts, and registry entries.

Removes Zurich Instruments LabOne [illegible] from your computer.

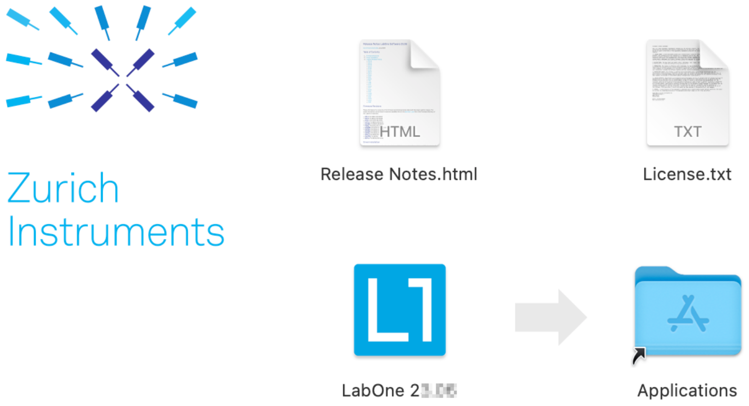
Figure 2.9: Repair of LabOne on Windows computers

After finishing the repair process, the normal uninstallation process described above can be triggered to uninstall LabOne.

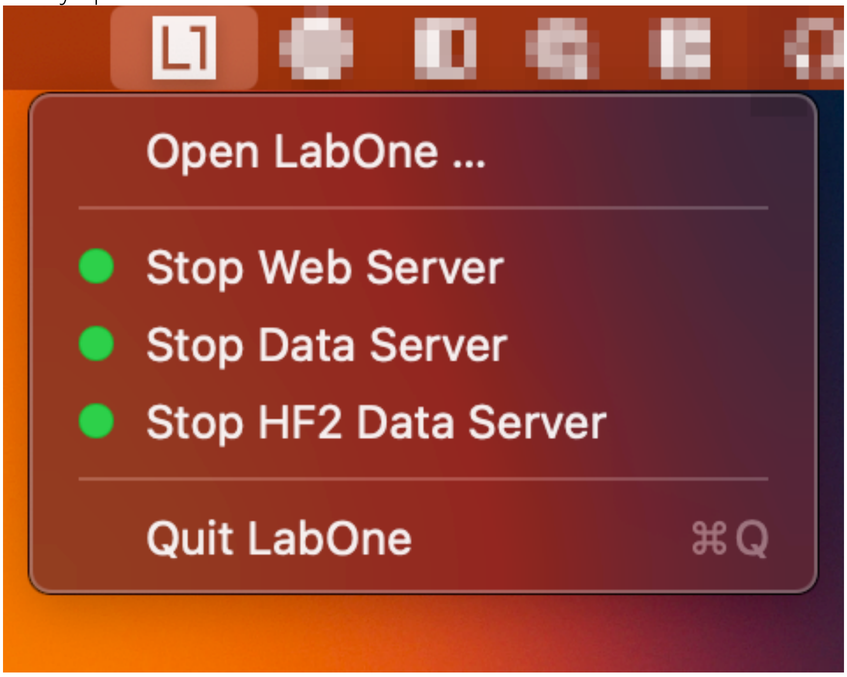
## 2.4.5. Installing LabOne on macOS

LabOne supports both Intel and ARM (M-series) architectures within a single universal disk image (DMG) file available in our Download Center.

- Download and double-click the DMG file to mount the image.



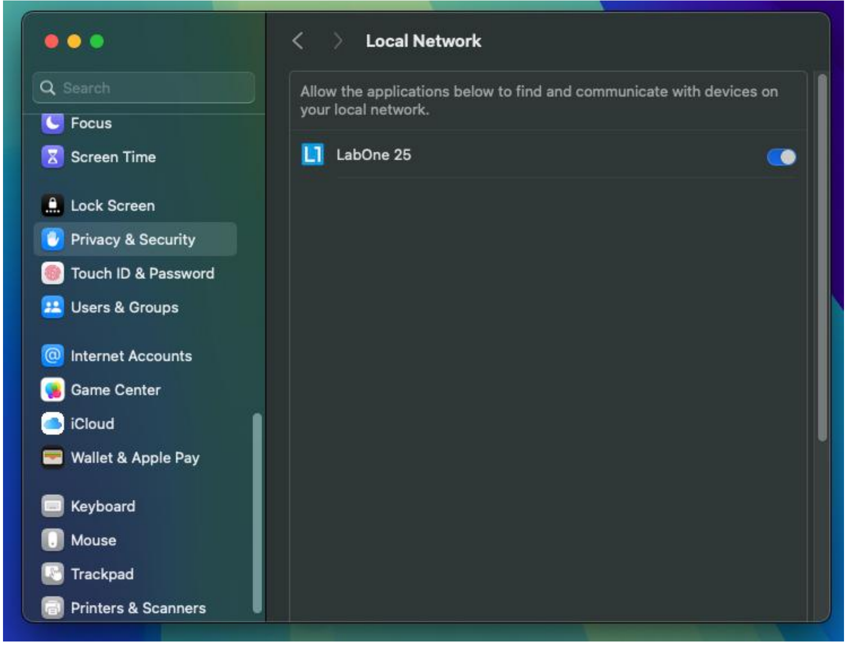
- The image contains a single LabOne application with all services needed.
- Once the application is started, a labone icon will appear in the menu bar. It allows the user to easily open a new session and shows the status of all services.



### Note

LabOne needs Local Network Access permissions. When LabOne is first started, a pop-up will appear asking to grant such permissions.

If you miss the pop-up, the permissions can also be enabled manually in Settings > Privacy & Security > Local Network.



Local Network

Allow the applications below to find and communicate with devices on your local network.

LabOne 25 ●

## 2.4.6. Uninstalling LabOne on macOS

To uninstall LabOne on macOS, simply drag the LabOne application to the trash bin.

## 2.4.7. Application Content

The LabOne application contains all resources available for macOS. This includes:

- The binaries for the Web Server and Data Servers.
- The binaries for the C, MATLAB, and LabVIEW APIs.
- An offline version of the user manuals.
- The latest firmware images for all instruments.

To access this content, right-click on the LabOne application and select "Show Package Contents". Then, go into Contents/Resources.

### Note

Since the application name contains a space, one needs to escape it when using the command line to access the contents: `cd /Applications/LabOne\ XX.XX.app/Contents/Resources`

## 2.4.8. Start LabOne Manually on the Command Line

To start the LabOne services like the data server and web server manually, one can use the command line.

The data server binary is called `ziDataServer` (`ziServer` for HF2 instruments) and is located at `Applications/LabOne\ XX.XX.app/Contents/Resources/DataServer/`.

The web server binary is called `ziWebServer` and is located at `Applications/LabOne\ XX.XX.app/Contents/Resources/DataServer/`.

### Note

No special command line arguments are needed to start the LabOne services. Use the `--help` argument to see all available options.

## 2.4.9. Installing LabOne on Linux

## 2.4.10. Requirements

Ensure that the following requirements are fulfilled before trying to install the LabOne software package:

1. LabOne software supports typical modern GNU/Linux distributions (Ubuntu 14.04+, CentOS 7+, Debian 8+). The minimum requirements are glibc 2.17+ and kernel 3.10+.
2. You have administrator rights for the system.
3. The correct version of the LabOne installation package for your operating system and platform have been downloaded from the Zurich Instruments Download Center:

   ```
   LabOneLinux<arch>-<release>.<revision>.tar.gz,
   ```

Please ensure you download the correct architecture (x86-64 or arm64) of the LabOne installer. The `uname` command can be used in order to determine which architecture you are using, by running:

```
uname -m
```

in a command line terminal. If the command outputs `x86_64` the x86-64 version of the LabOne package is required, if it displays `aarch64` the ARM64 version is required.

## 2.4.11. Linux LabOne Installation

Proceed with the installation in a command line shell as follows:

1. Extract the LabOne tarball in a temporary directory:

   ```
   tar xzvf LabOneLinux<arch>-<release>-<revision>.tar.gz
   ```

2. Navigate into the extracted directory.

   ```
   cd LabOneLinux<arch>-<release>-<revision>
   ```

3. Run the install script with administrator rights and proceed through the guided installation, using the default installation path if possible:

   ```
   sudo bash install.sh
   ```

   The install script lets you choose between the following three modes:
   - Type "a" to install the Data Server program, the Web Server program, documentation and APIs.
   - Type "u" to install `udev` support (only necessary if HF2 Instruments will be used with this LabOne installation and not relevant for other instrument classes).
   - Type "ENTER" to install both options "a" and "u".
4. Test your installation by running the software as described in the next section.

## 2.4.12. Running the Software on Linux

The following steps describe how to start the LabOne software in order to access and use your instrument in the User Interface.

1. Start the Web Server program at a command prompt:

   ```
   $ ziWebServer
   ```

2. Start an up-to-date web browser and enter the `127.0.0.1:8006` in the browser’s address bar to access the Web Server program and start the LabOne User Interface. The LabOne Web

Server installed on the PC listens by default on port number 8006 instead of 80 to minimize the probability of conflicts.

3. You can now start the LabOne User Interface as described in Subsection 2.5.5, "LabOne Software Start-up", on page 16 and choose an instrument to connect to via the Device Connection dialog shown in Figure 2.12, "Device Connection dialog", on page 17 .

### Important

Do not use two Data Server instances running in parallel; only one instance may run at a time.

## 2.4.13. Uninstalling LabOne on Linux

The LabOne software package copies an uninstall script to the base installation path (the default installation directory is `/opt/zi/`). To uninstall the LabOne package please perform the following steps in a command line shell:

1. Navigate to the path where LabOne is installed, for example, if LabOne is installed in the default installation path:

```
$ cd /opt/zi/
```

2. Run the uninstall script with administrator rights and proceed through the guided steps:

```
$ sudo bash uninstall_LabOne<arch>-<release>-<revision>.sh
```

# 2.5. Connecting to the Instrument

The Zurich Instruments VHFLI is operated using the LabOne software. After the installation as described in Section 2.4, "Software Installation", on page 5 , the instrument can be connected to the host computer using either the USB 3.0 or the 1 Gbit/s Ethernet (1GbE). Please use the respective cables supplied with the instrument. Once one of the physical connection achieved successfully, the LabOne software can recognize the instrument.

### Note

The following web browsers are supported (latest versions).

Chrome Firefox Opera Edge Safari

- Using the 1GbE port, it is possible to connect the instrument to an existing local area network (LAN) or establish a point-to-point connection to the host computer. For further details, see Subsection 2.5.13, "1GbE Connectivity", on page 24
- Using the USB port requires point-to-point connection to the host computer. For further information, see Subsection 2.5.12, "USB Connectivity", on page 23 .

### Note

It is recommended to use the 1GbE port for communicating with the instrument, especially for long-running experiments while measured signals are continuously acquired for an extended period of time. This is to avoid possible interruptions that the USB protocol might cause depending on the host computer's USB settings.

## 2.5.1. LabOne Software Architecture

The Zurich Instruments LabOne software gives quick and easy access to the instrument from a host PC. LabOne also supports advanced configurations with simultaneous access by multiple software clients (i.e., LabOne User Interface clients and/or API clients), and even simultaneous access by several users working on different computers. Here we give a brief overview of the architecture of the LabOne software. This will help to better understand the following chapters.

The software of Zurich Instruments equipment is server-based. The servers and other software components are organized in layers as shown in Figure 2.10 on page 15 .

- The lowest layer running on the PC is the LabOne Data Server, which is the interface to the connected instrument.
- The middle layer contains the LabOne Web Server, which is the server for the browser-based LabOne User Interface.
- The graphical user interface, together with the programming user interfaces, are contained in the top layer.

The architecture with one central Data Server allows multiple clients to access a device with synchronized settings. The following sections explain the different layers and their functionality in more detail.

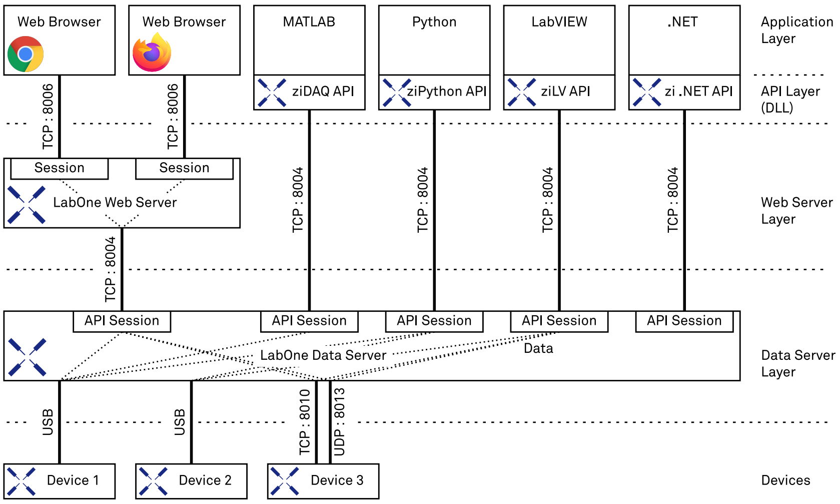


Figure 2.10: LabOne Software architecture

## 2.5.2. LabOne Data Server

The **LabOne Data Server** program is a dedicated server that is in charge of all communication to and from the device. The Data Server can control a single or also multiple instruments. It will distribute the measurement data from the instrument to all the clients that subscribe to it. It also ensures that settings changed by one client are communicated to other clients. The device settings are therefore synchronized on all clients. On a PC, only a single instance of a LabOne Data Server should be running.

## 2.5.3. LabOne Web Server

The LabOne Web Server is an application dedicated to serving up the web pages that constitute the LabOne user interface. The user interface can be opened with any device with a web browser. Since it is touch enabled, it is possible to work with the LabOne User Interface on a mobile device - like a tablet. The LabOne Web Server supports multiple clients simultaneously. This means that more than one session can be used to view data and to manipulate the instrument. A session could be running in a browser on the PC on which the LabOne software is installed. It could equally well be running in a browser on a remote machine.

With a LabOne Web Server running and accessing an instrument, a new session can be opened by typing in a network address and port number in a browser address bar. In case the Web Server runs on the **same** computer, the address is the localhost address (both are equivalent):

- `127.0.0.1:8006`
- `localhost:8006`

In case the Web Server runs on a **remote** computer, the address is the IP address or network name of the remote computer:

- `192.168.x.y:8006`
- `myPC.company.com:8006`

The most recent versions of the most popular browsers are supported: Chrome, Firefox, Edge, Safari and Opera.

## 2.5.4. LabOne API Layer

The instrument can also be controlled via the application program interfaces (APIs) provided by Zurich Instruments. APIs are provided in the form of libraries for the following programming environments:

- MATLAB
- Python
- LabVIEW
- .NET
- C

An extensive Python API and Python-based drivers are provided for the following frameworks:

- Zurich Instruments Python Toolkit
- Zurich Instruments QCoDeS Driver

The instrument can therefore be controlled by an external program, and the resulting data can be processed there. The device can be concurrently accessed via one or more of the APIs and via the user interface. This enables easy integration into larger laboratory setups. See the LabOne Programming Manual for further information. Using the APIs, the user has access to the same functionality that is available in the LabOne User Interface.

## 2.5.5. LabOne Software Start-up

This section describes the start-up of the LabOne User Interface which is used to control the VHFLI Instrument. If the LabOne software is not yet installed on the PC please follow the instructions in Section 2.4, "Software Installation", on page 5 . If the device is not yet connected please find more information in Subsection 2.5.11, "Visibility and Connection", on page 22 .

The LabOne User Interface start-up link can be found under the Windows 10/11 Start Menu. As shown in Figure 2.11 on page 16 , click on `Start Menu → Zurich Instruments LabOne`. This will open the User Interface in a new tab in your default web browser and start the LabOne Data Server and LabOne Web Server programs in the background. A detailed description of the software architecture is found in Subsection 2.5.1, "LabOne Software Architecture", on page 15 .

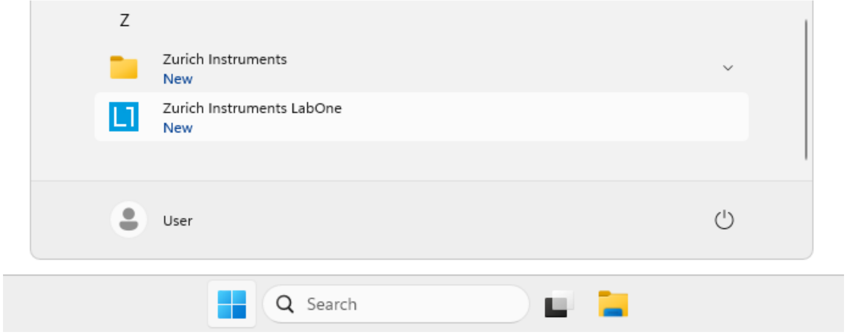


Figure 2.11: Link to the LabOne User Interface in the Windows 11 Start Menu

LabOne is an HTML5 browser-based program. This simply means that the user interface runs in a web browser and that a connection using a mobile device is also possible; simply specify the IP address (and port 8006) of the PC running the user interface.

## Note

By creating a shortcut to Google Chrome on your desktop with the Target `path\to\chrome.exe -app=http://127.0.0.1:8006` set in Properties you can run the LabOne User Interface in Chrome in application mode, which improves the user experience by removing the unnecessary browser controls.

After starting LabOne, the Device Connection dialog Figure 2.12 on page 17 is shown to select the device for the session. The term "session" is used for an active connection between the user interface and the device. Such a session is defined by device settings and user interface settings. Several sessions can be started in parallel. The sessions run on a shared LabOne Web Server. A detailed description of the software architecture can be found in the Subsection 2.5.1, "LabOne Software Architecture", on page 15 .

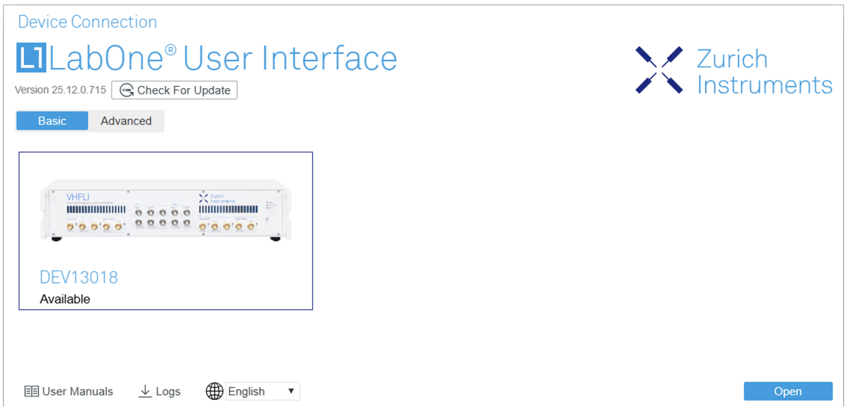
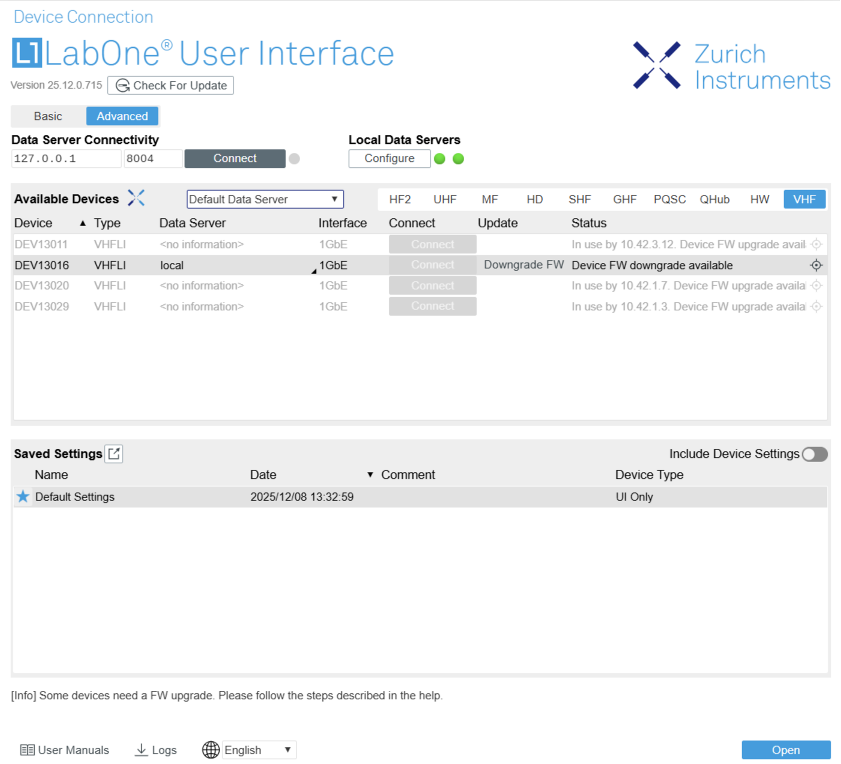


Figure 2.12: Device Connection dialog

The Device Connection dialog opens in the Basic view by default. In this view, all devices that are available for connection are represented by an icon with serial number and status information. If required, a button appears on the icon to perform a firmware upgrade. Otherwise, the device can be connected by a double click on the icon, or a click on the Open button at the bottom right of the dialog.

In some cases it’s useful to switch to the Advanced view of the Device Connection dialog by clicking on the "Advanced" button. The Advanced view offers the possibility to select custom device and UI settings for the new session and gives further connectivity options that are particularly useful for multi-instrument setups.



| Device | Type | Data Server | Interface | Connect | Update | Status |
|---|---|---|---|---|---|---|
| DEV13011 | VHFLI | <no information> | 1GbE | Connect | | In use by 10.42.3.12. Device FW upgrade avail |
| DEV13016 | VHFLI | local | 1GbE | Connect | Downgrade FW | Device FW downgrade available |
| DEV13020 | VHFLI | <no information> | 1GbE | Connect | | In use by 10.42.1.7. Device FW upgrade availa |
| DEV13029 | VHFLI | <no information> | 1GbE | Connect | | In use by 10.42.1.3. Device FW upgrade availa |

Saved Settings

Include Device Settings

| Name | Date | Comment | Device Type |
|---|---|---|---|
| Default Settings | 2025/12/08 13:32:59 | | UI Only |

[Info] Some devices need a FW upgrade. Please follow the steps described in the help.

Figure 2.13: Device Connection dialog (Advanced view)

The Advanced view consists of three parts:

- Data Server Connectivity
- Available Devices
- Saved Settings

The Available Devices table has a display filter, usually set to **Default Data Server**, that is accessible by a drop-down menu in the header row of the table. When changing this to **Local Data Servers**, the Available Devices table will show only connections via the Data Server on the host PC and will contain all instruments directly connected to the host PC via USB or to the local network via 1GbE. When using the **All Data Servers** filter, connections via Data Servers running on other PCs in the network also become accessible. Once your instrument appears in the Available Devices table, perform the following steps to start a new session:

1. Select an instrument in the **Available Devices** table.
2. Select a setting file in the **Saved Settings** list unless you would like to use the Default Settings.
3. Start the session by clicking on Open

## Note

By default, opening a new session will only load the UI settings (such as plot ranges), but not the device settings (such as signal amplitude) from the saved settings file. In order to include the device settings, enable the **Include Device Settings** checkbox. Note that this can affect existing sessions since the device settings are shared between them.

## Note

In case devices from other Zurich Instruments series (UHF, HF2, MF, HDAWG, PQSC, GHF, or SHF) are used in parallel, the list in **Available Devices** section can contain those as well.

The following sections describe the functionality of the **Device Connection** dialog in detail.

## 2.5.6. Data Server Connectivity

The Device Connection dialog represents a Web Server. However, on start-up the Web Server is not yet connected to a LabOne Data Server. With the **Connect/Disconnect** button the connection to a Data Server can be opened and closed.

This functionality can usually be ignored when working with a single VHFLI Instrument and a single host computer. Data Server Connectivity is important for users operating their instruments from a remote PC, i.e., from a PC different to the PC on which the Data Server is running or for users working with multiple instruments. The Data Server Connectivity function then gives the freedom to connect the Web Server to one of several accessible Data Servers. This includes Data Servers running on remote computers, and also Data Servers running on an MF Series instrument.

In order to work with a UHF, HF2, HDAWG, PQSC, GHF, or SHF instrument remotely, proceed as follows. On the computer directly connected to the instrument (Computer 1) open a User Interface session and change the Connectivity setting in the Config tab to "From Everywhere". On the remote computer (Computer 2), open the Device Connection dialog by starting up the LabOne User Interface and then go to the Advanced view by clicking on Advanced on the top left of the dialog. Change the display filter from Default Data Server to All Data Servers by opening the drop-down menu in the header row of the Available Devices table. This will make the Instrument connected to Computer 1 visible in the list. Select the device and connect to the remote Data Server by clicking on Connect. Then start the User Interface as described above.

## Note

When using the filter "All Data Servers", take great care to connect to the right instrument, especially in larger local networks. Always identify your instrument based on its serial number in the form DEV0000, which can be found on the instrument back panel.

## 2.5.7. Available Devices

The Available Devices table gives an overview of the visible devices. A device is ready for use if either marked free or connected. The first column of the list holds the **Enable** button controlling the connection between the device and a Data Server. This button is greyed out until a Data Server is connected to the LabOne Web Server using the Connect button. If a device is connected to a Data Server, no other Data Server running on another PC can access this device.

The second column indicates the serial number and the third column shows the instrument type. The fourth column shows the host name of the LabOne Data Server controlling the device. The next column shows the interface type. For VHFLI Instruments the interfaces USB or 1GbE are available and are listed if physically connected. The LabOne Data Server will scan for the available devices and interfaces every second. If a device has just been switched on or physically connected it may take up to 20 s before it becomes visible to the LabOne Data Server.

Table 2.5: Device Status Information

| | |
|---|---|
| Available | The device is not in use by any LabOne Data Server and can be connected by clicking the Enable button. Alternatively, a session can also be started by clicking on the **Open** button, without prior connection. |
| In use by | The device is in use by a LabOne Data Server. As a consequence the device cannot be accessed by the specified interface. To access the device a disconnect is needed. The additional message "FW upgrade available" or "FW downgrade available" may also be displayed in this state. |
| Connected | The device is connected to a LabOne Data Server, either on the same PC (indicated as local) or on a remote PC (indicated by its IP address). The user can start a session to work with that device. |
| Device FW upgrade required | The firmware is out of date and must be upgraded before the device can be used. Please first upgrade the firmware by clicking on the Upgrade FW button as described in Section 2.6, "Software Update", on page 28 . |
| Device FW upgrade available. Please update | The firmware is out of date but the device can still be used. It is highly recommended to upgrade the firmware by clicking on the Upgrade FW button as described in Section 2.6, "Software Update", on page 28 . |
| Device FW downgrade available | The firmware of the device is newer than the version supplied with the installed LabOne software. This could be due to reverting to an earlier LabOne version. The device can still be used but it is also possible to downgrade to the older firmware version if for any reason this is necessary. Click on the Downgrade FW button to downgrade the firmware. It is strongly advised to upgrade LabOne instead of downgrading the firmware. |
| Device FW upgrade required. Please use USB firmware upgrade utility | The firmware of UHFLI/UHFQA is too old to be updated from the Device Connection dialog. Please first upgrade the firmware using the USB Firmware Upgrade Utility provided with LabOne software. |
| Device not yet ready | The device is visible and starting up. When the device is ready it will be flagged as Available. |

## 2.5.8. Saved Settings

Settings files can contain both UI and device settings. UI settings control the structure of the LabOne User Interface, e.g. the position and ordering of opened tabs. Device settings specify the set-up of a device. The device settings persist on the device until the next power cycle or until overwritten by loading another settings file.

The columns are described in Table 2.6 on page 19 . The table rows can be sorted by clicking on the column header that should be sorted. The default sorting is by time. Therefore, the most recent settings are found on top. Sorting by the favorite marker or setting file name may be useful as well.

Table 2.6: Column Descriptions

| | |
|---|---|
| ☆ ★ | Allows favorite settings files to be grouped together. By activating the stars adjacent to a settings file and clicking on the column heading, the chosen files will be grouped together at the top or bottom of the list accordingly. The favorite marker is saved to the settings file. When the LabOne user interface is started next time, the row will be marked as favorite again. |
| Name | The name of the settings file. In the file system, the file name has the extension .md. |

| | |
|---|---|
| Date | The date and time the settings file was last written. |
| Comment | Allows a comment to be stored in the settings file. By clicking on the comment field a text can be typed in which is subsequently stored in the settings file. This comment is useful to describe the specific conditions of a measurement. |
| Device Type | The instrument type with which this settings file was saved. |

## Special Settings Files

Certain file names have the prefix "last_session_". Such files are created automatically by the LabOne Web Server when a session is terminated either explicitly by the user, or under critical error conditions, and save the current UI and device settings. The prefix is prepended to the name of the most recently used settings file. This allows any unsaved changes to be recovered upon starting a new session.

If a user loads such a last session settings file the "last_session_" prefix will be cut away from the file name. Otherwise, there is a risk that an auto-save will overwrite a setting which was saved explicitly by the user.

The settings file with the name "Default Settings" contains the default UI settings. See button description in Table 2.7 on page 20 .

Table 2.7: Button Descriptions

| | |
|---|---|
| **Open** | The settings contained in the selected settings file will be loaded. The button "Include Device Settings" controls whether only UI settings are loaded, or if device settings are included. |
| **Include Device Settings** | Controls which part of the selected settings file is loaded upon clicking on Open. If enabled, both the device and the UI settings are loaded. |
| **Auto Start** | Skips the session dialog at start-up if selected device is available. The default UI settings will be loaded with unchanged device settings. |

### Note

The user setting files are saved to an application-specific folder in the directory structure. The best way to manage these files is using the File Manager tab.

### Note

The factory default UI settings can be customized by saving a file with the name "default_ui" in the Config tab once the LabOne session has been started and the desired UI setup has been established. To use factory defaults again, the "default_ui" file must be removed from the user setting directory using the File Manager tab.

### Note

Double clicking on a device row in the Available Devices table is a quick way of starting the default LabOne UI. This action is equivalent to selecting the desired device and clicking the **Open** button.

Double clicking on a row in the Saved Settings table is a quick way of loading the LabOne UI with those UI settings and, depending on the "Include Device Settings" checkbox, device settings. This action is equivalent to selecting the desired settings file and clicking the **Open** button.

## 2.5.9. Tray Icon

When LabOne is started, a tray icon appears by default in the bottom right corner of the screen, as shown in the figure below. By right-clicking on the icon, a new web server session can be opened quickly, or the LabOne Web and Data Servers can be stopped by clicking on Exit. Double-clicking the icon also opens a new web server session, which is useful when setting up a connection to multiple instruments, for example.

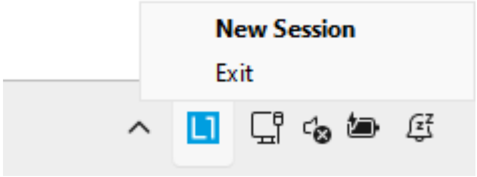


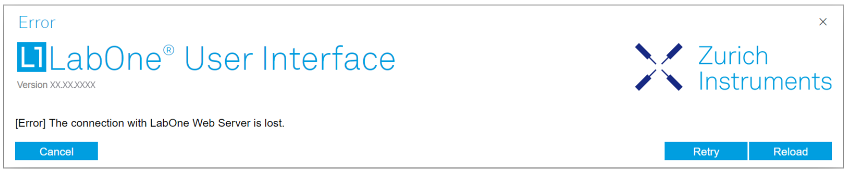
Figure 2.14: LabOne Tray Icon in Windows 11

### 2.5.10. Messages

The LabOne Web Server will show additional messages in case of a missing component or a failure condition. These messages display information about the failure condition. The following paragraphs list these messages and give more information on the user actions needed to resolve the problem.

### Lost Connection to the LabOne Web Server

In this case the browser is no longer able to connect to the LabOne Web Server. This can happen if the Web Server and Data Server run on different PCs and a network connection is interrupted. As long as the Web Server is running and the session did not yet time out, it is possible to just attach to the existing session and continue. Thus, within about 15 seconds it is possible with **Retry** to recover the old session connection. The **Reload** button opens the Device Connection dialog shown in Figure 2.12 on page 17 . The figure below shows an example of the Connection Lost dialog.

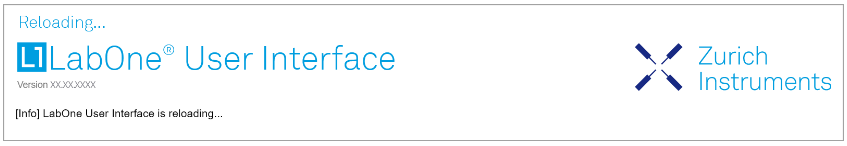


[Error] The connection with LabOne Web Server is lost.

Figure 2.15: Dialog: Connection Lost

### Reloading...

If a session error cannot be handled, the LabOne Web Server will restart to show a new Device Connection dialog as shown in Figure 2.12 on page 17 . During the restart a window is displayed indicating that the LabOne User Interface will reload. If reloading does not happen the same effect can be triggered by pressing F5 on the keyboard. The figure below shows an example of this dialog.

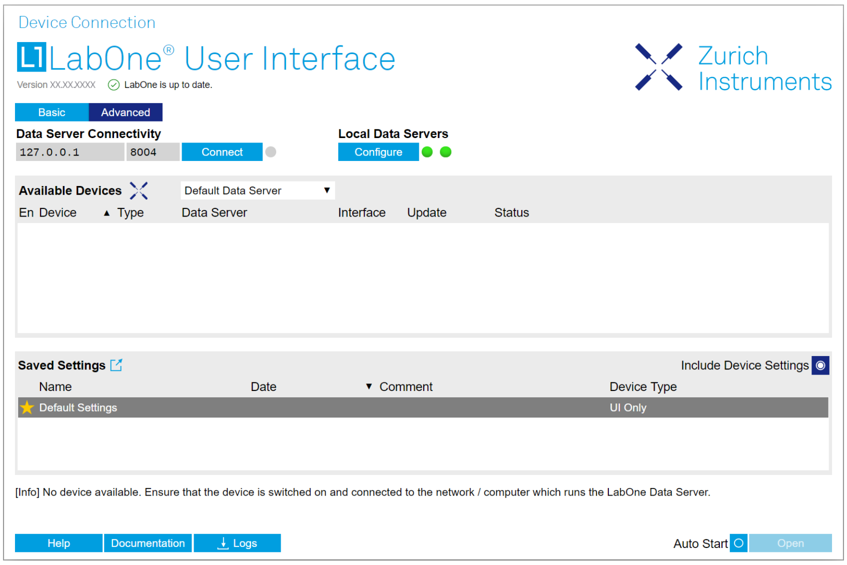


[Info] LabOne User Interface is reloading...

Figure 2.16: Dialog: Reloading

### No Device Discovered

An empty "Available Devices" table means that no devices were discovered. This can mean that no LabOne Data Server is running, or that it is running but failed to detect any devices. The device may be switched off or the interface connection fails. For more information on the interface between device and PC see Subsection 2.5.11, "Visibility and Connection", on page 22 . The figure below shows an example of this dialog.



Device Connection

LabOne® User Interface

Version XX.XX.XXXX ✓ LabOne is up to date.

Basic | Advanced

Data Server Connectivity: 127.0.0.1 8004 Connect ●

Local Data Servers: Configure ● ●

Available Devices — Default Data Server ▼

| En | Device | ▲ Type | Data Server | Interface | Update | Status |
|---|---|---|---|---|---|---|

Saved Settings — Include Device Settings ◉

| Name | Date | ▼ Comment | Device Type |
|---|---|---|---|
| ★ Default Settings | | | UI Only |

[Info] No device available. Ensure that the device is switched on and connected to the network / computer which runs the LabOne Data Server.

Help | Documentation | Logs — Auto Start ○ Open

Figure 2.17: No Device Discovered

## No Device Available

If all the devices in the "Available Devices" table are shown grayed, this indicates that they are either in use by another Data Server, or need a firmware upgrade. For firmware upgrade see Section 2.6, "Software Update", on page 28 . If all the devices are in use, access is not possible until a connection is relinquished by another Data Server.

## 2.5.11. Visibility and Connection

There are several ways to connect the instrument to a host computer. The device can either be connected by Universal Serial Bus (USB) or by 1 Gbit/s Ethernet (1GbE). The USB connection is a point-to-point connection between the device and the PC on which the Data Server runs. The 1GbE connection can be a point-to-point connection or an integration of the device into the local network (LAN). Depending on the network configuration and the installed network card, one or the other connectivity is better suited.

If an instrument is connected to a network, it can be accessed from multiple host computers. To manage the access to the instrument, there are two different connectivity states: visible and connected. It is important to distinguish if an instrument is just physically connected over 1GbE or actively controlled by the LabOne Data Server. In the first case the instrument is visible to the LabOne Data Server. In the second case the instrument is logically connected.

Figure 2.18, "Connectivity Example", on page 23 shows some examples of possible configurations of computer-to-instrument connectivity.

- Data Server on PC 1 is connected to device 1 (USB) and device 2 (USB).
- Data Server on PC 2 is connected to device 4 (TCP/IP).
- Data Server on PC 3 is connected to device 5.
- The device 3 is free and visible to PC 1 and PC 2 over TCP/IP.
- Devices 2 and 4 are physically connected by TCP/IP and USB interface. Only one interface is logically connected to the Data Server.

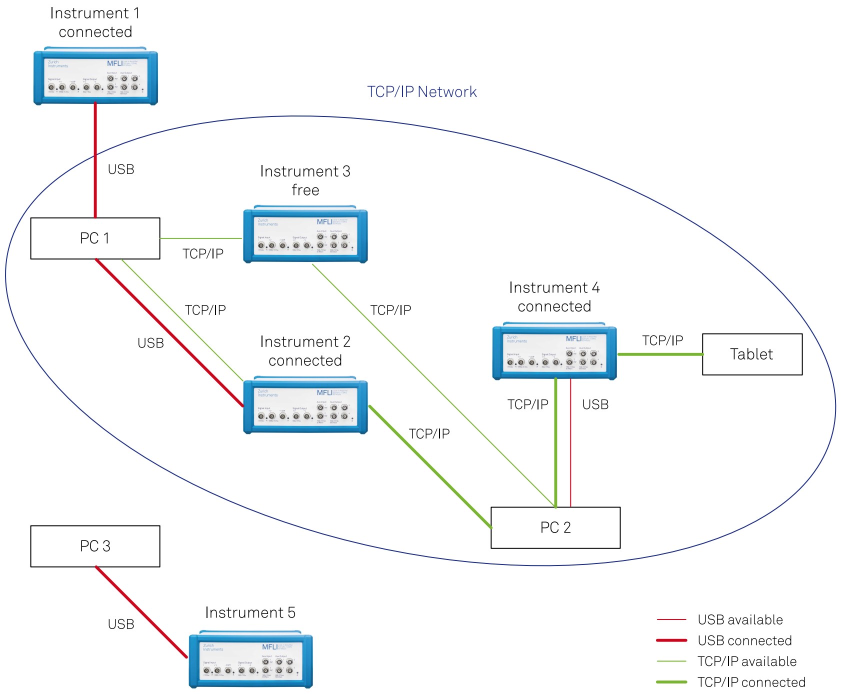


Figure 2.18: Connectivity Example

## Visible Instruments

An instrument is visible if the Data Server can identify it. On a TCP/IP network, several PCs running a Data Server will detect the same instrument as visible, i.e., discover it. If a device is discovered, the LabOne Data Server can initiate a connection to access the instrument. Only a single Data Server can be connected to an instrument at a time.

## Connected Instrument

Once connected to an instrument, the Data Server has exclusive access to that instrument. If another Data Server from another PC already has an active connection to the instrument, the instrument is still visible but cannot be connected.

Although a Data Server has exclusive access to a connected instrument, the Data Server can have multiple clients. Because of this, multiple browser and API sessions can access the instrument simultaneously.

## 2.5.12. USB Connectivity

To control the device over USB, connect the instrument with the supplied USB 3.0 cable to the PC on which the LabOne Software is installed.

### Warning

Make sure the cable is a USB 3.0 cable with "SuperSpeed" support. A USB 2.0 cable will also fit in the device USB socket but it won't sustain the device throughput.

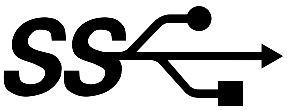


Figure 2.19: The USB "SuperSpeed" logo should appear on the USB cable

Also make sure that the USB port on the computer supports USB 3. This is the case if the USB port is USB-C, if it is blue or if it has a "SuperSpeed" logo.

### Warning

When using a laptop, ensure that it is connected to power and that power savings options are disabled. Battery saving options can cause the streaming of data at high sampling rate to become unreliable.

The software will automatically use the USB interface for controlling the device if available. If the USB connection is not available, the 1GbE connection may be selected. It is possible to enforce or exclude a specific interface connection.

## 2.5.13. 1GbE Connectivity

There are three methods for connecting to the device via 1GbE:

- Multicast DHCP
- Multicast point-to-point (P2P)
- Static Device IP

Multicast DHCP is the simplest and preferred connection method. Other connection methods can become necessary when using network configurations that conflict with local policies.

### Multicast DHCP

The most straightforward TCP/IP connection method is to rely on a network configuration to recognize the instrument. When connecting the instrument to a local area network (LAN), the DHCP server will assign an IP address to the instrument like to any PC in the network. In case of restricted networks, the network administrator may be required to register the device on the network by means of the MAC address. The MAC address is indicated on the back panel of the instrument. The LabOne Data Server will detect the device in the network by means of a multicast.

If the network configuration does not support multicast, or if the host computer has other network cards installed, it is necessary to use a static IP setup as described below. The instrument is configured to accept the IP address from the DHCP server, or to fall back to the IP address `192.168.1.10` if it does not get the address from the DHCP server.

Requirements:

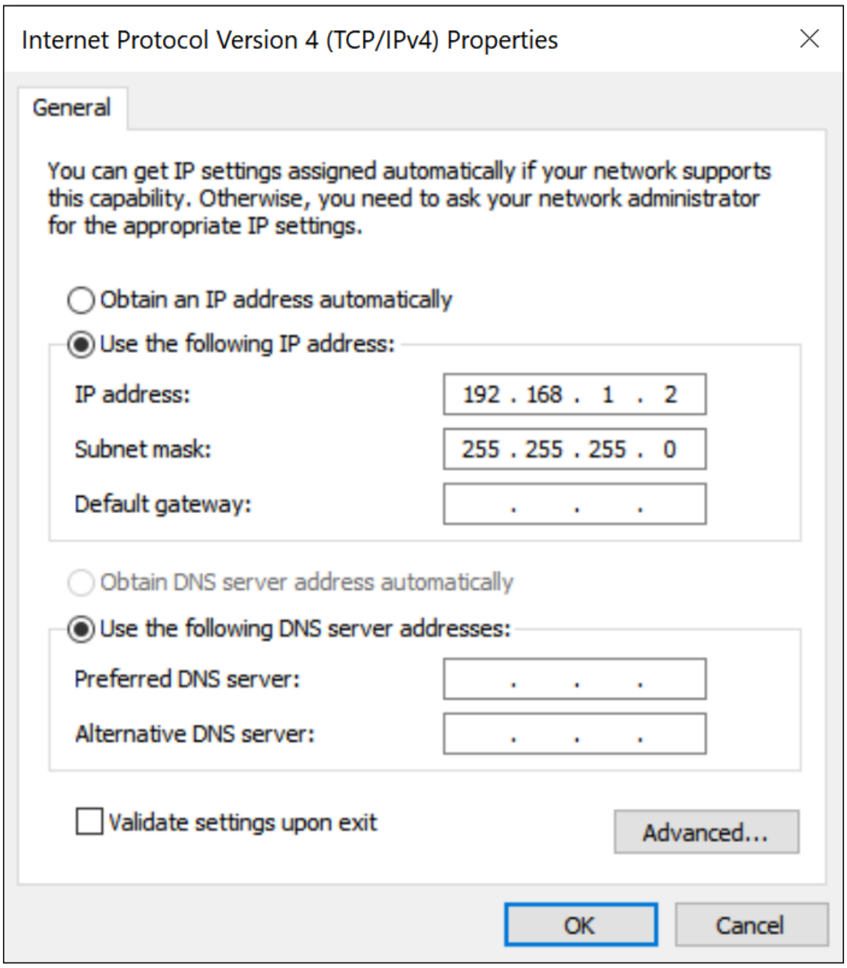
- Network supports multicast

### Multicast Point-to-Point

Setting up a point-to-point (P2P) network consisting only of the host computer and the instrument avoids problems related to special network policies. Since it is nonetheless necessary to stay connected to the internet, it is recommended to install two network cards in the computer, one of which is used for internet connectivity, the other can be used for connecting to the instrument. Alternatively, internet connectivity can be established via wireless LAN.

In such a P2P network the IP address of the host computer needs to be set to a static value, whereas the IP address of the device can be left dynamic.

1. Connect the 1GbE port of the network card that is dedicated for instrument connectivity directly to the 1GbE port of the instrument
2. Set this network card to static IP in TCP/IPv4 using the address `192.168.1.n`, where n=[2..9] and the mask `255.255.255.0`. (On Windows go to `Control Panel → Internet Options → Network and Internet → Network and Sharing Center → Local Area Connection → Properties`).



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 1 . 2

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternative DNS server: . . .

□ Validate settings upon exit

Figure 2.20: Static IP configuration for the host computer

3. Start up the LabOne User Interface normally. If your instrument does not show in the list of Available Devices, the reason may be that your network card does not support multicast. In that case, see "Static Device IP", on page 27 .

Requirements:

- Two network cards needed for additional connection to internet
- Network card of PC supports multicast
- Network card connected to the device must be in static IP4 configuration

## Note

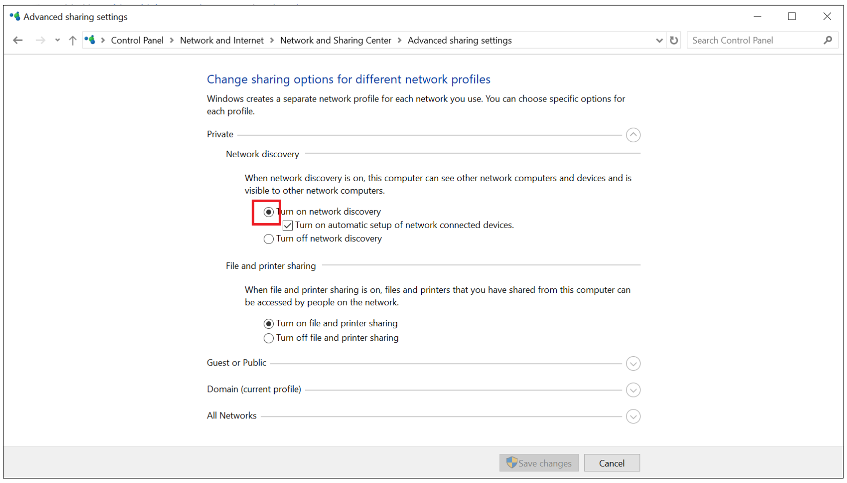
A power cycle of the instrument is required if it was previously connected to a network that provided an IP address to the instrument.

## Note

Only IP v4 is currently supported. There is no support for IP v6.

## Note

If the instrument is detected by LabOne but the connection can not be established, the reason can be the firewall blocking the connection. It is then recommended to change the P2P connection from Public to Private. On Windows this is achieved by turning on network discovery in the Private tab of the network's advanced sharing settings as shown in the figure below.



Advanced sharing settings
Control Panel > Network and Internet > Network and Sharing Center > Advanced sharing settings
Search Control Panel

Change sharing options for different network profiles

Windows creates a separate network profile for each network you use. You can choose specific options for each profile.

Private

Network discovery

When network discovery is on, this computer can see other network computers and devices and is visible to other network computers.

● Turn on network discovery
☑ Turn on automatic setup of network connected devices.
○ Turn off network discovery

File and printer sharing

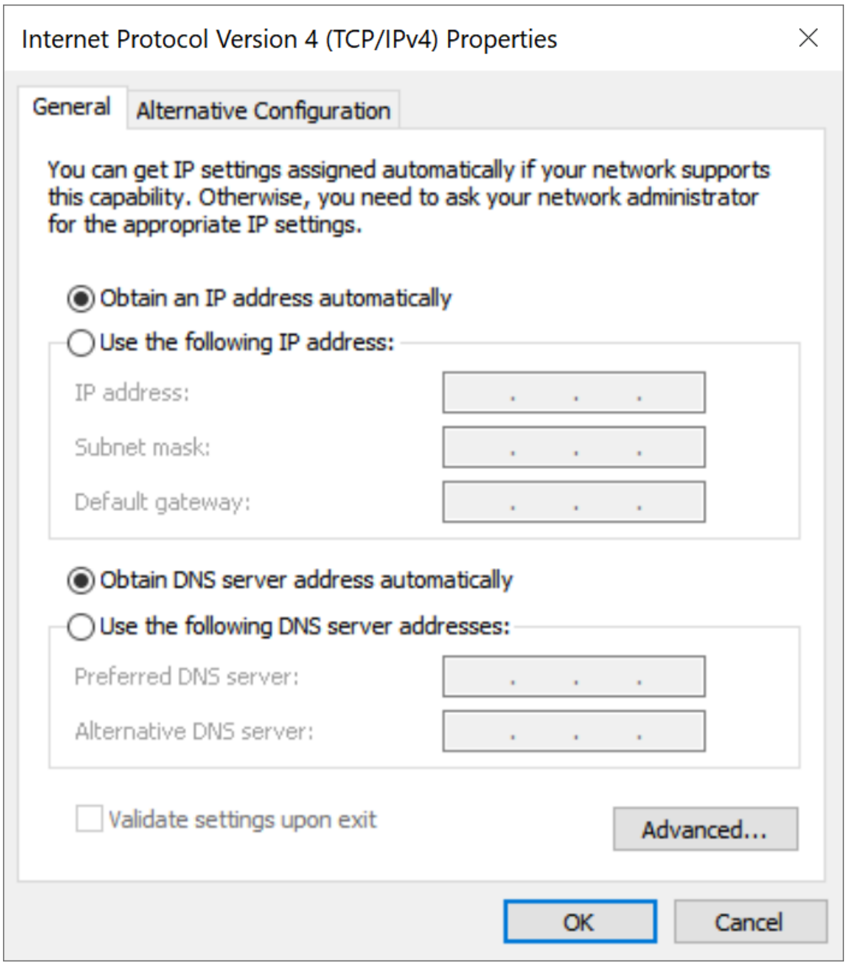
When file and printer sharing is on, files and printers that you have shared from this computer can be accessed by people on the network.

● Turn on file and printer sharing
○ Turn off file and printer sharing

Figure 2.21: Turn on network discovery for Private P2P connection

## Warning

Changing the IP settings of your network adapters manually can interfere with its later use, as it cannot be used anymore for network connectivity until it is configured again for dynamic IP.



Internet Protocol Version 4 (TCP/IPv4) Properties
General
Alternative Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically
○ Use the following IP address:

IP address: . . .
Subnet mask: . . .
Default gateway: . . .

● Obtain DNS server address automatically
○ Use the following DNS server addresses:

Preferred DNS server: . . .
Alternative DNS server: . . .

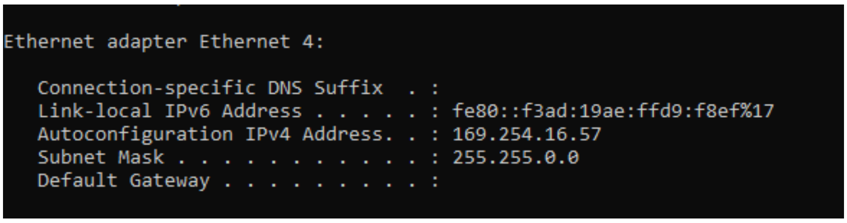
□ Validate settings upon exit

Figure 2.22: Dynamic IP configuration for the host computer

## Static Device IP

Although it is highly recommended to use dynamic IP assignment method in the host network of the instrument, there may be cases where the user wants to assign a static IP to the instrument. For instance, when the host network only contains Ethernet switches and hubs but no Ethernet routers are included, there is no DHCP server to dynamically assign an IP to the instrument. It is still advised to add an Ethernet router to the network and benefit from dynamic IP assignment; however, if a router is not available, the instrument can be configured to work with a static IP.

Note that the static IP assigned to the instrument must be within the same range of the IP assigned to the host computer. Whether the host computer's IP is assigned statically or by a fallback mechanism, one can find this IP by running the command `ipconfig` or `ipconfig/all` in the operating system's terminal. As an example, Figure 2.23 on page 27 shows the outcome of running `ipconfig` in the terminal.



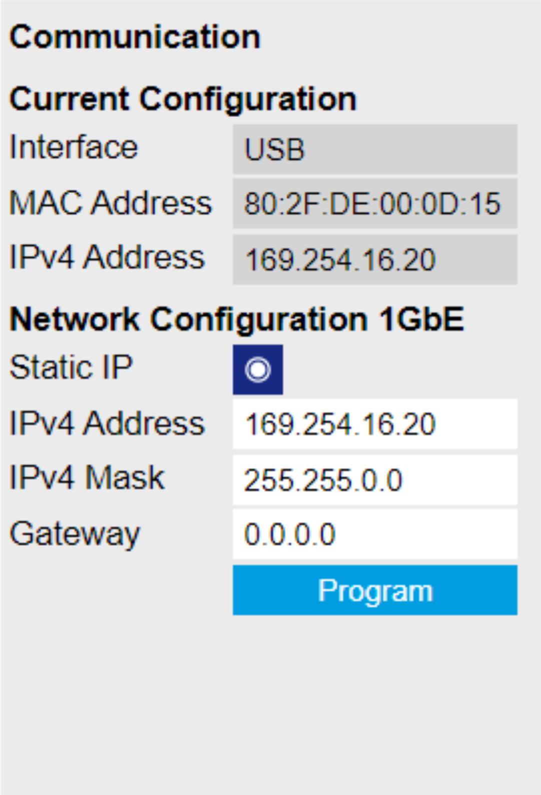
```
Ethernet adapter Ethernet 4:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f3ad:19ae:ffd9:f8ef%17
   Autoconfiguration IPv4 Address. . : 169.254.16.57
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . :
```

Figure 2.23: IP and subnet mask of host computer

It shows the network adapter of the host computer can be reached via the IP `169.254.16.57` and it uses a subnet mask of `255.255.0.0`. To make sure that the instrument is visible to this computer, one needs to assign a static IP of the form `169.254.x.x` and the same subnet mask to the instrument. To do so, the user should follow the instructions below.

1. Attach the instrument using an Ethernet cable to the network where the user's computer is hosted.
2. Attach the instrument via a USB cable to the host computer and switch it on.
3. Open the LabOne user interface (UI) and connect to the instrument via USB.
4. Open the "Device" tab of the LabOne UI and locate the "Communication" section as shown in Figure 2.24, "Configuration of static IP in LabOne UI", on page 28 .
5. Write down the desired static IP, e.g. `169.254.16.20`, into the numeric field "IPv4 Address".
6. Add the same subnet mask as the host computer, e.g. `255.255.0.0` to the numeric field "IPv4 Mask".
7. You can leave the field "Gateway" as `0.0.0.0` or change to be similar to the IP address but ending with `1`, e.g. `169.254.16.1`.
8. Enable the radio button for "Static IP".
9. Press the button "Program" to save the new settings to the instruments.
10. Power cycle the instrument and remove the USB cable. The instrument should be visible to LabOne via Ethernet connection.



**Communication**

**Current Configuration**

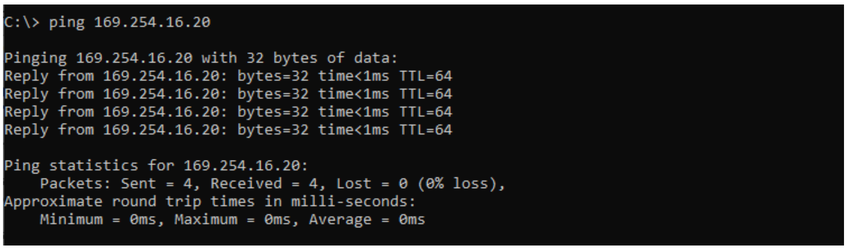
| | |
|---|---|
| Interface | USB |
| MAC Address | 80:2F:DE:00:0D:15 |
| IPv4 Address | 169.254.16.20 |

**Network Configuration 1GbE**

| | |
|---|---|
| Static IP | ● |
| IPv4 Address | 169.254.16.20 |
| IPv4 Mask | 255.255.0.0 |
| Gateway | 0.0.0.0 |

Figure 2.24: Configuration of static IP in LabOne UI

To make sure the IP assignment is done properly, one can use the command `ping` to check if the instrument can be reached through the network using its IP address. Figure 2.25 on page 28 shows the outcome of `ping` when the instrument is visible via the IP `169.254.16.20`.



```
C:\> ping 169.254.16.20

Pinging 169.254.16.20 with 32 bytes of data:
Reply from 169.254.16.20: bytes=32 time<1ms TTL=64
Reply from 169.254.16.20: bytes=32 time<1ms TTL=64
Reply from 169.254.16.20: bytes=32 time<1ms TTL=64
Reply from 169.254.16.20: bytes=32 time<1ms TTL=64

Ping statistics for 169.254.16.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Figure 2.25: Instrument visible through pinging

If set properly according to the instructions above, the instrument will use the same static IP configurations after each power cycle.

# 2.6. Software Update

## 2.6.1. Overview

It is recommended to regularly update the LabOne software on the VHFLI Instrument to the latest version. In case the Instrument has access to the internet, this is a very simple task and can be done with a single click in the software itself, as shown in Subsection 2.6.2, "Updating LabOne using Automatic Update Check", on page 28 . If you use one of the LabOne APIs with a separate installer, don't forget to update this part of the software, too.

## 2.6.2. Updating LabOne using Automatic Update Check

Updating the software is done in two steps. First, LabOne is updated on the PC by downloading and installing the LabOne software from the Zurich Instruments downloads page, as shown in Section 2.4, "Software Installation", on page 5 . Second, the instrument firmware needs to be updated from the Device Connection dialog after starting up LabOne. This is shown in Subsection 2.6.3, "Updating the Instrument Firmware ", on page 29 . In case "Periodically check for updates" has been enabled during the LabOne installation and LabOne has access to the internet, a notification will appear on the Device Connection dialog whenever a new version of the software is available for download. This setting can later be changed in the Config tab of the LabOne user interface. In case automatic update check is disabled, the user can manually check for updates at any time by clicking on the button Check For Update in the Device Connection dialog. In case an update is found, clicking on the button "Update Available" shown in Figure 2.26 on page 29 will start a download of the latest LabOne installer for Windows or Linux, see Figure 2.27 on page 29 . After download, proceed as explained in Section 2.4, "Software Installation", on page 5 to update LabOne.

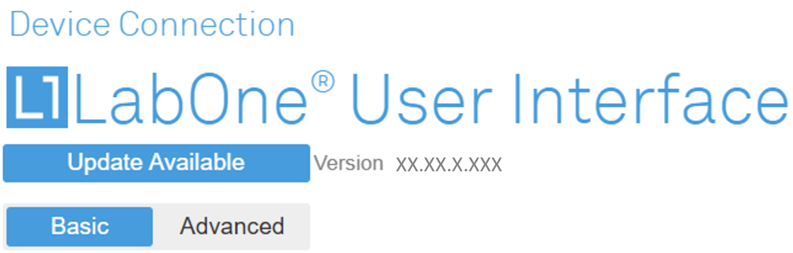
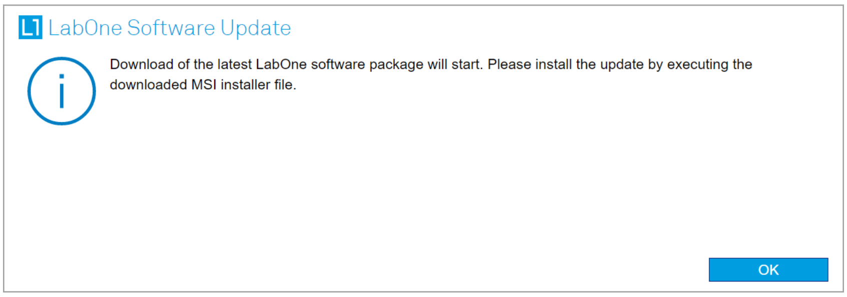


Figure 2.26: Device Connection dialog: LabOne update available



Download of the latest LabOne software package will start. Please install the update by executing the downloaded MSI installer file.

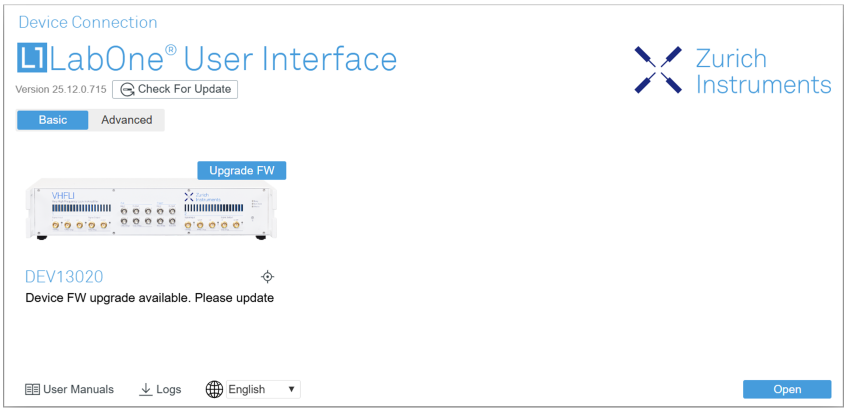
Figure 2.27: Download LabOne MSI using Automatic Update Check feature

## 2.6.3. Updating the Instrument Firmware

The LabOne software consists of both software that runs on your PC and software that runs on the instrument. In order to distinguish between the two, the latter will be called firmware for the rest of this document. When upgrading to a new software release, it's also necessary to update the instrument firmware.

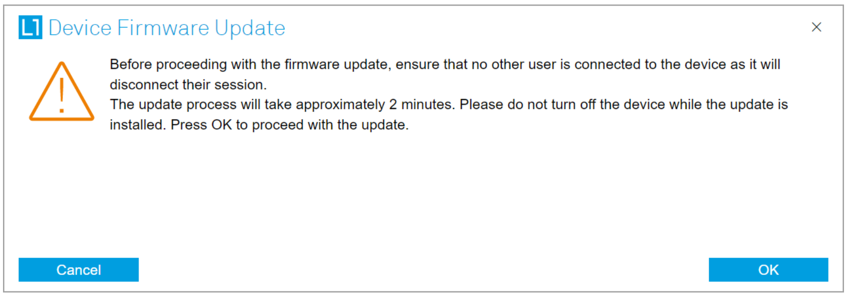
If the firmware needs an update, this is indicated in the Device Connection dialog of the LabOne user interface under Windows.

In the Basic view of the dialog, there will be a button "Upgrade FW" appearing together with the instrument icon as shown in Figure 2.28 on page 29 . In the Advanced view, there will be a link "Upgrade FW" in the Update column of the Available Devices table. Click on **Upgrade FW** to open the firmware update start-up dialog shown in Figure 2.29 on page 30 . The firmware upgrade takes approximately 2 minutes.



Device FW upgrade available. Please update

Figure 2.28: Device Connection dialog with available firmware update



Device Firmware Update

Before proceeding with the firmware update, ensure that no other user is connected to the device as it will disconnect their session.
The update process will take approximately 2 minutes. Please do not turn off the device while the update is installed. Press OK to proceed with the update.

Figure 2.29: Device Firmware Update start-up dialog

### Important

Do not disconnect the USB or 1GbE cable to the Instrument or power-cycle the Instrument during a firmware update.

If you encounter any issues while upgrading the instrument firmware, please contact Zurich Instruments at support@zhinst.com.

# 2.7. Troubleshooting

This section aims to help the user solve and avoid problems while using the software and operating the instrument.

## 2.7.1. Common Problems

Your VHFLI Instrument is an advanced piece of laboratory equipment which has many more features and capabilities than a traditional lock-in amplifier. In order to benefit from these, the user needs access to a large number of settings in the LabOne User Interface. The complexity of the settings might overwhelm a first-time user, and even expert users can get surprised by certain combinations of settings. To avoid problems, it's good to use the possibility to save and load settings in the Config Tab. This allows one to keep an overview by operating the instrument based on known configurations. This section provides an easy-to-follow checklist to solve the most common mishaps.

Table 2.8: Common Problems

| Problem | Check item |
|---|---|
| The software cannot be installed or uninstalled | Please verify you have administrator/root rights. |
| The software cannot be updated | Please use the Modify option in Windows Apps & Features functionality. In the software installer select Repair, then uninstall the old software version, and install the new version. |
| The Instrument does not turn on | Please verify the power supply connection and inspect the fuse. The fuse holder is integrated in the power connector on the back panel of the instrument. |
| The Instrument can't be connected over USB | Please verify that the instrument is connected through the "USB 1" port. The port labeled "USB 2" is not currently supported and will be enabled with a future LabOne release. |
| The Instrument has a high input noise floor (when connected to host computer by USB) | the USB cable connects the Instrument ground to computer ground, which might inject some unwanted noise to the measurements results. In this case it is recommended to use the Ethernet connection which is galvanically isolated using a UTP Cat 5 or 6 cable (UTP stands for "unshielded twisted pair"). |
| The Instrument performs poorly at low frequencies (below 100 kHz) | the signal inputs of the instrument might be set to AC operation. Please verify to turn off the AC switch in the Lock-in Tab or In / Out Tab. |
| The Instrument performs poorly during operation | the demodulator filters might be set too wide (too much noise) or too narrow (slow response) for your application. Please verify if the demodulator filter settings match your frequency versus noise plan. |

| Problem | Check item |
|---|---|
| The Instrument performs poorly during operation | clipping of the input signal may be occurring. This is detectable by monitoring the red LEDs on the front panel of the instrument or the Input Overflow (OVI) flags on the STATUS_TAB of the user interface. It can be avoided by adding enough margin on the input range setting (for instance 50% to 70% of the maximum signal peak). |
| The Instrument performs strangely when working with the GHF-MF Multi-frequency Option | it is easily possible to turn on more signal generators than intended. Check the generated Signal Output with the integrated oscilloscope and check the number of simultaneously activated oscillator voltages. |
| The Instrument performs close to specification, but higher performance is expected | After 2 years since the last calibration, a few analog parameters are subject to drift. This may cause inaccurate measurements. Zurich Instruments recommends re-calibration of the Instrument every 2 years. |
| The Instrument measurements are unpredictable | Please check the Status Tab to see if there is any active warning (red flag), or if one has occurred in the past (yellow flag). |
| The Instrument does not generate any output signal | verify that signal output switch has been activated in the Lock-in Tab or in the In / Out Tab. |
| The sample stream from the Instrument to the host computer is not continuous | Check the communication (COM) flags in the status bar. The three flags indicate occasional sample loss, packet loss, or stall. Sample loss occurs when a sampling rate is set too high (the instrument sends more samples than the interface and the host computer can absorb). The packet loss indicates an important failure of the communications to the host computer and compromises the behavior of the instrument. Both problems are prevented by reducing the sample rate settings. The stall flag indicates that a setting was actively changed by the system to prevent UI crash. |
| The LabOne User Interface does not start | Verify that the LabOne Data Server (`ziDataServer.exe`) and the LabOne Web Server (`ziWebServer.exe`) are running via the Windows Task Manager. The Data Server should be started automatically by `ziService.exe` and the Web Server should be started upon clicking "Zurich Instruments LabOne" in the Windows Start Menu.<br>If both are running, but clicking the Start Menu does not open a new User Interface session in a new tab of your default browser then try to create a new session manually by entering `127.0.0.1:8006` in the address bar of your browser. |
| The user interface does not start or starts but remains idle | Verify that the Data Server has been started and is running on your host computer. |
| The user interface is slow and the web browser process consumes a lot of CPU power | Make sure that the hardware acceleration is enabled for the web browser that is used for LabOne. For the Windows operating system, the hardware acceleration can be enabled in `Control Panel → Display → Screen Resolution`. Go to Advanced Settings and then Trouble Shoot. In case you use a NVIDIA graphics card, you have to use the NVIDIA control panel. Go to Manage 3D Settings, then Program Settings and select the program that you want to customize. |
| The Instrument does not appear in the Available Devices list in the LabOne User Interface or is not detected by the API (when connected to the local network through the 1GbE port) | Verify that the Instrument is properly connected to your network. If it is connected but still not visible, there may be a network configuration issue. Try connecting the host computer directly to the Instrument via USB. See Subsection 2.5.11, "Connecting to the Instrument", on page 22 for more information on connectivity and IP assignment. |

## 2.7.2. Location of the Log Files

The most recent log files of the LabOne Web and Data Server programs are most easily accessed by clicking on Logs in the Figure 2.12, "LabOne Device Connection dialog", on page 17 of the user interface. The Device Connection dialog opens on software start-up or upon clicking on Session Manager in the Config tab of the user interface.

The location of the Web and Data Server log files on disk are given in the sections below.

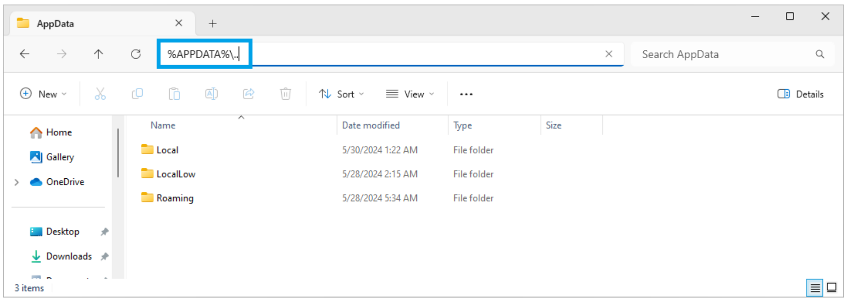
## Windows

The Web and Data Server log files on Windows can be found in the following directories.

- LabOne Data Server (`ziDataServer.exe`): `C:\Windows\ServiceProfiles\LocalService\AppData\Local\Temp\Zurich Instruments\LabOne\ziDataServerLog`
- LabOne Web Server (`ziWebServer.exe`): `C:\Users\[USER]\AppData\Local\Temp\Zurich Instruments\LabOne\ziWebServerLog`

### Note

The `C:\Users\[USER]\AppData` folder is hidden by default under Windows. A quick way of accessing it is to enter `%AppData%\..` in the address bar of the Windows File Explorer.



%APPDATA%\..

| Name | Date modified | Type | Size |
|---|---|---|---|
| Local | 5/30/2024 1:22 AM | File folder | |
| LocalLow | 5/28/2024 2:15 AM | File folder | |
| Roaming | 5/28/2024 5:34 AM | File folder | |

3 items

Figure 2.30: Using the

## Linux and macOS

The Web and Data Server log files on Linux or macOS can be found in the following directories.

- LabOne Data Server (`ziDataServer`): `/tmp/ziDataServerLog_[USER]`
- LabOne Web Server (`ziWebServer`): `/tmp/ziWebServerLog_[USER]`

## 2.7.3. Prevent web browsers from sleep mode

It often occurs that an experiment requires a long-time signal acquisition; therefore, the setup including the measurement instrument and LabOne software are left unattended. By default, many web browsers go to a sleep mode after a certain idle time which results in the loss of acquired data when using the web-based user interface of LabOne for measurement. Although it is recommended to take advantage of LabOne APIs in these situations to automate the measurement process and avoid using web browsers for data recording, it is still possible to adjust the browser settings to prevent it from entering the sleep mode. Below, you will find how to modify the settings of your preferred browser to ensure a long-run data acquisition can be implemented properly.

## Edge

1. Open **Settings** by typing `edge://settings` in the address bar
2. Select **System** from the icon bar.
3. Find the **Never put these sites to sleep** section of the **Optimized Performance** tab.
4. Add the IP address and the port of LabOne Webserver, e.g., `127.0.0.1:8006` or `192.168.73.98:80` to the list.

## Chrome

1. While LabOne is running, open a tab in Chrome and type `chrome://discards` in the address bar.
2. In the shown table listing all the open tabs, find LabOne and disable its **Auto Discardable** feature.
3. This option avoids discarding and refreshing the LabOne tab as long as it is open. To disable this feature permanently, you can use an extension from the Chrome Webstore.

## Firefox

1. Open **Advanced Preferences** by typing `about:config` in the address bar.
2. Look for `browser.tabs.unloadOnLowMemory` in the search bar.
3. Change it to **false** if it is **true**.

## Opera

1. Open **Settings** by typing `opera://settings` in the address bar.
2. Locate the **User Interface** section in the **Advanced** view.
3. Disable the **Snooze inactive tabs to save memory** option and restart Opera.

## Safari

1. Open **Debug** menu.
2. Go to **Miscellaneous Flags**.
3. Disable **Hidden Page Timer Throttling**.

# 3. Functional Overview

This chapter provides the overview of the features offered by the VHFLI Lock-in Amplifier. The first section contains a summary of the hardware and software features. The following section details the front panel and the back panel of the measurement instrument. The last section provides ordering support.

## 3.1. Features

### 3.1.1. Lock-in Operating Modes

- Internal reference mode
- External reference mode
- Dual-lock-in operation (two independent lock-in amplifiers in the same box)
- Triple-harmonic mode (simultaneous measurement at three harmonic frequencies)
- Arbitrary frequency mode (with VHF-MD option, simultaneous measurement at up to 8 arbitrary frequencies)

### 3.1.2. Signal Inputs

- 2 low-noise voltage inputs with variable range, selectable from ±5 mV to ±2 V peak (50 Ω)
- 2 low-noise current inputs with variable range, selectable from ±10 nA to ±10 mA
- Frequency range DC - 50 MHz (200 MHz with VHF-F200M option)
- Selectable AC/DC coupling for Voltage Inputs

### 3.1.3. Signal Outputs

- 2 low-noise voltage outputs, DC - 50 MHz (200 MHz with the VHF-F200M option)
- Variable output range, selectable from ±5 mV to ±5 V peak (into 50 Ω)

### 3.1.4. Demodulators & Reference

- Up to 8 dual-phase demodulators
- Up to 8 programmable numerical oscillators
- Up to 2 external reference signals
- Up to 2 input and up to 2 output trigger signals
- Individually programmable demodulator filters
- 128-bit internal processing
- 64-bit resolution demodulator sample
- 48-bit internal reference resolution

### 3.1.5. Auxiliary Input and Outputs

- 4 auxiliary outputs for user-defined signals, > 30 MHz bandwidth, 14 bit
- 2 auxiliary inputs, general purpose

### 3.1.6. High-speed Connectivity

- BNC connectors on front and SMA connectors on back panel
- USB-C high-speed host interface
- LAN/Ethernet 1 Gbit/s controller interface
- DIO: 32-bit digital input-output port
- Clock input/output connectors (10/100 MHz)

### 3.1.7. Extensive Time and Frequency Domain Analysis Tools

- Numeric tool
- Plotter
- Oscilloscope
- Sweeper and Frequency response analyzer
- FFT spectrum analyzer
- Data Acquisition tool
- Timeline Module

### 3.1.8. Software Features

- Web-based, high-speed LabOne® user interface with multi-instrument control
- Data server with multi-client support
- API for Python, MATLAB®, LabVIEW™, .NET, and C.

## 3.2. Front Panel Tour

The front panel BNC connectors and control LEDs are arranged as shown in Figure 3.1 on page 35 and listed in Table 3.1 on page 35 .

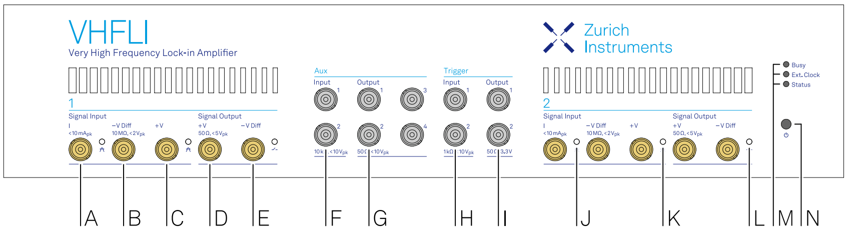


Figure 3.1: VHFLI Lock-in Amplifier front panel

Table 3.1: VHFLI Lock-in Amplifier front panel description

| Position | Label / Name | Description |
|---|---|---|
| A | Signal Input I | single-ended current input |
| B | Signal Input −V Diff | voltage input |
| | | single-ended mode: floating |
| | | differential mode: negative voltage input |
| C | Signal Input +V | voltage input |
| | | single-ended mode: single-ended voltage input |
| | | differential mode: positive voltage input |
| D | Signal Output +V | voltage output |
| | | single-ended mode: single-ended voltage output |
| | | differential mode: positive voltage output |
| E | Signal Output −V Diff | voltage output |
| | | single-ended mode: internally shorted to ground |
| | | differential mode: negative voltage output |
| F | Aux Inputs | auxiliary inputs 1 to 2 |
| G | Aux Outputs | auxiliary outputs 1 to 4 |
| H | Trig Inputs | trigger inputs 1 to 2 |
| I | Trig Outputs | trigger outputs 1 to 2, 3.3 V TTL |
| J | Signal Input I Over | this red LED indicates that the current input signal saturates the A/D converter and therefore the current input range must be increased or the signal must be attenuated |

| Position | Label / Name | Description |
|---|---|---|
| K | Signal Input V Over | this red LED indicates that the voltage input signal saturates the A/D converter and therefore the voltage input range must be increased or the signal must be attenuated |
| L | Signal Output ON | this blue LED indicates that the signal output is actively driven by the instrument |
| M | multicolor LEDs | off<br>Instrument off or uninitialized<br>blink<br>all LEDs blink for 5 seconds → indicator used by the Identify Device functionality |
| | Busy | unused |
| | Ext. Clock | off<br>10/100 MHz External Clock Signal not present/detected<br>blue<br>10/100 MHz External Clock Signal is present and locked on to<br>yellow<br>10/100 MHz External Clock Signal present, but not locked on to<br>red<br>10/100 MHz External Clock Signal present, but lock failed |
| | ZSync | unused |
| | Status | off<br>Instrument off or uninitialized<br>blue<br>Instrument is initialized and has no warnings or errors<br>yellow<br>Instrument has warnings<br>red<br>Instrument has errors |
| N | ⏻<br>Soft power button | Power button with incorporated status LED<br>off<br>Instrument off and disconnected from mains power<br>blue<br>- flashing rapidly (>1/sec): Firmware is starting<br>- flashing slowly (<1/sec): Firmware ready, waiting for connection<br>- constant: Instrument ready and active connection over USB or Ethernet<br>red<br>- breathing: Instrument off but connected to mains power → safe to power off using the rear panel switch, or restart using the soft power button<br>- flashing: Instrument booting up<br>- constant: Fatal error occurred |

## 3.3. Back Panel Tour

The back panel is the main interface for power, control, service and connectivity to other ZI instruments. Please refer to Figure 3.2 on page 37 and Table 3.2 on page 37 for the detailed description of the items.

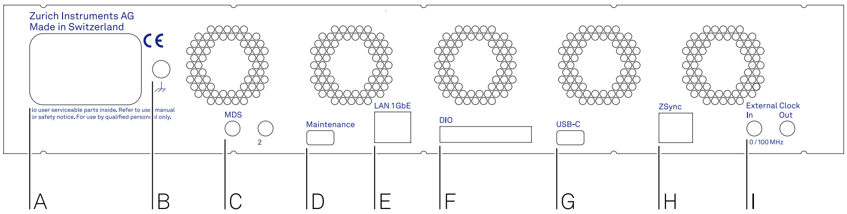


Figure 3.2: VHFLI Lock-in Amplifier back panel

Table 3.2: VHFLI Lock-in Amplifier back panel description

| Position | Label / Name | Description |
|---|---|---|
| A | AC 100 - 240 V | Power inlet, fuse holder, and power switch |
| B | ⏚<br>Earth ground | 4 mm banana jack connector for earth ground, electrically connected to the chassis and the earth pin of the power inlet |
| C | MDS 1 & 2 | SMA: bidirectional TTL ports for multi-device synchronization |
| D | Maintenance USB-C | Universal Serial Bus (USB) C port<br>**Attention:** This port offers limited data throughput and the USB-C port (see position G) is recommended for normal operation |
| E | LAN 1GbE | 1 Gbit LAN connector for instrument control and data acquisition |
| F | DIO 32bit | 32-bit digital input/output (DIO) connector |
| G | USB-C | Universal Serial Bus (USB) 3.0 port for instrument control and data acquisition |
| H | ZSync Primary | unused<br>**Attention:** This is not an Ethernet plug, connection to an Ethernet network might damage the instrument |
| I | External Clk In and Out | External clock (10 MHz/100 MHz) Input and Output for synchronization with other instruments |

# 3.4. Ordering Guide

Table 3.3 on page 37 provides an overview of the available VHFLI products. Upgradeable features are options that can be purchased anytime without the need to send the Instrument back to Zurich Instruments.

Table 3.3: VHFLI Instrument product codes for ordering

| Product code | Product name | Description | Field upgrade possible |
|---|---|---|---|
| VHFLI | VHFLI Lock-in Amplifier | base lock-in amplifier | - |
| VHF-F200M | VHF-F200M Frequency extension 200 MHz | option | yes |
| VHF-MD | VHF-MD Multi-demodulator | option | yes |
| VHF-MOD | VHF-MOD AM/FM Modulation | option | yes[1] |
| VHF-PID | VHF-PID Quad PID/PLL Controller | option | yes |
| VHF-AWG | VHF-AWG Arbitrary Waveform Generator | option | yes |

[1] Requires VHF-MD Multi-demodulator option

# 4. Tutorials

The tutorials in this chapter have been created to allow users to become more familiar with the basic technique of lock-in amplification, the operation of host-based lock-in amplifiers, the LabOne web browser based user interface, as well as some more advanced lock-in measurement techniques. In order to successfully carry out the tutorials, users are required to have certain laboratory equipment and basic equipment handling knowledge.

### Note

For all tutorials, you must have LabOne installed as described in the Chapter 2, "Getting Started", on page 1 .

## 4.1. Simple Loop

This tutorial is for people with no or little prior experience with Zurich Instruments lock-in amplifiers. By using a very basic measurement setup, this tutorial shows the most fundamental working principles of the VHFLI Instrument and the LabOne GUI in a step-by-step hands-on approach.

### Note

This tutorial is applicable to all VHFLI Instruments. No specific options are required.

### 4.1.1. Preparation

In this tutorial, you are asked to generate a single-ended signal with the VHFLI Instrument and measure that generated signal with the same instrument using an internal reference. This is done by connecting Signal Output +V to Signal Input +V of one of the two channels with a short BNC cable (ideally < 30 cm). Alternatively, it is possible to connect the generated signal at Signal Output +V to an oscilloscope by using a T-piece and an additional BNC cable. Figure 4.1 on page 38 displays a sketch of the hardware setup.

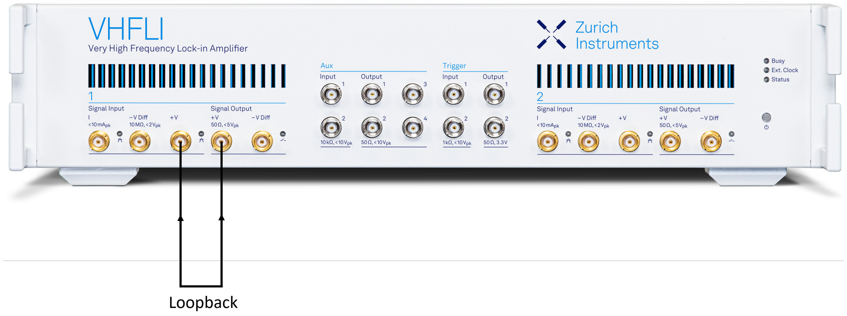


Figure 4.1: Setup for the Simple Loop tutorial.

Make sure the VHFLI is powered on and successfully connected to the controlling PC (see Chapter 2, "Getting Started", on page 1 for details). The tutorial can be started with the default instrument configuration (e.g. after a power cycle) and the default user interface settings (i.e. as is after pressing F5 in the browser). Once LabOne is started, the Setup Workspace is shown on the screen. In case some blocks are already present on the canvas, you can remove them by clicking on the “Delete all blocks” button in the top center of the Setup workspace.

### 4.1.2. Generate and visualize the Test Signal

In the Setup workspace, click on the Blocks button (the "+" symbol in the top bar) to open the dropdown menu. Select Signal Output 1 from the submenu as shown in Figure 4.2 on page 39 . This will add the Signal Output 1 block to the canvas, allowing you to generate the test signal.

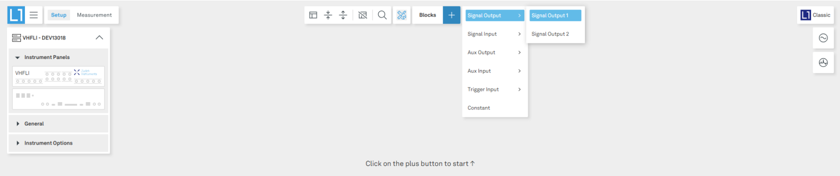
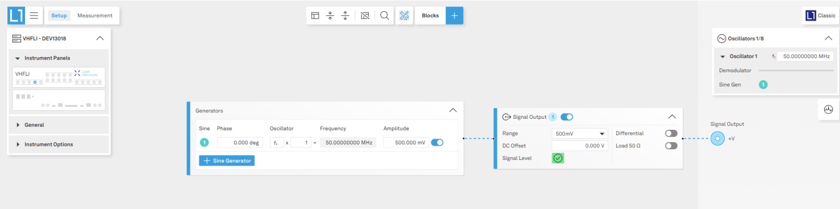


Figure 4.2: Adding the Signal Output 1 block via the Blocks menu.

After adding the block, configure the signal output to generate a 50 MHz sine wave with an amplitude of 500 mV by enabling the sine generator and setting the Range to 500 mV. Ensure the output is switched on by enabling the corresponding toggle. The oscillator settings will appear on the right-side Oscillators panel; see Figure 4.3 on page 39 .


Generators

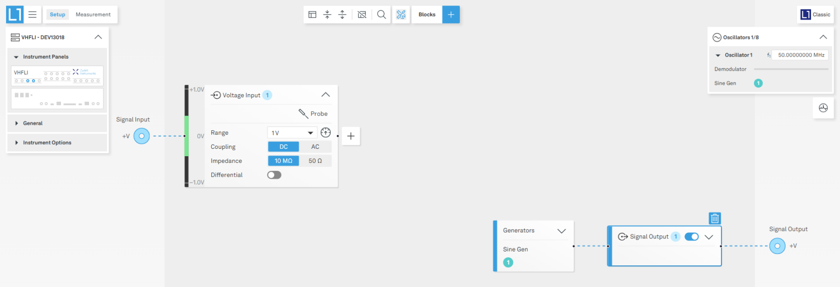
| Sine | Phase | Oscillator | Frequency | Amplitude |
|---|---|---|---|---|
| 1 | 0.000 deg | f₁ x 1 = | 50.00000000 MHz | 500.000 mV |

Signal Output 1
Range: 500mV
DC Offset: 0.000 V
Differential
Load 50 Ω
Signal Level

Oscillators 1/8
Oscillator 1 f₁ 50.00000000 MHz
Demodulator
Sine Gen 1

Figure 4.3: Configuring Signal Output with the sine generator set to 50 MHz and 500 mV amplitude. The oscillator settings are visible in the side panel.

To begin the demodulation process, first add the Voltage Input 1 block to the canvas by clicking on the "+" button on the top bar and selecting the corresponding Signal Input channel from the top bar. Once the block k is on the canvas, you can configure the input settings such as range, coupling (DC/ AC), and impedance, to match your experimental requirements. For this tutorial, set the input range to 1.0 V, and be sure to have the AC, 50 Ω, Diff and Float buttons unchecked; see Figure 4.4 on page 39 .

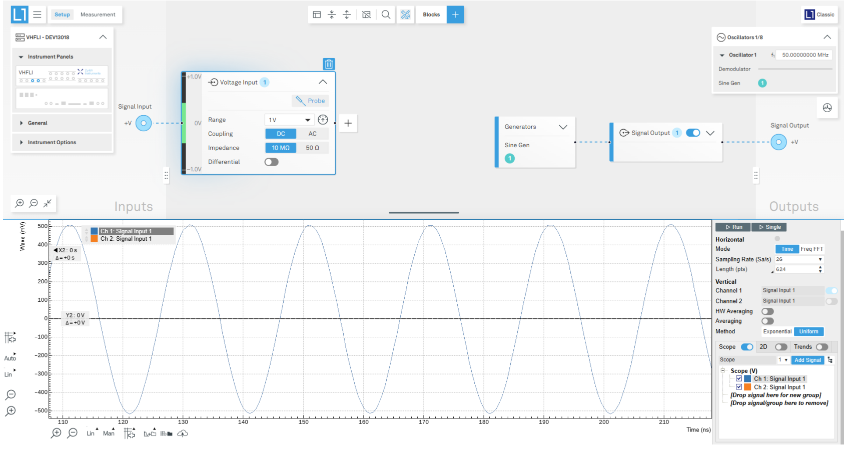

Voltage Input 1
Range: 1 V
Coupling: DC AC
Impedance: 10 MΩ 50 Ω
Differential

Oscillators 1/8
Oscillator 1 f₁ 50.00000000 MHz
Demodulator
Sine Gen 1

Figure 4.4: Adding a Voltage Input block and configuring input parameters.

The range setting ensures that the analog amplification on the Voltage Input +V is set such that the resolution of the input analog-to-digital converter is used efficiently without clipping the signal. This optimizes the dynamic range.

The incoming signal can now be observed over time by using the Scope tool, by clicking on the corresponding probe icon.


Horizontal
Mode: Time Freq FFT
Sampling Rate (Sa/s): 2G
Length (pts): 624

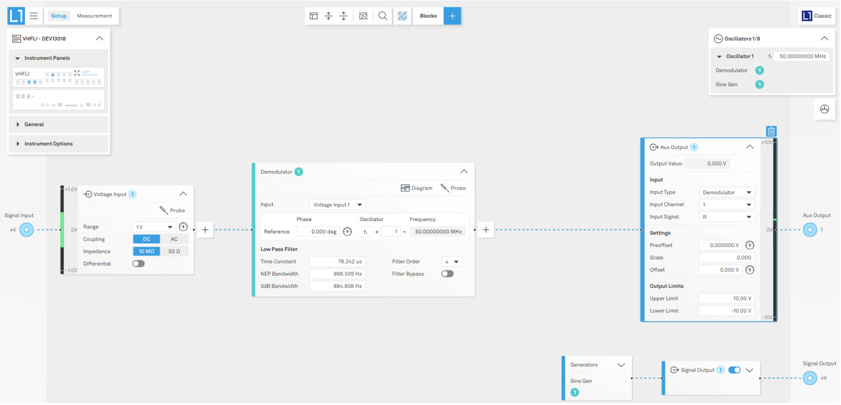
Vertical
Channel 1: Signal Input 1
Channel 2: Signal Input 1
HW Averaging
Averaging
Method: Exponential Uniform

Scope 2D Trends
Scope 1 Add Signal
Scope (V)
☑ Ch 1: Signal Input 1
☑ Ch 2: Signal Input 1
[Drop signal here for new group]
[Drop signal/group here to remove]

Figure 4.5: Observing the input signal using the probe feature. The input sine wave is visible in the scope window.

### 4.1.3. Demodulate the Test Signal

Now, we are ready to use the VHFLI to demodulate the input signal and measure its amplitude and phase. To this end, you add a Demodulator block and connect it to the signal input by clicking on the "+" on the right-hand side of the Voltage Input block. On the Demodulator block, you can adjust the demodulator's low pass filter settings, including time constant, bandwidth, and filter order, to optimize the signal processing for your measurement needs. In this example a 3dB low-pass filter bandwidth of 1 kHz is used, with a 4th order filter. The demodulator uses the same internal oscillator set up in the signal generation steps. The measurement result can then be routed to the Auxiliary Output (by adding the corresponding connection to the demodulator block) or streamed internally to the host computer.



Voltage Input 1
Range: 1V
Coupling: DC
Impedance: 10 MΩ
Differential: off

Demodulator 1
Input: Voltage Input 1
Reference — Phase: 0.000 deg; Oscillator: f1 x 1; Frequency: 50.00000000 MHz
Low Pass Filter
Time Constant: 78.242 us
NEP Bandwidth: 998.505 Hz
3dB Bandwidth: 884.808 Hz
Filter Order: 4
Filter Bypass: off

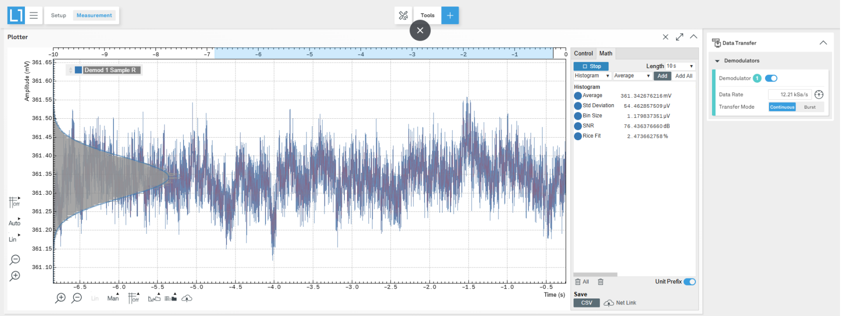
Oscillators 1/8
Oscillator 1: f1 50.00000000 MHz
Demodulator 1
Sine Gen 1

Aux Output 1
Output Value: 0.000 V
Input
Input Type: Demodulator
Input Channel: 1
Input Signal: R
Settings
Preoffset: 0.000000 V
Scale: 0.000
Offset: 0.000 V
Output Limits
Upper Limit: 10.00 V
Lower Limit: -10.00 V

Figure 4.6: Block diagram representing both the signal generation and demodulation data flow.

### 4.1.4. Plot the measurement results

To view and analyze your measurement results, switch to the Measurement workspace by clicking on the Measurement tab at the top left of the interface. In this workspace, you have access to a variety of powerful visualization tools. For example, you can use the Plotter as shown in Figure 4.7 on page 40 to display the time trace of your demodulated data, allowing you to observe amplitude variations directly over time. Alternatively, you can select the Spectrum Analyzer to gain insights into the frequency domain representation of your measured signals, or other tools, by clicking on the "+" button in the top center part of the interface.



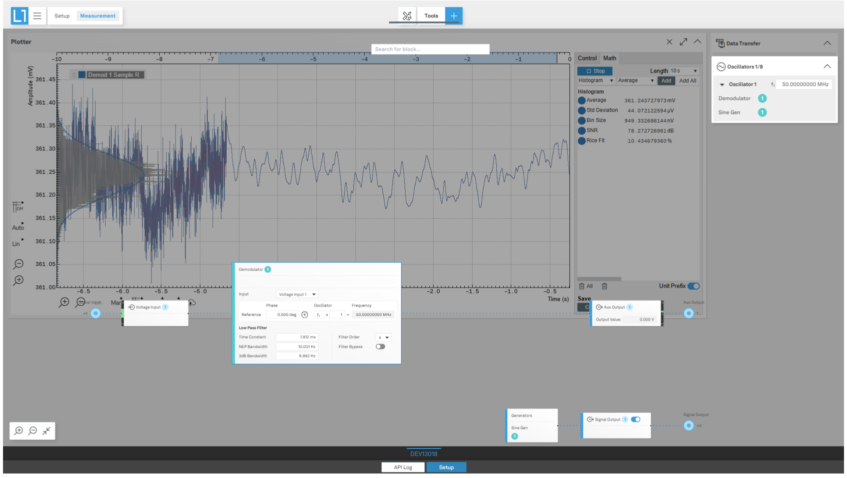
Control / Math
Length: 10 s
Histogram
Average: 361.342676216 mV
Std Deviation: 54.462857509 µV
Bin Size: 1.179837351 µV
SNR: 76.436376660 dB
Rice Fit: 2.473662758 %

Data Transfer
Demodulators
Demodulator 1: on
Data Rate: 12.21 kSa/s
Transfer Mode: Continuous

Figure 4.7: Plotter showing the amplitude of the demodulated signal in the time domain; data transfer is enabled on the right.

On the right-side panel, make sure to enable the Data Transfer toggle for the relevant demodulator channel to ensure your data is sent to the visualization tools. It is important to set a suitable Data Rate for data transfer. This rate should be matched to the low pass filter setting of your demodulator to achieve optimal sampling; note that you can click the button next to the data rate setting to let the software automatically adjust the data rate for you. Configuring these settings appropriately ensures you capture all relevant measurement information without excessive data volume or with insufficient sampling.

The average value of the measured signal in the plotter is about 351 mV, as expected from a 500 mV peak to peak sinusoidal input, taking onto account the typical √2 factor stemming from the demodulation process. Additional insights, as well as statistics on the signal, can be gained thanks to the Math sub-tab within the Plotter instance. See Figure 4.8 on page 41 for an example of the Measurement workspace.



Length 10 s

Histogram ▾ Average ▾

Histogram

| | |
|---|---|
| Average | 361.243727973 mV |
| Std Deviation | 44.072122594 µV |
| Bin Size | 949.332686144 nV |
| SNR | 78.272726961 dB |
| Rice Fit | 10.434679360 % |

Oscillators 1/8

Oscillator 1 $f_1$ 50.00000000 MHz

Demodulator 1

Sine Gen 1

Demodulator 1

Input: Voltage Input 1

| | Phase | Oscillator | Frequency |
|---|---|---|---|
| Reference | 0.000 deg | $f_1$ x 1 = | 50.00000000 MHz |

Low Pass Filter

| | | | |
|---|---|---|---|
| Time Constant | 7.812 ms | Filter Order | 4 |
| NEP Bandwidth | 10.001 Hz | Filter Bypass | ○ |
| 3dB Bandwidth | 8.862 Hz | | |

Aux Output 1

Output Value: 0.000 V

Sine Gen 1

Signal Output 1

Figure 4.8: Overlaying the setup in the measurement workspace, enabling real-time tuning and visualization.

For real-time parameter adjustments (e.g. different value of the low pass filter time constant) and immediate feedback, you can overlay your setup diagram within the measurement workspace by clicking "Setup" in the lower part of the screen (see Figure 4.8 on page 41 ). This allows you to tweak parameters live, with changes reflected instantly in your chosen visualization tool. In the example in Figure 4.8 on page 41 , the low-pass filter bandwidth is changed to a value of about 10 Hz, and the corresponding effect on the demodulated data is reflected in the underlying plotter. Alternatively, you can add the setup as a standalone block in the measurement workspace by clicking on the “+” button and selecting “Setup”.

# 5. Specifications

## Important

Unless otherwise stated, all specifications apply after 30 minutes of instrument warm-up.

## Important

Important changes in the specification parameters are explicitly mentioned in the revision history of this document.

## 5.1. General Specifications

Table 5.1: General and storage

| Parameter | Description |
|---|---|
| Storage temperature | −25 °C to +65 °C |
| Storage relative humidity (non-condensing) | up to 95% RH |
| Operating temperature | +5 °C to +40 °C |
| Operating relative humidity (non-condensing) | up to 90% RH |
| Specification temperature | +18 °C to +28 °C |
| Power consumption | 100 W (max 140 W) |
| Operating environment | IEC61010, indoor location, installation category II, pollution degree 2 |
| Operating altitude | up to 2000 m |
| Power inlet fuses | 250 V, 2 A, fast acting, 5 × 20 mm |
| Power supply AC line | 100–240 V (±10%), 50/60 Hz, 1.4 A |
| Dimensions (W × D × H) | 32.5 × 44.5 × 10 cm (with handle); 19-inch rack compatible |
| Weight | 6.3 kg (14 lb) |
| PC operating systems | See LabOne Compatibility |
| Recommended calibration interval | 2 years |

Table 5.2: Signal inputs

| Parameter | Description |
|---|---|
| Number of Lock-in Units | 2 |
| Channel type | 1 Voltage (differential and single-ended) and 1 Current per Lock-in Unit |
| Frequency range | DC–50 MHz (DC–200 MHz with VHF-F200M option) |
| A/D conversion | 14 bit, 2 GSa/s |
| Connectors | BNC |
| Input impedance | 50 Ω or 10 MΩ, selectable |
| Input voltage noise | < 3 nV/√Hz (> 50 kHz) |
| Input current noise | 50 fA/√Hz @ 1 kHz, 10 nA input range |

| Parameter | Description |
|---|---|
| Voltage input range AC+DC | ±5 mV to ±2 V |
| Voltage input AC coupling cutoff frequency | ~200 kHz (50 Ω) |
| Voltage input DC offset (max) | ±0.5 mV or 1% of range, whichever is bigger |
| Current input ranges | ±10 nA to ±10 mA |
| Voltage input harmonic distortion | < –70 dBc (1, 0.1, 0.01 V ranges); < –55 dBc (1 mV range) |
| Voltage input non-harmonic distortion | < –70 dBc (1, 0.1 V ranges); < –60 dBc (0.01 V range); < –45 dBc (1 mV range) |

Table 5.3: Current signal input performance

| Current input range | Transimpedance gain | Bandwidth (–3 dB) | Input impedance at DC | Input noise (typical) |
|---|---|---|---|---|
| 10 mA | | > 80 MHz | | 400 pA/√Hz @ 1 kHz |
| 1 mA | | 50 MHz | | 45 pA/√Hz @ 10 MHz |
| 100 µA | | 50 MHz | | 10 pA/√Hz @ 1 MHz |
| 10 µA | | 10 MHz | | 700 fA/√Hz @ 100 kHz |
| 1 µA | | 10 MHz | | 530 fA/√Hz @ 100 kHz |
| 100 nA | | 100 kHz | | 70 fA/√Hz @ 1 kHz |
| 10 nA | | 100 kHz | | 50 fA/√Hz @ 1 kHz |

Table 5.4: Signal outputs

| Parameter | Description |
|---|---|
| Number of output channels | 2 |
| Output type | Voltage, differential and single-ended |
| Frequency range | 30 MHz (0.1 dB), > 230 MHz (3 dB) |
| Output ranges | ±5 mV, ±50 mV, ±500 mV, ±5 V |
| D/A conversion | 14 bit, 2 GSa/s |
| Connectors | BNC |
| Signal Adder | Auxiliary input (digital adder) |
| Voltage output impedance | 50 Ω |
| Voltage output non-harmonic distortion | ≤ –60 dBc (5 mV range) |
| Voltage output noise floor | 40 nV/√Hz (5V range), 4 nV/√Hz (5 mV range) |
| Voltage output amplitude accuracy | 1% |
| Voltage output phase noise | –130 dBc/Hz (100 Hz offset), –140 dBc/Hz (1 kHz offset) @ 10 MHz |
| Voltage output Common Mode Rejection Ration (CMMR) | -44 dB or better |
| Voltage output non-harmonic distortion | -60 dBc or better (5 mV range) |

Table 5.5: Reference and triggers

| Parameter | Description |
|---|---|
| Triggers | 2 inputs, 2 outputs |
| Trigger connectors | BNC (front panel) |
| Reference frequency resolution | 7.1 µHz |
| Reference phase angle resolution | 2.7 µº |
| Trigger inputs impedance | low: 50 Ω, high: 1 kΩ |
| Trigger inputs levels | Low impedance: ±5 V, high: ±10 V |
| Trigger inputs hysteresis | >60 mV typ. |
| Trigger outputs impedance | 50 Ω, 1 kΩ |
| Trigger outputs levels | 3.3 V TTL |
| Internal oscillator frequency | 10/100 MHz |
| Internal oscillator initial accuracy | max ±1 ppm |
| Internal oscillator aging | 3 ppm max. over 10 years |
| Internal oscillator short term stability | 1e-9 1/s (Allan Variance) |
| Internal oscillator temperature coefficient | ±50 ppb, 0 ºC to +50 ºC |
| Internal oscillator phase noise | Max. -120 dBc/Hz @100 Hz, -145 dBc/Hz @1 kHz |

Table 5.6: Demodulators

| Parameter | Description |
|---|---|
| Frequency range | DC–50 MHz (DC–200 MHz with VHF-F200M option) |
| Number of demodulators | 2 dual-phase (8 with VHF-MD option) |
| Number of oscillators | 2 (8 with VHF-MD option) |
| Dynamic reserve | > 120 dB |
| Input signals | Voltage input, current input, auxiliary inputs, constant |
| Output sample rate on LAN/USB 3.0 | Up to 2 MSa/s total (all demodulators); 25 MSa/s in triggered mode |
| Output sample rate on Auxiliary outputs | 50 MSa/s per high-speed auxiliary output; 14 bit |
| Filter time constant | 14 ns to 21 s |
| Filter bandwidth | 3.2 mHz to 11 MHz |
| Filter slope | 6, 12, 18, 24 dB/Oct |
| Number of harmonics | 1 to 131071 |

Table 5.7: Auxiliary ports

| Parameter | Description |
|---|---|
| Auxiliary inputs | 2 channels, ±10 V, > 150 MHz BW; digital adder with voltage inputs |
| Auxiliary outputs | 4 channels, ±10 V, > 30 MHz BW |
| Connectors | BNC |
| Auxiliary outputs signals | R, Θ, X, Y, or user-defined |
| Amplitude | ± 10 V, ±1 V, ± 100 mV (in and out) |
| A/D conversion | 14 bit, 2 GSa/s |
| D/A conversion | 14 bit, 2 GSa/s |
| Auxiliary inputs impedance | 50 Ω |
| Auxiliary outputs impedance | 50 Ω |
| Auxiliary outputs drive current | 100 mA |

| Parameter | Description |
|---|---|
| Auxiliary outputs resolution | 0.012 mV |

Table 5.8: Maximum ratings

| Parameter | Lower | Upper |
|---|---|---|
| Damage threshold Current Input | -5 V | +5 V |
| Damage threshold Voltage Input +V/-V Diff | -5 V | +5 V |
| Damage threshold Signal Output | -6 V | +6 V |
| Damage threshold Trig In (50 Ω) | -5 V | 5 V |
| Damage threshold Trig In (high impedance) | -11 V | 11 V |
| Damage threshold Trig Out | -0.7 V | 4 V |
| Damage threshold Aux In | -12 V | + 12 V |
| Damage threshold Aux Out | -12 V | + 12 V |
| Damage threshold External Clock In (DC) | -3 V | +3 V |
| Damage threshold External Clock Out (DC) | -3 V | +3 V |
| DIO In / Out in default configuration 3.3 V CMOS/TTL | -0.5 V | 3.8 V |

## Important

Operation outside the Damage Thresholds can cause permanent device damage. Damage Threshold does not imply functional operation of the device at these or any other conditions beyond those listed under specifications. If used outside the specifications but within the Damage Thresholds, the device can not be fully functional, and this can affect device reliability, functionality, performance, and shorten the device lifetime.

Table 5.9: Connectivity and others

| Parameter | Description |
|---|---|
| Host connection | LAN/Ethernet 1 Gbit/s; USB 3.0 |
| Digital I/O | 32 bits, 50 MHz |
| Clocking | 10 MHz or 100 MHz input and output |
| External clock input connector | SMA |
| External clock input impedance | 50 Ω |
| External clock input voltage | 10 MHz: 200 mV min, 1.4 V max; 100 MHz: 180 mV min, 1.4 V max |
| Internal clock output connector | SMA |
| Internal clock output impedance | 50 Ω (AC-coupled) |
| Internal clock output amplitude | 2 Vpp min, 5 Vpp max |

Table 5.10: Scope

| Parameter | Description |
|---|---|
| Input channels | Voltage input, current input, auxiliary inputs |
| Modes | Time domain, frequency domain (FFT) |
| Number of display channels | 4 |
| Trigger channels | Sig In, Curr. In, Aux In, Trig In |
| Trigger mode | Edge |
| Sampling rates | 15.3 kSa/s to 2 GSa/s |
| Vertical resolution | 14 bits |

| Parameter | Description |
|---|---|
| BW limit mode, vertical resolution increase | Sample decimation, averaging |
| Cursor math | Location, area, wave, peak, tracking, histogram |

Table 5.11: Spectrum Analyser

| Parameter | Description |
|---|---|
| Centre frequency range | 0–50 MHz (200 MHz w/ F200M option) |
| Spectrum modes | FFT(X+iY), FFT(R), FFT(θ), FFT((dθ/dt)/2π) |
| Statistical options | Amplitude, spectral density, power |
| Averaging modes | None, exponential moving average |
| Max number of samples per spectrum | 8.4 MSa |
| Max span | 1.5 MSa/s |
| Window functions | Rectangular, Hann, Hamming, Blackman Harris, Flat Top, Exponential, cosine, cosine squared |
| Cursor math | Location, area, tracking, plot area, peak, trough, histogram |

Table 5.12: Sweeper

| Parameter | Description |
|---|---|
| Scan parameters | Oscillator frequency; demodulated phase shift; auxiliary offset, signal output amplitude, phase and offset; PID setpoint, MOD carrier amplitude, index and sideband amplitude |
| Sweep ranges | Full range, linear and logarithmic |
| Parameter sweep resolution | Arbitrary, defined by start/stop value and number of points |
| Display parameters | Demodulator output (X, Y, R, θ, f) |
| Display options | Single plot, dual plot (e.g., Bode plot), multi-trace |
| Statistical options | Amplitude, spectral density, power |
| Preset measurement modes | Parameter sweep, noise amplitude, FRA, 3-omega-sweep |

Table 5.13: Arithmetic Unit

| Parameter | Description |
|---|---|
| Number of units | 2 |
| Input signals | Demodulator output (X, Y, R), constant |
| Operations | Addition, multiplication, division |
| Scaling factors | Fixed coefficient |
| Output signals | PID |
| Data rate to computer | N/A |

Table 5.14: Modulation analysis option

| Parameter | Description |
|---|---|
| Carrier/Modulation frequency range | 7.1 µHz – 50 MHz / 7.1 µHz – 50 MHz (200 MHz w/ F200M) |
| Sideband frequency | fs = m * fc ± n * fm |
| m,n: harmonic analysis | m,n = 1 to max harmonic |
| AM modulation index | h_AM = A_m / A_c |
| Amplitude of modulation signal | A_c + A_m < V_range |

| Parameter | Description |
|---|---|
| FM modulation index | h_FM = f_p / f_m |

Table 5.15: PID/PLL Option

| Parameter | Description |
|---|---|
| Controllers | 4 PID controllers with PLL capability |
| PLL frequency range | 1 Hz – 50 MHz (200 MHz with the VHF-F200M) |
| PLL Programmable center frequency | 1 Hz – 50 MHz (200 MHz w/ VHF-F200M) |
| Programmable PLL bandwidth | — |
| Typical PLL phase noise (at 100 Hz) | — |
| PID (regulation or closed-loop) bandwidth | — |
| PID Total update rate | up to 25 MSa/s |
| PID inputs | X, Y, R, and Θ, DFRT (amplitude AU); (More in following L1 updates) |
| PID outputs | Signal output amplitudes, Osc. freq. (More in following L1 updates) |

# 6. Device Node Tree

This chapter contains reference documentation for the settings and measurement data available on VHFLI Instruments. Whilst other chapters in this manual describes many of these settings in terms of the features available in the LabOne User Interface, this chapter describes them on the device level and provides a hierarchically organized and comprehensive list of device functionality.

Since these settings and data streams may be written and read using the LabOne APIs (Application Programming Interfaces) this chapter is of particular interest to users who would like to perform measurements programmatically via LabVIEW, Python, MATLAB, .NET or C.

Please see:

- Section 6.1, "Introduction", on page 48 for an introduction of how the instrument's settings and measurement data are organized hierarchically in the Data Server's so-called "Node Tree".
- Section 6.2, "Reference Node Documentation", on page 51 for a reference list of the settings and measurement data available on VHFLI Instruments, organized by branch in the Node Tree.

## 6.1. Introduction

This chapter provides an overview of how an instrument's configuration and output is organized by the Data Server.

All communication with an instrument occurs via the Data Server program the instrument is connected to (see Subsection 2.5.1, "LabOne Software Architecture", on page 15 for an overview of LabOne's software components). Although the instrument's settings are stored locally on the device, it is the Data Server's task to ensure it maintains the values of the current settings and makes these settings (and any subscribed data) available to all its current clients. A client may be the LabOne User Interface or a user's own program implemented using one of the LabOne Application Programming Interfaces, e.g., Python.

The instrument's settings and data are organized by the Data Server in a file-system-like hierarchical structure called the node tree. When an instrument is connected to a Data Server, its device ID becomes a top-level branch in the Data Server's node tree. The features of the instrument are organized as branches underneath the top-level device branch and the individual instrument settings are leaves of these branches.

For example, the auxiliary outputs of the instrument with device ID "dev1000" are located in the tree in the branch:

```
/dev1000/auxouts/
```

In turn, each individual auxiliary output channel has its own branch underneath the "AUXOUTS" branch.

```
/dev1000/auxouts/0/
/dev1000/auxouts/1/
/dev1000/auxouts/2/
/dev1000/auxouts/3/
```

Whilst the auxiliary outputs and other channels are labelled on the instrument's panels and the User Interface using 1-based indexing, the Data Server's node tree uses 0-based indexing. Individual settings (and data) of an auxiliary output are available as leaves underneath the corresponding channel's branch:

```
/dev1000/auxouts/0/demodselect
/dev1000/auxouts/0/limitlower
/dev1000/auxouts/0/limitupper
/dev1000/auxouts/0/offset
/dev1000/auxouts/0/outputselect
/dev1000/auxouts/0/preoffset
/dev1000/auxouts/0/scale
/dev1000/auxouts/0/value
```

These are all individual node paths in the node tree; the lowest-level nodes which represent a single instrument setting or data stream. Whether the node is an instrument setting or data-stream and which type of data it contains or provides is well-defined and documented on a per-node basis in the Reference Node Documentation section in the relevant instrument-specific user manual. The different properties and types are explained in Subsection 6.1.1, "Node Properties and Data Types ", on page 49 .

For instrument settings, a Data Server client modifies the node's value by specifying the appropriate path and a value to the Data Server as a (path, value) pair. When an instrument's setting is changed in the LabOne User Interface, the path and the value of the node that was changed are displayed in the Status Bar in the bottom of the Window. This is described in more detail in Subsection 6.1.2, "Exploring the Node Tree", on page 49 .

### Module Parameters

LabOne Core Modules, such as the Sweeper, also use a similar tree-like structure to organize their parameters. Please note, however, that module nodes are not visible in the Data Server's node tree; they are local to the instance of the module created in a LabOne client and are not synchronized between clients.

## 6.1.1. Node Properties and Data Types

A node may have one or more of the following properties:

| Property | Description |
|---|---|
| Read | Data can be read from the node. |
| Write | Data can be written to the node. |
| Setting | The node corresponds to a writable instrument configuration. The data of these nodes are persisted in snapshots of the instrument and stored in the LabOne XML settings files. |
| Streaming | A node with the read attribute that provides instrument data, typically at a user-configured rate. The data is usually a more complex data type, for example demodulator data is returned as `ZIDemodSample`. A full list of streaming nodes is available in the Programming Manual in the Chapter Instrument Communication. Their availability depends on the device class (e.g. MF) and the option set installed on the device. |

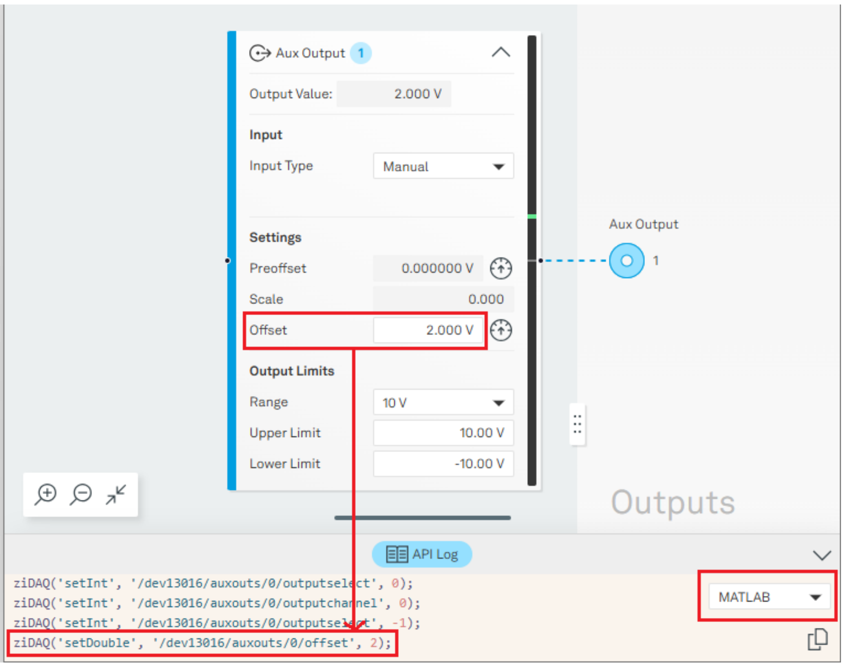
A node may contain data of the following types:

| Type | Description |
|---|---|
| Integer | Integer data. |
| Double | Double precision floating point data. |
| String | A string array. |
| Integer (enumerated) | As for Integer, but the node only allows certain values. |
| Composite data type | For example, `ZIDemodSample`. These custom data types are structures whose fields contain the instrument output, a timestamp and other relevant instrument settings such as the demodulator oscillator frequency. Documentation of custom data types is available in |

## 6.1.2. Exploring the Node Tree

### In the LabOne User Interface

A convenient method to learn which node is responsible for a specific instrument setting is to check the Command Log history in the bottom of the LabOne User Interface. The command in the Status Bar gets updated every time a configuration change is made. Figure 6.1 on page 50 shows how the equivalent MATLAB command is displayed after modifying the value of the auxiliary output 1's offset. The format of the LabOne UI's command history can be configured in the "API Log" section of LabOne User Interface. The commands can be displayed in MATLAB, Python and .NET programming languages. The entire history generated in the current UI session can be viewed by clicking the "API Log" button.



Output Value: 2.000 V

Input Type: Manual

Preoffset: 0.000000 V

Scale: 0.000

Offset: 2.000 V

Range: 10 V

Upper Limit: 10.00 V

Lower Limit: -10.00 V

```
ziDAQ('setInt', '/dev13016/auxouts/0/outputselect', 0);
ziDAQ('setInt', '/dev13016/auxouts/0/outputchannel', 0);
ziDAQ('setInt', '/dev13016/auxouts/0/outputselect', -1);
ziDAQ('setDouble', '/dev13016/auxouts/0/offset', 2);
```

MATLAB

Figure 6.1: When a device's configuration is modified in the LabOne User Interface, the API Log section displays the equivalent command to perform the same configuration via a LabOne programming interface.

## In a LabOne Programming Interface

A list of nodes (under a specific branch) can be requested from the Data Server in an API client using the `listNodes` command (MATLAB, Python, .NET) or `ziAPIListNodes()` function (C API). Please see each API's command reference for more help using the `listNodes` command. To obtain a list of all the nodes that provide data from an instrument at a high rate, so-called streaming nodes, the `streamingonly` flag can be provided to `listNodes`. More information on data streaming and streaming nodes is available in the LabOne Programming Manual.

The detailed descriptions of nodes that is provided in Section 6.2, "Reference Node Documentation", on page 51 is accessible directly in the LabOne MATLAB or Python programming interfaces using the "help" command. The `help` command is `daq.help(path)` in Python and `ziDAQ('help', path)` in MATLAB. The command returns a description of the instrument node including access properties, data type, units and available options. The "help" command also handles wildcards to return a detailed description of all nodes matching the path. An example is provided below.

```
    daq = zhinst.core.ziDAQServer('localhost', 8004, 6)
    daq.help('/dev1000/auxouts/0/offset')
    # Out:
    # /dev1000/auxouts/0/OFFSET#
    # Add the specified offset voltage to the signal after scaling. Auxiliary
Output
    # Value = (Signal+Preoffset)*Scale + Offset
    # Properties: Read, Write, Setting
    # Type: Double
    # Unit: V
```

## 6.1.3. Data Server Nodes

The Data Server has nodes in the node tree available under the top-level `/zi/` branch. These nodes give information about the version and state of the Data Server the client is connected to. For example, the nodes:

- `/zi/about/version`
- `/zi/about/revision`

are read-only nodes that contain information about the release version and revision of the Data Server. The nodes under the `/zi/devices/` list which devices are connected, discoverable and visible to the Data Server.

The nodes:

- `/zi/config/open`
- `/zi/config/port`

are settings nodes that can be used to configure which port the Data Server listens to for incoming client connections and whether it may accept connections from clients on hosts other than the localhost.

Nodes that are of particular use to programmers are:

- `/zi/debug/logpath` - the location of the Data Server's log in the PC's file system,
- `/zi/debug/level` - the current log-level of the Data Server (configurable; has the Write attribute),
- `/zi/debug/log` - the last Data Server log entries as a string array.

The Global nodes of the LabOne Data Server are listed in the LabOne API User Manual.

# 6.2. Reference Node Documentation

This section describes all the nodes in the data server’s node tree organized by branch.

## 6.2.1. AU

### /dev..../au/amplitudes/n/result/max

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the maximum normalized output result value. It can be between -1 and 1.

### /dev..../au/amplitudes/n/result/min

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the minimum normalized output result value. It can be between -1 and 1.

### /dev..../au/amplitudes/n/result/over

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | Dependent |

Reports the number of times the output result of the arithmetic unit was overloaded.

### /dev..../au/amplitudes/n/result/value

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | Dependent |

Output result value of the arithmetic unit, updated at a low rate.

### /dev..../au/amplitudes/n/scalings/n

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Double |
| Unit: | None |

Scaling value applied to the product of two selected inputs of the arithmetic unit.

### /dev..../au/amplitudes/n/sources/n/channel

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Integer (64 bit) |
| Unit: | None |

Selects the channel of the selected input source.

### /dev..../au/amplitudes/n/sources/n/select

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Integer (enumerated) |
| Unit: | None |

Selects the input source for the arithmetic unit.

| | |
|---|---|
| -1 | "const_1": Select a constant 1 as input. |
| 0 | "demod_x": Select Demod X as input. |
| 1 | "demod_y": Select Demod Y as input. |
| 2 | "demod_r": Select Demod R as input. |

## 6.2.2. AUXINS

### /dev..../auxins/n/autorange

| | |
|---|---|
| Properties: | Read, Write |
| Type: | Integer (64 bit) |
| Unit: | None |

Automatic adjustment of the Range to approximately two times the maximum signal input amplitude.

### /dev..../auxins/n/delaycompensation/delay

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Double |
| Unit: | s |

Compensation for signal propagation delays to achieve a flatter phase response. The default value compensates for the propagation delay resulting from the signal output to the auxiliary input with a short cable in between. Every additional meter of cable would add ~5 ns propagation delay, which can be compensated using this node.

### /dev..../auxins/n/delaycompensation/reset

| | |
|---|---|
| Properties: | Read, Write |
| Type: | Integer (64 bit) |
| Unit: | None |

Resets the compensation delay to the default value.

### /dev..../auxins/n/max

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the maximum normalized voltage measured on this channel. It can be between -1 and 1. To prevent signal clipping and overvoltage, it is advised to keep it between -0.9 and 0.9.

### /dev..../auxins/n/min

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the minimum normalized voltage measured on this channel. It can be between -1 and 1. To prevent signal clipping and overvoltage, it is advised to keep it between -0.9 and 0.9.

### /dev..../auxins/n/on

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enables the auxiliary input.

### /dev..../auxins/n/range

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | dBm |

Sets the range of the Auxiliary Input power. The instrument has a fixed number of ranges with 5 dBm steps and will round the entered value to the closest available range.

### /dev..../auxins/n/value

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | V |

Voltage present at the auxiliary input. The value is filtered and updated at a low rate.

## 6.2.3. AUXOUTS

### /dev..../auxouts/n/limitlower

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Lower limit for the signal at the Auxiliary Output. A smaller value will be clipped.

### /dev..../auxouts/n/limitupper

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Upper limit for the signal at the Auxiliary Output. A larger value will be clipped.

### /dev..../auxouts/n/max

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the maximum normalized voltage generated on this channel. It can be between -1 and 1. To prevent signal clipping and overvoltage, it is advised to keep it between -0.9 and 0.9.

### /dev..../auxouts/n/min

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the minimum normalized voltage generated on this channel. It can be between -1 and 1. To prevent signal clipping and overvoltage, it is advised to keep it between -0.9 and 0.9.

### /dev..../auxouts/n/offset

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Add the specified offset voltage to the signal after scaling. Auxiliary Output Value = (Signal+Preoffset)*Scale + Offset

### /dev..../auxouts/n/outputchannel

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Selects the channel of the selected signal source.

### /dev..../auxouts/n/outputselect

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Select the signal source to be represented on the Auxiliary Output.

| | |
|---|---|
| -1 | "manual": Select Manual as the output option. |
| 0 | "demod_x": Select Demod X as the output option. |
| 1 | "demod_y": Select Demod Y as the output option. |
| 2 | "demod_r": Select Demod R as the output option. |
| 3 | "demod_theta": Select Demod Theta as the output option. |
| 5 | "pid": Select PID Out as the output option. |
| 7 | "au_amplitude": Select the Arithmetic Unit for amplitude values as the output option. |
| 9 | "pid_shift": Select PID Shift as the output option. |
| 10 | "pid_error": Select PID Error as the output option. |

### /dev..../auxouts/n/preoffset

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | Dependent |

Add a pre-offset to the signal before scaling is applied. Auxiliary Output Value = (Signal+Preoffset)*Scale + Offset

### /dev..../auxouts/n/range

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Sets the output voltage range.

### /dev..../auxouts/n/scale

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | None |

Multiplication factor to scale the signal. Auxiliary Output Value = (Signal+Preoffset)*Scale + Offset

### /dev..../auxouts/n/value

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | V |

Voltage present on the Auxiliary Output. Auxiliary Output Value = (Signal+Preoffset)*Scale + Offset

## 6.2.4. CLOCKBASE

### /dev..../clockbase

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | Hz |

Returns the internal clock frequency of the device.

## 6.2.5. CURRINS

### /dev..../currins/n/autorange

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Automatic adjustment of the Range to approximately two times the maximum current input amplitude.

### /dev..../currins/n/delaycompensation/delay

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | s |

Compensation for signal propagation delays to achieve a flatter phase response. The default value compensates for the propagation delay resulting from the signal output to the current input with a short cable in between. Every additional meter of cable would add ~5 ns propagation delay, which can be compensated using this node.

### /dev..../currins/n/delaycompensation/reset

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Resets the compensation delay to the default value.

### /dev..../currins/n/max

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the maximum normalized current measured on this channel. It can be between -1 and 1. To prevent signal clipping and overcurrent, it is advised to keep it between -0.9 and 0.9.

### /dev..../currins/n/min

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the minimum normalized current measured on this channel. It can be between -1 and 1. To prevent signal clipping and overcurrent, it is advised to keep it between -0.9 and 0.9.

### /dev..../currins/n/on

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enables the current input.

### /dev..../currins/n/range

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | A |

Defines the gain of the current input amplifier. The range should exceed the incoming signal by roughly a factor of two including a potential DC offset. The instrument selects the next higher available range relative to a value inserted by the user. A suitable choice of this setting optimizes the accuracy and signal-to-noise ratio by ensuring that the full dynamic range of the input ADC is used.

## 6.2.6. DEMODS

### /dev..../demods/n/adcselect

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Selects the input signal for the demodulator.

| | |
|---|---|
| 0 | "sigin0", "signal_input0": Sig In 1 |
| 1 | "sigin1", "signal_input1": Sig In 2 |
| 8 | "auxin0", "auxiliary_input0": Aux In 1 |
| 9 | "auxin1", "auxiliary_input1": Aux In 2 |
| 16 | "currin0", "current_input0": Curr In 1 |
| 17 | "currin1", "current_input1": Curr In 2 |
| 177 | "demod_constant_input": Demodulate a constant input. This results in a sine wave of the frequency specified by the demodulator's oscillator with an amplitude of 1 (at lower frequencies; higher frequencies will be attenuated). The maximum possible frequency is limited by the demodulator sampling rate and bandwidth; use demodulator order 1 for the least attenuation in demodulator output. This signal may be used with the auxiliary outputs or PID for advanced measurement and control tasks. When the demodulator output is written to an auxiliary output, the resulting signal can also be used as an additional output channel (for low frequencies). |

### /dev..../demods/n/bypass

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Allows to bypass the demodulator low-pass filter, thus increasing the bandwidth.

| | |
|---|---|
| 0 | "disabled": disabled |
| 1 | "enabled": enabled |

### /dev..../demods/n/freq

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | Hz |

Indicates the frequency used for demodulation.The demodulation frequency is calculated as oscillator frequency times the harmonic factor. When the MOD option is used linear combinations of oscillator frequencies including the harmonic factors define the sideband frequencies.

### /dev..../demods/n/harmonic

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Multiplies the selected oscillator's frequency by an integer. If the demodulator is used as a phase detector in external reference mode (PLL), the effect is that the internal oscillator locks to the external frequency divided by the integer factor.

### /dev..../demods/n/order

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Selects the filter roll off between 6 dB/oct and 24 dB/oct.

| | |
|---|---|
| 1 | 1st order filter 6 dB/oct |
| 2 | 2nd order filter 12 dB/oct |
| 3 | 3rd order filter 18 dB/oct |
| 4 | 4th order filter 24 dB/oct |

### /dev..../demods/n/oscselect

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Assignes an oscillator to the demodulator. Number of available oscillators depends on the installed options.

| | |
|---|---|
| 0 | Oscillator 1 |
| 1 | Oscillator 2 |
| 2 | Oscillator 3 |
| 3 | Oscillator 4 |
| 4 | Oscillator 5 |
| 5 | Oscillator 6 |
| 6 | Oscillator 7 |
| 7 | Oscillator 8 |

### /dev..../demods/n/phaseadjust

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Adjust the demodulator phase automatically in order to read 0 degrees.

### /dev..../demods/n/phaseshift

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | deg |

Applies phase shift to the reference input of the demodulator.

### /dev..../demods/n/rate

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | 1/s |

Defines the demodulator sampling rate, the number of samples that are sent to the host computer per second. A rate of about 7-10 higher as compared to the filter bandwidth usually provides sufficient aliasing suppression. This is also the rate of data received by LabOne Data Server and saved to the computer hard disk. This setting has no impact on the sample rate on the auxiliary outputs connectors. Note: the value inserted by the user may be approximated to the nearest value supported by the instrument.

### /dev..../demods/n/sample

| | |
|---|---|
| **Properties:** | Read, Stream |
| **Type:** | ZIVectorData |
| **Unit:** | Dependent |

Contains streamed demodulator samples with sample interval defined by the demodulator data rate.

### /dev..../demods/n/timeconstant

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | s |

Sets the integration time constant or in other words, the cutoff frequency of the demodulator low pass filter.

### /dev..../demods/n/trigger/source

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Selects the trigger input for the demodulator.

| | |
|---|---|
| 0 | "trigin1": Trigger input 1. |
| 1 | "trigin2": Trigger input 2. |
| 2 | "trigin3": Trigger input 3. |
| 3 | "trigin4": Trigger input 4. |
| 1024 | "swtrig0", "software_trigger0": Software Trigger 1. |

## 6.2.7. EXTREFS

### /dev..../extrefs/n/adcselect

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Indicates the input signal selection for the selected demodulator.

| | |
|---|---|
| 0 | "sigin0", "signal_input0": Signal Input 1 is connected to the corresponding demodulator. |
| 1 | "sigin1", "signal_input1": Signal Input 2 is connected to the corresponding demodulator. |
| 8 | "auxin0", "auxiliary_input0": Auxiliary Input 1 is connected to the corresponding demodulator. |
| 9 | "auxin1", "auxiliary_input1": Auxiliary Input 2 is connected to the corresponding demodulator. |
| 16 | "currin0", "current_input0": Current Input 1 is connected to the corresponding demodulator. |
| 17 | "currin1", "current_input1": Current Input 2 is connected to the corresponding demodulator. |

### /dev..../extrefs/n/automode

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

This defines the type of automatic adaptation of parameters in the PID used for Ext Ref.

| | |
|---|---|
| 2 | "low_bandwidth", "pid_coeffs_filter_low_bw": The PID coefficients, the filter bandwidth and the output limits are automatically set using a low bandwidth. |
| 3 | "high_bandwidth", "pid_coeffs_filter_high_bw": The PID coefficients, the filter bandwidth and the output limits are automatically set using a high bandwidth. |
| 4 | "all", "pid_coeffs_filter_auto_bw": The PID coefficient, the filter bandwidth and the output limits are dynamically adapted. |

### /dev..../extrefs/n/demodselect

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Indicates the demodulator connected to the extref channel.

### /dev..../extrefs/n/enable

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enables the external reference.

### /dev..../extrefs/n/locked

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Indicates whether the external reference is locked.

### /dev..../extrefs/n/oscselect

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Indicates which oscillator is being locked to the external reference.

## 6.2.8. FEATURES

### /dev..../features/code

| | |
|---|---|
| **Properties:** | Write |
| **Type:** | String |
| **Unit:** | None |

Node providing a mechanism to enter feature codes into the instrument.

### /dev..../features/devtype

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Returns the device type.

### /dev..../features/options

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Returns enabled options.

### /dev..../features/serial

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Device serial number.

## 6.2.9. GENERATORS

### /dev..../generators/n/freq

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | Hz |

Indicates the frequency used for output signal generation. The frequency is calculated as oscillator frequency times the harmonic factor. When the MOD option is used linear combinations of oscillator frequencies including the harmonic factors define the sideband frequencies.

### /dev..../generators/n/harmonic

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Multiplies the selected oscillator's frequency by an integer.

### /dev..../generators/n/oscselect

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Assignes an oscillator to the generator. Number of available oscillators depends on the installed options.

### /dev..../generators/n/phaseshift

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | deg |

Applies phase shift to the reference input of the generator.

## 6.2.10. MODS

### /dev..../mods/n/carrier/amplitude

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Set the carrier amplitude

### /dev..../mods/n/carrier/enable

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enable the modulation

### /dev..../mods/n/carrier/harmonic

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Set the harmonic of the carrier frequency. 1 = Fundamental

### /dev..../mods/n/carrier/inputselect

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Select Signal Input for the carrier demodulation

| | |
|---|---|
| 0 | "sigin0", "signal_input0": Sig In 1 |

### /dev..../mods/n/carrier/order

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Selects the filter roll off between 6 dB/oct and 48 dB/oct for carrier demodulation

| | |
|---|---|
| 1 | 1st order filter 6 dB/oct |
| 2 | 2nd order filter 12 dB/oct |
| 3 | 3rd order filter 18 dB/oct |
| 4 | 4th order filter 24 dB/oct |

### /dev..../mods/n/carrier/oscselect

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Select the oscillator for the carrier signal.

| | |
|---|---|
| 0 | Oscillator 1 |
| 1 | Oscillator 2 |
| 2 | Oscillator 3 |
| 3 | Oscillator 4 |

### /dev..../mods/n/carrier/phaseshift

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | degree |

Phase shift applied to the reference input of the carrier demodulator and also to the carrier signal on the Signal Outputs

### /dev..../mods/n/carrier/timeconstant

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | s |

Sets the integration time constant or in other words, the cutoff frequency of the low-pass filter for the carrier demodulation.

### /dev..../mods/n/enable

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enables the modulation.

### /dev..../mods/n/freqdev

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | Hz |

FM mode peak deviation value.

### /dev..../mods/n/freqdevenable

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

In FM mode, choose to work with either modulation index or peak deviation. The modulation index equals peak deviation divided by modulation frequency.

| | |
|---|---|
| 0 | "modulation_index": Use modulation index. |
| 1 | "peak_deviation": Use peak deviation. |

### /dev..../mods/n/index

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | None |

FM modulation index: The modulation index equals peak deviation divided by modulation frequency.

### /dev..../mods/n/mode

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Select the modulation mode.

| | |
|---|---|
| 0 | "am": AM Modulation |
| 1 | "fm": FM Modulation |
| 2 | "manual": Manual |

## /dev..../mods/n/output

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Select Signal Output.

| | |
|---|---|
| 0 | "none": None |
| 1 | "sigout0", "signal_output0": Signal Output 1 |

## /dev..../mods/n/sidebands/n/amplitude

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Set the amplitude of the sideband components.

## /dev..../mods/n/sidebands/n/enable

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enable the signal generation for the respective sideband

## /dev..../mods/n/sidebands/n/harmonic

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Set harmonic of the sideband frequencies. 1 = fundamental

## /dev..../mods/n/sidebands/n/inputselect

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Select Signal Input for the sideband demodulation

| | |
|---|---|
| 0 | "sigin0", "signal_input0": Sig In 1 |

## /dev..../mods/n/sidebands/n/mode

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Enabling of the first sideband and selection of the position of the sideband relative to the carrier frequency for manual mode.

| | |
|---|---|
| 0 | "off": Off: First sideband is disabled. The sideband demodulator behaves like a normal demodulator. |
| 1 | "upper": C + M: First sideband to the right of the carrier |
| 2 | "lower": C - M: First sideband to the left of the carrier |

### /dev..../mods/n/sidebands/n/order

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Selects the filter roll off between 6 dB/oct and 48 dB/oct for sideband demodulation

| | |
|---|---|
| 1 | 1st order filter 6 dB/oct |
| 2 | 2nd order filter 12 dB/oct |
| 3 | 3rd order filter 18 dB/oct |
| 4 | 4th order filter 24 dB/oct |

### /dev..../mods/n/sidebands/n/oscselect

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Select the oscillator for the second sideband.

| | |
|---|---|
| 0 | Oscillator 1 |
| 1 | Oscillator 2 |
| 2 | Oscillator 3 |
| 3 | Oscillator 4 |

### /dev..../mods/n/sidebands/n/phaseshift

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | degree |

Phase shift applied to the reference input of the sideband demodulator and also to the sideband signal on the Signal Outputs

### /dev..../mods/n/sidebands/n/timeconstant

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | s |

Sets the integration time constant or in other words, the cutoff frequency of the low-pass filter for the sideband demodulation.

## 6.2.11. OSCS

### /dev..../oscs/n/freq

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | Hz |

Frequency control for each oscillator.

## 6.2.12. PIDS

### /dev..../pids/n/center

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | Dependent |

Sets the center value for the PID output. After adding the Center value, the signal is clamped to Center + Lower Limit and Center + Upper Limit.

### /dev..../pids/n/d

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | Dependent |

PID derivative gain.

### /dev..../pids/n/demod/adcselect

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Indicates the signal source which is connected to the chosen input demodulator channel.

| | |
|---|---|
| 0 | "sigin0", "signal_input0": Signal Input 1 is connected to the corresponding demodulator. |
| 1 | "sigin1", "signal_input1": Signal Input 2 is connected to the corresponding demodulator. |
| 8 | "auxin0", "auxiliary_input0": Auxiliary Input 1 is connected to the corresponding demodulator. |
| 9 | "auxin1", "auxiliary_input1": Auxiliary Input 2 is connected to the corresponding demodulator. |
| 16 | "currin0", "current_input0": Current Input 1 is connected to the corresponding demodulator. |
| 17 | "currin1", "current_input1": Current Input 2 is connected to the corresponding demodulator. |

### /dev..../pids/n/demod/harmonic

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Multiplier of the for the reference frequency of the current demodulator.

### /dev..../pids/n/demod/order

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Selects the filter roll off between 6 dB/oct and 48 dB/oct of the current demodulator.

| | |
|---|---|
| 1 | 1st order filter 6 dB/oct |
| 2 | 2nd order filter 12 dB/oct |
| 3 | 3rd order filter 18 dB/oct |
| 4 | 4th order filter 24 dB/oct |
| 5 | 5th order filter 30 dB/oct |
| 6 | 6th order filter 36 dB/oct |
| 7 | 7th order filter 42 dB/oct |
| 8 | 8th order filter 48 dB/oct |

### /dev..../pids/n/demod/timeconstant

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Double |
| Unit: | s |

Defines the characteristic time constant (cut off) of the demodulator filter used as an input.

### /dev..../pids/n/dlimittimeconstant

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Double |
| Unit: | s |

The cutoff of the low-pass filter for the D limitation given as time constant. When set to 0, the low-pass filter is disabled.

### /dev..../pids/n/enable

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Integer (64 bit) |
| Unit: | None |

Enable the PID controller.

### /dev..../pids/n/error

| | |
|---|---|
| Properties: | Read |
| Type: | Double |
| Unit: | Dependent |

Error = Set point - PID Input.

### /dev..../pids/n/i

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Double |
| Unit: | Dependent |

PID integral gain I.

### /dev..../pids/n/input

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Integer (enumerated) |
| Unit: | None |

Select input source of PID controller.

| | |
|---|---|
| 0 | "demod_x": Demodulator X |
| 1 | "demod_y": Demodulator Y |
| 2 | "demod_r": Demodulator R |
| 3 | "demod_theta": Demodulator Theta |
| 7 | "au_amplitude": Arithmetic Unit for amplitude values |

### /dev..../pids/n/inputchannel

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Integer (64 bit) |
| Unit: | None |

Select input channel of PID controller.

### /dev..../pids/n/keepint

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | Dependent |

If enabled, the accumulated integral error is maintained upon restart of the PID. If is disabled, the integral error is set to zero when the PID is disabled.

### /dev..../pids/n/limitlower

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | Dependent |

Sets the lower limit for the PID output. After adding the Center value, the signal is clamped to Center + Lower Limit and Center + Upper Limit.

### /dev..../pids/n/limitupper

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | Dependent |

Sets the upper limit for the PID output. After adding the Center value, the signal is clamped to Center + Lower Limit and Center + Upper Limit.

### /dev..../pids/n/mode

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Sets the operation mode of the PID module.

| | |
|---|---|
| 0 | "pid": PID |
| 1 | "pll": PLL (phase locked loop) |
| 2 | "extref": ExtRef (external reference) |

### /dev..../pids/n/output

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Select output of the PID controller.

| | |
|---|---|
| 0 | "sigout0_amp", "signal_output0_amplitude": Driving Signal Output 1 amplitudes |
| 1 | "sigout1_amp", "signal_output1_amplitude": Driving Signal Output 2 amplitudes |
| 2 | "oscillator_frequency": Controlling any of the internal oscillator frequencies |

### /dev..../pids/n/outputchannel

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Select the output channel of the driven output of PID controller.

### /dev..../pids/n/p

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | Dependent |

PID Proportional gain P.

### /dev..../pids/n/phaseunwrap

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enables the phase unwrapping to track phase errors past the +/-180 degree boundary and increase PLL bandwidth.

### /dev..../pids/n/pll/automode

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

This defines the type of automatic adaptation of parameters in the PID.

| | |
|---|---|
| 0 | "no_adaption": No automatic adaption. |
| 1 | "pid_coeffs": The PID coefficients are automatically set based on the filter parameters. |
| 2 | "pid_coeffs_filter_low_bw": The PID coefficients, the filter bandwidth and the output limits are automatically set using a low bandwidth. |
| 3 | "pid_coeffs_filter_high_bw": The PID coefficients, the filter bandwidth and the output limits are automatically set using a high bandwidth. |

### /dev..../pids/n/pll/locked

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Indicates when the PID, configured as PLL, is locked.

### /dev..../pids/n/rate

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | 1/s |

PID sampling rate and update rate of PID outputs. Needs to be set substantially higher than the targeted loop filter bandwidth.

### /dev..../pids/n/setpoint

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | Dependent |

PID controller setpoint

### /dev..../pids/n/shift

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | Dependent |

Difference between the current output value Out and the Center. Shift = PError + IInt(Error, dt) + D*dError/dt

### /dev..../pids/n/stream/burstlen

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Defines how many (complex) samples should be acquired with each trigger.

### /dev..../pids/n/stream/droppedvectors

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Indicates the number of times pid streaming data (vectors) where dropped leading to gaps in the acquired signals.

### /dev..../pids/n/stream/effectiverate

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | 1/s |

Current rate of the PID stream data sent to PC. Defined based on Max Rate.

### /dev..../pids/n/stream/enable

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Enables the data acquisition for the corresponding PID stream.

| | |
|---|---|
| 0 | "off": OFF: PID stream inactive |
| 1 | "on": ON: PID stream active |

### /dev..../pids/n/stream/missedtrigbusy

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Indicates the number of times the acquisition unit was busy (still recording a previous burst) and a trigger was omitted.

### /dev..../pids/n/stream/missedtrigfull

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Indicates the number of times the memory was full and a trigger was omitted.

### /dev..../pids/n/stream/rate

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | 1/s |

Target Rate for PID output data sent to PC. This value defines the applied decimation for sending data to the PC. It does not affect any other place where PID data are used.

### /dev..../pids/n/stream/sample

| | |
|---|---|
| **Properties:** | Read, Stream |
| **Type:** | ZIVectorData |
| **Unit:** | Dependent |

Contains streamed pid samples.

### /dev..../pids/n/stream/trigger/delay

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | None |

[empty]

### /dev..../pids/n/stream/trigger/mode

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Selects the trigger mode.

| | |
|---|---|
| 1 | "rising_edge": PID data is streamed to the host computer on the trigger's rising edge. |
| 2 | "falling_edge": PID data is streamed to the host computer on the trigger's falling edge. |
| 3 | "both_edge": PID data is streamed to the host computer on both trigger's edges. |

### /dev..../pids/n/stream/trigger/source

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Selects the trigger input for the PID.

| | |
|---|---|
| 0 | "trigin1": Trigger input 1. |
| 1 | "trigin2": Trigger input 2. |
| 2 | "trigin3": Trigger input 3. |
| 3 | "trigin4": Trigger input 4. |
| 1024 | "swtrig0", "software_trigger0": Software Trigger 1. |

### /dev..../pids/n/stream/trigger/triggeracq

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Enables the triggered acquisition.

| | |
|---|---|
| 0 | "continuous": Continuous PID data acquisition (triggering is disabled). |
| 1 | "triggered": Triggered PID data acquisition. |

### /dev..../pids/n/value

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | Dependent |

Gives the current PID output value.

## 6.2.13. REFTRIGGENS

### /dev..../reftriggens/n/dutycycle

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | None |

Configures the duty cycle of the reference trigger signal as a rational number between 0 and 1.

### /dev..../reftriggens/n/freq

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | Hz |

Indicates the frequency used for the reference trigger generation. The frequency is calculated as oscillator frequency times the harmonic factor.

### /dev..../reftriggens/n/harmonic

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Multiplies the frequency of the assigned oscillator to the reference trigger generator by an integer factor.

### /dev..../reftriggens/n/oscselect

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Assignes an oscillator to the selected reference trigger generator. Number of available oscillators depends on the installed options.

### /dev..../reftriggens/n/phaseshift

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | deg |

Applies phase shift to the reference input of the reference trigger generator.

## 6.2.14. SCOPES

### /dev..../scopes/n/averaging/count

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Configures the number of Scope measurements to average.

### /dev..../scopes/n/averaging/enable

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Integer (64 bit) |
| Unit: | None |

Enables averaging of Scope measurements.

### /dev..../scopes/n/channels/n/enable

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Integer (64 bit) |
| Unit: | Dependent |

Enables recording for this Scope channel.

### /dev..../scopes/n/channels/n/inputselect

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Integer (enumerated) |
| Unit: | None |

Selects the scope input signal.

| | |
|---|---|
| 0 | "sigin0", "signal_input0": Signal Input Channel 1 |
| 1 | "sigin1", "signal_input1": Signal Input Channel 2 |
| 8 | "auxin0", "auxiliary_input0": Aux Input Channel 1 |
| 9 | "auxin1", "auxiliary_input1": Aux Input Channel 2 |
| 16 | "currin0", "current_input0": Current Input Channel 1 |
| 17 | "currin1", "current_input1": Current Input Channel 2 |

### /dev..../scopes/n/channels/n/wave

| | |
|---|---|
| Properties: | Read |
| Type: | ZIVectorData |
| Unit: | Dependent |

Contains the acquired Scope measurement data.

### /dev..../scopes/n/enable

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Integer (64 bit) |
| Unit: | None |

Enables the acquisition of scope shots.

### /dev..../scopes/n/length

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Integer (64 bit) |
| Unit: | None |

Defines the length of the recorded Scope shot in number of samples.

### /dev..../scopes/n/rate

| | |
|---|---|
| Properties: | Read, Write, Setting |
| Type: | Double |
| Unit: | 1/s |

Defines the scope's sampling rate. Note: the value inserted by the user may be approximated to the nearest value supported by the instrument.

## /dev..../scopes/n/segments/count

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Specifies the number of segments to be recorded in device memory. The maximum scope shot size is given by the available memory divided by the number of segments. This functionality requires the DIG option.

## /dev..../scopes/n/segments/enable

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enable segmented scope recording. This allows for full bandwidth recording of scope shots with a minimum dead time between individual shots. This functionality requires the DIG option.

## /dev..../scopes/n/single

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Puts the Scope into single shot mode.

## /dev..../scopes/n/trigger/channel

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Selects the trigger source signal.

| | |
|---|---|
| 0 | "trigin1": Trigger Input 1. |
| 1 | "trigin2": Trigger Input 2. |
| 32 | "sigin0", "signal_input0": Signal Input 1. |
| 33 | "sigin1", "signal_input1": Signal Input 2. |
| 40 | "auxin0", "auxiliary_input0": Auxiliary Input 1. |
| 41 | "auxin1", "auxiliary_input1": Auxiliary Input 2. |
| 48 | "currin0", "current_input0": Current Input 1. |
| 49 | "currin1", "current_input1": Current Input 2. |
| 1024 | "swtrig0", "software_trigger0": Software Trigger 1. |

## /dev..../scopes/n/trigger/delay

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | s |

The delay of a Scope measurement. A negative delay results in data being acquired before the trigger point. The resolution is 2 ns.

## /dev..../scopes/n/trigger/edge

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Sets on which slope of the trigger signal the scope should trigger.

| | |
|---|---|
| 1 | "rising": Rising edge triggered. |
| 2 | "falling": Falling edge triggered. |
| 3 | "both": Triggers on both the rising and falling edge. |

### /dev..../scopes/n/trigger/enable

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

When triggering is enabled scope data are acquired every time the defined trigger condition is met.

| | |
|---|---|
| 0 | "off": OFF: Continuous scope shot acquisition |
| 1 | "on": ON: Trigger based scope shot acquisition |

### /dev..../scopes/n/trigger/hysteresis

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Defines the voltage the source signal must deviate from the trigger level before the trigger is rearmed again. Set to 0 to turn it off. The sign is defined by the Edge setting.

### /dev..../scopes/n/trigger/level

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Defines the trigger level.

## 6.2.15. SIGINS

### /dev..../sigins/n/ac

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Defines the input coupling for the Signal Inputs. AC coupling inserts a high-pass filter.

| | |
|---|---|
| 0 | "dc": OFF: DC coupling |
| 1 | "ac": ON: AC coupling |

### /dev..../sigins/n/autorange

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Automatic adjustment of the Range to approximately two times the maximum signal input amplitude.

### /dev..../sigins/n/delaycompensation/delay

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | s |

Compensation for signal propagation delays to achieve a flatter phase response. The default value compensates for the propagation delay resulting from the signal input and output and a short cable in between. Every additional meter of cable would add ~5 ns propagation delay, which can be compensated using this node.

## /dev..../sigins/n/delaycompensation/reset

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Resets the compensation delay to the default value.

## /dev..../sigins/n/diff

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Switches between single ended (OFF) and differential (ON) measurements.

| | |
|---|---|
| 0 | "off": OFF: Single ended voltage input |
| 1 | "on": ON: Differential voltage input |

## /dev..../sigins/n/imp50

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Switches between 50 Ohm (ON) and 10 M Ohm (OFF).

| | |
|---|---|
| 0 | "10_MOhm": OFF: 10 M Ohm |
| 1 | "50_Ohm": ON: 50 Ohm |

## /dev..../sigins/n/max

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the maximum normalized voltage measured on this channel. It can be between -1 and 1. To prevent signal clipping and overvoltage, it is advised to keep it between -0.9 and 0.9.

## /dev..../sigins/n/min

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the minimum normalized voltage measured on this channel. It can be between -1 and 1. To prevent signal clipping and overvoltage, it is advised to keep it between -0.9 and 0.9.

## /dev..../sigins/n/on

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enables the signal input.

### /dev..../sigins/n/range

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Defines the gain of the analog input amplifier. The range should exceed the incoming signal by roughly a factor two including a potential DC offset. The instrument selects the next higher available range relative to a value inserted by the user. A suitable choice of this setting optimizes the accuracy and signal-to-noise ratio by ensuring that the full dynamic range of the input ADC is used.

## 6.2.16. SIGOUTS

### /dev..../sigouts/n/diff

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Switch between single-ended output (OFF) and differential output (ON). In differential mode the signal swing is defined between Signal Output +V / -V.

### /dev..../sigouts/n/generators/n/amplitude

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Sets the amplitude of the generator output.

### /dev..../sigouts/n/generators/n/enable

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enables (1) or disables (0) the generator output.

### /dev..../sigouts/n/imp50

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Select the load impedance between 50 Ohm and HiZ. The impedance of the output is always 50 Ohm. For a load impedance of 50 Ohm the displayed voltage is half the output voltage to reflect the voltage seen at the load.

| | |
|---|---|
| 0 | "high_impedance": HiZ |
| 1 | "50_Ohm": 50 Ohm |

### /dev..../sigouts/n/max

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the maximum normalized voltage generated on this channel. It can be between -1 and 1. To prevent signal clipping and overvoltage, it is advised to keep it between -0.9 and 0.9.

### /dev..../sigouts/n/min

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

Indicates the minimum normalized voltage generated on this channel. It can be between -1 and 1. To prevent signal clipping and overvoltage, it is advised to keep it between -0.9 and 0.9.

### /dev..../sigouts/n/offset

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Defines the DC voltage that is added to the dynamic part of the output signal.

### /dev..../sigouts/n/on

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enabling/Disabling the Signal Output. Corresponds to the blue LED indicator on the instrument front panel.

### /dev..../sigouts/n/range

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Sets the output voltage range.

## 6.2.17. STATS

### /dev..../stats/physical/channels/n/temperatures/n

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | °C |

Temperature readings of the of the analogue front-end channels.

### /dev..../stats/physical/channels/n/voltages/n

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | V |

Voltage readings of the of the analogue front-end channels.

### /dev..../stats/physical/currents/n

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | mA |

Internal current measurements.

### /dev..../stats/physical/fanspeeds/n

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | RPM |

Speed of the internal cooling fans for monitoring.

### /dev..../stats/physical/fpga/aux

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | V |

Supply voltage of the FPGA.

### /dev..../stats/physical/fpga/core

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | V |

Core voltage of the FPGA.

### /dev..../stats/physical/fpga/pstemp

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | °C |

Internal temperature of the FPGA's processor system.

### /dev..../stats/physical/fpga/temp

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | °C |

Internal temperature of the FPGA.

### /dev..../stats/physical/overtemperature

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

This flag is set to 1 if the temperature of the FPGA exceeds 85°C. It will be reset to 0 after a restart of the device.

### /dev..../stats/physical/powermonitors/n/current

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | A |

Current of the corresponding rail monitored on the motherboard.

### /dev..../stats/physical/powermonitors/n/voltage

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | V |

Voltage of the corresponding rail monitored on the motherboard.

### /dev..../stats/physical/powersupplies/n/current

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | A |

Current of the corresponding main power supply rail. May read 0 in case the particular rail does not support current measurement.

### /dev..../stats/physical/powersupplies/n/temperature

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | °C |

Temperature of the corresponding main power supply rail. May read 0 in case the particular rail does not support temperature measurement.

### /dev..../stats/physical/powersupplies/n/voltage

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | V |

Voltage of the corresponding main power supply rail. May report the nominal value in case the particular rail does not support voltage measurement.

### /dev..../stats/physical/temperatures/n

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | °C |

Internal temperature measurements.

## 6.2.18. STATUS

### /dev..../status/flags/binary

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

A set of binary flags giving an indication of the state of various parts of the device. Reserved for future use.

### /dev..../status/time

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

The current timestamp.

## 6.2.19. SYSTEM

### /dev..../system/activeinterface

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Currently active interface of the device.

### /dev..../system/boardrevisions/n

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Hardware revision of the FPGA base board

### /dev..../system/clocks/referenceclock/in/freq

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | Hz |

Indicates the frequency of the reference clock.

### /dev..../system/clocks/referenceclock/in/source

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

The intended reference clock source. When the source is changed, all the instruments connected with ZSync links will be disconnected. The connection should be re-established manually.

| | |
|---|---|
| 0 | "internal": The internal clock is intended to be used as the frequency and time base reference. |
| 1 | "external": An external clock is intended to be used as the frequency and time base reference. Provide a clean and stable 10 MHz or 100 MHz reference to the appropriate back panel connector. |

### /dev..../system/clocks/referenceclock/in/sourceactual

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

The actual reference clock source.

| | |
|---|---|
| 0 | "internal": The internal clock is used as the frequency and time base reference. |
| 1 | "external": An external clock is used as the frequency and time base reference. |

### /dev..../system/clocks/referenceclock/in/status

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Status of the reference clock.

| | |
|---|---|
| 0 | "locked": Reference clock has been locked on. |
| 1 | "error": There was an error locking onto the reference clock signal. |
| 2 | "busy": The device is busy trying to lock onto the reference clock signal. |

### /dev..../system/clocks/referenceclock/out/enable

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enable clock signal on the reference clock output.

### /dev..../system/clocks/referenceclock/out/freq

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | Hz |

Select the frequency of the output reference clock. Only 10 MHz and 100 MHz are allowed.

### /dev..../system/fpgarevision

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

HDL firmware revision.

### /dev..../system/fwlog

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Returns log output of the firmware.

### /dev..../system/fwlogenable

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enables logging to the fwlog node.

### /dev..../system/fwrevision

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Revision of the device-internal controller software.

### /dev..../system/fx3revision

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

USB firmware revision.

### /dev..../system/identify

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Setting this node to 1 will cause the device to blink the power led for a few seconds.

### /dev..../system/kerneltype

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Returns the type of the data server kernel (mdk or hpk).

### /dev..../system/nics/n/defaultgateway

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | String |
| **Unit:** | None |

Default gateway configuration for the network connection.

### /dev..../system/nics/n/defaultip4

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | String |
| **Unit:** | None |

IPv4 address of the device to use if static IP is enabled.

### /dev..../system/nics/n/defaultmask

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | String |
| **Unit:** | None |

IPv4 mask in case of static IP.

### /dev..../system/nics/n/gateway

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Current network gateway.

### /dev..../system/nics/n/ip4

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Current IPv4 of the device.

### /dev..../system/nics/n/mac

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Current MAC address of the device network interface.

### /dev..../system/nics/n/mask

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | String |
| **Unit:** | None |

Current network mask.

### /dev..../system/nics/n/saveip

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

If written, this action will program the defined static IP address to the device.

### /dev..../system/nics/n/static

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enable this flag if the device is used in a network with fixed IP assignment without a DHCP server.

### /dev..../system/powerconfigdate

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Contains the date of power configuration (format is: (year << 16) | (month << 8) | day)

### /dev..../system/properties/centerfreqstepsize

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | Hz |

Step size for center frequency changes.

### /dev..../system/properties/freqresolution

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

The number of bits used to represent a frequency.

### /dev..../system/properties/freqscaling

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

The scale factor to use to convert a frequency represented as a freqresolution-bit integer to a floating point value.

### /dev..../system/properties/maxdemodrate

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | 1/s |

The maximum demodulator rate that can be set. Only relevant for lock-in amplifiers.

### /dev..../system/properties/maxfreq

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Double |
| **Unit:** | None |

The maximum oscillator frequency that can be set.

### /dev..../system/properties/maxtimeconstant

| | |
|---|---|
| Properties: | Read |
| Type: | Double |
| Unit: | s |

The maximum demodulator time constant that can be set. Only relevant for lock-in amplifiers.

### /dev..../system/properties/minfreq

| | |
|---|---|
| Properties: | Read |
| Type: | Double |
| Unit: | None |

The minimum oscillator frequency that can be set.

### /dev..../system/properties/mintimeconstant

| | |
|---|---|
| Properties: | Read |
| Type: | Double |
| Unit: | s |

The minimum demodulator time constant that can be set. Only relevant for lock-in amplifiers.

### /dev..../system/properties/negativefreq

| | |
|---|---|
| Properties: | Read |
| Type: | Integer (64 bit) |
| Unit: | None |

Indicates whether negative frequencies are supported.

### /dev..../system/properties/timebase

| | |
|---|---|
| Properties: | Read |
| Type: | Double |
| Unit: | s |

Minimal time difference between two timestamps. The value is equal to 1/(maximum sampling rate).

### /dev..../system/shutdown

| | |
|---|---|
| Properties: | Read, Write |
| Type: | Integer (64 bit) |
| Unit: | None |

Sending a '1' to this node initiates a shutdown of the operating system on the device. It is recommended to trigger this shutdown before switching the device off with the hardware switch at the back side of the device.

### /dev..../system/stall

| | |
|---|---|
| Properties: | Read, Write |
| Type: | Integer (64 bit) |
| Unit: | None |

Indicates if the network connection is stalled.

### /dev..../system/swtriggers/n/single

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Issues a single software trigger event.

### /dev..../system/update

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Requests update of the device firmware and bitstream from the dataserver.

## 6.2.20. TRIGINS

### /dev..../trigins/n/autothreshold

| | |
|---|---|
| **Properties:** | Read, Write |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Automatically adjust the trigger threshold. The level is set to fall in the center of the applied transitions.

### /dev..../trigins/n/imp50

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Trigger input impedance: When on, the trigger input impedance is 50 Ohm, when off 1 kOhm.

| | |
|---|---|
| 0 | "1_kOhm": OFF: 1 kOhm |
| 1 | "50_Ohm": ON: 50 Ohm |

### /dev..../trigins/n/level

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Double |
| **Unit:** | V |

Trigger voltage level at which the trigger input toggles between low and high. Use 50% amplitude for digital input and consider the trigger hysteresis.

### /dev..../trigins/n/value

| | |
|---|---|
| **Properties:** | Read |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Shows the value of the digital Trigger Input. The value is integrated over a period of 100 ms. Values are: 1: low; 2: high; 3: was low and high in the period.

## 6.2.21. TRIGOUTS

### /dev..../trigouts/n/on

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (64 bit) |
| **Unit:** | None |

Enables the trigger output.

### /dev..../trigouts/n/source

| | |
|---|---|
| **Properties:** | Read, Write, Setting |
| **Type:** | Integer (enumerated) |
| **Unit:** | None |

Select the signal assigned to the trigger output.

| | |
|---|---|
| 80 | "ref_trig_gen0": Reference Trigger Generator 1. |
| 81 | "ref_trig_gen1": Reference Trigger Generator 2. |