

# SHFPPC User Manual

Parametric Pump Controller

# SHFPPC User Manual

Zurich Instruments AG

Revision 24.07

Copyright © 2008-2024 Zurich Instruments AG

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.

LabVIEW is a registered trademark of National Instruments Inc. MATLAB is a registered trademark of The MathWorks, Inc. All other trademarks are the property of their respective owners.

# Table of Contents

Declaration of Conformity .....	
1. Change Log.....	1
2. Getting Started .....	2
2. 1. Quick Start Guide.....	2
2. 2. Inspect the Package Contents .....	3
2. 3. Handling and Safety Instructions .....	4
2. 4. Software Installation .....	5
2. 5. Connecting to the Instrument.....	12
2. 6. Software Update .....	26
2. 7. Troubleshooting.....	27
3. Functional Overview .....	30
3. 1. Features.....	30
3. 2. Front Panel Tour .....	31
3. 3. Back Panel Tour .....	32
3. 4. Ordering Guide.....	32
4. Specifications.....	33
4. 1. General Specifications .....	33
4. 2. Analog Interface Specifications .....	34
5. Device Node Tree .....	36
5. 1. Introduction .....	36
5. 2. Reference Node Documentation .....	39

# CE Declaration of Conformity



The manufacturer

Zurich Instruments  
Technoparkstrasse 1  
8005 Zurich  
Switzerland

declares that the product

SHFPPC, Super High Frequency Parametric Pump Controller

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

Zurich, October 20<sup>th</sup>, 2022

Flavio Heer, CTO

# UKCA Declaration of Conformity



The manufacturer

Zurich Instruments  
Technoparkstrasse 1  
8005 Zurich Switzerland

declares that the product

SHFPPC, Super High Frequency Parametric Pump Controller

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, EN 55011:2016, EN 55011:2016/A1:2017, EN 55011:2016/A11:2020 (Group 1, Class A equipment)
S.I. 2016/1101 (Electrical Equipment (Safety) Regulations)	EN 61010-1:2010, EN 61010-1:2010/A1:2019, 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 20<sup>th</sup>, 2022

A handwritten signature in black ink that reads 'Flavio Heer'.

Flavio Heer, CTO

# 1. Change Log

## 1.1. Release 24.07

**Release date:** 31-Jul-2024

## 1.2. Release 24.04

**Release date:** 30-Apr-2024

## 1.3. Release 24.01

**Release date:** 31-Jan-2024

## 1.4. Release 23.10

**Release date:** 30-Nov-2023

- Initial release of the SHFPPC user manual.

## 2. Getting Started

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

- A Quick Start Guide for the impatient
- Inspecting the package content and accessories
- List of essential handling and safety instructions
- Connecting to the SHFPPC Instrument
- Handy list of troubleshooting guidelines

This chapter is delivered as a hard copy with the instrument upon delivery. It is also the first chapter of the SHFPPC User Manual.

### 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:

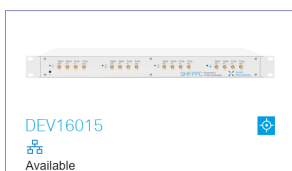
1. Inspect the package content. Besides the Instrument there should be a country-specific power cable, a USB cable, an Ethernet cable and a hard copy of the user manual [Getting Started](#)
2. Check the Handling and Safety Instructions in [Handling and Safety Instructions](#).
3. Download and install the latest LabOne software from the [Zurich Instruments Download Center](#). Choose the download file that fits your computer (e.g. Windows with 64-bit addressing). For more detailed information see [Software Installation](#).
4. Connect the Instrument to the power line. Turn it on and connect it to a switch in the LAN using the Ethernet cable.
5. 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.

Device Connection

 LabOne® User Interface

Version 23.10.49450 [Check For Update](#)

[Basic](#) [Advanced](#)



 Zurich Instruments

6. 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 [Connecting to the Instrument](#).

If any problems occur whilst setting up the instrument and software please see the [Troubleshooting](#) at the end of this chapter.

The functional description of the LabOne User Interface [Functional Overview](#) provides a general introduction to the various tools and settings tabs with tables in each section describing every UI element. 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.

#### Note

The SHFPPC needs to warm up for 30 minutes after power-up.

## 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 SHFPPC Instrument
- You have received 1 power cord with a power plug suited to your country
- You have received 1 USB 2.0 cable and/or 1 LAN cable (category 5/6 required)
- You have received a printed version of the "Getting Started" section
- 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 SHFPPC

	2 or 4-channel SHFPPC
	the power cord (e.g. EU norm)
	the USB 2.0 cable
	the power inlet, with power switch
	the LAN / Ethernet cable (category 5/6 required)
	the "Next Calibration" sticker on the back panel of the instrument
	the MAC address sticker on the back panel of the instrument

The SHFPPC 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 SHFPPC 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.

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.

Depending on the equipment attached to the SHFPPC instrument, levels of electromagnetic emissions exceeding the standards tested against can occur.

This product is designed for use solely in industrial and laboratory environments or, if expressly permitted, also in the field and must not be used in any way that may cause personal injury or property damage.

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.

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.
Electromagnetic environment	This equipment has been certified to conform with industrial electromagnetic environment as defined in EN 61326-1:2020. 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 <a href="#">Specifications</a> 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.
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 <a href="#">Specifications</a> section. Do not block the ventilator opening on the back or the air intake on the chassis side 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	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 accessed 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 sockets on the back panel labeled "ZSync" are not intended for Ethernet LAN connection. Connecting an Ethernet device to these sockets 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 <a href="#">Specifications</a> 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](#).

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 SHFPPC 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 [Software Update](#).

### 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 SHFPPC 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.

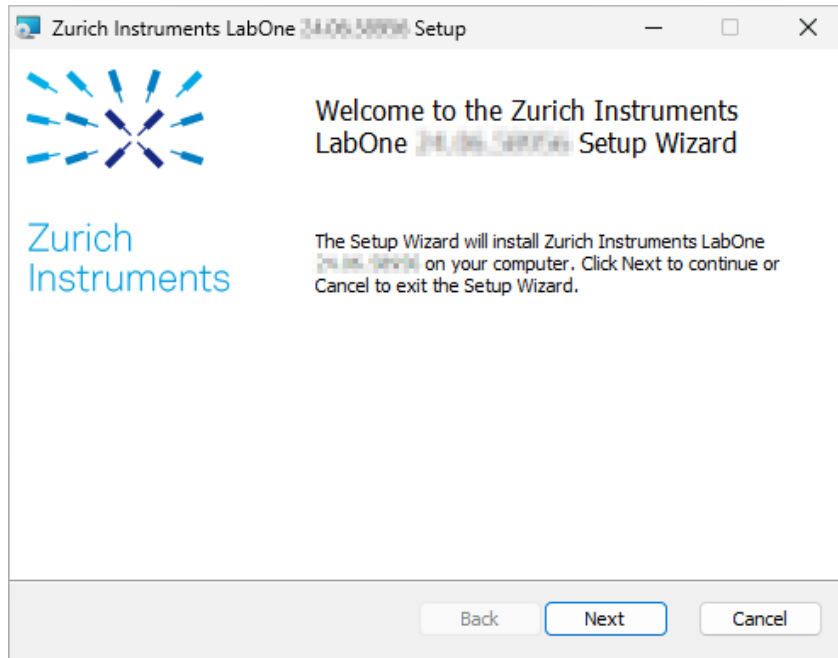


Figure 2.1: Installation welcome screen

- 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 SHFPPC Instrument the "SHFPPC 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.

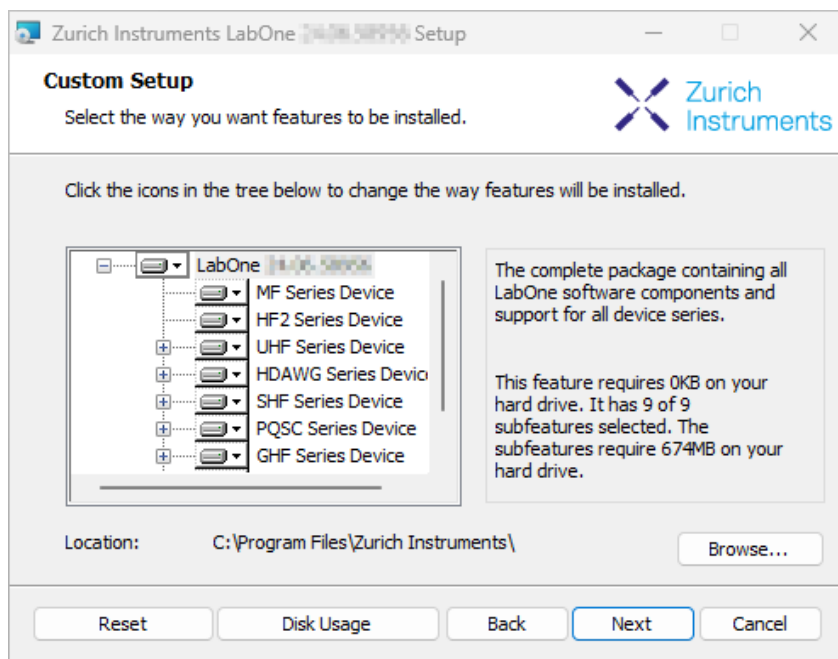


Figure 2.2: Custom setup screen

- 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.

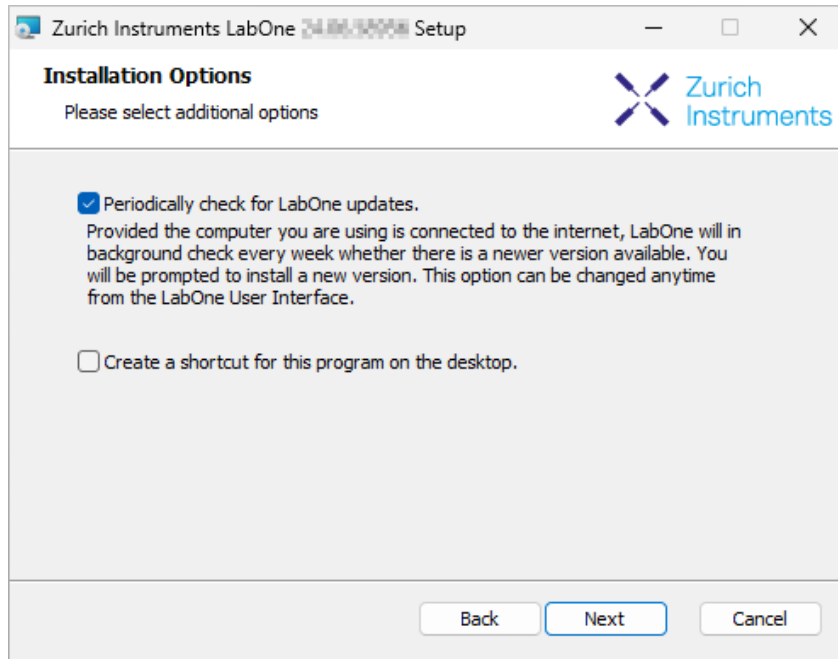


Figure 2.3: Automatic update check

- 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.

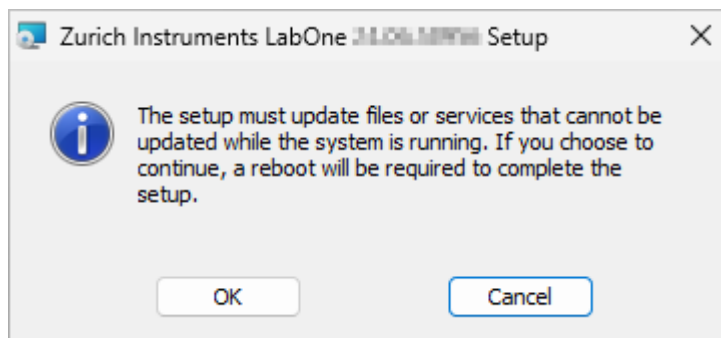


Figure 2.4: Installation reboot request

- 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**.

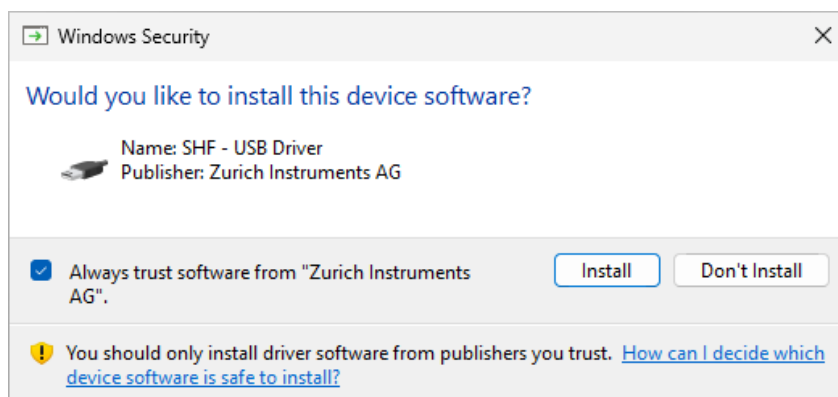


Figure 2.5: Installation driver acceptance

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

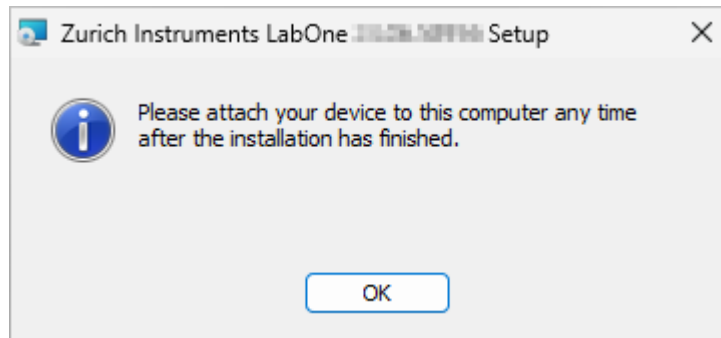


Figure 2.6: Installation completion screen

3. Click **Finish** to close the Zurich Instruments LabOne installer.
4. You can now start the LabOne User Interface as described in [LabOne Software Start-up](#) and choose an instrument to connect to via the Device Connection dialog shown in [Device Connection dialog](#).

## Warning

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

### 2.4.3. Start LabOne Manually on 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 [LabOne Software Start-up](#). The advantage of using the command line is being able to observe and change the behavior of the Web and Data Servers. To start the Servers manually, open a command-line terminal (Command Prompt, PowerShell (Windows) or Bash (Linux)). For Windows, the current working directory needs to be the installation directory of the Web Server and Data Server. They are installed in the Program Files folder (usually: C:\Program Files) under \Zurich Instruments\LabOne in the WebServer and DataServer folders, respectively. The Web Server and Data Server (ziDataServer) are started by running the respective executable in each folder. Please be aware that only one instance of the Web Server can run at a time per computer. The behavior of the Servers can be changed by providing command line arguments. For a detailed list of all arguments see the command line help text:

```
$ ziWebServer --help
```

For the Data Server:

```
$ ziDataServer --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. Before running the Webserver from the terminal, the user needs to ensure there is no other instance of Webserver running in the background. This can be checked using the Tray Icon as shown below.

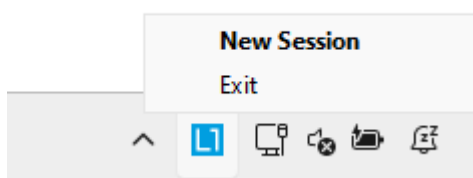


Figure 2.7: LabOne Tray Icon in Windows 11

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"
```

## 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.

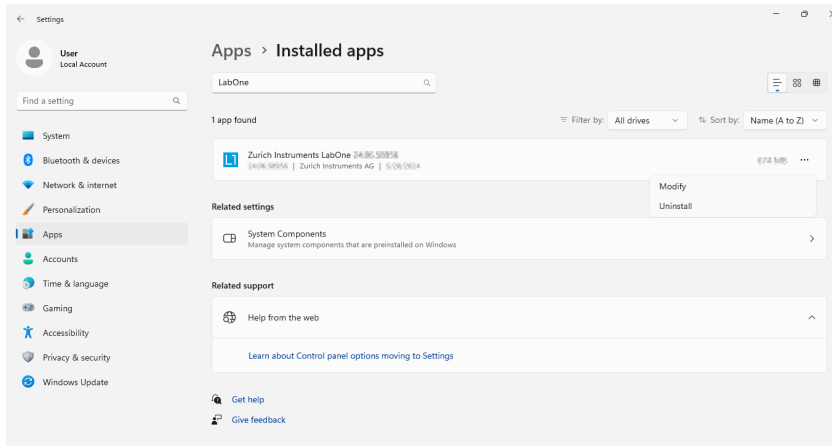


Figure 2.8: Uninstallation of LabOne on Windows computers

### 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. This will open the LabOne installation wizard with the option "Repair" as shown in Figure 2.9.

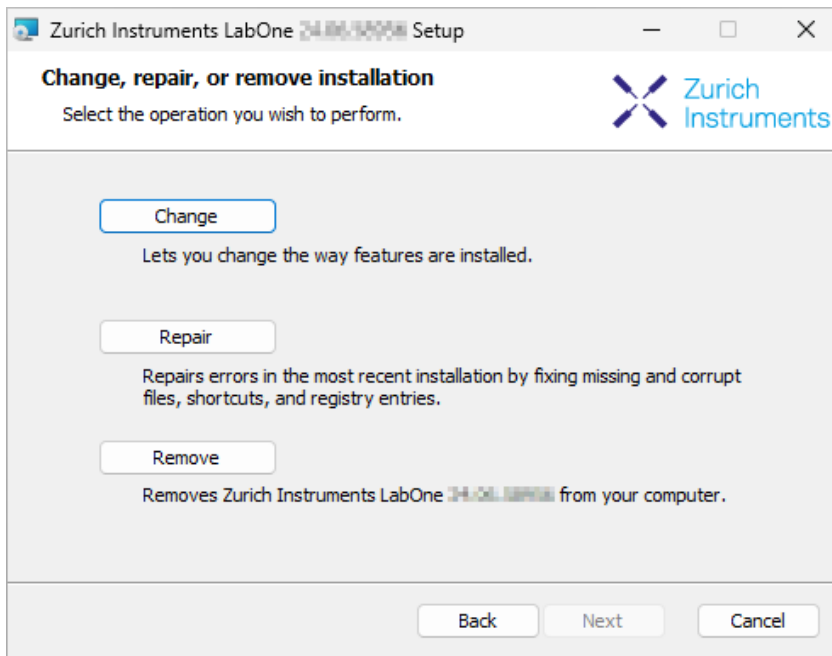


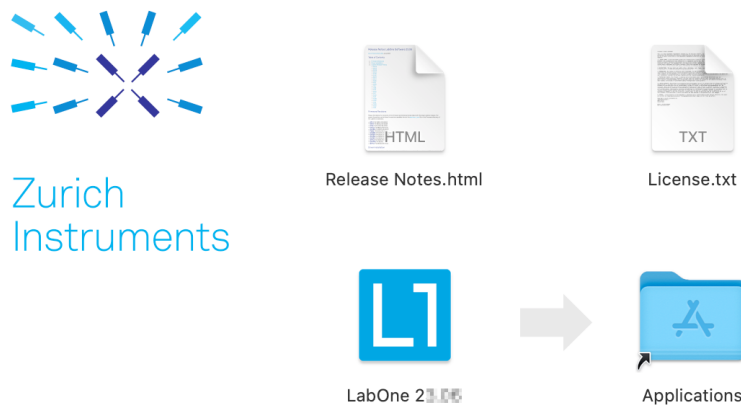
Figure 2.9: Repair of LabOne on Windows computers

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

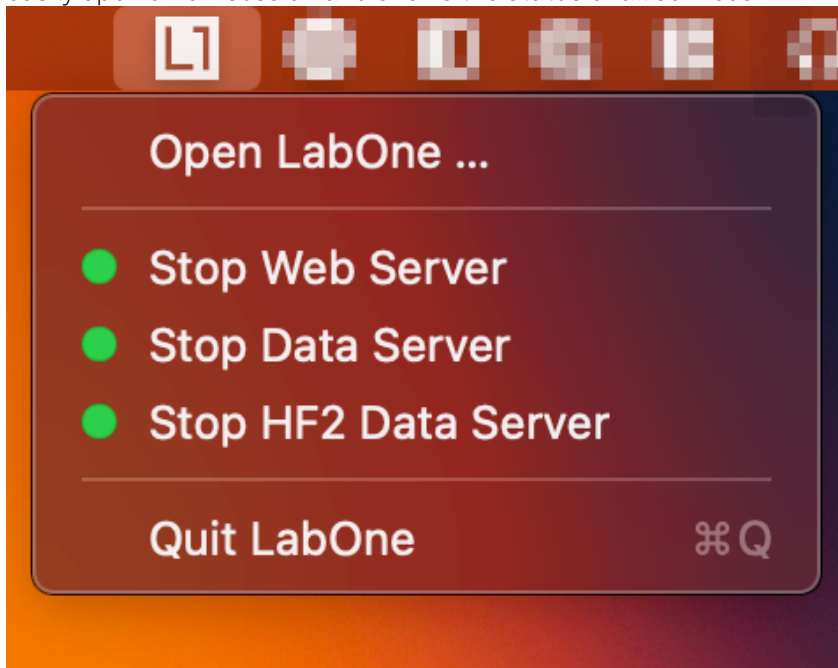
### 2.4.4. Installing LabOne on macOS

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.



### 2.4.5. Uninstalling LabOne on macOS

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

### 2.4.6. 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`

### 2.4.7. 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.

### 2.4.8. Installing LabOne on Linux

### 2.4.9. 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.

### 2.4.10. 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:



## 2.5. Connecting to the Instrument

- 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.

### 2.4.11. 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 [LabOne Software Start-up](#) and choose an instrument to connect to via the Device Connection dialog shown in [Device Connection dialog](#).

### Important

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

### 2.4.12. 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 SHFPPC is operated using the LabOne software. After installation of LabOne, the instrument can be connected to a PC by using either the Universal Serial Bus (USB) cable or the 1 Gbit/s Ethernet (1GbE) LAN cable supplied with the instrument. The LabOne software is controlled via a web browser once suitable physical and logical connections to the instrument have been made.

## Note

The following web browsers are supported (latest versions)



- When using 1GbE, integrate the instrument physically into an existing local area network (LAN) by connecting the instrument to a switch in the LAN using an Ethernet cable. The instrument can then be accessed from a web browser running on any computer in the same LAN with LabOne installed. The Ethernet connection can also be point-to-point. This requires some adjustment of the network card settings of the host computer. Depending on the network configuration and the installed network card, one or the other connection scheme is better suited.
- Using the USB connection to physically connect to the instrument requires the installation of a USB driver on Windows computers. This driver is included in the LabOne software installer and will be installed on the host computer as part of the LabOne installation wizard.

### 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.

- 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.

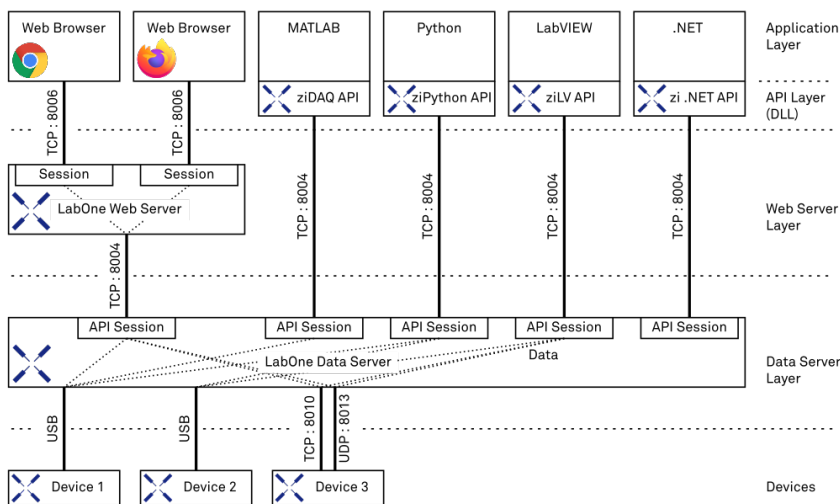


Figure 2.10: LabOne Software architecture

## 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.

## 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.

## 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 DLLs for the following programming environments:

- MATLAB
- Python
- LabVIEW
- .NET
- C

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.

### 2.5.2. LabOne Software Start-up

This section describes the start-up of the LabOne User Interface which is used to control the PQSC Instrument. If the LabOne software is not yet installed on the PC please follow the instructions in [Software Installation](#). If the device is not yet connected please find more information in [Visibility and Connection](#).

The LabOne User Interface start-up link can be found under the Windows 10/11 Start Menu. As shown in [Figure 2.11](#), 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 [LabOne Software Architecture](#).

## 2.5. Connecting to the Instrument

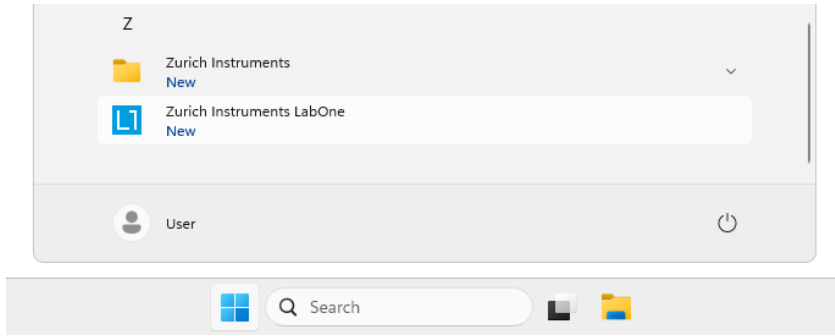


Figure 2.11: Link to the LabOne User Interface in the Windows 11 Start Menu

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](#) 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 [LabOne Software Architecture](#).

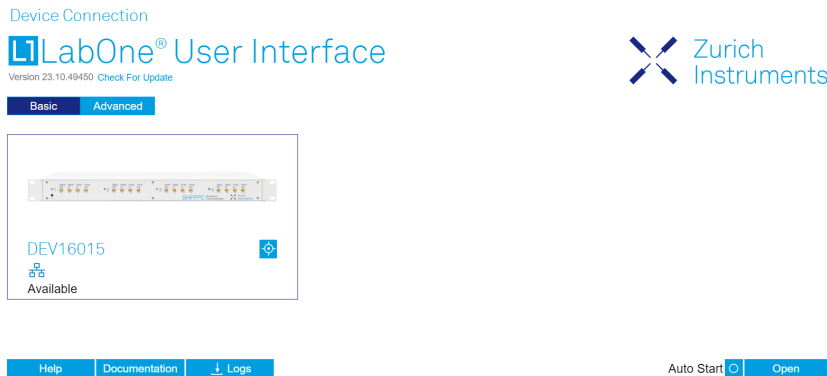


Figure 2.12: Device Connection dialog

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.

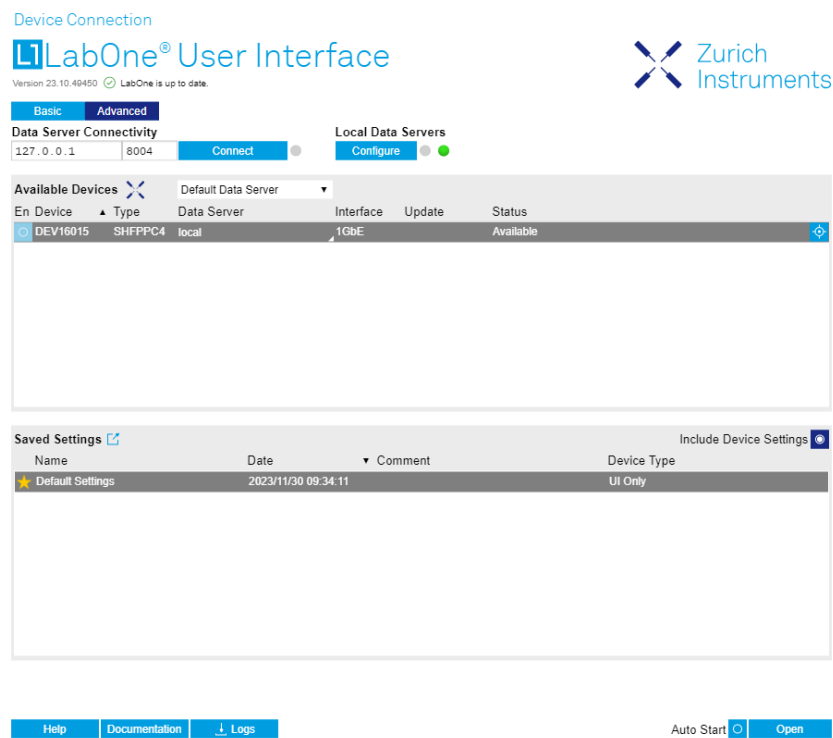


Figure 2.13: Device Connection dialog (Advanced view)

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.

## 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 PQSC 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.

## 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 PQSC 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.

If a firmware update of the instrument is available, the second to last column will show a button, and a simple click on it will update the instrument. The last column indicates the status of the device. explains the meaning of some of the possible device statuses.

Table 2.5: Device Status Information


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.
Free	The device is not in use by any LabOne Data Server and can be connected by clicking the <b>Open</b> button.
In Use	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.
Device FW upgrade required/available	The firmware of the device is out of date. Please first upgrade the firmware as described in <a href="#">Software Update</a> .
Device not yet ready	The device is visible and starting up.

## 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](#). 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](#).

Table 2.7: Button Descriptions

<b>Open</b>	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.
<b>Include Device Settings</b>	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.
<b>Auto Start</b>	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.

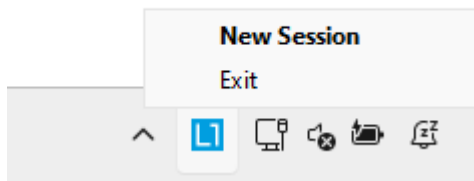
## Note

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.

## 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.

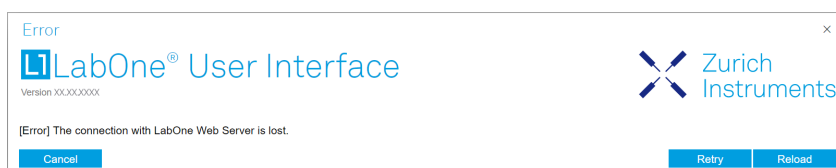


## Messages

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

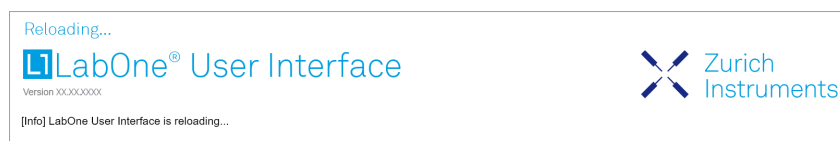
In this case the browser is no longer able to connect to the LabOne Web Server. This can happen if the Web Server and Data Server run on different PCs and a network connection is interrupted. As long as the Web Server is running and the session did not yet time out, it is possible to just attach to the 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](#). The figure below shows an example of the Connection Lost dialog.





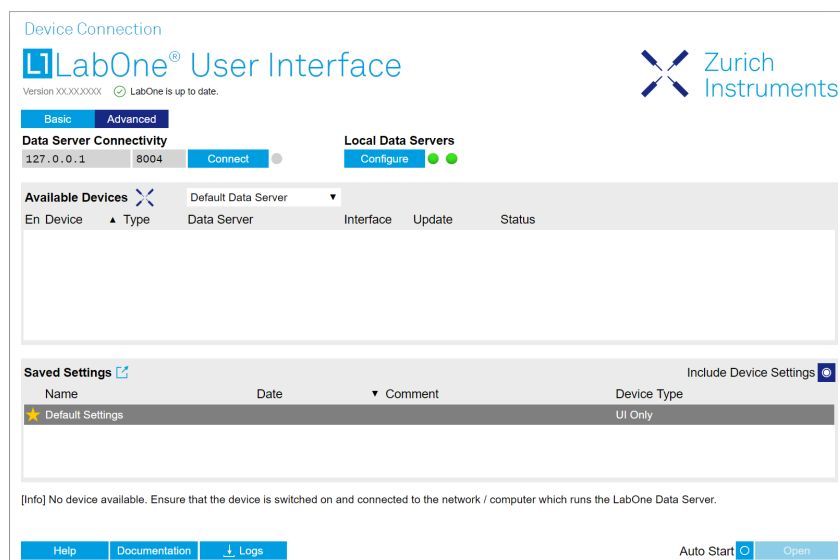
## Reloading...

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](#). 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.



## 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 [Visibility and Connection](#). The figure below shows an example of this dialog.



## 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 [Software Update](#). If all the devices are in use, access is not possible until a connection is relinquished by another Data Server.

### 2.5.3. 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.

## 2.5. Connecting to the Instrument

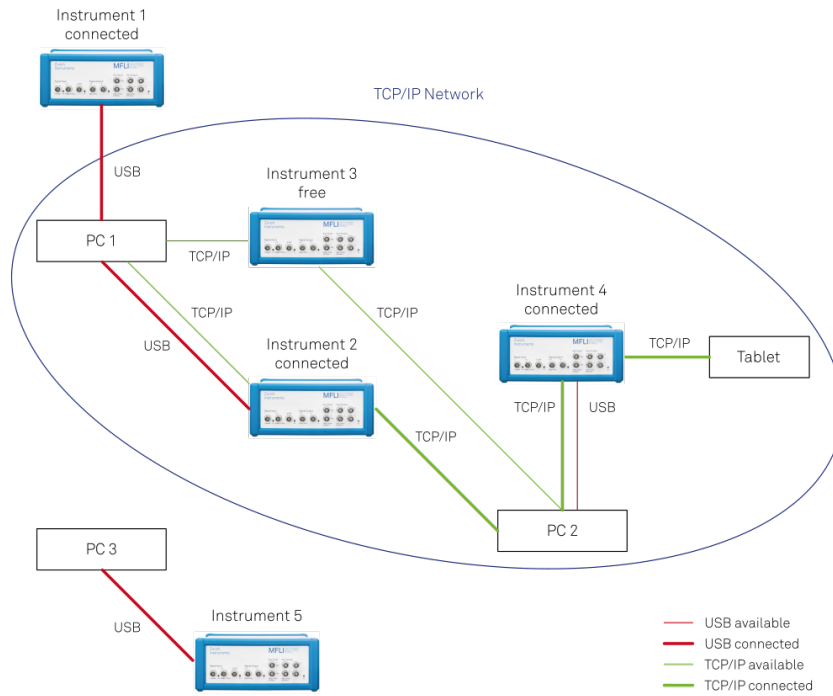


Figure 2.14: Connectivity

Figure 2.14 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.

## 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. Like this, multiple browser and API sessions can access the instrument simultaneously.

### 2.5.4. 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**, see [Static IP configuration for the host computer](#) (go to **Control Panel** → **Internet Options** → **Network and Internet** → **Network and Sharing Center** → **Local Area Connection** → **Properties**).
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 use a static device IP as described in the following section.

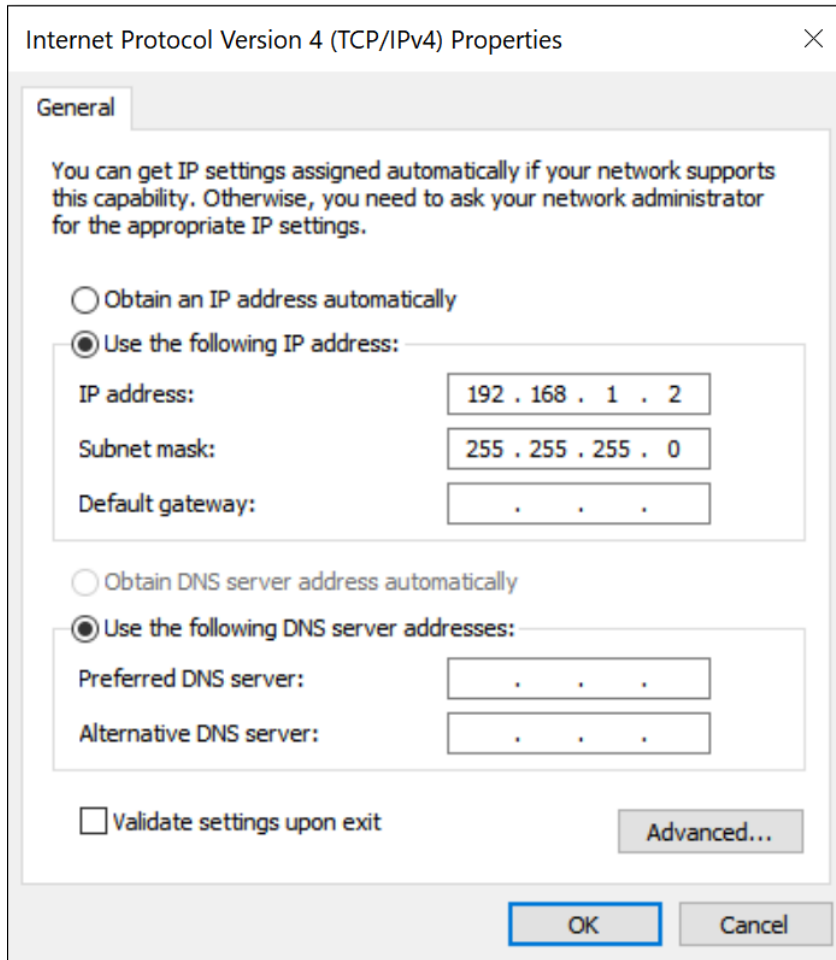


Figure 2.15: Static IP configuration for the host computer

## 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 a 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.

## 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.

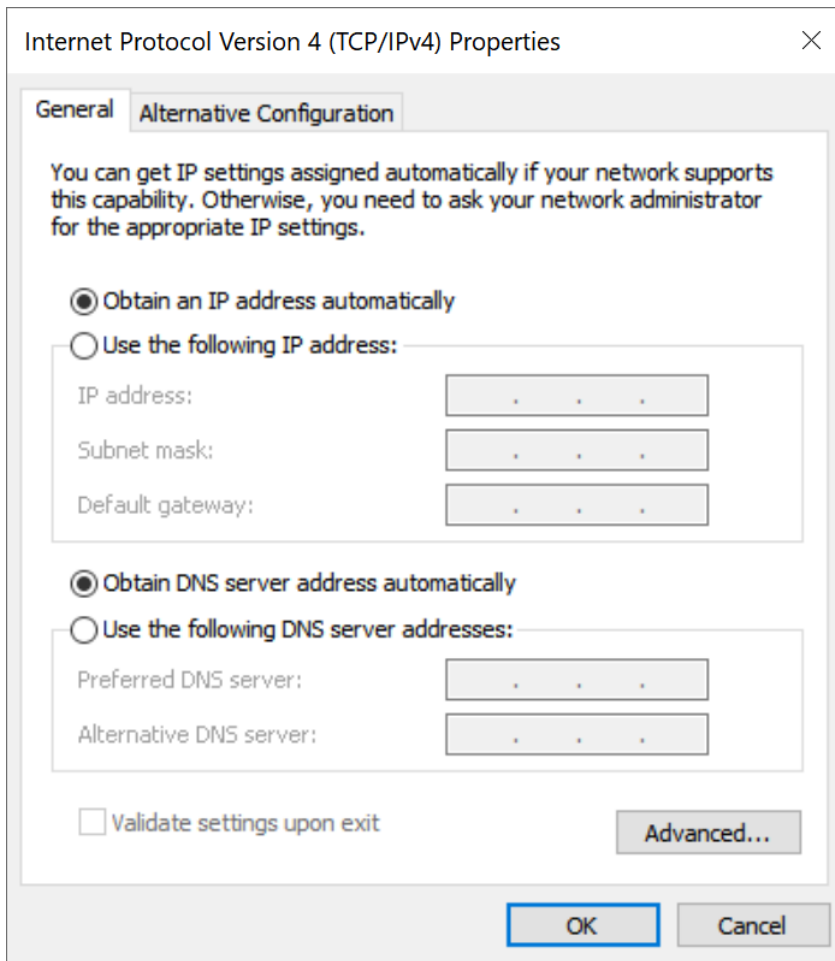


Figure 2.16: Dynamic IP configuration for the host computer

## 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 2.17](#) 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 2.17: 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 [Configuration of static IP in LabOne UI](#).
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.

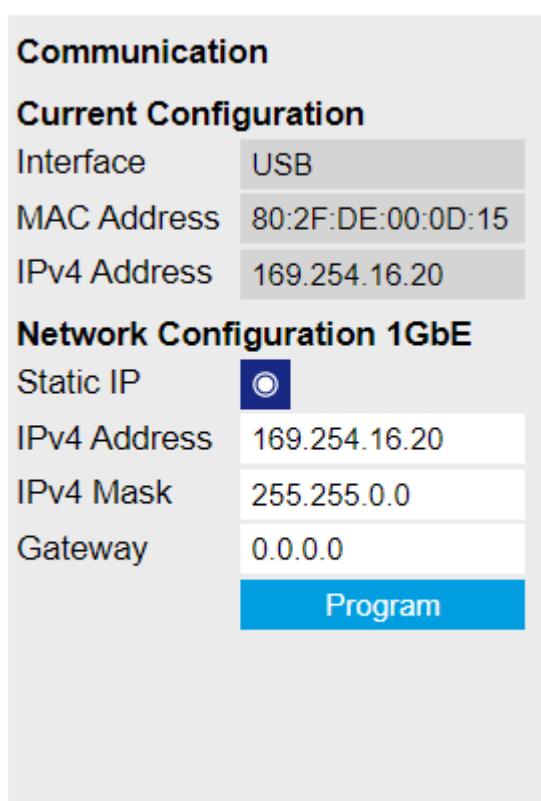


Figure 2.18: Configuration of static IP in LabOne UI

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 2.19](#) 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 2.19: 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.

## Fallback Device IP

When configured to a dynamic address, but no DHCP server is present in the network, e.g., device connected directly to a PC, the instrument falls back on an IP address in the local link IP range that is **169.254.x.x**. If the host computer has also an IP address within the same range, the instrument becomes visible to the LabOne data server running on the host computer. This way, there is no need to go through the process described above to assign a static IP to the instrument.

## 2.6. Software Update

### 2.6.1. Overview

It is recommended to regularly update the LabOne software on the SHFPPC 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 [Updating LabOne using Automatic Update Check](#). 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 [Software Installation](#). Second, the instrument firmware needs to be updated from the Device Connection dialog after starting up LabOne. This is shown in [Updating the Instrument Firmware](#). 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 2.20](#) will start a download the latest LabOne installer for Windows or Linux, see [Figure 2.21](#). After download, proceed as explained in [Software Installation](#) to update LabOne.



Figure 2.20: Device Connection dialog: LabOne update available

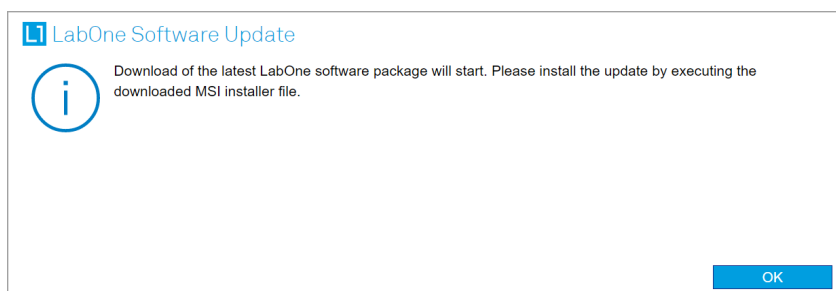


Figure 2.21: Download LabOne MSI using Automatic Update Check feature

### 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 2.22. 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 2.23. The firmware upgrade takes approximately 2 minutes.

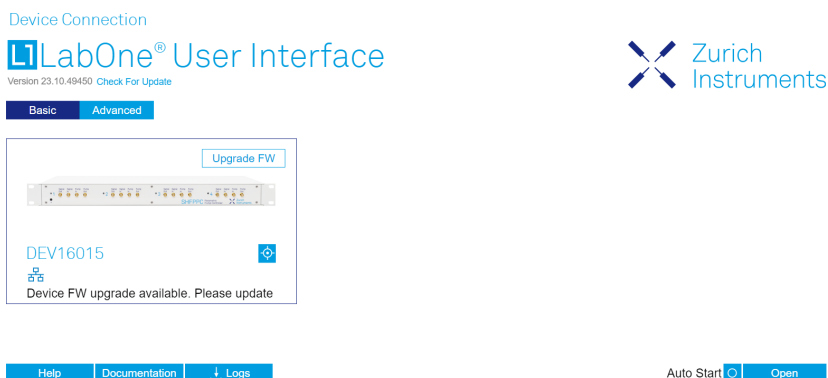


Figure 2.22: Device Connection dialog with available firmware update

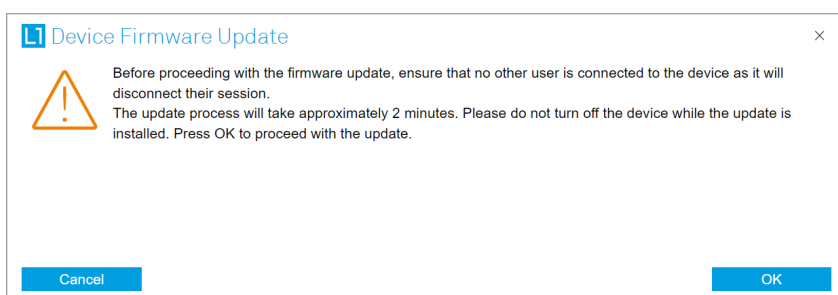


Figure 2.23: Device Firmware Update start-up dialog

## 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](mailto: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 SHFPPC Instrument is an advanced piece of laboratory equipment which has many features and capabilities. In order to benefit from these, the user needs access to a large number of settings in the API or 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. 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.



Problem	Check item
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.
The LabOne User Interface does not start	Verify that the LabOne Data Server ( <b>ziDataServer.exe</b> ) and the LabOne Web Server ( <b>ziWebServer.exe</b> ) are running via the Windows Task Manager. The Data Server should be started automatically by <b>ziService.exe</b> and the Web Server should be started upon clicking "Zurich Instruments LabOne" in the Windows Start Menu. 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 <b>127.0.0.1:8006</b> 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 <b>Control Panel → Display → Screen Resolution</b> . 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.

## 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 **LabOne Device Connection** dialog 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.

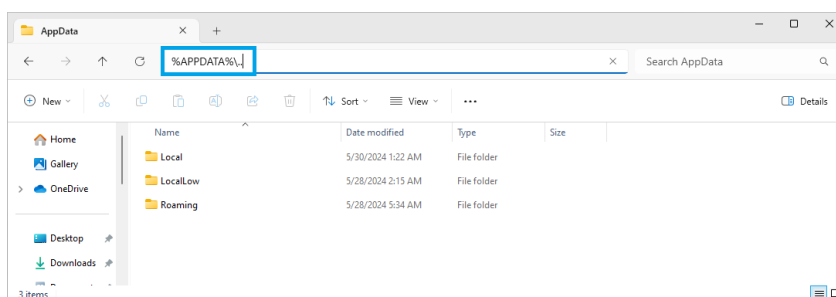


Figure 2.24: Using the

## 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 provided by the SHFPPC Instrument. The first section contains the description of the functional diagram and the hardware and software feature list. The next section details the front panel and the back panel of the measurement instrument. The following section provides product selection and ordering support.

## 3.1. Features

The SHFPPC Instrument consists of two or four pump and control channels for operating parametric amplifiers. Each Parametric Pump Channel consists of Pump Tone Generation and a Readout Signal Conditioning unit processing analog signals (dark blue color), and additional internal units that process digital data (light blue color). Here, the Pump Tone Generation unit generates both pump and probe tones for the control and characterization of Josephson parametric amplifiers, while the Readout Signal Conditioning unit consists of an amplification stage for the measurement readout signal as well as an interferometric pump tone cancellation circuit. The interface units that connect to the front panel are depicted on the left-hand side and the units at the back panel are depicted on the right-hand side of the Figure. Arrows between the panels and the interface units indicate selected physical connections and data flow.

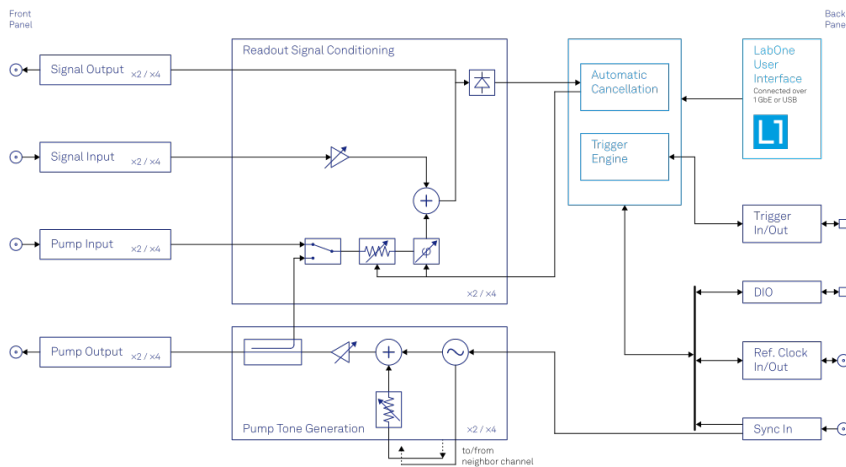


Figure 3.1: SHFPPC instrument functional diagram

### 3.1.1. Parametric Pump Channels

The Parametric Pump channel has signal generation and signal conditioning functionality.

#### Pump Tone Generation

- Low-noise pump tone outputs, recommended frequency range 4 - 12 GHz
- Calibrated output power, up to 20 dBm
- Integrated probe tone, recommended frequency range 4 - 8.5 GHz, combined with pump tone on a single output

#### Readout Signal Conditioning

- Low-noise readout signal amplification stage with 10 dB amplification, recommended frequency range 5 - 9 GHz
- Interferometric pump tone cancellation, recommended frequency range 5 - 10 GHz

## High-speed Connectivity

- SMA connectors on front and back panel for triggers, signals, and external clock
- LAN/Ethernet 1 Gbit/s controller interface
- USB 2.0 controller interface
- Clock input/output connectors (10/100 MHz)

## Software Features

- LabOne Graphic User Interface: Web-based with multi-instrument control
- Zurich Instruments LabOne Q software for high-level programming of quantum computing experiments.
- Data server with multi-client support
- LabOne APIs, including Python, C, LabVIEW, MATLAB, .NET
- Turnkey software and firmware features for fast system tune-up

## 3.2. Front Panel Tour

The front panel SMA connectors and control LEDs are arranged as shown in Figure 3.2 and listed in Table 3.1.

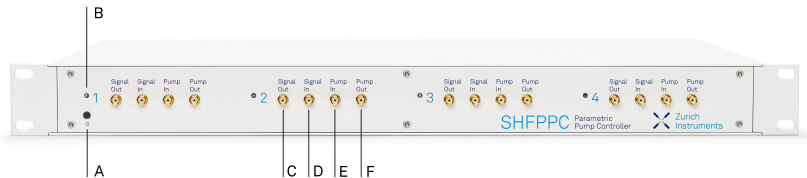


Figure 3.2: SHFPPC Parametric Pump Controller front panel

Table 3.1: SHFPPC Parametric Pump Controller front panel description

Position	Label / Name	Description
A	Soft power button	Power button with incorporated status LED off Instrument off and disconnected from mains power blue <ul style="list-style-type: none"> <li>- <b>dim:</b> Firmware is starting</li> <li>- <b>bright:</b> Instrument ready and active connection over Ethernet or USB</li> </ul>
B	Channel LED	Status indication for Parametric Pump Channel off RF Synthesizer disabled, no RF signal on Pump Out port blue RF Synthesizer enabled, pump tone or probe tone applied at Pump Out port orange RF Synthesizer enabled, in use to generate probe tone for twin channel; No RF signal applied at Pump Out port
C	Signal Out	Readout signal output
D	Signal In	Readout signal input
E	Pump In	External pump tone input
F	Pump Out	Parametric Pump Output

## 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 3.3](#) and [Table 3.2](#) for the detailed description of the items.

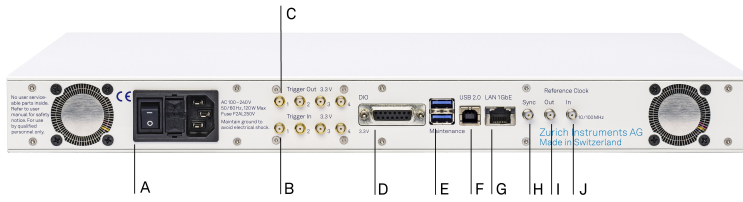


Figure 3.3: SHFPPC Parametric Pump Controller front panel

Table 3.2: SHFPPC Parametric Pump Controller front panel description

Position	Label / Name	Description
A	AC 100 - 240 V	Power inlet, fuse holder, and power switch
B	Trigger In	SMA: unidirectional CMOS/TTL ports for trigger input, 3.3 V
C	Trigger Out	SMA: unidirectional CMOS/TTL ports for trigger output, 3.3 V
D	DIO	16-bit digital input/output (DIO) connector
E	Maintenance	Universal Serial Bus (USB) port for maintenance → do not use for standard operation
F	USB 2.0	Universal Serial Bus (USB) port for instrument control
G	LAN 1GbE	1 Gbit LAN connector for instrument control
H	Sync	Sync Input; no user functionality
I	Reference Clock Out	External Reference Clock Output (10 MHz/100 MHz)
J	Reference Clock In	External Reference Clock Input (10 MHz/100 MHz)

## 3.4. Ordering Guide

[Table 3.3](#) provides an overview of the available SHFPPC products and options.

Table 3.3: SHFPPC Instrument product codes for ordering

Product code	Product name	Description	Field upgrade possible
SHFPPC2	SHFPPC Parametric Pump Controller 2-Channel Configuration	Base instrument with 2 PPC channels	no
SHFPPC4	SHFPPC Parametric Pump Controller 4-Channel Configuration	Base instrument with 4 PPC channels	no

Table 3.4: Product selector SHFPPC

Channel	Feature	SHFPPC2	SHFPPC4
Parametric Pump	Readout Channels	2	4
	Synthesizers	2	4
	Frequency range	recommended: 4-12 GHz	
Shared resources	Total number of Markers/Triggers	4/4	
	ZSync capability	no	
	USB 2.0	yes	
	LAN 1 Gbit/s	yes	

# 4. 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.

## 4.1. General Specifications

Table 4.1: General and storage

Parameter	Min	Typ	Max
storage temperature	-25 °C	-	65 °C
storage relative humidity (non-condensing)	-	-	95%
operating temperature	5 °C	-	40 °C
operating relative humidity (non-condensing)	-	-	90%
specification temperature	18 °C	-	28 °C
power consumption	-	-	120 W
operating environment	IEC61010, indoor location, installation category II, pollution degree 2		
operating altitude	up to 2000 meters		
power inlet fuses	250 V, 2 A, fast acting, 5 x 20 mm		
power supply AC line	100-240 V ( $\pm 10\%$ ), 50/60 Hz		
dimensions (width x depth x height)	44.0 x 38.0 x 4.4 cm (no handle), 17.3 x 15.0 x 1.7 inch, 19 inch rack compatible		
weight	6 kg (14 lb)		
recommended calibration interval	2 years		

Table 4.2: Maximum ratings

Parameter	Details	Min	Typ	Max
Damage threshold Pump Out	voltage level	-15 V	-	+15 V
Damage threshold Pump In	voltage level	-15 V	-	+15 V
	RF power	-	-	+20 dBm
Damage threshold Signal Out	voltage level	-15 V	-	+15 V
Damage threshold Signal In	voltage level	-15 V	-	+15 V
	RF power	-	-	+20 dBm
Damage threshold Trig Out	voltage level	-0.5 V	-	+3.8 V
Damage threshold Trig In	voltage level (50 $\Omega$ input impedance)	-0.5 V	-	+5.5 V
Damage threshold DIO Out	voltage level	-0.5 V	-	+3.8 V
Damage threshold DIO In	voltage level	-0.5 V	-	+6.5 V

Parameter	Details	Min	Typ	Max
Damage threshold External Clk Out	voltage level	-3 V	-	+3 V
Damage threshold External Clk In	voltage level	-15 V	-	+15 V
	RF power	-	-	+15 dBm

Table 4.3: Host computer requirements

Parameter	Description
supported Windows operating systems	Windows 10, 11 on x86-64
supported macOS operating systems	macOS 10.11+ on x86-64 and ARMv8
supported Linux distributions	GNU/Linux (Ubuntu 14.04+, CentOS 7+, Debian 8+) on x86-64 and ARMv8
supported processors	x86-64 (Intel, AMD), ARMv8 (e.g., Raspberry Pi 4 and newer, Apple M-series)

## 4.2. Analog Interface Specifications

Table 4.4: Pump Tone Generation (Parametric Pump Channels)

Parameter	Details	Min	Typ	Max	
Output impedance	-	-	50 $\Omega$	-	
Output coupling	-	AC			
Synthesizer frequency range	Pump Tone	-	4...12 GHz	2.5...12 GHz	
	Probe Tone	-	4...8.5 GHz	2.5...10 GHz	
Synthesizer frequency step	-	1 kHz	-	-	
Nominal output level range	Pump Tone	HIGH	-	0...20 dBm	-
		LOW	-	-20...0 dBm	-
	Probe Tone <sup>1</sup>	at 6 GHz	-	-35...-50 dBc	-
Maximum output level	Pump Tone, ALC enabled and low-pass filter disabled	4...8 GHz <sup>2</sup>	-	20 dBm	-
		8...9 GHz	-	18 dBm	-
		9...10 GHz	-	14 dBm	-
		10...11 GHz	-	12 dBm	-
		11...12 GHz	-	8 dBm	-
Output level error	Pump Tone	-	1.5 dB	-	
	Probe Tone	-	3 dB	-	
Output level step	Pump Tone	ALC enabled	-	0.01 dB	-
	Probe Tone	-	-	0.5 dB	-
Maximum output level drift with temperature	Pump Tone	ALC enabled	-	0.02 dB / °C	-
Maximum output level drift over time	Pump Tone	ALC enabled, at 25 °C	-	0.02 dB / 24h	-
Output phase noise	10 GHz	offset 1 kHz	-	-95 dBc/ Hz	-

Parameter	Details	Min	Typ	Max
	offset 10 kHz	-	-105 dBc/Hz	-
	offset 1 MHz	-	-115 dBc/Hz	-
	offset 10 MHz	-	-145 dBc/Hz	-

Table 4.5: Readout Signal Conditioning (Parametric Pump Channels)

Parameter	Details	Min	Typ	Max
Signal In impedance	-	-	50 Ω	-
Signal In coupling	-	AC		
Signal Out impedance	-	-	50 Ω	-
Signal Out coupling	-	AC		
Pump In impedance	-	-	50 Ω	-
Pump In coupling	-	AC		
Signal In to Signal Out frequency range	-	-	5...9 GHz	-
Signal In to Signal Out gain	-	-	10 dB	-
Signal In to Signal Out noise figure	8 GHz	-	5 dB	-
Signal In maximum level	8 GHz	-	-7 dBm	-
Pump tone cancellation frequency range	-	-	5...10 GHz	-
Cancellation tone attenuation span	-	-	50 dB	-
Cancellation tone phase shift span	-	360 deg	-	-

Table 4.6: Other Inputs and Outputs

Parameter	Details	Min	Typ	Max
Reference clock input	-	SMA on back panel		
Reference clock input impedance	-	-	50 Ω	-
Reference clock input coupling	-	AC		
Reference clock input frequency	-	10 / 100 MHz		
Reference clock input level range	-	-	-10...+14 dBm	-
Reference clock output	-	SMA on back panel		
Reference clock output impedance	-	-	50 Ω	-
Reference clock output coupling	-	AC		
Reference clock output frequency	-	10/100 MHz		
Reference clock output level	10MHz	-	+8 dBm	-
	100 MHz	-	+7 dBm	-

1. Relative to pump tone level, for a pump tone frequency of 6 GHz and a pump tone level of 10 dBm ↩
2. The upper end of each listed frequency range is excluded from the interval ↩



# 5. Device Node Tree

This chapter contains reference documentation for the settings and measurement data available on SHFPPC Instruments. Whilst [Functional Description](#) 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:

- [Introduction](#) for an introduction of how the instrument's settings and measurement data are organized hierarchically in the Data Server's so-called "Node Tree".
- [Reference Node Documentation](#) for a reference list of the settings and measurement data available on SHFPPC Instruments, organized by branch in the Node Tree.

## 5.1. 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 [LabOne Software Architecture](#) 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 [Node Properties and Data Types](#).

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 [Exploring the Node Tree](#).

## 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.

### 5.1.1. 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 <b>ZIDemodSample</b> . 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.
Pipelined	If the sequence pipeliner mode is off the value set to the node is applied immediately. Otherwise, it goes to the staging area of the sequence pipeliner instead. Multiple pipelined nodes can be programmed as part of a job definition, that is finalized by writing a one to the relevant <b>commit</b> node.

A node may contain data of the following types:

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, <b>ZIDemodSample</b> . 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

### 5.1.2. 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 5.1](#) 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 Config Tab (MATLAB, Python and .NET are available). The entire history generated in the current UI session can be viewed by clicking the "Show Log" button.

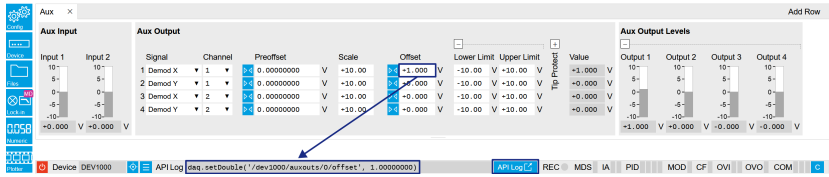


Figure 5.1: When a device's configuration is modified in the LabOne User Interface, the Status Bar displays the equivalent command to perform the same configuration via a LabOne programming interface. Here, the MATLAB code to modify auxiliary output 1's offset value is provided. When "Show Log" is clicked the entire configuration history is displayed in a new browser tab.

## 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 [Reference Node Documentation](#) 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
```

### 5.1.3. 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 Instrument Communication chapter of the LabOne Programming Manual

## 5.2. Reference Node Documentation

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

### 5.2.1. CLOCKBASE

#### `/dev..../clockbase`

Properties: Read  
 Type: Double  
 Unit: Hz

Returns the internal clock frequency of the device.

### 5.2.2. FEATURES

#### `/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.

### 5.2.3. PPCHANNELS

#### `/dev..../ppchannels/n/cancellation/attenuation`

Properties: Read, Write, Setting  
 Type: Double  
 Unit: dB

Sets the attenuation applied to the cancellation tone.

**/dev..../ppchannels/n/cancellation/on**

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

Enables pump tone cancellation.

**/dev..../ppchannels/n/cancellation/phaseshift**

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

Sets the phase shift applied to the cancellation tone.

**/dev..../ppchannels/n/cancellation/signalpower**

**Properties:** Read  
**Type:** Double  
**Unit:** dBm

Total signal power after cancellation. Provides an indication of how well the pump tone has been suppressed. This value is read-only.

**/dev..../ppchannels/n/cancellation/source**

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

Selects the source for the cancellation tone.

0 "internal": Self-generated pump tone via internal loop-back.  
 1 "external": External pump tone via the PUMP IN connector.

**/dev..../ppchannels/n/cancellation/sourcefreq**

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

Sets the frequency of the externally sourced cancellation tone. This value is locked to the self-generated pump tone frequency when internal loop-back is enabled.

**/dev..../ppchannels/n/synthesizer/probe/freq**

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

Sets the frequency of the generated probe tone.

**/dev..../ppchannels/n/synthesizer/probe/maxpower**

**Properties:** Read  
**Type:** Double  
**Unit:** dBm

Maximum probe tone power the instrument can provide at current operation point (depends on channel's pump tone settings, and probe tone frequency). This value is read-only.

**/dev..../ppchannels/n/synthesizer/probe/minpower**

**Properties:** Read  
**Type:** Double  
**Unit:** dBm

Minimum probe tone power the instrument can provide at current operation point (depends on channel's pump tone settings, and probe tone frequency). This value is read-only.

**/dev..../ppchannels/n/synthesizer/probe/on**

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

Enables probe tone output.

**/dev..../ppchannels/n/synthesizer/probe/power**

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

Sets the output power of the generated probe tone.

**/dev..../ppchannels/n/synthesizer/pump/alc**

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

Enables automatic level control (ALC) for pump tone output.

**/dev..../ppchannels/n/synthesizer/pump/filter**

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

Enables the integrated low-pass filter on the pump tone signal path. Filter cut-off frequency is automatically adjusted based on pump tone frequency.

**/dev..../ppchannels/n/synthesizer/pump/freq**

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

Sets the frequency of the generated pump tone.

**/dev..../ppchannels/n/synthesizer/pump/maxpower**

**Properties:** Read  
**Type:** Double  
**Unit:** dBm

Maximum pump tone power the instrument can provide at current operation point (depends on ALC mode, low-pass filter mode, power range setting, and frequency). This value is read-only.

**/dev..../ppchannels/n/synthesizer/pump/minpower**

**Properties:** Read  
**Type:** Double  
**Unit:** dBm

Minimum pump tone power the instrument can provide at current operation point (depends on ALC mode, low-pass filter mode, power range setting, and frequency). This value is read-only.

**/dev..../ppchannels/n/synthesizer/pump/on**

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

Enables pump tone output.

**/dev..../ppchannels/n/synthesizer/pump/power**

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

Sets the output power of the generated pump tone.

**/dev..../ppchannels/n/synthesizer/pump/range**

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

Sets the channel's RF synthesizer output power range mode. A precise indication of the instrument's power limits in the individual modes is available by query of the MAXPOWER and MINPOWER nodes.

- 0 "autorange": AUTO: The instrument automatically selects the range based on requested pump tone power.
- 1 "lowrange": LOW: The instrument is configured for lower power output.
- 2 "highrange": HIGH: The instrument is configured for higher power output.

**/dev..../ppchannels/n/synthesizer/status**

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

Current operational mode of the channel's RF synthesizer board. This value is read-only.

- 0 "pump": Synthesizer is configured for pump tone output.
- 1 "probe": Synthesizer is providing a probe tone to the adjacent channel.

**5.2.4. STATS****/dev..../stats/cmdstream/bandwidth**

**Properties:** Read  
**Type:** Double  
**Unit:** Mbit/s

Command streaming bandwidth usage on the physical network connection between device and data server.

**/dev..../stats/cmdstream/bytesreceived**

**Properties:** Read  
**Type:** Integer (64 bit)  
**Unit:** B

Number of bytes received on the command stream from the device since session start.

**/dev..../stats/cmdstream/bytessent**

**Properties:** Read  
**Type:** Integer (64 bit)  
**Unit:** B

Number of bytes sent on the command stream from the device since session start.

**/dev..../stats/cmdstream/packetslost**

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

Number of command packets lost since device start. Command packets contain device settings that are sent to and received from the device.

**/dev..../stats/cmdstream/packetsreceived**

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

Number of packets received on the command stream from the device since session start.

**/dev..../stats/cmdstream/packetsst**

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

Number of packets sent on the command stream to the device since session start.

**/dev..../stats/cmdstream/pending**

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

Number of buffers ready for receiving command packets from the device.

**/dev..../stats/cmdstream/processing**

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

Number of buffers being processed for command packets. Small values indicate proper performance. For a TCP/IP interface, command packets are sent using the TCP protocol.



**/dev..../stats/datastream/bandwidth**

**Properties:** Read  
**Type:** Double  
**Unit:** Mbit/s

Data streaming bandwidth usage on the physical network connection between device and data server.

**/dev..../stats/datastream/bytesreceived**

**Properties:** Read  
**Type:** Integer (64 bit)  
**Unit:** B

Number of bytes received on the data stream from the device since session start.

**/dev..../stats/datastream/packetslost**

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

Number of data packets lost since device start. Data packets contain measurement data.

**/dev..../stats/datastream/packetsreceived**

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

Number of packets received on the data stream from the device since session start.

**/dev..../stats/datastream/pending**

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

Number of buffers ready for receiving data packets from the device.

**/dev..../stats/datastream/processing**

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

Number of buffers being processed for data packets. Small values indicate proper performance. For a TCP/IP interface, data packets are sent using the UDP protocol.

**/dev..../stats/physical/ppchannels/n/alctrl**

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

[empty]

**/dev..../stats/physical/ppchannels/n/alcpower**

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

[empty]

**/dev..../stats/physical/ppchannels/n/cancelpower**

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

[empty]

**/dev..../stats/physical/ppchannels/n/temperatures/n**

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

[empty]

**/dev..../stats/physical/ppchannels/n/voltages/n**

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

[empty]

**/dev..../stats/physical/voltages/n**

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

Internal voltage measurements.

## 5.2.5. STATUS

**/dev..../status/fifolevel**

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

USB FIFO level: Indicates the USB FIFO fill level inside the device. When 100%, data is lost

**/dev..../status/flags/packetlosstcp**

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

Flag indicating if tcp packages have been lost.

**/dev.... /status/flags/packetlossudp**

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

Flag indicating if udp packages have been lost.

**/dev.... /status/time**

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

The current timestamp.

## 5.2.6. 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 Reference clock has been locked on.  
 1 There was an error locking onto the reference clock signal.  
 2 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. When the clock output is turned on or off, all the instruments connected with ZSync links will be disconnected. The connection should be re-established manually.

**/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. When the frequency is changed, all the instruments connected with ZSync links will be disconnected. The connection should be re-established manually.

**/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/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/interfacespeed**

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

Speed of the currently active interface (USB only).

**/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/owner**

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

Returns the current owner of the device (IP).

**/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/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/update**

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

Requests update of the device firmware and bitstream from the dataserver.