PQSC User Manual

Zurich
Instruments

PQSC User Manual

Zurich Instruments AG

Revision 20.07.0

Revision History

Revision 20.07.0, 28-Aug-2020:

The document was updated to comply with the changes of the 20.07 LabOne release.

Highlights of the changes and additions to the PQSC product are:

- Support synchronization with the UHFQA
- Add realtime fast feedback architecture
- Add Lookup Table decoder for feedback
- General stability improvements

A more detailed list of all technical changes can be found in the LabOne release notes.

Revision 20.01.0, 28-Feb-2020:

First version of PQSC User Manual.

Table of Contents

Declaration of Conformity

The manufacturer

Zurich Instruments
Technoparkstrasse 1
8005 Zurich
Switzerland

declares that the product

PQSC Programmable Quantum System Controller/s

fulfills the requirements of the European guidelines

- 2014/30/EU Electromagnetic Compatibility
- 2014/35/EU Low Voltage Directive
- 2011/65/EU, 2015/863/EU, 2017/2102/EU Restriction of Hazardous Substances (RoHS)
- 1907/2006/EC Registration, Evaluation, Authorization, and Restriction of Chemicals (REACH)

The assessment was performed using the directives according to Table 1.

Table 1. Conformity table

EN 61326-1:2012	Electrical equipment for measurement, control and laboratory use EMC requirements - Part 1: General requirements - Emission: CISPR11, IEC 61000-3-2/3 - Immunity: IEC 61000-4-2/3/4/5/6/8/11
IEC 61010-1:2010 + AMD1:2016 IEC 61010-1 National Deviations for EU, US, CA	Safety requirements for electrical equipment for measurement, control and laboratory use

For the assessment of electromagnetic compatibility, the limits of radio interference for Class B equipment as well as the immunity to interference for operation in industry have been used as a basis.

Chapter 1. Getting Started

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

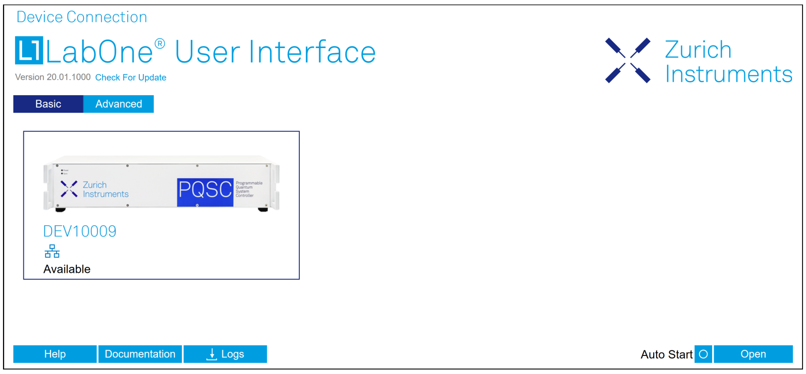
- A Quick Start Guide for the impatient
- Inspecting the package content and accessories
- List of essential handling and safety instructions
- Connecting to the PQSC Instrument
- Handy list of troubleshooting guidelines

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

1.1. 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 Chapter 1.
2. Check the Handling and Safety Instructions in Section 1.3.
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 Section 1.4.
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.



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 Section 1.5.

If any problems occur whilst setting up the instrument and software please see the troubleshooting section at the end of this chapter.

Chapter 4 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 web page will serve as a valuable resource that is constantly updated and expanded.

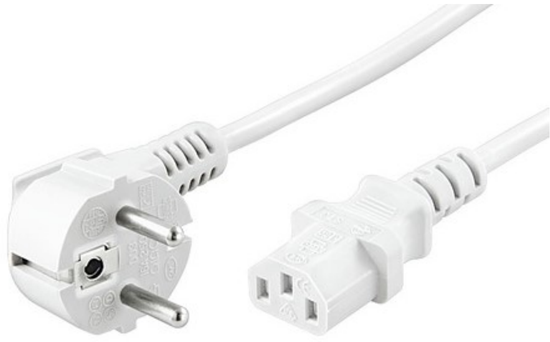
1.2. Inspect the Package Contents

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:

- You have received 1 Zurich Instruments PQSC Instrument
- You have received 1 power cord with a power plug suited to your country
- You have received 1 USB 3.0 cable and/or 1 LAN cable (category 5/6 required) and at least 1 ZSync cables. These are used to connect the ZSync ports of Zurich Instruments devices
- You have received a printed version of the "Getting Started" section
- The "Next Calibration" sticker on the rear panel of the Instrument indicates approximately 2 years ahead in time. Zurich Instruments recommends calibration intervals of 2 years
- The MAC address of the instrument is displayed on a sticker on the back panel

Table 1.1. Package contents for the PQSC Instrument

<table>
<tr><td colspan="2">
</td></tr>
<tr><td>
the power cord (e.g. EU norm)</td><td>
the USB 3.0 cable</td></tr>
</table>

 the power inlet, with power switch and fuse holder	 the LAN / Ethernet cable (category 5/6 required)
 the ZSync cable	
 the "Next Calibration" sticker on the back panel of the instrument	 the MAC address sticker on the back panel of the instrument

The PQSC Instrument is equipped with a multi-mains switched power supply, and therefore can be connected to most power systems in the world.

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 at `<support@zhinst.com>`.

1.3. Handling and Safety Instructions

The PQSC Instrument is a sensitive piece of electronic equipment, which under no circumstances should 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 cancels 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.

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 affect correct 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 1.2. 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.
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 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, 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 the correct interpretation of the acquired measurement data.

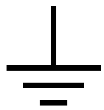
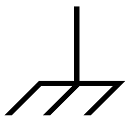
High voltage transients due to inductive loads	When measuring devices with high inductance, take adequate measures to protect the Signal Input connectors against the high voltages of inductive load switching transients. These voltages can exceed the maximum voltage ratings of the Signal Inputs and lead to damage.
Location and ventilation	This instrument or system is intended for indoor use in an installation category II and pollution degree 2 environment as per IEC 61010-1. Do not operate or store the instrument outside the ambient conditions specified in . 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 socket 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 at the instrument outside the ambient conditions specified in .
Handling	Do not drop the Instrument, handle with due care, do not store liquids on the device as there is a chance of spilling and 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.

When 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 by email at <support@zhinst.com>.

Table 1.3. 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 only with a power cord specified for this product and certified for the country of use.
Instrument emits abnormal noise, smell, or sparks	Switch off the Instrument immediately to prevent large damage.
Instrument is damaged	Switch off the Instrument immediately and secure it against unintended operation.

Table 1.4. Symbols

	Earth ground
	Chassis ground
	Caution. Refer to accompanying documentation

1.4. Software Installation

The PQSC Instrument is operated from a host computer with the LabOne software. To install the LabOne software on a PC, administrator rights are required. In order to simply run the software later, a regular user account is sufficient. Instructions for downloading the correct version of the software packages from the Zurich Instruments website are described below in the platform dependent sections. It is recommended to regularly update to the latest software version provided by Zurich Instruments. Thanks to the Automatic Update check feature, the update can be initiated with a single click from within the user interface as shown in Section 1.6.

1.4.1. Installing LabOne on Windows

The installation packages for Zurich Instruments LabOne software are available as Windows installer .msi packages. The software is available on the Zurich Instruments download page, www.zhinst.com/downloads. Please ensure that you have administrator rights for the PC where the software is to be installed and that you download the correct software installer for the PC's processor architecture (32-bit or 64-bit), for help see the section called "Determining PC Architecture on Microsoft Windows". See www.zhinst.com/labone/compatibility for a comprehensive list of supported Windows systems.

Determining PC Architecture on Microsoft Windows

In case you are unsure which Windows architecture you are using, it can be checked as follows:

- Windows 7: `Control panel → System and Security → System/System type`
- Windows 8: `Control panel → System → System/System type`
- Windows 10: `Settings → System → About/System type`

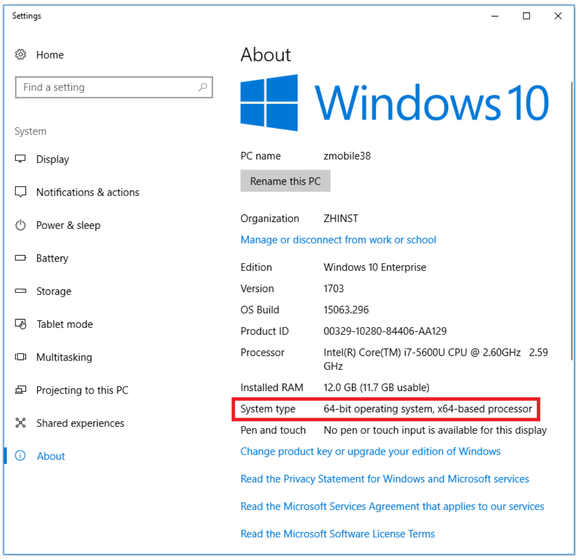


Figure 1.1. Find out the OS addressing architecture (32-bit or 64-bit)

Windows LabOne Installation

1. The PQSC 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 `LabOne32/64-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.

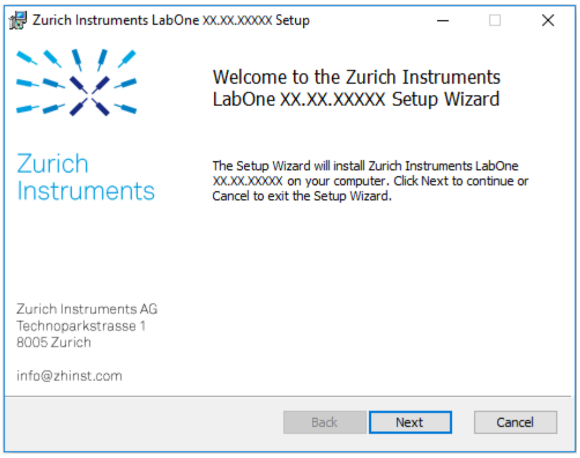


Figure 1.2. Installation welcome screen

- 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 PQSC Instrument the "PQSC Series Device, Web Server" and "API" features are required. Please install the features for other device classes as well as required. If you would like to install shortcuts on your desktop area enable the feature "Desktop Shortcuts". To proceed click the **Next** button.

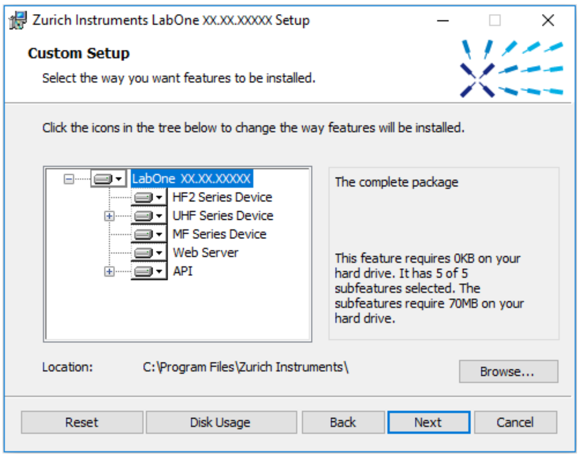


Figure 1.3. Custom setup screen

- Select whether the software should periodically check for updates. The software will not update automatically even with enabled periodic check for updates. This setting can later be changed in the user interface. To proceed click the **Next** button.

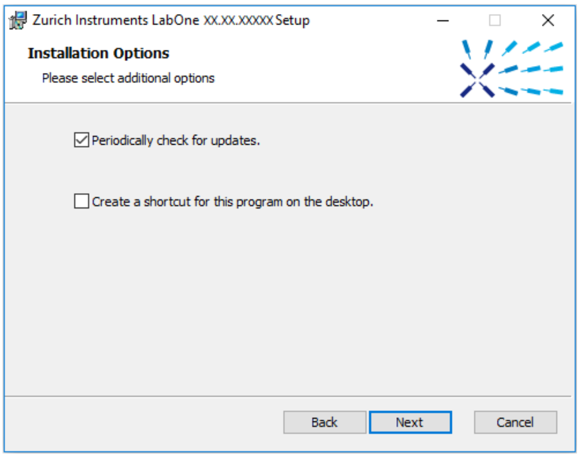


Figure 1.4. Automatic update check

- Click the **Install** button to start the installation process.
- Windows will ask up to two times to reboot the computer. Make sure you have no unsaved work on your computer. Actually a reboot is practically never required, so that one may safely click **OK**.

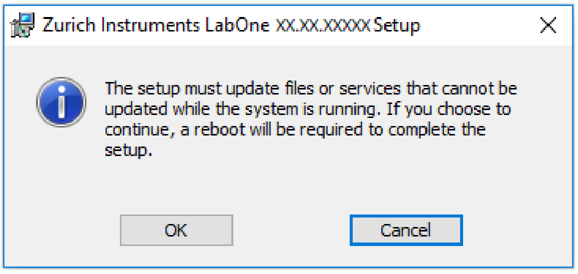


Figure 1.5. Installation reboot request

- On Windows Server 2008 and Windows 7 it is required to confirm the installation of up to 2 drivers from the trusted publisher Zurich Instruments. Click on **Install**.

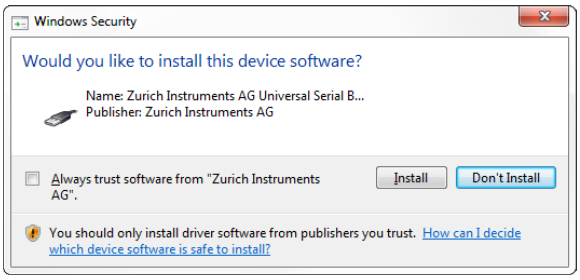


Figure 1.6. Installation driver acceptance

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

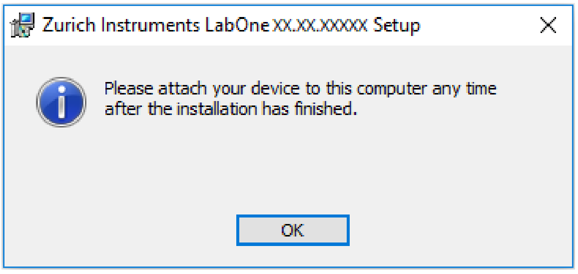


Figure 1.7. Installation completion screen

3. Click **Finish** to close the Zurich Instruments LabOne installer.
4. You can now start the LabOne User Interface as described in Section 1.5.2 and choose an instrument to connect to via the Device Connection dialog shown in Figure 1.10.

Warning

Do not install drivers from another source and therefore not trusted as originating from Zurich Instruments.

1.4.2. Installing LabOne on Linux

Requirements

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

1. Officially, Ubuntu 14.04 LTS and 16.04 LTS (amd64) are supported although in practice LabOne software may work on other platforms. Please ensure that you are using a Linux distribution that is compatible with Ubuntu/Debian.
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 downloads page:
   - LabOneLinux<arch>-<release>.<revision>.tar.gz, for example:

     ```
     LabOneLinux32/64-16.12.41721.tar.gz
     ```

   Please ensure you download the correct architecture (32-bit/64-bit) 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 "`x686`" the 32-bit version of the LabOne package is required, if it displays "`x86_64`" the 64-bit version is required.

Linux LabOne Installation

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

1. Extract the LabOne tarball in a temporary directory:

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

2. Navigate into the extracted directory.

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

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

```
sudo bash install.sh
```

The install script lets you choose between the following three modes:

- Type "a" to install the Data Server program, the Web Server program, documentation and APIs.
- Type "u" to install `udev` support (only necessary if HF2 Instruments will be used with this LabOne installation and not relevant for other instrument classes).
- Type "ENTER" to install both options "a" and "u".

4. Test your installation by running the software as described in the next section.

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 Section 1.5.2 and choose an instrument to connect to via the Device Connection dialog shown in Figure 1.10.

Important

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

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
```

1.4.3. Start LabOne Manually on the Command Line

After installing the LabOne software, the Web and Data Server can be started manually using the command-line. The more common way to start LabOne under windows is described in Section 1.5.2. The advantage of using the command line is being able to observe and change the behaviour of the Web and Data Server. 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 and Data Server. They are installed in the Program Files Folder (usually: C:\Program Files) under \Zurich Instruments \LabOne in the WebServer or DataServer folder respectively. The Web and Data Server (ziDataServer) is started by running the respective executable in each folder. Please be aware that only one instance of each Server may run at a time per computer. The behaviour of the Server can be changed by providing command line arguments. For the Web Server the following arguments are possible:

```
Options:
  -h [ --help ]              Produce help message
  --ip arg (=0.0.0.0)        IP address to which the webserver should listen
  --port arg                 The ports on which the webserver should listen,
                             default is 8006 if none is specified
  -w [ --websocket ] arg (=1)  Enable WebSocket use, if available, for data
                             streaming to the browser. Default is on, will
                             use AJAX if switched off.
  --discovery arg (=1)       Enable discovery based session dialog. Default
                             is on.
  --multicast-hops arg (=1)  Set number of IP multicast hops
  --server-ip arg (=127.0.0.1) IP address of the ziDataServer
  --server-port arg (=8004)  The port on of the ziDataServer
  -r [ --resource-path ] arg  Home directory for the web server (read access
                             rights)
  -d [ --data-path ] arg     Data directory for the web server (write access
                             rights)
  -s [ --setting-path ] arg  Setting directory for the web server (write
                             access rights)
  -l [ --log-path ] arg      Log directory for the web server (write access
                             rights)
  -D [ --doc-path ] arg      Documentation directory for the web server (read
                             access rights)
  --firmware-path arg        Directory containing device firmware (read
                             access rights)
  --api-log arg (=268435711)  API command log mask
  -a [ --auto-start ] arg (=0) Start browser page automatically
  --debug arg (=3)           Set the debug level (trace:0, info:1, debug:2,
                             warning:3, error:4, fatal:5, status:6)
  --api-level arg (=6)       Stick to the given ziAPI version: 1 = ziAPI v1;
                             4 = ziAPI v4; 5 = ziAPI v5; 6 = ziAPI v6
                             (default)
  --hide-console arg (=0)    Start process without console window.
  --wait-on-exit arg (=0)    Wait for key press on exit of the server
  -i [ --info ] arg          Output requested information and exit, available
                             arguments are: product, version, revision
  --dir-watching arg (=1)    If true (default), enable monitoring of changes
                             to directories and files used by LabOne. Affects
                             the Config and File Manager tabs.
```

For the Data Server the following arguments are possible:

```
Will log to directory '/tmp/ziDataServerLog_danielw'
Options:
  -h [ --help ]             Produce help message
  --port arg (=8004)        The port on which the server should listen
  --open-override arg (=0)  Start the server listening on connections from
                            outside
  --debug arg (=3)          Set the debug level (trace:0, info:1, debug:2,
                            warning:3, error:4, fatal:5, status:6)
  --device-ip arg           Set static IP for the device, or specify 'usb' to
                            force connection via USB
  --device-port arg (=8010)  Set TCP/IP port for the device
  --interface-ip arg (=1)   Enables devices search on ip via multicast
  --interface-usb arg (=1)  Enables devices search on usb
  --interface-pcie arg (=0)  Enables devices search on PCIe
  --auto-connect arg (=1)   Auto-connects a device if on USB or PCIe.
```

```
--discovery arg (=1)       Enable UDP multicast data server discovery.
                           Default is on.
--multicast-hops arg (=1)  Set number of IP multicast hops
--device-serial arg        Connects to a given device serial (devnnn). Will
                           automatically detect if not specified.
--buffer-size arg (=65536) Session buffer size (>=32768)
--firmware-path arg        Directory containing device firmware (read access
                           rights)
-l [ --log-path ] arg      Log directory for the server (write access rights)
--wait-on-exit arg (=0)    Wait for key press on exit of the server
--hide-console arg (=0)    Start process without console window.
-i [ --info ] arg          Output requested information and exit, available
                           arguments are: product, version, revision
```

1.5. Connecting to the Instrument

The Zurich Instruments PQSC 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)

Chrome Firefox Opera Edge Safari

In order to physically connect to the instrument, integrate the instrument 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 device in the same LAN. 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 an RNDIS driver on the host computer. For PC users, this driver included in the LabOne software installer. The driver is available online for Mac users.

1.5.1. 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 1.8. 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. This 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.

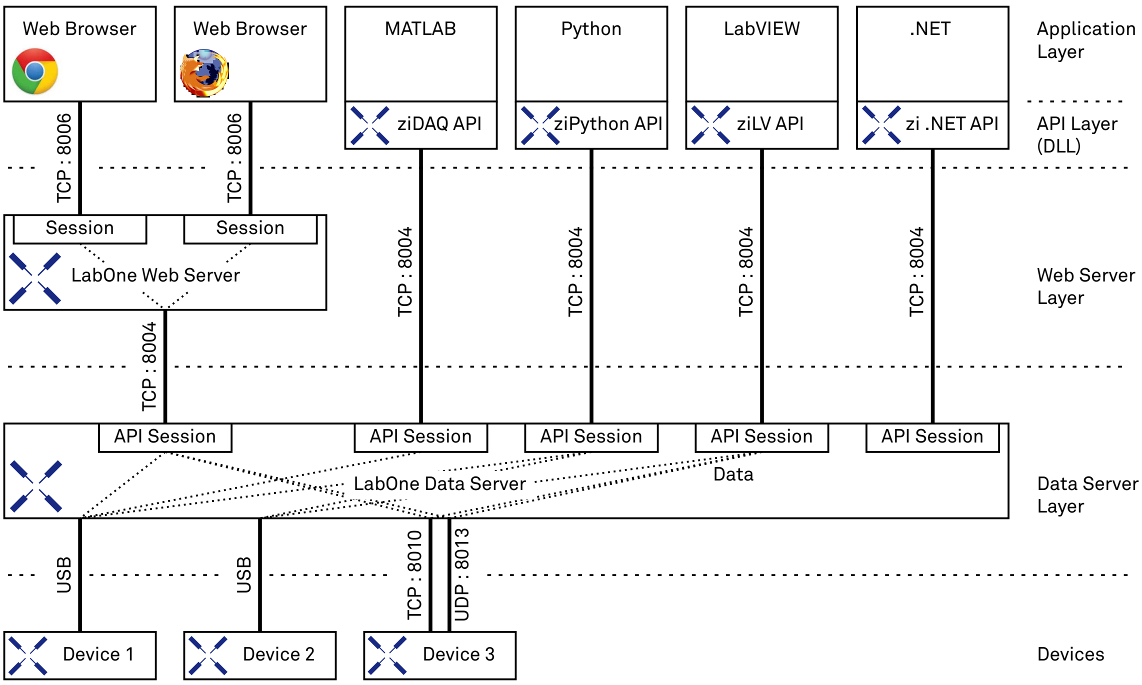


Figure 1.8. Software architecture

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. That is to say 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.

1.5.2. 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 Section 1.4 Software Installation. If the device is not yet connected please find more information in Section 1.5 Device Connectivity.

The LabOne User Interface start-up link can be found under the Windows 10 Start Menu[1] as shown in Figure 1.9: 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 Section 1.5.1.

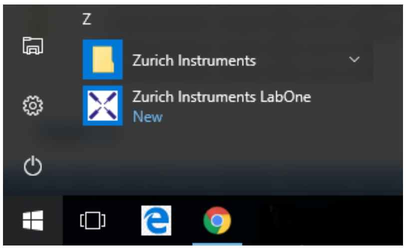


Figure 1.9. Link to the LabOne User Interface in the Windows 10 Start Menu

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 run the LabOne User Interface in Chrome

[1] Under Windows 7 and 8, the LabOne User Interface start-up link can be found in `Start Menu → all programs / all apps → Zurich Instruments LabOne`.

in application mode which improves the user experience by removing the unnecessary browser controls.

After starting LabOne, the Device Connection dialog in Figure 1.10 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 Section 1.5.1.

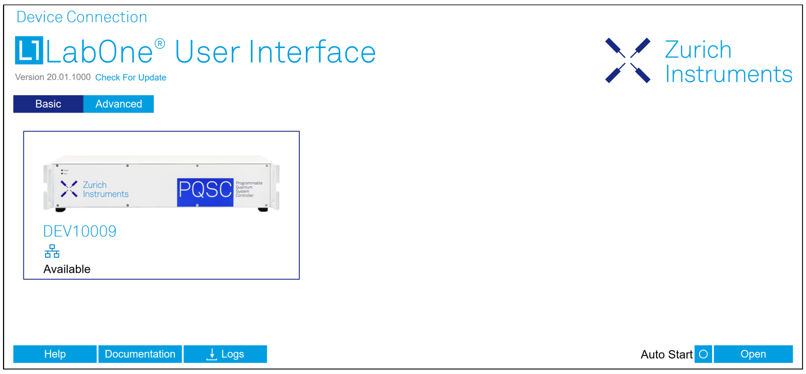


Figure 1.10. Device Connection dialog

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 which are particularly useful for multi-instrument setups.

The Advanced view consists of three parts: Data Server Connectivity, Available Devices, and 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, also connections via Data Servers running on other PCs in the network 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

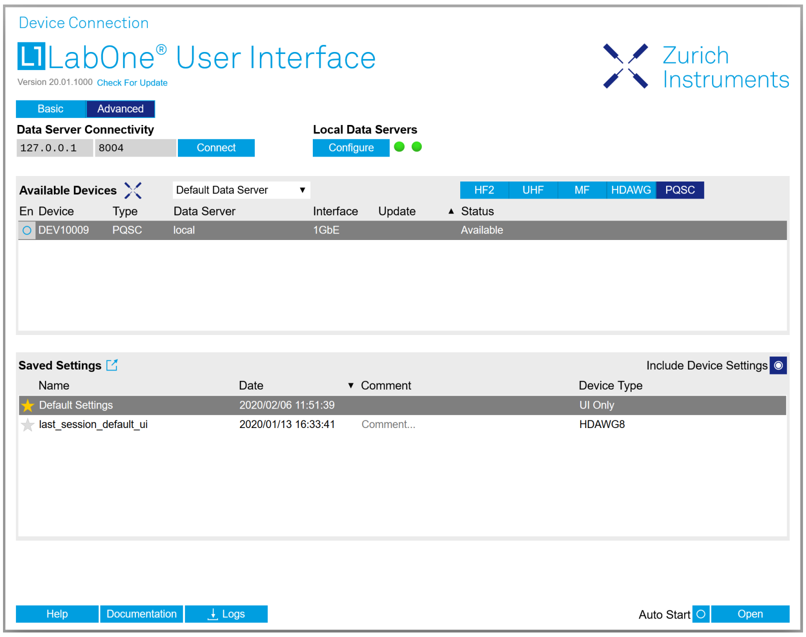


Figure 1.11. Device Connection dialog (Advanced view)

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, HD) are used in parallel, the list of 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 where 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, or HDAWG 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", cf. Section 4.4.

On the remote computer (Computer 2), open the Device Connection dialog by starting up the LabOne User Interface. 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 of 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 UHF Instruments the interfaces USB or 1GbE are available. The interface is listed if it is physically connected. For MF series instruments the interface is indicated as PCIe in case the Data Server is running on the instrument, even if the physical connection between PC and instrument is USB or 1GbE. PCIe corresponds to the interface between the embedded PC and the measurement unit inside the MF instrument. 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.

The last column indicates the status of the device. Table 1.5 explains the meaning of some of the possible device statuses.

Table 1.5. 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 **Open** 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 Section 1.6.2.
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 1.6. 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 1.6. 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 .xml.
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 1.7.

Table 1.7. Button Descriptions

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

Note

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

Note

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

Note

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

Double clicking on a row in the Saved Settings table is a quick way of loading the LabOne UI with the 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. Via a right click 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 e.g. useful when setting up a connection to multiple instruments.

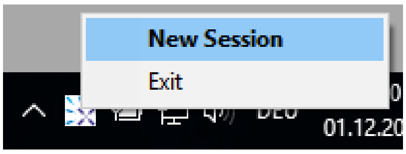


Figure 1.12. LabOne Tray Icon in Windows 10

Messages

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

Lost Connection to the LabOne Web Server

In this case the browser is no longer able to connect to the LabOne Web Server. This can happen if the Web Server and Data Server run on different PCs and a network connection is interrupted. As long as the Web Server is running and the session did not yet time out, it is possible to just attach to the existing session and continue. Thus, within about 15 seconds it is possible with **Retry** to recover the old session connection. The **Reload** button opens the Device Connection dialog shown in Figure 1.10. The figure below shows an example of the Connection Lost dialog.

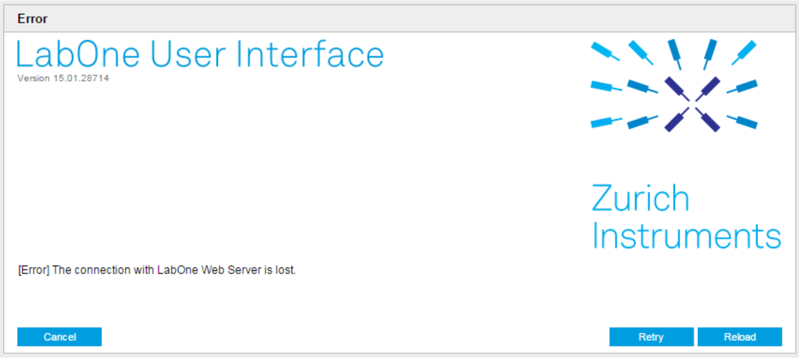


Figure 1.13. Dialog: Connection Lost

Reloading...

If a session error cannot be handled, the LabOne Web Server will restart to show a new Device Connection dialog as shown in Figure 1.10. 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.

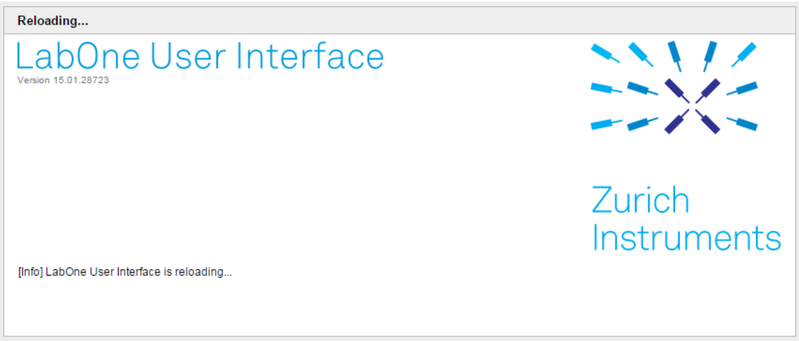


Figure 1.14. Dialog: Reloading

No Device Discovered

An empty "Available Devices" table means that no devices were discovered. This can mean that no LabOne Data Server is running, or that it is running but failed to detect any devices. The device may be switched off or the interface connection fails. For more information on the interface between device and PC see Section 1.5. The figure below shows an example of this dialog.

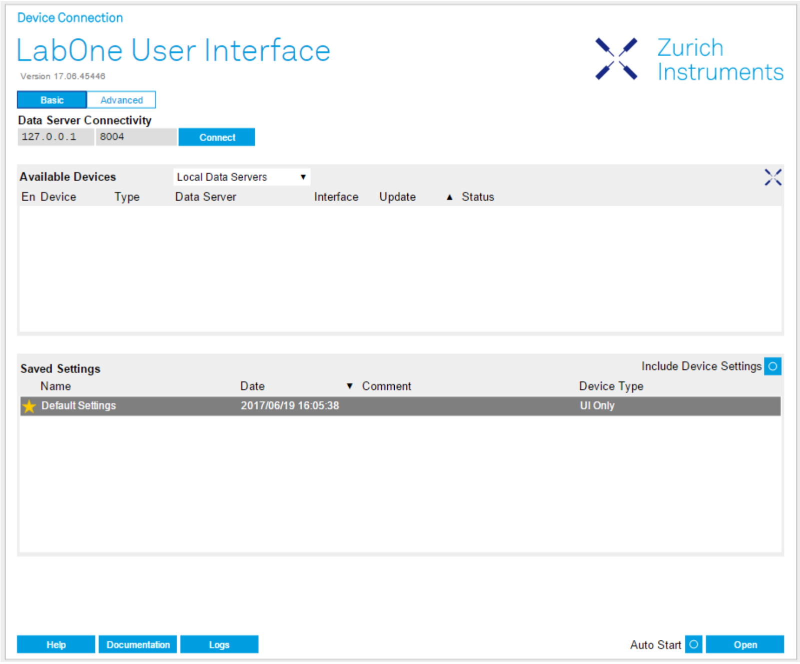


Figure 1.15. No Device Discovered

No Device Available

If all the devices in the "Available Devices" table are shown grayed, this indicates that they are either in use by another Data Server, or need a firmware upgrade. For firmware upgrade see Section 1.6.2. If all the devices are in use, access is not possible until a connection is relinquished by the another Data Server.

1.5.3. 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.

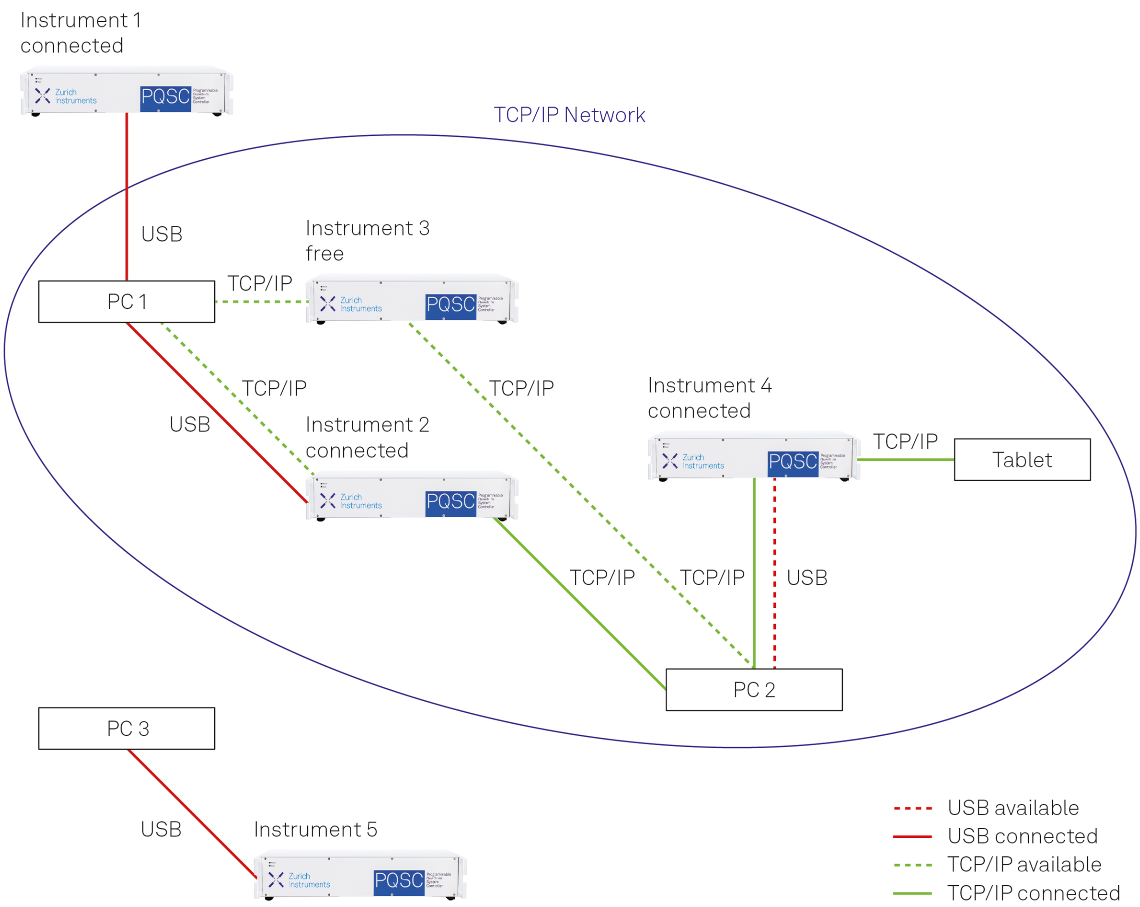


Figure 1.16. Connectivity

Figure 1.16 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.

1.5.4. 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 Figure 1.17 (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.

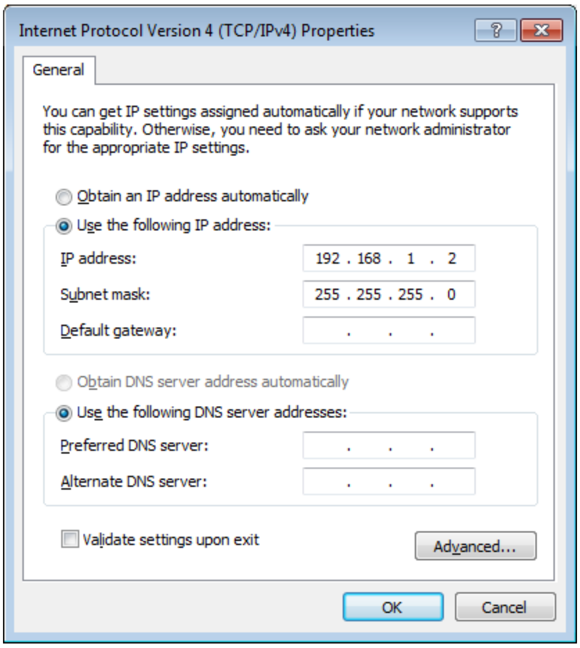


Figure 1.17. Static IP configuration for the host computer

Requirements

- Two networks 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.

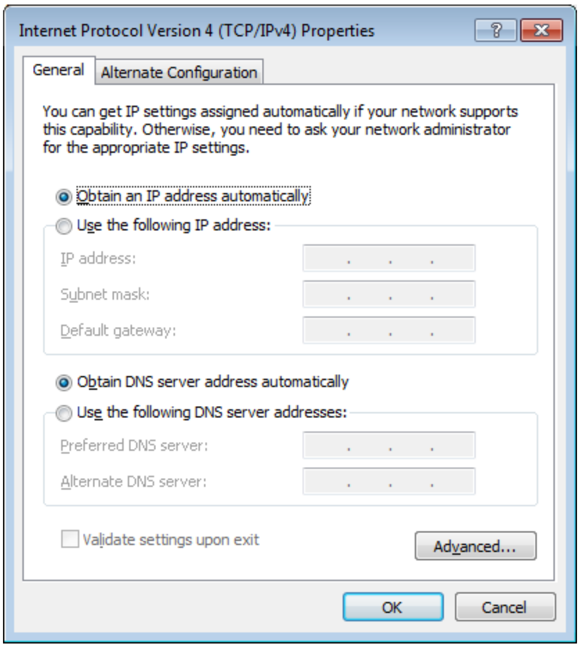


Figure 1.18. Dynamic IP configuration for the host computer

Static Device IP

Using a static IP address for the host computer is necessary to set up a point-to-point network. On top of that, a static device IP configuration can be necessary in the rare cases in which the network card does not support multicast.

1. Connect the 1GbE port of the network card that is dedicated for device connectivity directly to the 1GbE port of the instrument.
2. In the Device tab of the LabOne user interface, enable the setting "Static IP" with the IP Address 192.168.1.10, click on "Program", and restart the instrument using the soft power button
3. Modify the shortcut of the LabOne User Interface and LabOne Data Server in the Windows Start menu. Right-click and go to Properties, then add the following command line argument to the Target field: `--device-ip 192.168.1.10`.

   The LabOne User Interface shortcut Target field should look like this:

   ```
   "C:\Program Files\Zurich Instruments\LabOne\WebServer\ziWebServer.exe"
   --auto-start=1 --server-port=8004 --resource-path "C:\Program Files
   \Zurich Instruments\LabOne\WebServer\html\\" --device-ip 192.168.1.10
   ```

   The LabOne Data Server shortcut Target field should look like this:

   ```
   "C:\Program Files\Zurich Instruments\LabOne\DataServer
   \ziDataServer.exe" --device-ip 192.168.1.10
   ```

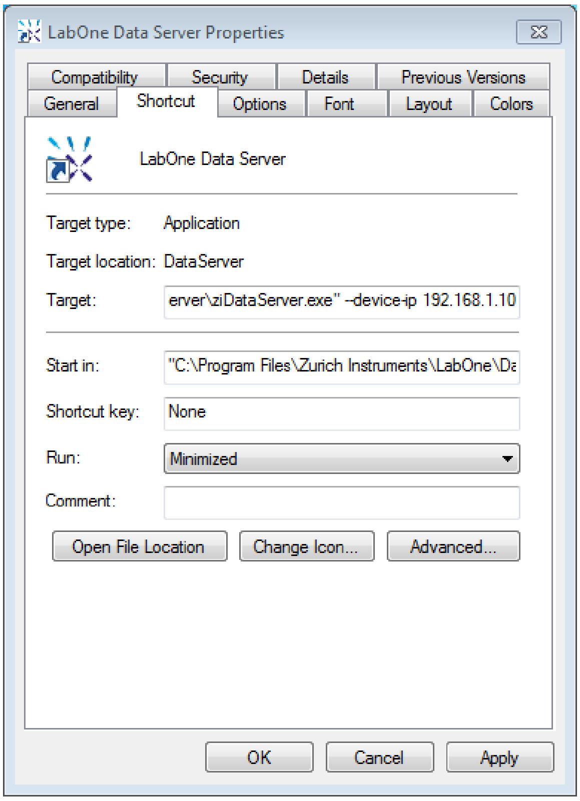


Figure 1.19. Static IP shortcut modification

4. (Optional) To verify the connection between the host computer and the PQSC Instrument, open a DOS command window and ping the IP address entered above

Requirements

- Device IP must be known
- Needs network administrator support on networks with dynamic IP configuration

1.6. Software Update

1.6.1. Updating LabOne using Automatic Update Check

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 1.20 will start a download the latest LabOne installer for Windows or Linux, see Figure 1.21. After download, proceed as explained in Section 1.4 to update LabOne.

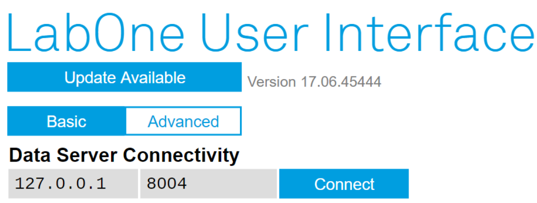


Figure 1.20. Device Connection dialog: LabOne update available

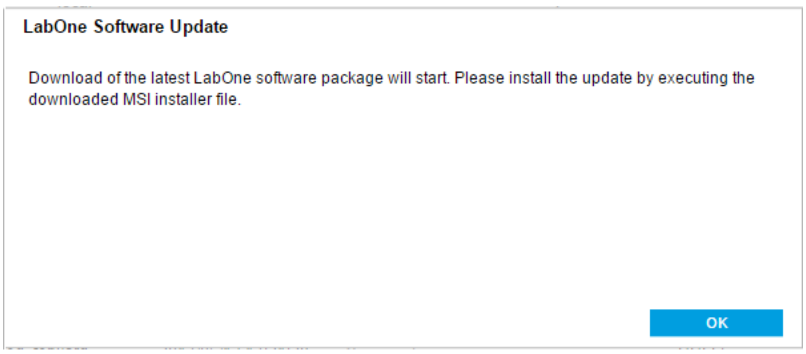


Figure 1.21. Download LabOne MSI using Automatic Update Check feature

1.6.2. 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 1.22. In the Advanced view, there will be a link "Upgrade FW" in the Update column of the Available Devices table. Click on Upgrade FW or Upgrade FW , respectively, to open the firmware update start-up dialog shown in Figure 1.23. The firmware upgrade takes approximately 2 minutes.

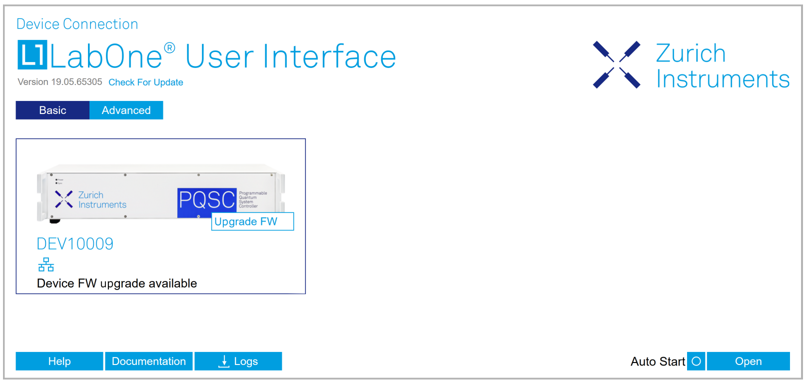


Figure 1.22. Device Connection dialog with available firmware update

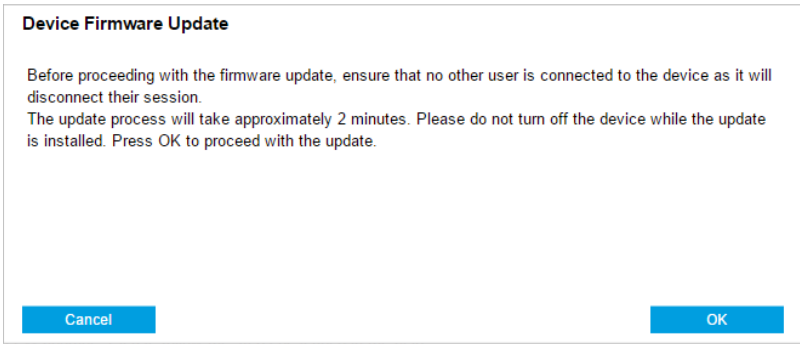


Figure 1.23. Device Firmware Update start-up dialog

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 whilst upgrading the instrument firmware, please contact Zurich Instruments at support@zhinst.com.

1.7. Troubleshooting

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

1.7.1. Common Problems

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

The software cannot be installed or uninstalled: please verify you have Windows administrator rights.

The software cannot be updated: please use the Modify option in Windows Apps & Features functionality. In the software installer select Repair, then uninstall the old software version, and install the new version.

The Instrument does not turn on: please verify the power supply connection and inspect the fuse. The fuse holder is integrated in the power connector on the back panel of the instrument.

The sample stream from the Instrument to the host computer is not continuous: check the communication (COM) flags in the status bar. The three flags indicate occasional sample loss, packet loss, or stall. Sample loss occurs when a sampling rate is set too high (the instruments sends more samples than the interface and the host computer can absorb). The packet loss indicates an important failure of the communications to the host computer and compromises the behavior of the instrument. Both problems are prevented by reducing the sample rate settings. The stall flag indicates that a setting was actively changed by the system to prevent UI crash.

The LabOne User Interface does not start (when running the LabOne on a PC): verify that the LabOne Data Server (`ziServer.exe` for HF2 or `ziDataServer.exe` for other instruments) and the LabOne Web Server (`ziWebServer.exe`) are running via the Windows Task Manager. The Data Server should be started automatically by `ziService.exe` and the Web Server should be started upon clicking "Zurich Instruments LabOne" in the Windows Start Menu. If both are running, but clicking the Start Menu does not open a new User Interface session in a new tab of your default browser then try to create a new session manually by entering `127.0.0.1:8006` in the address bar of your browser.

The user interface is slow and the web browser process consumes a lot of CPU power: make sure that the hardware acceleration is enabled for the web browser that is used for LabOne. For the Windows operating system, the hardware acceleration can be enabled in Control Panel\Display \Screen Resolution. Go to Advanced Settings and then Trouble Shoot. In case you use a NVIDIA graphics card, you have to use the NVIDIA control panel. Go to Manage 3D Settings, then Program Settings and select the program that you want to customize.

1.7.2. 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 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.

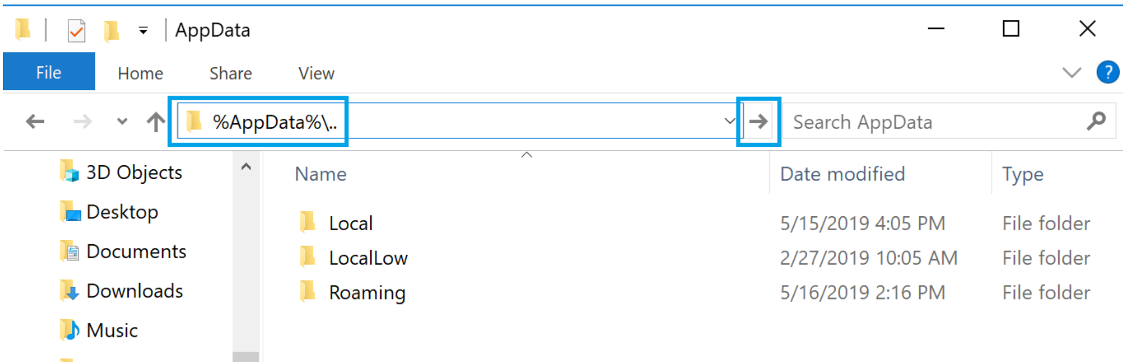


Figure 1.24. Using the `%AppData%\..` shortcut in Windows Explorer to access the hidden folder.

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.exe`):

  `/tmp/ziWebServerLog_[USER]`

Chapter 2. Functional Overview

This chapter provides the overview of the features provided by the PQSC Instrument. The first section contains the description of the graphical overview and the hardware and software feature list. The following section details the front panel and the back panel of the measurement instrument.

2.1. Features

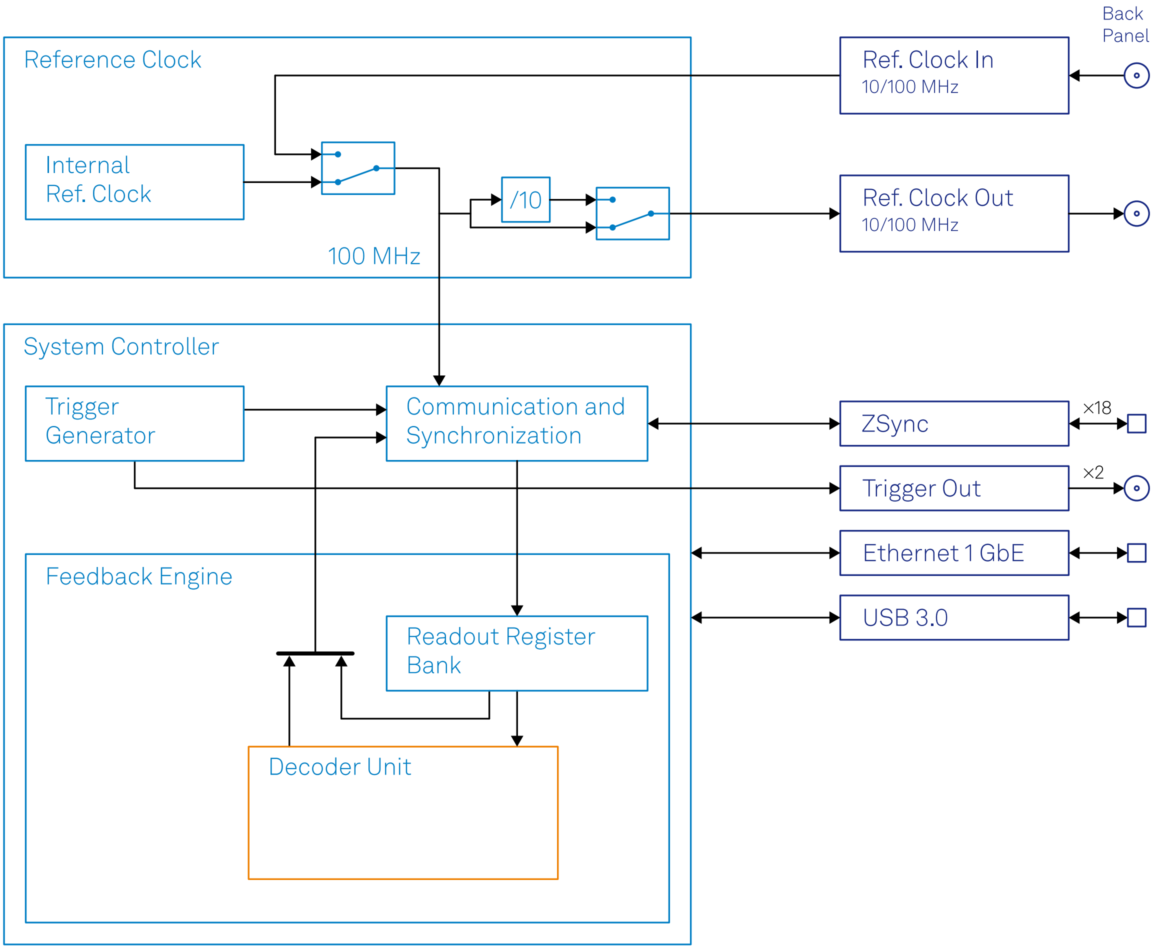


Figure 2.1. PQSC instrument functional diagram

The PQSC Instrument according to Figure 2.1 consists of several internal units (light blue color) surrounded by several interface units (dark blue color) and the back panel on the right-hand side. The orange blocks are optional units that can be either ordered at the beginning or upgraded later. The arrows between the panels and the interface units indicates selected physical connections and the data flow.

The PQSC comes with 18 ZSync ports to connect with the Zurich Instruments HDAWG for qubit control and with the Zurich Instruments UHFQA for qubit readout. This scalable architecture supports setups with more than 100 accurately synchronized AWG channels, and provides status monitoring to ensure quality and reliability of qubit tune-up routines. The ZSync links distribute the system clock to all instruments and synchronize all instruments to sub-nanosecond levels. Further, the links provide a bidirectional data interface to send qubit readout results to the PQSC for central processing, and to send trigger signals to the slave instruments to initiate synchronized actions. The ZSync links adhere to strict real-time behavior: all data transfers are predictable to single-clock-cycle precision. This enables the implementation of rapid tune-up procedures, syndrome decoding, and error correction routines. The LabOne control software provides a high-level interface to all instruments in the system and comes with APIs for Python, C, MATLAB®, LabVIEW®, and .NET.

FPGA

- Type: Xilinx® UltraScale+™ XCZU15EG-2I
- System logic cells: 747k
- CLB flip-flops: 682k
- CLB LUTs: 341k
- DSP slices: 3,528
- Block RAM: 26.2 Mb
- UltraRAM: 31.5 Mb

CPUs and memory

- Application processor: Quad ARM® Cortex TM-A53 up to 1,333 MHz
- Real-time processor: Dual ARM® Cortex TM-R5 up to 533 MHz
- SDRAM: 4 GB DDR4 with ECC

Clock

- Input frequency: Auto-detect 10 MHz / 100 MHz
- Input coupling: 50 Ω, SMA connector
- Output frequency: Switchable 10 MHz / 100 MHz
- Output amplitude: >1 Vpp in 50 Ω

Connectivity

- Host connection: LAN / Ethernet, 1 Gbit/s, USB 3.0, JTAG over USB 2.0 for Xilinx® ChipScope™ access
- device connection: 18 ZSync ports
- ZSync communication bandwidth: Down-stream 200 MB/s, Up-stream 100 MB/s
- ZSync communication latency: < 100 ns
- Trigger: 2 trigger inputs, 2 trigger outputs, 3.3 V TTL on SMA connector
- Digital I/O: 32 bits, 3.3 V TTL, general purpose

Software Features

- Web-based, high-speed user interface with multi-instrument control
- Data server with multi-client support
- API for C, LabVIEW, MATLAB, Python based instrument programming

2.2. Front Panel Tour

The front panel Control LEDs are arranged as shown in Figure 2.2 and listed in Table 2.1.

Figure 2.2. PQSC Programmable Quantum System Controller front panel

Table 2.1. PQSC Instrument front panel description

Position	Label / Name	Description
A	Power	device status LED
		blinking, blue: the instrument is ready to connect
		steady glow, blue: the instrument has an active connection over USB or Ethernet
		steady glow, yellow: the FPGA is being configured
		blinking, purple: the instrument firmware is being updated
		blinking, red: the instrument firmware is starting up
B	Sync	system synchronization LED. This status combines the status of the 18 ZSync ports on the back panel (labelled M)
		off: no instrument has been detected on any of the 18 ZSync ports on the back panel (labelled M). Ensure that the connected instrument is turned on and configured properly
		blue: link is established on any of the 18 ZSync ports on the back panel (labelled M)
		blinking: data traffic on any of the 18 ZSync ports on the back panel (labelled M)
		yellow: busy, establishing a connection on any of the 18 ZSync ports on the back panel (labelled M)
		pink: a transmitter collision occurred on any of the 18 ZSync ports on the back panel (labelled M)
		red: a cable is plugged in but the link is not established or an error was detected on any of the 18 ZSync ports on the back panel (labelled M). The error can be clock, receiver or transmitter related
		red: external reference clock is selected but the PQSC can not lock onto an external reference clock

2.3. Back Panel Tour

The back panel is the main interface for power, control, service and connectivity to other ZI instruments. Please refer to Figure 2.3 and Table 2.2 for the detailed description of the items.

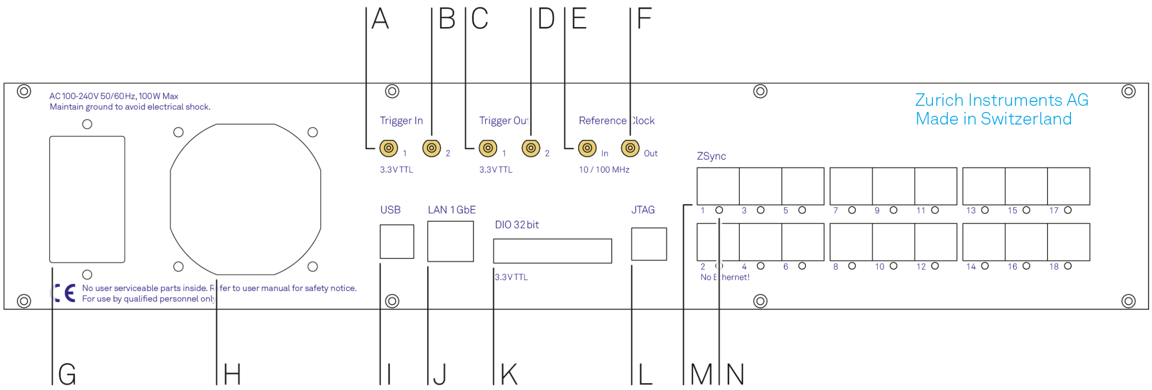


Figure 2.3. PQSC Instrument back panel

Table 2.2. PQSC Instrument back panel description

Position	Label / Name	Description
A	Trigger In 1	digital trigger input
B	Trigger In 2	digital trigger input
C	Trigger Out 1	digital trigger output
D	Trigger Out 2	digital trigger output
E	Reference Clock In	reference clock input (10 MHz / 100 MHz) for synchronization with other instruments
F	Reference Clock Out	reference clock output (10 MHz / 100 MHz) for synchronization with other instruments
G	AC 100 - 240 V	power inlet and power switch
H	-	ventilator (important: keep clear from obstruction)
I	USB	universal serial bus host computer connection
J	LAN 1 GbE	1 Gbit LAN connector for connection to the host computer
K	DIO 32bit 3.3 VTTL	32-bit digital input/output connector
L	JTAG	connector for programming and debugging the FPGA
M	ZSync	18 ZSync ports. Inter-instrument synchronization bus connector - attention: this is not an Ethernet plug, connection to an Ethernet network might damage the instrument.
N	ZSync	18 ZSync port synchronization LEDs. Shows the status of the corresponding ZSync port
		off: no instrument has been detected on the ZSync port. Ensure that the connected instrument is turned on and configured properly
		blue: link is established on the ZSync port
		blinking: data traffic on the ZSync port

Position	Label / Name	Description
		yellow: busy, establishing a connection on the ZSync port
		pink: a transmitter collision occurred on the ZSync port
		red: a cable is plugged in but the link is not established or an error was detected on the ZSync port. The error can be clock, receiver or transmitter related
		red: external reference clock is selected but the PQSC can not lock onto an external reference clock

Chapter 3. Tutorials

The tutorials in this chapter have been created to allow users to become more familiar with the operation of the Programmable Quantum System Controller. In order to successfully carry out the tutorials, it's assumed that users have certain laboratory equipment and basic equipment handling knowledge.

Note

For all tutorials, you must have LabOne installed as described in the Getting Started Chapter. If you upgraded from a previous LabOne version, please be sure that all the devices run the same version of the firmware.

3.1. Synchronization of multiple HDAWGs

Note

This tutorial is applicable to the PQSC when used with multiple HDAWG.

3.1.1. Goals and Requirements

The goal of this tutorial is to demonstrate the multi-HDAWG synchronization with the PQSC. We demonstrate how to synchronize the clock reference of multiple HDAWG and how to synchronously start them. This tutorial assumes that you are already familiar with the HDAWG, otherwise, please do the tutorial 'Basic Waveform Playback' from the HDAWG user manual first. In order to visualize the multi-channel signals, an oscilloscope with sufficient bandwidth and channel number is required.

The equipment list is given below.

- 1 PQSC
- 2 or more HDAWGs
- 1 oscilloscope (min. 2 channels, recommended 4, bandwidth 500 MHz or more)
- 1 Ethernet switch
- 1 Ethernet cable per instrument (supplied with your PQSC and HDAWGs)
- 1 ZSync cable per HDAWG (supplied with your HDAWGs)
- 2 SMA coaxial cables
- 2 adaptors BNC male to SMA female

3.1.2. Preparation

Connect the cables as illustrated below. Make sure that the instruments are powered on and connected by Ethernet to your local area network (LAN) where the host computer resides. After starting LabOne, the default web browser opens with the LabOne graphical user interface. It's advised to open a tab in the browser for each instrument.

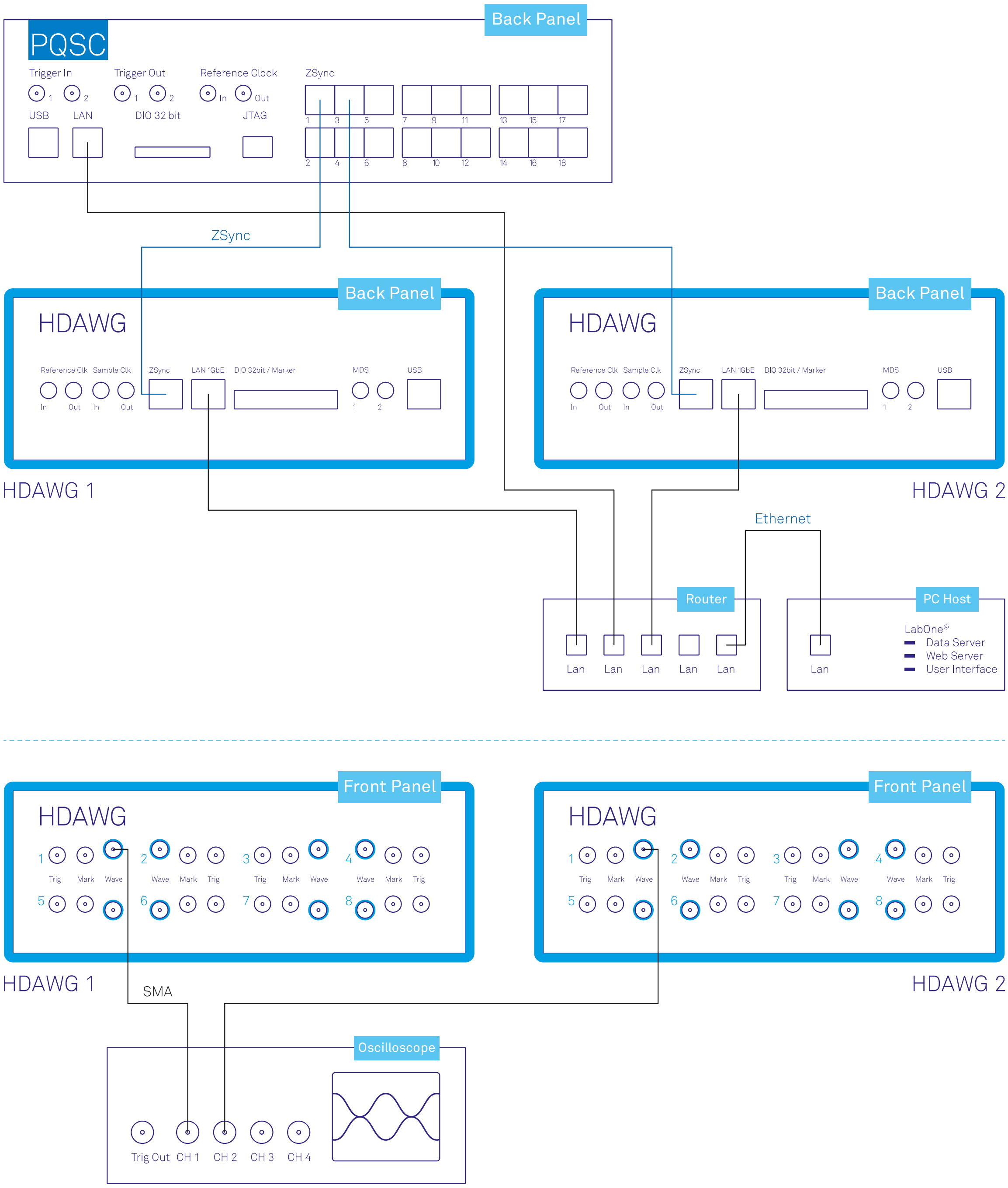


Figure 3.1. Connections for the multiple HDAWG synchronization tutorial

The tutorial can be started with the default instrument configuration (e.g. after a power cycle) and the default user interface settings (e.g. as is after pressing F5 in the browser).

3.1.3. Multi device synchronization

The first step to enable the device synchronization is to enable the ZSync clock and triggers on the HDAWGs. The following table summarizes the necessary settings. It should be repeated for each HDAWG connected to the PQSC.

Table 3.1. Settings: enable the ZSync clock on the HDAWG

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Device		Configuration		Reference clock Source	ZSync
Device		Configuration		Sample Clock Frequency (Hz)	2.4G
DIO		Digital I/O		Mode	QCCS

Figure 3.2. LabOne UI HDAWG: Device and DIO tabs

After changing the selector, the 'Status' LED will turn yellow for few seconds and then green again to signal that the HDAWG successfully locked to the reference clock provided by the PQSC over ZSync.

Then, check that the PQSC correctly recognized the HDAWGs. On the back of the instrument or in the 'Ports' tab of the PQSC verify that the status LED of the used ZSync ports turned blue and the serial number of the HDAWG is displayed. You may assign an alias to for each instrument to easily recognize it later.

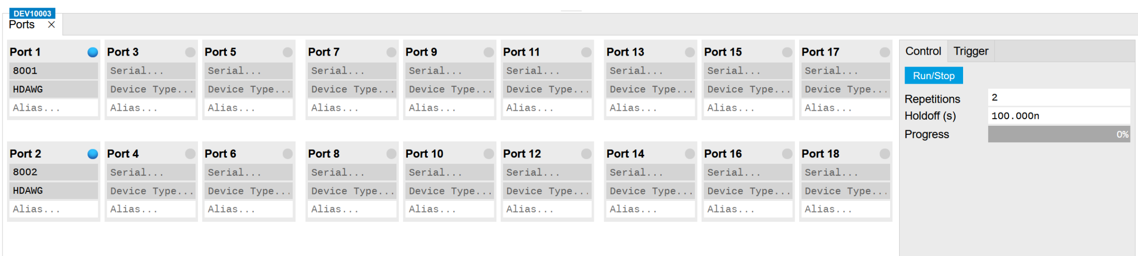


Figure 3.3. LabOne UI PQSC: Ports tab

3.1.4. Multi device synchronous triggering

To have a synchronous start of the HDAWGs, the PQSC needs to generate start triggers over ZSync and the HDAWGs have to wait for them and then start the execution. In the HDAWGs, these signals are internally routed over the DIO interface, which will be unavailable for normal usage.

We configure the sequencers to play a square waveform as soon as the trigger from the PQSC is received. The necessary settings are summarized in the following table. This must be done for each HDAWG connected to the PQSC.

Table 3.2. Settings: configure the HDAWG sequencers

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Output		Waveform Generators			4x2 channels
Output		Waveform Generators	1	Output Amplitude Wave 1	1.0
Output		Waveform Generators	1	Modulation	OFF
Output		Wave Outputs	1	Range	1V
Output		Wave Outputs	1	Enable	ON

The following sequence should be loaded in the sequencer and then started:

```
wave w = ones(64);

while(true) {
  waitZSyncTrigger();
  playWave(w);
}
```

The AWG status LED will turn yellow, meaning that is ready and waiting for the trigger.

The scope should be configured as following:

Table 3.3. Settings: configure the external scope

Scope Setting	Value / State
Ch1/Ch2 enable	ON
Ch1/Ch2 range	0.2 V/div
Timebase	20 ns/div
Trigger source	Ch1
Trigger level	200 mV
Run / Stop	ON

Finally, we configure the periodic trigger generation in the Ports tab of the PQSC and then start it by clinking on Run/Stop button.

Table 3.4. Settings: configure the periodic trigger generation on the PQSC

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Ports	Control			Repetitions	2

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Ports	Control			Holdoff (s)	100n

On the scope we can now see two pulses with both channels aligned in time. The inter-channel alignment can be further adjusted by changing the delay of each HDAWG Wave output in the field "Output > Wave Outputs > Delay (s)". The two pulses are spaced by 100 ns as specified by the Holdoff time.

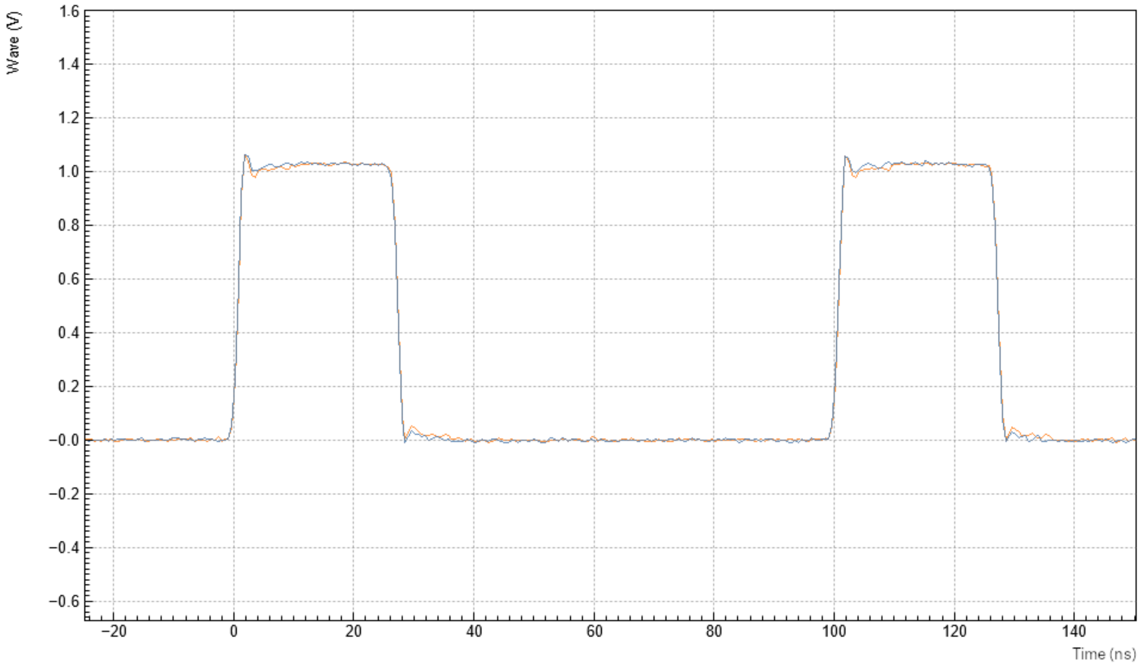


Figure 3.4. Pulses as generated by the two HDAWG and captured by the scope

3.2. Synchronization of HDAWGs and UHFQA

Note

This tutorial is applicable to the PQSC when used with multiple HDAWG and UHFQA.

3.2.1. Goals and Requirements

The goal of this tutorial is to demonstrate the multi-HDAWG and UHFQA synchronization with the PQSC for signal generation and signal acquisition. We demonstrate how to synchronize the clock reference and triggers of all the instruments, how to synchronously start the signal generation and how to align with the signal acquisition. This tutorial assumes that you are already familiar with the PQSC and the HDAWG, otherwise, please follow first the tutorial for multi-HDAWG synchronization in Section 3.1. In order to visualize the multi-channel signals, an oscilloscope with sufficient bandwidth and channel number is required.

The equipment list is given below.

- 1 PQSC
- 2 or more HDAWGs
- 1 UHFQA
- 1 External 10 MHz reference clock
- 1 oscilloscope (min. 4 channels, bandwidth 500 MHz or more)
- 1 Ethernet switch
- 1 Ethernet cable per instrument (supplied with your PQSC, HDAWGs and UHFQA)
- 1 ZSync cable per HDAWG (supplied with your HDAWGs)
- 1 DIO cable with level adapter (supplied with your UHFQA)
- 6 SMA coaxial cables
- 1 BNC coaxial cables
- 1 power splitter SMA for the reference clock
- 3 adaptors BNC male to SMA female

3.2.2. Preparation

Connect the cables as illustrated below. The cables connecting the 10 MHz reference clock to the PQSC and the UHFQA must have the same length. Make sure that the instruments are powered on and connected by Ethernet to your local area network (LAN) where the host computer resides. After starting LabOne, the default web browser opens with the LabOne graphical user interface. It’s advised to open a tab in the browser for each instrument.

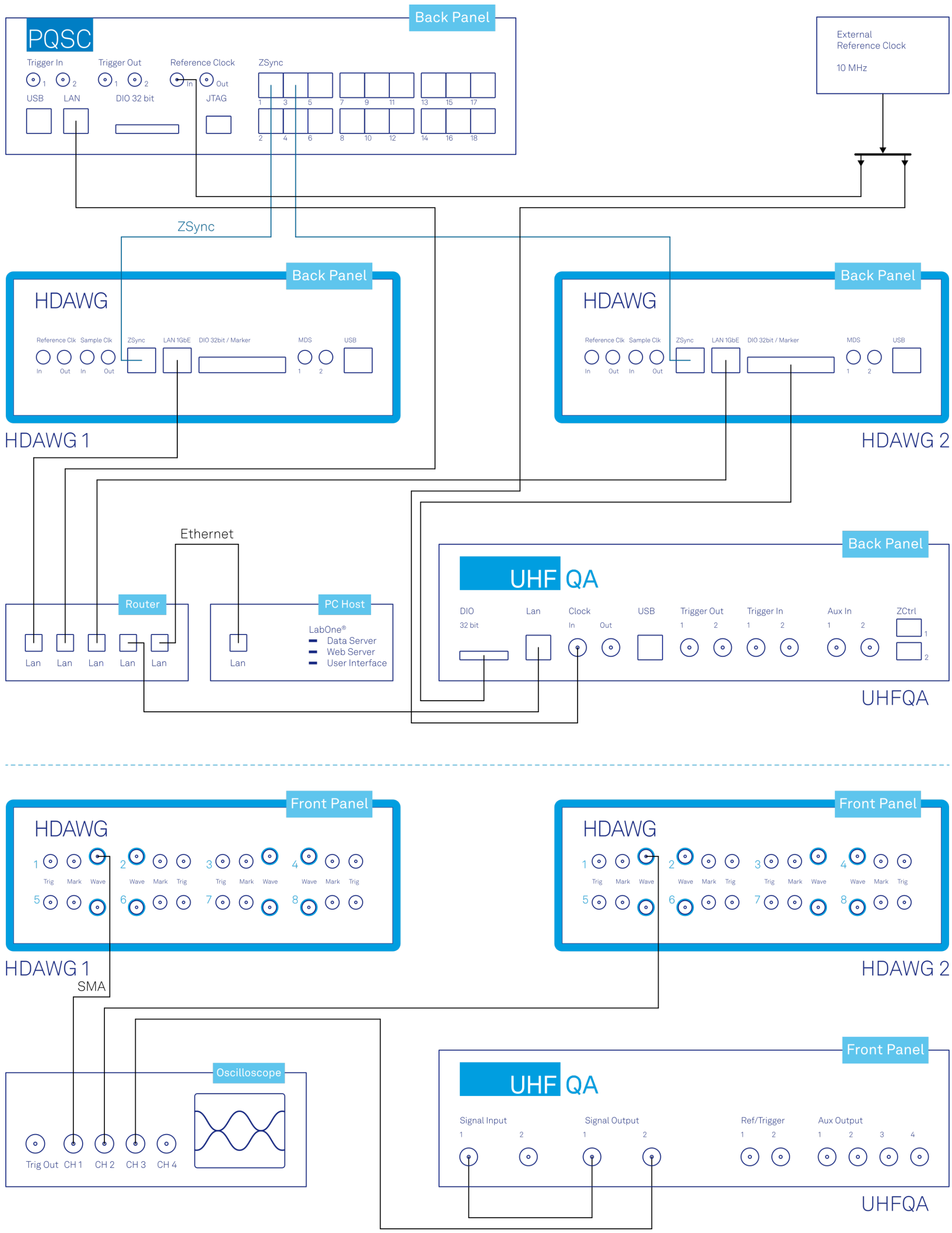


Figure 3.5. Connections for the multiple HDAWG/UHFQA synchronization tutorial

The tutorial can be started with the default instrument configuration (e.g. after a power cycle) and the default user interface settings (e.g. as is after pressing F5 in the browser).

3.2.3. Multi device synchronization

The first step to enable the device synchronization is to distribute the reference clocks to the instruments and enable the triggering. The PQSC and the UHFQA need an external 10 MHz reference clock, while the HDAWGs receive their reference clock over ZSync. It's important to first enable the external reference clock on the PQSC and then on the HDAWGs/UHFQA, since a change of the clocking in the PQSC will cause a disconnection of the devices connected over ZSync. The following tables summarizes the necessary settings for each instrument.

Table 3.5. Settings: enable the external reference clock on the PQSC

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Device		Configuration		Reference clock Input Source	External

Table 3.6. Settings: enable the ZSync clock on the HDAWGs

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Device		Configuration		Reference clock Source	ZSync
Device		Configuration		Sample Clock Frequency (Hz)	2.4G
DIO		Digital I/O		Mode	QCCS

Table 3.7. Settings: enable the external reference clock on the UHFQA

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Device		Configuration		Settings Clock Source	10MHz

On the PQSC and the HDAWGs, after changing the selector, the 'Status' LED will turn yellow for few seconds and then green again to signal that the instruments successfully locked to the reference clock.

Then, check that the PQSC correctly recognized the HDAWGs. On the back of the instrument or in the Ports tab of the PQSC verify that the status LED of the used ZSync ports turned blue and the serial number of the HDAWG is displayed. You may assign an alias to for each instrument to easily recognize it later. The UHFQA is not visible on the PQSC.

The next step is to enable the distribution of the triggers from the PQSC to the other instruments. The HDAWGs receive triggers over the ZSync port directly from the PQSC. The UHFQA receives them indirectly via the HDAWG over the DIO port. Here the HDAWG serves as a bridge to the PQSC. First we enable the interface; the following tables summarizes the necessary settings for the UHFQA and the HDAWG 2. The HDAWG 1 doesn't need any further configuration since it has no UHFQA connected and thus does not have to operate as a bridge.

Table 3.8. Settings: configure the DIO interface on the UHFQA

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DIO		Digital I/O		Mode	Manual
DIO		Digital I/O	All	Drive	OFF
DIO		Digital I/O		Clock	Internal 50 MHz

Table 3.9. Settings: configure the DIO interface on the HDAWG 2

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DIO		Digital I/O		Interface	LVCMOS

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DIO		Digital I/O	31...24	Drive	ON
DIO		Digital I/O	23...16	Drive	ON
DIO		Digital I/O	15...8	Drive	OFF
DIO		Digital I/O	7...0	Drive	OFF

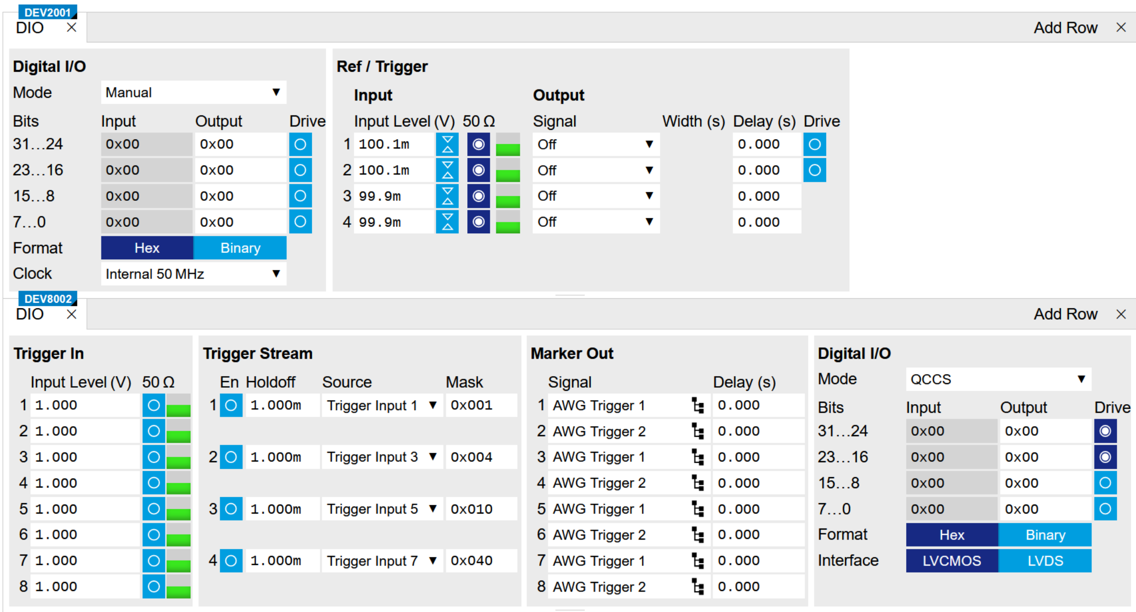


Figure 3.6. LabOne UI HDAWG and UHFQA: DIO tabs

Next, the AWG sequencers on the HDAWGs and the UHFQA need to be configured with the right trigger signal assignment. A complete description of the signals on the DIO port can be found in the HDAWG or UHFQA manuals. The following table summarizes the correct assignments for this tutorial.

Table 3.10. Signal assignment on DIO

Signal	UHFQA	HDAWG
VALID Polarity	HIGH	HIGH
VALID Index	16	0
STROBE Slope	None	None

The settings for the HDAWG can be applied via the user interface in the AWG Sequencer tab (Trigger sub-tab), but for the UHFQA these settings are currently only accessible via the API. The following Python script applies the settings from the table above to all instruments. Please refer to the LabOne Programming Manual to find user instructions for the Python API, and replace the serial numbers in `hdawg_serials` and `uhfqas_serials` in the script with the serial numbers of your instruments.

```
# Add here all the serial of your HDAWGs and UHFQA
hdawgs_serials = ['dev8001', 'dev8002']
uhfqas_serials = ['dev2001']
host = 'localhost'

import zhinst.ziPython
```

```
daq = zhinst.ziPython.ziDAQServer(host, 8004, 6)
for hdawg_serial in hdawgs_serials:
    daq.connectDevice(hdawg_serial, '1gbe')
    for i in range(4):
      daq.setInt(f'/{hdawg_serial}/awgs/{i}/dio/strobe/slope', 0)    #None
      daq.setInt(f'/{hdawg_serial}/awgs/{i}/dio/valid/polarity', 2)  #High
      daq.setInt(f'/{hdawg_serial}/awgs/{i}/dio/valid/index', 0)
for uhfqa_serial in uhfqas_serials:
    daq.connectDevice(uhfqa_serial, '1gbe')
    daq.setInt(f'/{uhfqa_serial}/awgs/0/dio/strobe/slope', 0)    #None
    daq.setInt(f'/{uhfqa_serial}/awgs/0/dio/valid/polarity', 2)  #High
    daq.setInt(f'/{uhfqa_serial}/awgs/0/dio/valid/index', 16)
```

Now all the sequencers are ready to receive triggers issued by the PQSC synchronously.

3.2.4. Multi device signal generation

We configure the AWG sequencers to play a square waveform as soon as the trigger from the PQSC is received. The necessary settings are summarized in the following tables.

Table 3.11. Settings: configure the HDAWG sequencers

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Output		Waveform Generators			4x2 channels
Output		Waveform Generators	1	Output Amplitude Wave 1	1.0
Output		Waveform Generators	1	Modulation	OFF
Output		Wave Outputs	1	Range	2V
Output		Wave Outputs	1	Enable	ON

Table 3.12. Settings: configure the UHFQA sequencer

Tab	Sub-tab	Section	#	Label	Setting / Value / State
In / Out		Signal Outputs	2	50 Ω	ON
In / Out		Signal Outputs	2	Range	750 mV
In / Out		Signal Outputs	2	On	ON
AWG	Control			Rerun	OFF
AWG	Control	Output 2		Amplitude (FS)	1.0
AWG	Control	Output 2		Mode	Plain

The following sequence programs should be loaded in the sequencers and then started. For the UHFQA:

```
const WAVE_GRANULARITY_UHFQA = 24;
wave w = ones(WAVE_GRANULARITY_UHFQA*2);

while(true) {
  waitZSyncTrigger();
  playWave(2, w);
}
```

And for the HDAWG:

```
const WAVE_GRANULARITY_HDAWG = 32;
wave w = 0.75*ones(WAVE_GRANULARITY_HDAWG*2);

while(true) {
  waitZSyncTrigger();
  playWave(w);
}
```

The AWG status LED will turn yellow, meaning that is ready and waiting for the trigger. Configure the scope as described in Section 3.1. Finally, we configure the periodic trigger generation in the Ports tab of the PQSC and then start it by clinking on the Run/Stop button.

Table 3.13. Settings: configure the periodic trigger generation on the PQSC

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Ports	Control			Repetitions	2
Ports	Control			Holdoff (s)	100n

On the scope we can now see two identical pulses with both channels aligned in time. The inter-channel alignment can be further adjusted by changing the delay of the HDAWG Wave output in the field “Output > Wave Outputs > Delay (s)”.

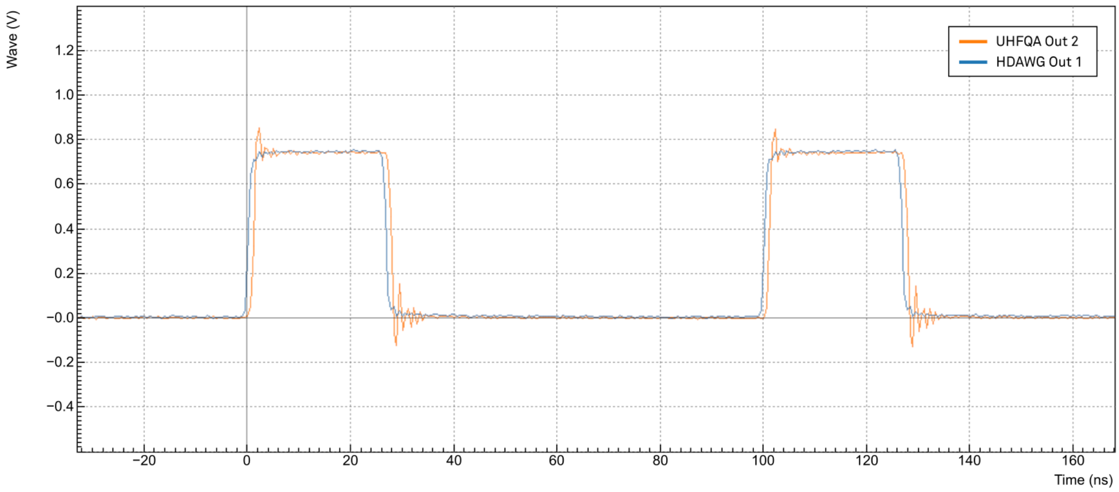


Figure 3.7. Pulses as generated by the HDAWG 2 and the UHFQA and captured by the scope

To obtain two identical pulses we had to adjust both the wave amplitude and range as well as the pulse length. The HDAWG range is the peak-to-peak voltage, while the UHFQA range is defined as the peak voltage, so there is factor of 2 to take into account. In the tutorial the waveform amplitude has been selected to be exactly 750 mV.

The sample rate of the two instruments are different, 2.4 GSa/s for the HDAWG and 1.8 GSa/s for the UHFQA. The common frequency is 600 MHz, so approximately every 1.66 ns they align. In other words, two channels align respectively every 4 samples and 3 samples, as shown in this sketch

Figure 3.8. Time alignment of the HDAWG and UHFQA

The waveform granularity is 16 samples on the HDAWG and 8 samples on the UHFQA. If we design our sequence programs such that they respect a waveform granularity of 32 samples on the HDAWG and 24 samples on the UHFQA, the output will be always aligned. In units of time, this correspond to a granularity of approximately 13.33 ns. The waveform playback instruction `playWave` should follow immediately after the instruction `waitZSyncTriggger()` to ensure the alignment. The playback should be also gapless, so it's necessary to avoid long `wait` instructions or too short waveform in complex loops.

To introduce efficient spacers, it's possible to use the `playZero` instruction. The length of the spacer pulse should respect the same granularity rules and match the length of the pulses played on the other instruments. For example, if in the previous example we want the pulses to occur one after the other, we can modify the UHFQA sequence program as follow:

```
const WAVE_GRANULARITY_UHFQA = 24;
wave w = ones(WAVE_GRANULARITY_UHFQA*2);

while(true) {
  waitZSyncTrigger();
  playZero(WAVE_GRANULARITY_UHFQA*2);
  playWave(2, w);
}
```

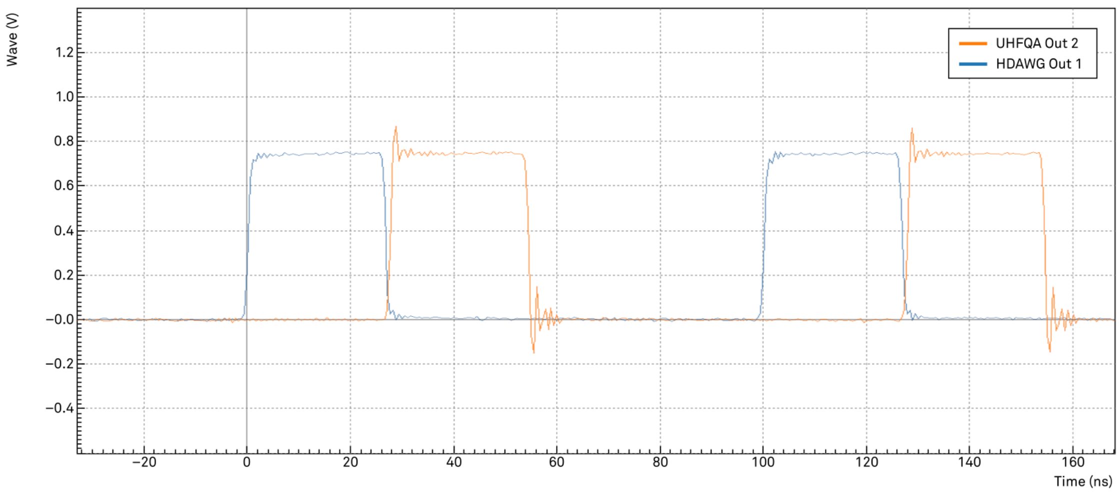


Figure 3.9. Shifted pulses with `playZero`

3.2.5. Multi-device signal generation and acquisition

The loopback connection on the UHFQA can be used to simulate the response of the device under test. In this example we use a single PQSC trigger to mark the start, the timing of the following signals will be controlled by the sequencers on the HDAWGs and the UHFQA. We simulate a simple experiment where a generic drive pulse is generated by the HDAWG. This pulse is immediately followed by a readout pulse generated by the UHFQA, which is also triggered for the readout. The control pulse is modulated at 150 MHz and the readout pulse is modulated at 50 MHz. Signal input 1 of the UHFQA is used to acquire the signal, input 2 is used to monitor it. Configure the PQSC as follow

Table 3.14. Settings: configure the start trigger generation on the PQSC

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Ports	Control			Repetitions	1

The following sequence programs should be uploaded to the AWG sequencers and then started. For the UHFQA:

```
const DEVICE_SAMPLE_RATE = 1.8e9;
const WAVE_GRANULARITY_UHFQA = 24;
const DRIVE_PULSE_LEN = WAVE_GRANULARITY_UHFQA*5;
const READOUT_LEN = WAVE_GRANULARITY_UHFQA*10;
const PAD_LEN = WAVE_GRANULARITY_UHFQA*100;
const READOUT_FREQ = 50e6;
const AVERAGES = 1024;


wave w_cosine = cosine(READOUT_LEN, 0, READOUT_FREQ*READOUT_LEN/DEVICE_SAMPLE_RATE);
wave w_sine = sine(READOUT_LEN, 0, READOUT_FREQ*READOUT_LEN/DEVICE_SAMPLE_RATE);

var i;

setTrigger(QA_INT_0);
while(true) {
  waitZSyncTrigger();
  for (i = 0; i < AVERAGES; i++) {
    playZero(DRIVE_PULSE_LEN);

    // Trigger the readout
    setTrigger(QA_INT_0 | AWG_INTEGRATION_TRIGGER | AWG_MONITOR_TRIGGER);
    setTrigger(QA_INT_0);

    playWave(1, w_cosine, 2, w_sine);
    playZero(PAD_LEN);
  }
}
```

And for the HDAWG:

```
//const DEVICE_SAMPLE_RATE = 2.4e9 already defined by the compiler
const WAVE_GRANULARITY_HDAWG = 32;
const DRIVE_PULSE_LEN = WAVE_GRANULARITY_HDAWG*5;
const READOUT_LEN = WAVE_GRANULARITY_HDAWG*10;
const PAD_LEN = WAVE_GRANULARITY_HDAWG*100;
const DRIVE_FREQ = 150e6;
const AVERAGES = 1024;

wave w_drive = cosine(DRIVE_PULSE_LEN, 0, DRIVE_FREQ*DRIVE_PULSE_LEN/
DEVICE_SAMPLE_RATE);
w_drive *= gauss(DRIVE_PULSE_LEN, DRIVE_PULSE_LEN/2, DRIVE_PULSE_LEN/6);

var i;
while(true) {
  waitZSyncTrigger();
  for (i = 0; i < AVERAGES; i++) {
    playWave(w_drive);

    // Trigger the readout (on the UHFQA)
    //setTrigger(QA_INT_0 | AWG_INTEGRATION_TRIGGER | AWG_MONITOR_TRIGGER);
    //setTrigger(QA_INT_0);

    playZero(READOUT_LEN);
    playZero(PAD_LEN);
  }
}
```

The generated signal shows two modulated pulses aligned in time. In contrast to the previous example, the two-fold repetition of the pulses is contolled by loops in the sequence programs and not by the periodic trigger generation on the PQSC.

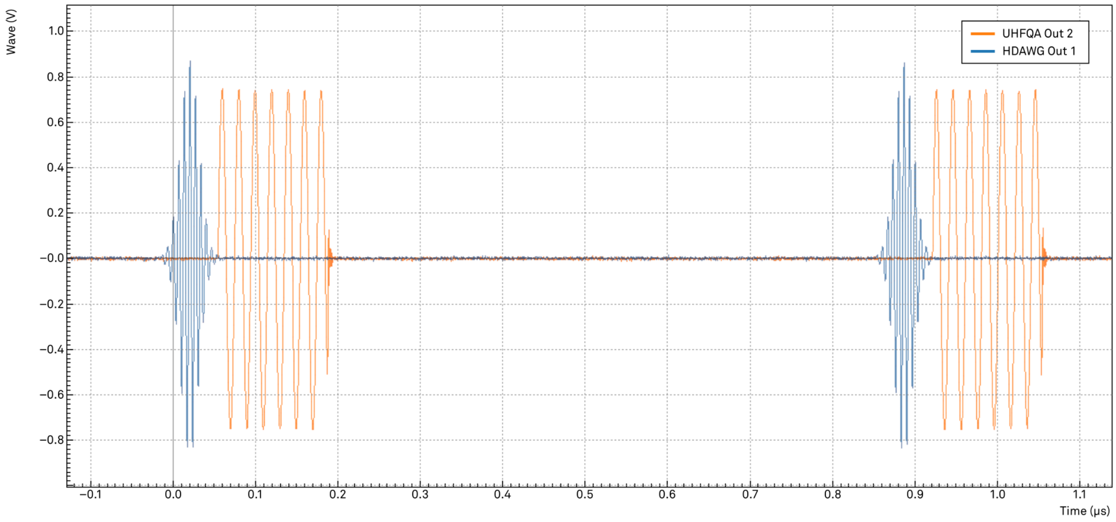


Figure 3.10. Control pulse and readout pulse generated by the HDAWG and the UHFQA

The UHFQA sequence additionally generates a trigger signal to start the QA Weighted Integration unit and the QA Input Monitor. While such trigger may look like a violation of the rule to have identical gapless sequences, it's indeed still valid. The execution time of these instructions is approximately 4 clock cycles of the sequencer, significantly less then the length of the previous waveform. The sequencer has enough time to do that while playing the first waveform, so the playback is gapless and the equal timing on the sequencers is respected. To acquire the signal on the channel 1 we have to configure the UHFQA as follows:

Table 3.15. Settings: configure the UHFQA for the readout

Tab	Sub-tab	Section	#	Label	Setting / Value / State
In / Out		Signal Outputs	1	50 Ω	ON
In / Out		Signal Outputs	1	Range	750 mV
In / Out		Signal Outputs	1	On	ON
AWG	Control	Output 1		Amplitude (FS)	1.0
AWG	Control	Output 1		Mode	Plain
QA Setup		Deskew	0,0	Matrix Representation	0.0
QA Setup		Deskew	1,0	Matrix Representation	0.0
QA Setup		Deskew	0,1	Matrix Representation	1.0
QA Setup		Deskew	1,1	Matrix Representation	1.0
QA Setup		Integration		Mode	Standard
QA Setup		Integration		Length	240
QA Setup		Integration		Trigger Signal	AWG Integration Trigger
QA Input	Control	Input Monitor		Length (Sample)	500
QA Input	Control	Input Monitor		Averages	1024
QA Input	Control	Weights / Generate		Amplitude	1

Tab	Sub-tab	Section	#	Label	Setting / Value / State
QA Input	Control	Weights / Generate		Frequency	50 M
QA Input	Control	Weights / Generate		Window Length	240
QA Input	Control	Weights / Generate		Channel	1
QA Input	Control	Weights / Generate		Set	click
QA Input	Control			Run/Stop	click

When the PQSC trigger is sent again, we can observe that the QA Input Monitor gets triggered by the sequencer. It acquires and average all the 1024 pulses. The trigger and the pulses are correctly aligned since they average out all correctly, otherwise the signal would have been distorted. The alignment can be further adjusted with the Delay setting in the QA Setup tab.

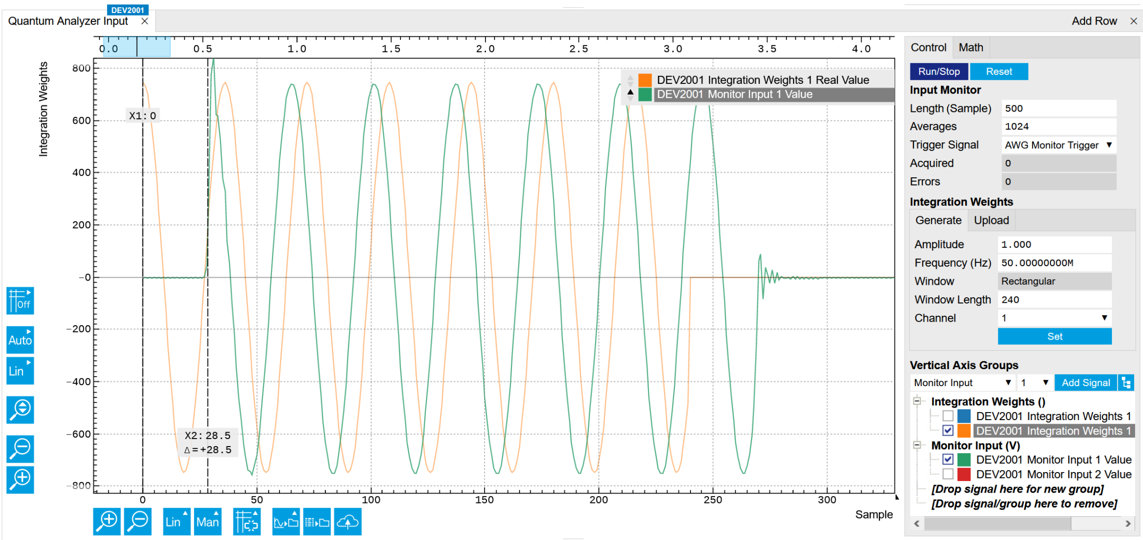


Figure 3.11. Signal as captured by the QA Monitor Input

This can be further verified with the weighted integration unit. Configure the QA Result tab as follows and start the PQSC trigger again. All 1024 results are identical up to the noise, as is evident from the scatter plot in the complex plane shown in Figure 3.12 .

Table 3.16. Settings: configure the UHFQA for the weighted integration readout

Tab	Sub-tab	Section	#	Label	Setting / Value / State
QA Result	Control	Result Wave		Source	Integration
QA Result	Control	Result Wave		Length (Sample)	1024
QA Result	Control	Result Wave		Averages	1
QA Result	Control			Run/Stop	click

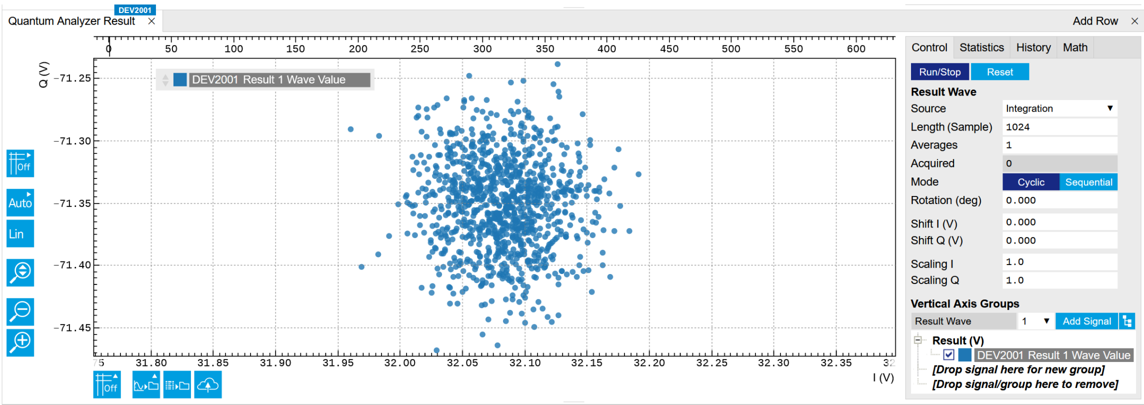


Figure 3.12. Signal as displayed in the QA Results tab

Chapter 4. Functional Description LabOne User Interface

This chapter gives a detailed description of the functionality available in the LabOne User Interface (UI) for the Zurich Instruments PQSC. LabOne provides a data server and a web server to control the Instrument with any of the most common web browsers (e.g. Firefox, Chrome, Edge, etc.). This platform-independent architecture supports interaction with the Instrument using various devices (PCs, tablets, smartphones, etc.) even at the same time if needed.

4.1. User Interface Overview

4.1.1. UI Nomenclature

This section provides an overview of the LabOne User Interface, its main elements and naming conventions. The LabOne User Interface is a browser-based UI provided as the primary interface to the PQSC instrument. Multiple browser sessions can access the instrument simultaneously and the user can have displays on multiple computer screens. Parallel to the UI the instrument can be controlled and read out by custom programs written in any of the supported languages (e.g. LabVIEW, MATLAB, Python, C) connecting through the LabOne APIs.

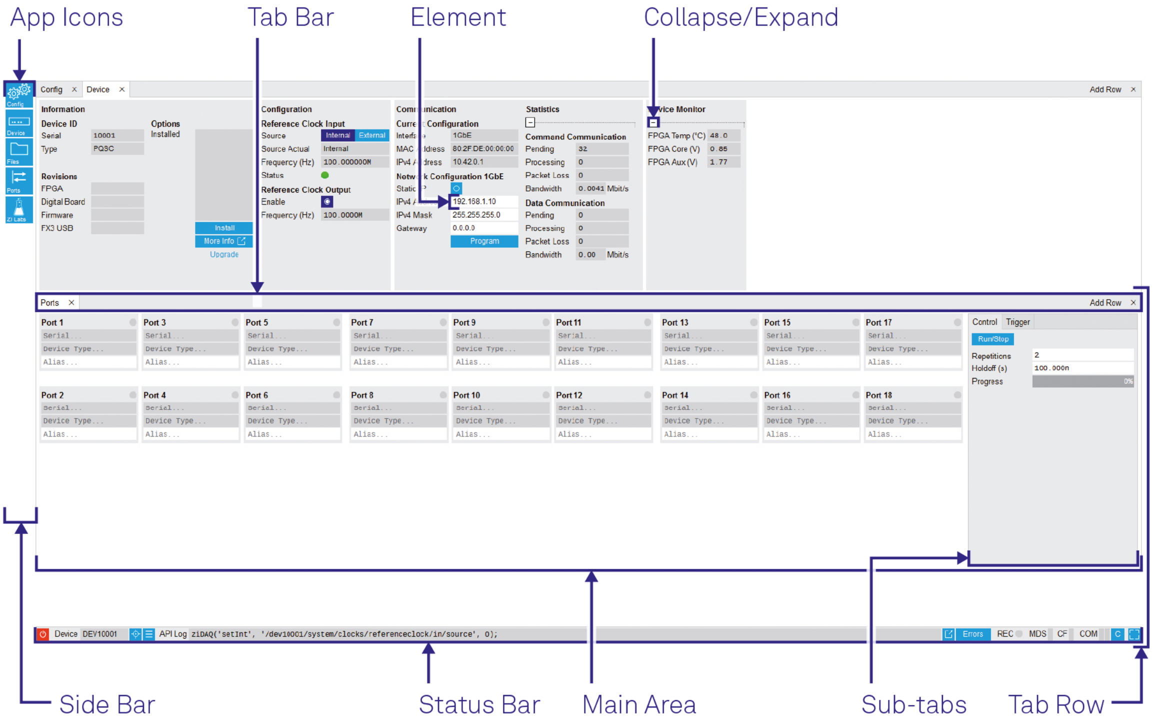


Figure 4.1. LabOne User Interface (default view)

Figure 4.1 shows the LabOne User Interface with the tabs opened by default after a new UI session has been started. The UI is by default divided into two tab rows, each containing a tab structure that gives access to the different LabOne tools. Depending on display size and application, tab rows can be freely added and deleted with the control elements on the right-hand side of each tab bar. Similarly the individual tabs can be deleted or added by selecting app icons from the side bar on the left. A click on an icon adds the corresponding tab to the display, alternatively the icon can be dragged and dropped into one of the tab rows. Moreover, tabs can be displaced by drag-and-drop within a row or across rows.

Table 4.1 gives brief descriptions and naming conventions for the most important UI items.

Table 4.1. LabOne User Interface features

Item name	Position	Description	Contains
side bar	left-hand side of the UI	contains app icons for each of the available	app icons

Item name	Position	Description	Contains
		tabs - a click on an icon adds or activates the corresponding tab in the active tab row	
status bar	bottom of the UI	contains important status indicators, warning lamps, device and session information and access to the command log	status indicators
main area	center of the UI	accommodates all active tabs – new rows can be added and removed by using the control elements in the top right corner of each tab row	tab rows, each consisting of tab bar and the active tab area
tab area	inside of each tab	provides the active part of each tab consisting of settings, controls and measurement tools	sections, plots, sub-tabs, unit selections

4.1.2. Unique Set of Analysis Tools

All instruments feature a comprehensive tool set for connecting and synchronizing multiple instruments.

The following table gives the overview of all app icons. Note that the selection of app icons depends on the upgrade options installed on a given instrument.

Table 4.2. Overview of app icons and short description

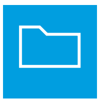
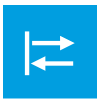
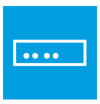
Control/Tool	Option/Range	Description
Files		Access settings and measurement data files on the host computer.
Config		Provides access to software configuration.
Ports		Control and status display of the 18 ZSync ports.
Device		Provides instrument specific settings.
ZI Labs		Experimental settings and controls.

Table 4.3 gives a quick overview over the different status bar elements along with a short description.

Table 4.3. Status bar description

Control/Tool	Option/Range	Description
Command log	last command	Shows the last command. A different formatting (Matlab, Python, ..) can be set in the config tab. The log is also saved in [User]\Documents \Zurich Instruments\LabOne \WebServer\Log
Show Log		Show the command log history in a separate browser window.
Errors	Errors	Display system errors in separate browser tab.
Device	devXXX	Indicates the device serial number.
Identify Device		When active, device LED blinks
MDS	grey/green/red/yellow	Multiple device synchronization indicator. Grey: Nothing to synchronize - single device on the UI. Green: All devices on the UI are correctly synchronized. Yellow: MDS sync in progress or only a subset of the connected devices is synchronized. Red: Devices not synchronized or error during MDS sync.
REC	grey/red	A blinking red indicator shows ongoing data recording (related to global recording settings in the Config tab).
CF	grey/yellow/red	Clock Failure - Red: present malfunction of the external 10 MHz reference oscillator. Yellow: indicates a malfunction occurred in the past.
COM	grey/yellow/red	Packet Loss - Red: present loss of data between the device and the host PC. Yellow: indicates a loss occurred in the past.
COM	grey/yellow/red	Sample Loss - Red: present loss of sample data between the device and the host PC. Yellow: indicates a loss occurred in the past.

Control/Tool	Option/Range	Description
C		Reset status flags: Clear the current state of the status flags
Full Screen		Toggles the browser between full screen and normal mode.

4.2. Feedback Data Processing

The PQSC provides a way to process readout result feedback actions in real time with minimal latency. Readout result are generated by the Quantum Analyzers, like the UHFQA, while the feedback data are received by Signal Generators, like the HDAWG. All the communications are done over ZSync.

The PQSC feedback architecture according to Figure 4.2 processes the incoming data in several stages. First, the incoming readout results are stored in the Readout Register Bank. The storage address is provided dynamically by the Quantum Analyzer. Then, a subset of the Readout Register Bank content can be forwarded directly to the signal generators or fed to a decoder for further processing.

Note

Only the ports 1-8 can be connected to a Quantum Analyzer; ports 9-18 are not enabled to receive data. All the ports are enabled to send data.

Note

The configuration of the feedback system is currently only accessible through the API.

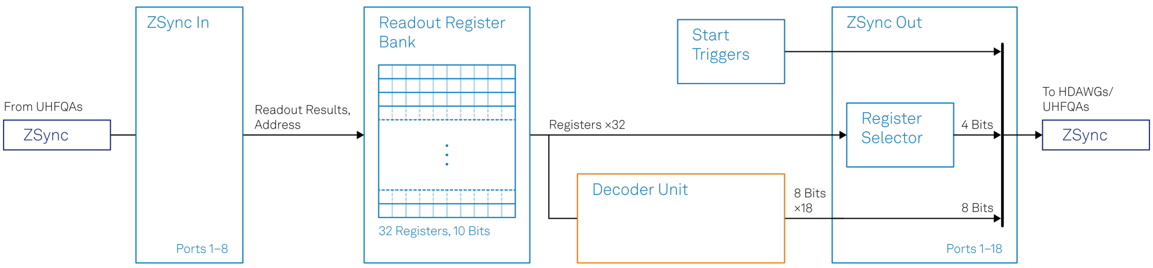


Figure 4.2. Block diagram of the PQSC feedback architecture

Readout Register Bank

The readout register bank is used to store multiple readout results measured by the Quantum Analyzers. The purpose of this memory is to provide a global point of access to all the last readouts done on multiple qubits. The register bank can be updated partially if not all the possible readouts are performed in a single cycle, for example if they are staggered. It's possible to store multiple repeated measurements by using a different register for each readout cycle. The readout register bank is not intended to acquire the measurements for offline analysis and storage, but only for processing and feedback by the PQSC itself and the other connected instruments.

The Readout Register bank consists of 32 readout registers, of which each can store up to 10 qubit readout results from one device. Every Quantum Analyzer connected to the PQSC writes into the readout register bank when it performs a readout. Only the qubits actually read are written into a register, the other bits are left untouched. The register address is specified in the sequencer by the Quantum Analyzer before the readout is started. Thus it can be changed between different readout events. From LabOne, the user can only clear all the registers before an experiment, the stored values are not accessible.

Register Forwarding

Readout register forwarding can be used to send a subset of the readout register bank content to the signal generators without further processing. Each ZSync output port can forward up to four freely chosen readout results, corresponding to four qubit readouts. Every port can be configured with his own set of forwarded results. The forwarded results are specified by the register number and index of the desired bit inside of the register. Whenever one of the four linked registers is updated, its content is automatically forwarded and a trigger for the receiving instrument is generated. This functionality is intended for feedback that require minimal latency and depend on the state of very few qubits. The typical example is the active qubit reset, when a qubit is brought to its ground state if it's in one excited state. The forwarded results are available in the bits 8-11 of the message sent over ZSync. That may change in future versions of LabOne.

4.2.1. Decoders

Quantum error correction codes require to perform a syndrome measurement to evaluate the eventual errors that corrupt the state of a set of entangled qubits. Such measurement is performed by reading a subset of the entangled qubits and from that deduce how to correct the error. This evaluation is performed by the Decoder unit. This unit has access to the whole readout register bank. It can generate an output as wide as one byte per port. The output of the decoder is available in the bits 0-7 of the message sent over ZSync. That may change in future versions of LabOne. The decoder is implemented as a lookup table decoder.

Lookup table decoder

The Lookup table decoder implement the error decoding function using lookup tables. In practice it's a map of every possible input, also called address, to a list of outputs. It can be evaluated in a short and constant time. The Figure 4.3 shows the data flow. The Source register selector is used to reduce the large set of the readout registers to a word of 16 bits that serves as the input table address. Similarly to the the register forwarding feature, the user can select 16 sources register numbers and indices of the desired bit inside of each register. Then this address is used to access the 4 lookup tables. The tables are programmed with an array of 2^{16} bytes. The input address will be used as index of the array. Each table will output 1 byte. Finally, for each ZSync output port, one of this 4 outputs is selected and forwarded to the receiving instrument. Whenever any of the readout results is updated, the lookup tables are evaluated and the outputs are sent together with a trigger event.

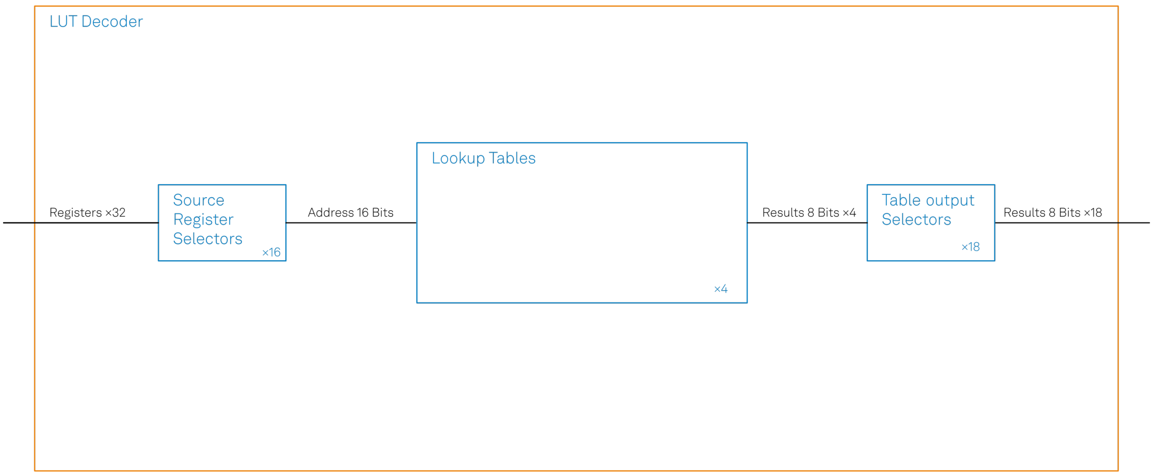


Figure 4.3. Block diagram of the Lookup Table decoder

4.3. Ports Tab

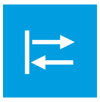
The Ports tab provides control and displays the status of the 18 ZSync ports. It is available on all PQSC instruments.

4.3.1. Features

- Display the state of the complete system at first glance
- Display port synchronization of all 18 ZSync ports
- Provide user-given names to connected instruments
- Control of triggers that are sent to the connected instruments

4.3.2. Description

Table 4.4. App icon and short description

Control/Tool	Option/Range	Description
Ports		Control and status display of the 18 ZSync ports.

The Ports tab (see Figure 4.4) is divided into two sections: The ZSync ports and two sub-tabs for Control and Trigger.

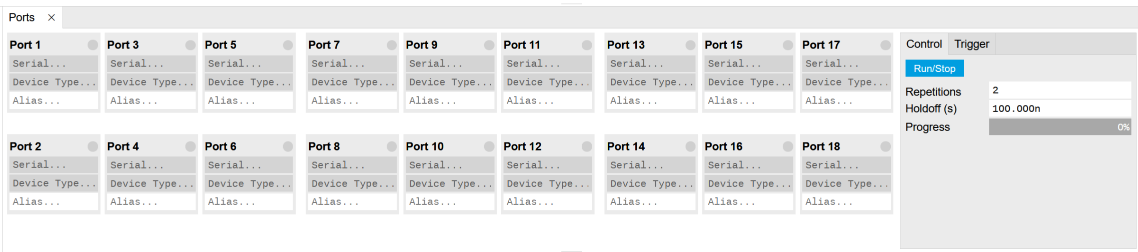


Figure 4.4. LabOne UI: Ports tab

The PQSC is always used in conjunction with other Zurich Instruments devices in a larger system, e.g. a Quantum Computing Control System (QCCS). The purpose of the Ports tab is to allow the user to understand the current state of the complete system at first glance. The main elements are the 18 ports, where the user can see information about the status and health of each connection, can provide a name to the connected instrument and can reset the connection. Furthermore the user can start the sending of the triggers and control the repetitions and the holdoff of the triggers.

Note: When using the PQSC together with HDAWGs, the user has to set up the connected HDAWG correctly to use the trigger information that it receives from the PQSC. Please refer to the multi-HDAWG synchronization tutorial in Section 3.1 for more details.

4.3.3. Functional Elements

Table 4.5. Ports tab

Control/Tool	Option/Range	Description
Connection Status	green/blue/yellow/red	Indicates the availability of the instrument connected to

Control/Tool	Option/Range	Description
		the port. Off: no Instrument detected. Yellow: connection to an instrument is in progress. Blue: connection to an instrument is ready or data is being sent to / received from an instrument. Red: an error has occurred on the connection to an instrument.
Serial		The device ID of the instrument connected to this port.
Device Type		The device type of the instrument connected to this port.
Alias		User-given name to the instrument connected to this port.
Run/Stop	Run/Stop	Starts sending triggers to all connected instruments over ZSync ports.
Repetitions		Sets the number of triggers sent over ZSync ports.
Holdoff	time in seconds	Sets the time between repeated triggers sent over ZSync ports.
Progress	0% to 100%	The percentage of repeated triggers sent over ZSync ports.
Enable	ON / OFF	Enable Trigger Out connector.
Source		Select the source for the Trigger Out connector.
	Start Trig	Generate a trigger when a trigger is sent over the selected ZSync port.
	Feedback	Generate a trigger when the PQSC sends feedback on the ZSync port.
Port		Select the ZSync port associated with the Trigger Out source.
Pulse Width	Time in seconds	Defines the minimal pulse width of the Trigger Out.

4.4. Config Tab

The Config tab provides access to all major LabOne settings and is available on all PQSC instruments.

4.4.1. Features

- define instrument connection parameters
- browser session control
- define UI appearance (grids, theme, etc.)
- store and load instrument settings and UI settings
- configure data recording

4.4.2. Description

The Config tab serves as a control panel for all general LabOne settings and is opened by default on start-up. Whenever the tab is closed or an additional one of the same type is needed, clicking the following icon will open a new instance of the tab.

Table 4.6. App icon and short description

Control/Tool	Option/Range	Description
Config		Provides access to software configuration.

The Config tab (see Figure 4.5) is divided into four sections to control connections, sessions, user interface appearance and data recording.

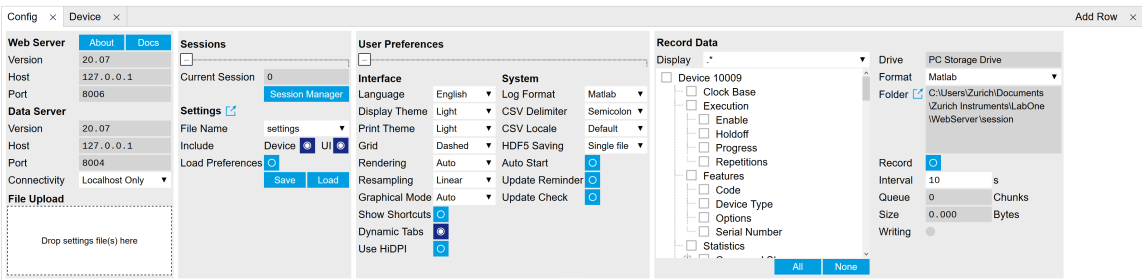


Figure 4.5. LabOne UI: Config tab

The Connection section provides information about connection and server versions. Access from remote locations can be restricted with the connectivity setting.

The Session section provides the session number which is also displayed in status bar. Clicking on Session Dialog opens the session dialog window (same as start up screen) that allows one to load different settings files as well as to connect to other instruments.

The Settings section allows one to load and save instrument and UI settings. The saved settings are later available in the session dialogue.

The User Preferences section contains the settings that are continuously stored and automatically reloaded the next time an PQSC instrument is used from the same computer account. For low ambient light conditions the use of the dark display theme is recommended (see Figure 4.6).

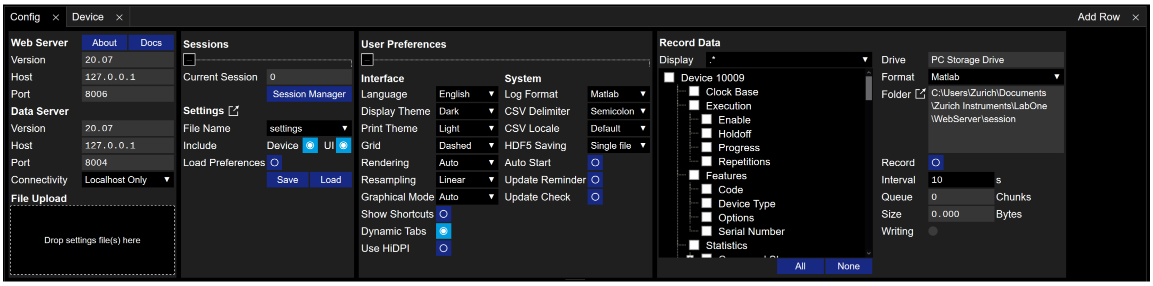


Figure 4.6. LabOne UI: Config tab - dark theme

4.4.3. Functional Elements

Table 4.7. Config tab

Control/Tool	Option/Range	Description
About		Get information about LabOne software.
Web Server Version and Revision	string	Web Server version and revision number
Host	default is localhost: 127.0.0.1	IP-Address of the LabOne Web Server
Port	4 digit integer	LabOne Web Server TCP/IP port
Data Server Version and Revision	string	Data Server version and revision number
Host	default is localhost: 127.0.0.1	IP-Address of the LabOne Data Server
Port	default is 8004	TCP/IP port used to connect to the LabOne Data Server.
Connect/Disconnect		Connect/disconnect the LabOne Data Server of the currently selected device. If a LabOne Data Server is connected only devices that are visible to that specific server are shown in the device list.
Status	grey/green	Indicates whether the LabOne User Interface is connected to the selected LabOne data server. Grey: no connection. Green: connected.
Connectivity	Localhost Only From Everywhere	Forbid/Allow to connect to this Data Server from other computers.
File Upload	drop area	Drag and drop files in this box to upload files. Clicking on the

Control/Tool	Option/Range	Description
		box opens a file dialog for file upload. Supported files: Settings (*.xml).
Session Id	integer number	Session identifier. A session is a connection between a client and LabOne Data Server.
Session Manager	Session Manager	Open the session manager dialog. This allows for device or session change. The current session can be continued by pressing cancel.
File Name	selection of available file names	Save/load the device and user interface settings to/from the selected file on the internal flash drive. The setting files can be downloaded/uploaded using the Files tab.
Include		Enable Save/Load of particular settings.
	No Include Settings	Please enable settings type to be included during Save/Load.
	Include Device	Enable Save/Load of Device settings.
	Include UI	Enable Save/Load of User Interface settings.
	Include UI and Device	Enable Save/Load of User Interface and Device settings.
	Include Preferences	Enable loading of User Preferences from settings file.
	Include UI, Device and Preferences	Enable Save/Load of User Interface, Device and User Preferences.
Save	Save	Save the user interface and device setting to a file.
Load	Load	Load the user interface and device setting from a file.
Language		Choose the language for the tooltips.
Display Theme	Light	Choose theme of the user interface.
	Dark	
Plot Print Theme	Light	Choose theme for printing SVG plots.
	Dark	
Plot Grid	Dashed	Select active grid setting for all SVG plots.
	Solid	
	None	

Control/Tool	Option/Range	Description
Plot Rendering		Select rendering hint about what tradeoffs to make as the browser renders SVG plots. The setting has impact on rendering speed and plot display for both displayed and saved plots.
	Auto	Indicates that the browser shall make appropriate tradeoffs to balance speed, crisp edges and geometric precision, but with geometric precision given more importance than speed and crisp edges.
	Optimize Speed	The browser shall emphasize rendering speed over geometric precision and crisp edges. This option will sometimes cause the browser to turn off shape anti-aliasing.
	Crisp Edges	Indicates that the browser shall attempt to emphasize the contrast between clean edges of artwork over rendering speed and geometric precision. To achieve crisp edges, the user agent might turn off anti-aliasing for all lines and curves or possibly just for straight lines which are close to vertical or horizontal.
	Geometric Precision	Indicates that the browser shall emphasize geometric precision over speed and crisp edges.
Resampling Method		Select the resampling interpolation method. Resampling corrects for sample misalignment in subsequent scope shots. This is important when using reduced sample rates with a time resolution below that of the trigger.
	Linear	Linear interpolation
	PCHIP	Piecewise Cubic Hermite Interpolating Polynomial
Show Shortcuts	ON / OFF	Displays a list of keyboard and mouse wheel shortcuts for manipulating plots.

Control/Tool	Option/Range	Description
Dynamic Tabs	ON / OFF	If enabled, sections inside the application tabs are collapsed automatically depending on the window width.
Graphical Mode	Auto	Select the display mode for the graphical elements. Auto format will select the format which fits best the current window width.
	Expanded	
	Collapsed	
Log Format	Telnet	Choose the command log format. See status bar and [User]\Documents \Zurich Instruments\LabOne \WebServer\Log
	Matlab	
	Python	
	.NET	
CSV Delimiter	Comma	Select which delimiter to insert for CSV files.
	Semicolon	
	Tab	
CSV Locale	Default locale. Dot for the decimal point and no digit grouping, e.g. 1005.07	Select the locale used for defining the decimal point and digit grouping symbols in numeric values in CSV files. The default locale uses dot for the decimal point and no digit grouping, e.g. 1005.07. The system locale uses the symbols set in the language and region settings of the computer.
	System locale. Use the symbols set in the language and region settings of the computer	
HDF5 Saving	Single file. All measurements go in one file	For HDF5 file format only: Select whether each measurement should be stored in a separate file, or whether all measurements should be saved in a single file.
	Multiple files. Each measurement goes in a separate file	
Auto Start	ON / OFF	Skip session manager dialog at start-up if selected device is available. In case of an error or disconnected device the session manager will be reactivated.
Update Reminder	ON / OFF	Display a reminder on start-up if the LabOne software wasn't updated in 180 days.
Update Check	ON / OFF	Periodically check for new LabOne software over the internet.
Drive		Select the drive for data saving.

Control/Tool	Option/Range	Description
Format	Matlab	File format of recorded and saved data.
	CSV	
	SXM (Nanonis)	
Open Folder		Open recorded data in the system File Explorer.
Folder	path indicating file location	Folder containing the recorded data.
Save Interval	Time in seconds	Time between saves to disk. A shorter interval means less system memory consumption, but for certain file formats (e.g. MATLAB) many small files on disk. A longer interval means more system memory consumption, but for certain file formats (e.g. MATLAB) fewer, larger files on disk.
Queue	integer number	Number of data chunks not yet written to disk.
Size	integer number	Accumulated size of saved data in the current session.
Record	ON / OFF	Start and stop saving data to disk as defined in the selection filter. Length of the files is determined by the Window Length setting in the Plotter tab.
Writing	grey/green	Indicates whether data is currently written to disk.
Display	filter or regular expression	Display specific tree branches using one of the preset view filters or a custom regular expression.
Tree	ON / OFF	Click on a tree node to activate it.
All		Select all tree elements.
None		Deselect all tree elements.

4.5. Device Tab

The Device tab is the main settings tab for the connected instrument and is available on all PQSC instruments.

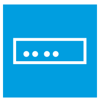
4.5.1. Features

- Option and upgrade management
- External clock referencing (10/100 MHz)
- Instrument connectivity parameters
- Device monitor

4.5.2. Description

The Device tab serves mainly as a control panel for all settings specific to the instrument that is controlled by LabOne in this particular session. Whenever the tab is closed or an additional one of the same type is needed, clicking the following icon will open a new instance of the tab.

Table 4.8. App icon and short description

Control/Tool	Option/Range	Description
Device		Provides instrument specific settings.

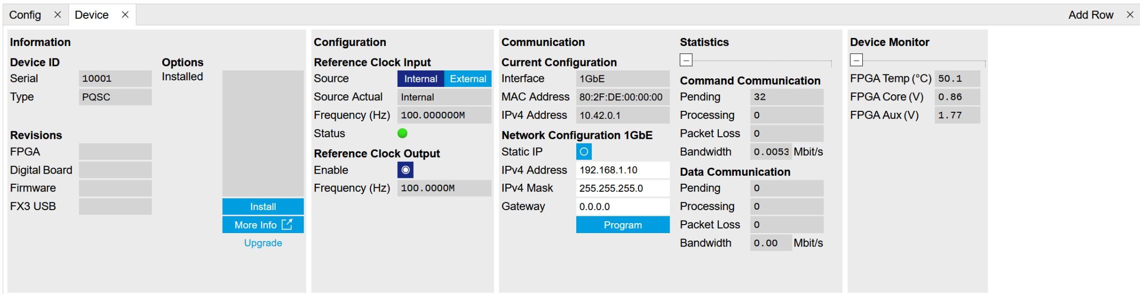


Figure 4.7. LabOne UI: Device tab

The Information section provides details about the instrument hardware and indicates the installed upgrade options. This is also the place where new options can be added by entering the provided option key.

The **Configuration** section allows one to change the reference from the internal clock to an external 10/100 MHz reference. The reference is to be connected to the Clock Input on the instrument back panel. The section also allows one to select a frequency of 10 or 100 MHz of the reference clock output, which is generated at the Clock Output on the instrument back panel

Note

Any change to the reference clock setting, either input and output, will disconnect all the devices connected over ZSync. The connections will not be automatically re-established and that should be done manually on every instrument.

The **Communication** section offers access to the instruments TCP/IP settings.

The **Statistics** section gives an overview on communication statistics.

Note

Packet loss on command streaming over TCP or USB: command packets should never be lost as it creates an invalid state.

The **Device Monitor** section is collapsed by default and generally only needed for servicing. It displays vitality signals of some of the instrument's hardware components.

4.5.3. Functional Elements

Table 4.9. Device tab

Control/Tool	Option/Range	Description
Serial	1-4 digit number	Device serial number
Type	string	Device type
FPGA	integer number	HDL firmware revision.
Digital Board	version number	Hardware revision of the FPGA base board.
Firmware	integer number	Revision of the device internal controller software.
FX3 USB	version number	USB firmware revision.
Installed Options	short names for each option	Options that are installed on this device.
Install	Install	Click to install options on this device. Requires a unique feature code and a power cycle after entry.
More Information		Display additional device information in a separate browser tab.
Upgrade Device Options		Display available upgrade options.
Input Reference Clock Source		Selects internal or external reference clock source. When the source is changed, all the instruments connected with ZSync links will be disconnected.
	Internal	The internal 100MHz clock is used as the frequency and time base reference.
	External	An external clock is intended to be used as the frequency and time base reference. Provide a clean and stable 10MHz or 100MHz reference

Control/Tool	Option/Range	Description
		to the appropriate back panel connector.
Actual Input Reference Clock Source		Currently active internal or external reference clock source.
	Internal	Internal 100MHz clock is actually used as the frequency and time base reference.
	External	An external clock is actually used as the frequency and time base reference.
Input Reference Clock Frequency		Indicates the frequency of the input reference clock.
Input Reference Clock Status		Indicates the status of the input reference clock. Green: locked. Yellow: the device is busy trying to lock onto the input reference clock signal. Red: there was an error locking onto the input reference clock signal. The instrument is currently not operational.
Output Reference Clock Enable		Enable clock signal on the reference clock output.
Output Reference Clock Frequency		Selects the frequency of the output reference clock to be 10MHz or 100MHz.
Interface		Active interface between device and data server. In case multiple options are available, the priority as indicated on the left applies.
MAC Address	80:2F:DE:xx:xx:xx	MAC address of the device. The MAC address is defined statically, cannot be changed and is unique for each device.
IPv4 Address	default 192.168.1.10	Current IP address of the device. This IP address is assigned dynamically by a DHCP server, defined statically, or is a fall-back IP address if the DHCP server could not be found (for point to point connections).
Static IP	ON / OFF	Enable this flag if the device is used in a network with fixed IP assignment without a DHCP server.
IPv4 Address	default 192.168.1.10	Static IP address to be written to the device.

Control/Tool	Option/Range	Description
IPv4 Mask	default 255.255.255.0	Static IP mask to be written to the device.
Gateway	default 192.168.1.1	Static IP gateway
Program	Program	Click to program the specified IPv4 address, IPv4 Mask and Gateway to the device.
Pending	integer value	Number of buffers ready for receiving command packets from the device.
Processing	integer value	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.
Packet Loss	integer value	Number of command packets lost since device start. Command packets contain device settings that are sent to and received from the device.
Bandwidth	numeric value	Command streaming bandwidth usage on the physical network connection between device and data server.
Pending	integer value	Number of buffers ready for receiving data packets from the device.
Processing	integer value	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.
Packet Loss	integer value	Number of data packets lost since device start. Data packets contain measurement data.
Bandwidth	numeric value	Data streaming bandwidth usage on the physical network connection between device and data server.

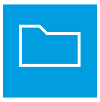
4.6. File Manager Tab

4.6.1. Features

- File preview for settings files and log files

4.6.2. Description

Table 4.10. App icon and short description

Control/Tool	Option/Range	Description
Files		Access settings and measurement data files on the host computer.

The Files tab (see Figure 4.8) provides three windows for exploring. The left window allows one to browse through the directory structure, the center window shows the files of the folder selected in the left window, and the right window displays the content of the file selected in the center window, e.g. a settings file or log file.

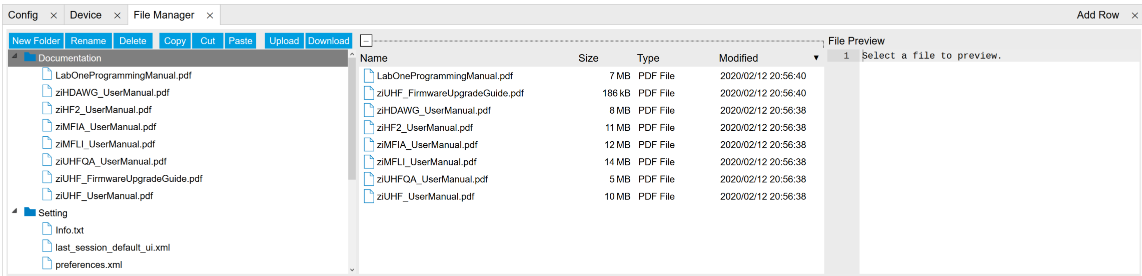


Figure 4.8. LabOne UI: File Manager tab

4.6.3. Functional Elements

Table 4.11. File tab

Control/Tool	Option/Range	Description
New Folder	New Folder	Create new folder at current location.
Rename	Rename	Rename selected file or folder.
Delete	Delete	Delete selected file(s) and/or folder(s).
Copy	Copy	Copy selected file(s) and/or folder(s) to Clipboard.
Cut	Cut	Cut selected file(s) and/or folder(s) to Clipboard.
Paste	Paste	Paste file(s) and/or folder(s) from Clipboard to the selected directory.

Control/Tool	Option/Range	Description
Upload	Upload	Upload file(s) and/or folder(s) to the selected directory.
Download	Download	Download selected file(s) and/or folder(s).

4.7. ZI Labs Tab

The ZI Labs tab contains experimental LabOne functionalities added by the ZI development team. The settings found here are often relevant to special applications, but have not yet found their definitive place in one of the other LabOne tabs. Naturally this tab is subject to frequent changes, and the documentation of the individual features would go beyond the scope of this user manual. Clicking the following icon will open a new instance of the tab.

Table 4.12. App Icon and short description

Control/Tool	Option/Range	Description
ZI Labs		Experimental settings and controls.

Chapter 5. Specifications

Important

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

Important

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

5.1. General Specifications

Table 5.1. General and storage

Parameter	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	-	-	100 W
operating environment	IEC61010, indoor location, installation category II, pollution degree 2		
operating altitude	up to 2000 meters		
power supply AC line	100-240 V (±10%), 50/60 Hz		
dimensions with handles and feet	45.0 × 34.5 × 10.0 cm, 17.7 × 13.6 × 3.9 inch, 19 inch rack compatible		
weight	6.0 kg		
recommended calibration interval	2 years		

Table 5.2. Maximum ratings

Parameter	min	typ	max
damage threshold Trigger Out 1 and 2	−0.7 V	–	+4 V
damage threshold Trigger In 1 and 2	−0.7 V	–	+4 V
damage threshold Reference Clock Out (DC)	-4 V	–	+4 V
damage threshold Reference Clock In (AC, with DC offset 0V)	–	–	+13.5 dBm
damage threshold Reference Clock In (DC)	-4 V	–	+4 V
DIO In / Out in default configuration 3.3 V CMOS/TTL	-0.7 V	–	+4 V

Table 5.3. Host computer requirements

Parameter	Description
supported Windows operating systems	32-bit and 64-bit versions of Windows 10, 8.1, 7, XP[1]
supported Linux distributions	Ubuntu 16.04 (AMD64), 14.04 LTS (AMD64, i386)
minimum host computer requirements	Windows XP 32-bit Dual Core CPU with SSE2 support 4 GB DRAM

Parameter	Description
	1 Gbit/s Ethernet controller
recommended host computer requirements	Windows 10 64-bit or Linux 64-bit Quad Core CPU (i7) or better 8 GB DRAM or better 1 Gbit/s Ethernet controller SSD HD drive (for high-bandwidth data saving) USB 3.0 connection
supported processors (requiring SSE2)	AMD K8 (Athlon 64, Sempron 64, Turion 64, etc.), AMD Phenom, Intel Pentium 4, Xeon Celeron, Celeron D, Pentium M, Celeron M, Core, Core 2, Core i5, Core i7, Atom

[1]Software version is available for download but not officially supported.

5.2. Analog Interface Specifications

Table 5.4. Trigger and other outputs

Parameter	Details	min	typ	max
trigger outputs	-	2 SMA on back panel		
trigger output impedance	-	50 Ω		
trigger output voltage range	50 Ω impedance	0 V	-	3.3 V
reference clock output	-	SMA on back panel		
reference clock output amplitude	100 MHz into 50 Ω	1 Vpp		
reference clock output frequency	-	10 / 100 MHz		

Table 5.5. Inputs

Parameter	Details	min	typ	max
trigger inputs	-	2 SMA on back panel		
trigger input impedance	-	50 Ω		
trigger input voltage range	50 Ω impedance	0 V	-	3.3 V
trigger input threshold	-	-	0.5 V	-
reference clock input frequency	-	10 / 100 MHz		
reference clock input amplitude	-	0 dBm	-	+13 dBm

5.3. Digital Interface Specifications

Table 5.6. Digital Interfaces

Parameter	Description
host computer connection	USB 3.0, 1.6 Gbit/s
	1GbE, LAN / Ethernet, 1 Gbit/s
DIO port	4 x 8 bit, general purpose digital input/output port, 3.3 V TTL specification
ZSync peripheral port	18 connectors for Zurich Instruments proprietary bus to communicate with external peripherals, 1.2 Gbit/s, 2.4V LVDS specification

5.3.1. DIO Connector

The DIO port is a VHDCI 68 pin connector as introduced by the SPI-3 document of the SCSI-3 specification. It is a female connector that requires a 32 mm wide male connector. The DIO port features 32 bits that can be configured byte-wise as inputs or outputs.

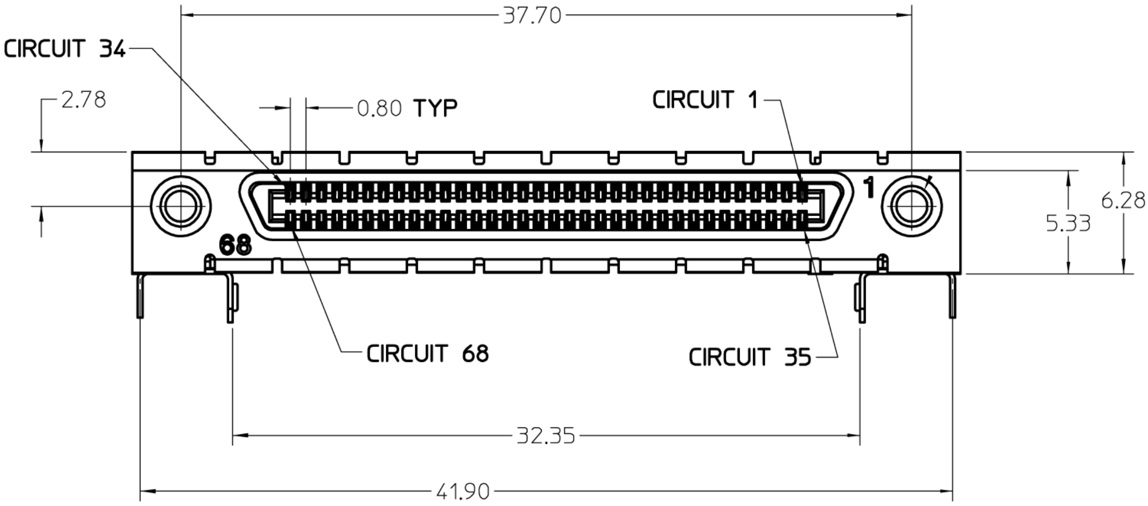


Figure 5.1. DIO HD 68 pin connector

Table 5.7. DIO pin assignment

Pin	Name	Description	Range specification
68	n/a	no signal assigned	3.3 V CMOS/TTL
67	DOL	DIO output latch, 50 MHz clock signal, the digital outputs are synchronized to the falling edge of this signal	3.3 V CMOS
66-59	DIO[31:24]	digital input or output (set by user)	output CMOS 3.3 V, input is CMOS/TTL
58-51	DIO[23:16]	digital input or output (set by user)	output CMOS 3.3 V, input is CMOS/TTL
50-43	DIO[15:8]	digital input or output (set by user)	output CMOS 3.3 V, input is CMOS/TTL

Pin	Name	Description	Range specification
42-35	DIO[7:0]	digital input or output (set by user)	output CMOS 3.3 V, input is CMOS/TTL
34-1	GND	digital ground	-

Chapter 6. Device Node Tree

This chapter contains reference documentation for the settings and measurement data available on PQSC Instruments. Whilst Chapter 4 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 organised and comprehensive list of device functionality.

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

Please see:

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

6.1. Introduction

This section 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 Section 1.5.1 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, it's device ID becomes a top-level branch in the Data Server's node tree. The features of the instrument are organised 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 "dev2006" are located in the tree in the branch:

/DEV2006/AUXOUTS/

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

/DEV2006/AUXOUTS/0/
/DEV2006/AUXOUTS/1/
/DEV2006/AUXOUTS/2/
/DEV2006/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:

/DEV2006/AUXOUTS/0/DEMODSELECT
/DEV2006/AUXOUTS/0/LIMITLOWER
/DEV2006/AUXOUTS/0/LIMITUPPER
/DEV2006/AUXOUTS/0/OFFSET
/DEV2006/AUXOUTS/0/OUTPUTSELECT
/DEV2006/AUXOUTS/0/PREOFFSET
/DEV2006/AUXOUTS/0/SCALE
/DEV2006/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 Section 6.2. The different properties and types are explained in Section 6.1.1.

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 Section 6.1.2.

Module Parameters

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

6.1.1. Node Properties and Data Types

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

Read	Data can be read from the node.
Write	Data can be written to the node.
Setting	A node with write attribute corresponding to an instrument configuration. The data in these nodes will be saved to and loaded from LabOne XML settings files.
Streaming	A node with the read attribute that provides instrument data, typically at a user-configured rate. The data is usually a more complex data type, for example demodulator data is returned as `ZIDemodSample`. .

A node may contain data of the following types:

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

6.1.2. Exploring the Node Tree

In the LabOne User Interface

A convenient method to learn which node is responsible for a specific instrument setting is to check the Command Log history in the bottom of the LabOne User Interface. The command in the Status Bar gets updated every time a configuration change is made. Figure 6.1 shows how the equivalent Matlab command is displayed after modifying the value of the auxilliary 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.

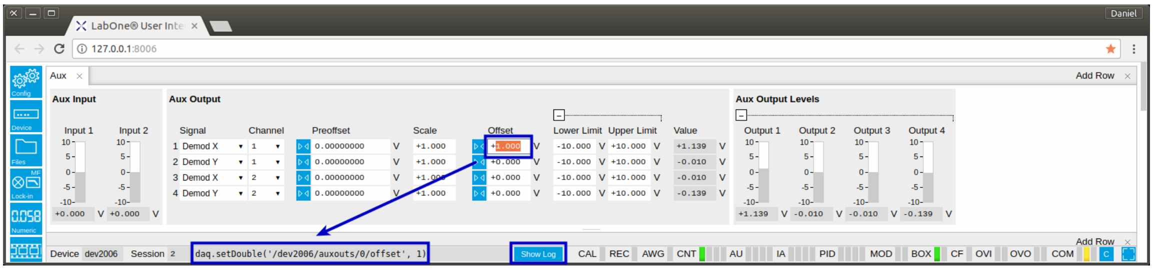


Figure 6.1. When a device's configuration is modified in the LabOne User Interface, the Status Bar displays the equivalent command to perform the same configuration via a LabOne programming interface. Here, the Matlab code to modify auxiliary output 1's offset value is provided. When "Show Log" is clicked 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 Section 6.2 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.ziPython.ziDAQServer('localhost', 8004, 6)
daq.help('/dev2006/auxouts/0/offset')
# Out:
# /DEV2006/AUXOUTS/0/OFFSET
# Add the specified offset voltage to the signal after scaling. Auxiliary Output
# Value = (Signal+Preoffset)*Scale + Offset
# Properties: Read, Write, Setting
# Type: Double
# Unit: V
```

6.1.3. Data Server Nodes

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

- /ZI/ABOUT/VERSION
- /ZI/ABOUT/REVISION

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

The nodes:

- /ZI/CONFIG/OPEN
- /ZI/CONFIG/PORT

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

Nodes that are of particualar use to programmers are:

- /ZI/DEBUG/LOGPATH - the location of the Data Server's log in the PC's filesystem,
- /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.

For documentation of all Data Server nodes see Section 6.2.

6.2. Reference Node Documentation

This section describes all the nodes in the data server's node tree organised by branch. See Section 6.1.1 for an explanation of node properties and types.

6.2.1. CLOCKBASE

/DEV..../CLOCKBASE

Properties: Read	Type: Double	Unit: Hz

Returns the internal clock frequency of the device.

6.2.2. EXECUTION

/DEV..../EXECUTION/ENABLE

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

Activate the trigger generation. Auto-resets to zero when done.

/DEV..../EXECUTION/HOLDOFF

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

Hold-off time between repeated triggers.

/DEV..../EXECUTION/PROGRESS

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

The fraction of the triggers generated so far.

/DEV..../EXECUTION/REPETITIONS

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

Number of triggers to be generated.

6.2.3. FEATURES

/DEV..../FEATURES/CODE

Properties: Read, Write	Type: String	Unit: None

Node providing a mechanism to write feature codes.

/DEV..../FEATURES/DEVTYPE

Properties: Read	Type: String	Unit: None

Returns the device type.

/DEV..../FEATURES/OPTIONS

Properties: Read	Type: String	Unit: None

Returns enabled options.

/DEV..../FEATURES/SERIAL

Properties: Read	Type: String	Unit: None

Device serial number.

6.2.4. FEEDBACK

/DEV..../FEEDBACK/REGISTERBANK/RESET

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

Clear all the readout registers.

/DEV..../FEEDBACK/DECODER/LUT/SOURCES/n/INDEX

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

The index of the bit in the readout register used as source address of the Lookup Table.

/DEV..../FEEDBACK/DECODER/LUT/SOURCES/n/REGISTER

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

The readout register used as source address of the Lookup Table.

/DEV..../FEEDBACK/DECODER/LUT/TABLES/n

Properties: Read, Write	Type: ZIVectorData	Unit: None

The lookup Table. A vector of 2^16 elements, each represents the output when its corresponding index is the source input address. Each element is an unsigned integer of 8 bits

6.2.5. 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/PACKETSSENT

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/OVERTEMPERATURE

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

This flag is set to 1 if the temperature of the FPGA exceeds 85°C. It will be reset to 0 after a restart of the device.

/DEV..../STATS/PHYSICAL/FPGA/AUX

Properties: Read	Type: Double	Unit: V

Supply voltage of the FPGA.

/DEV..../STATS/PHYSICAL/FPGA/CORE

Properties: Read	Type: Double	Unit: V

Core voltage of the FPGA.

/DEV..../STATS/PHYSICAL/FPGA/TEMP

Properties: Read	Type: Double	Unit: °C

Internal temperature of the FPGA.

6.2.6. 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/TIME

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

The current timestamp.

/DEV..../STATUS/FLAGS/BINARY

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

A set of binary flags giving an indication of the state of various parts of the device. Reserved for future use.

/DEV..../STATUS/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.

6.2.7. SYSTEM

/DEV..../SYSTEM/ACTIVEINTERFACE

Properties: Read	Type: String	Unit: None

Currently active interface of the device.

/DEV..../SYSTEM/FPGAREVISION

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

HDL firmware revision

/DEV..../SYSTEM/FWLOGENABLE

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

Enables logging to the fwlog node.

/DEV..../SYSTEM/FWREVISION

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

Revision of the device internal controller software

/DEV..../SYSTEM/FX3REVISION

Properties: Read	Type: String	Unit: None

USB firmware revision

/DEV..../SYSTEM/IDENTIFY

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

Setting this node to 1 will cause the device to blink the power led for a few seconds.

/DEV..../SYSTEM/INTERFACESPEED

Properties: Read Type: String Unit: None

Speed of the currently active interface (USB only).

/DEV..../SYSTEM/OWNER

Properties: Read Type: String Unit: None

Returns the current owner of the device (IP).

/DEV..../SYSTEM/PORTTCP

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

Returns the current TCP port used for communication to the dataserver.

/DEV..../SYSTEM/PORTUDP

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

Returns the current UDP port used for communication to the dataserver.

/DEV..../SYSTEM/POWERCONFIGDATE

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

Contains the date of power configuration (format is: (year << 16) | (month << 8) | day)

/DEV..../SYSTEM/SAVEPORTS

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

Flag indicating that the TCP and UDP ports should be saved.

/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 MFLI 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.

/DEV..../SYSTEM/BOARDREVISIONS/n

Properties: Read	Type: String	Unit: None

Hardware revision of the FPGA base board

/DEV..../SYSTEM/PROPERTIES/MAXFREQ

Properties: Read	Type: Double	Unit: None

The maximum oscillator frequency that can be set.

/DEV..../SYSTEM/PROPERTIES/MINFREQ

Properties: Read	Type: Double	Unit: None

The minimum oscillator frequency that can be set.

/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. Is equal to 1/(maximum sampling rate).

/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/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.

Allowed Values:

0 "internal": The internal clock is 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 10MHz or 100MHz 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.

Allowed Values:

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.

Allowed Values:

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 100 MHz is allowed.

6.2.8. TRIGGERS

/DEV..../TRIGGERS/OUT/n/ENABLE

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

Enable the Trigger Out.

/DEV..../TRIGGERS/OUT/n/PORT

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

The ZSync port associated with the Trigger Out source.

/DEV..../TRIGGERS/OUT/n/PULSEWIDTH

Properties: Read	Type: Double	Unit: s

Defines the minimum pulse width of the generated pulses.

/DEV..../TRIGGERS/OUT/n/SOURCE

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

Select the source of the trigger Out.

Allowed Values:

0 "start_trigger": Generate a trigger when a "start trigger" is sent over the selected ZSync.

1 "feedback": Generate a trigger when feedback is sent over the selected ZSync.

6.2.9. ZSYNCS

/DEV..../ZSYNCS/n/CONNECTION/ALIAS

Properties: Read, Write, Setting | Type: ZIVectorData | Unit: None

User-given name to the instrument connected to this port.

/DEV..../ZSYNCS/n/CONNECTION/DEVTYPE

Properties: Read | Type: ZIVectorData | Unit: None

The device type of the instrument connected to this port.

/DEV..../ZSYNCS/n/CONNECTION/SERIAL

Properties: Read | Type: ZIVectorData | Unit: None

The device ID of the instrument connected to this port.

/DEV..../ZSYNCS/n/CONNECTION/STATUS

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

The current status of the instrument connected to the port.

Allowed Values:

0 No connection

1 Connection in progress

2 Connected

3 Connection error

4 Data is being sent/received

/DEV..../ZSYNCS/n/OUTPUT/DECODER/ENABLE

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

Enable decoder output forwarding for a given port.

/DEV..../ZSYNCS/n/OUTPUT/DECODER/SOURCE

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

The index of the lookup table in the LUT decoder that is forwarded.

/DEV..../ZSYNCS/n/OUTPUT/REGISTERBANK/ENABLE

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

Enable readout registers forwarding for a given port.

/DEV..../ZSYNCS/n/OUTPUT/REGISTERBANK/SOURCES/n/INDEX

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

The index of the bit in the readout register forwarded.

/DEV..../ZSYNCS/n/OUTPUT/REGISTERBANK/SOURCES/n/REGISTER

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

The readout register forwarded.

6.2.10. ZI (Labone Data Server Nodes)

/ZI/ABOUT/COMMIT

Properties: Read	Type: String	Unit: None

Contains the commit hash of the source code used to build this version of the LabOne software.

/ZI/ABOUT/COPYRIGHT

Properties: Read	Type: String	Unit: None

Holds the copyright notice.

/ZI/ABOUT/DATASERVER

Properties: Read	Type: String	Unit: None

Contains information about the Zurich Instruments Data Server.

/ZI/ABOUT/FWREVISION

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

Contains the revision of the device firmware.

/ZI/ABOUT/REVISION

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

Contains the revision number of the Zurich Instruments Data Server.

/ZI/ABOUT/VERSION

Properties: Read | Type: String | Unit: None

Contains the version of the LabOne software.

/ZI/CLOCKBASE

Properties: Read | Type: Double | Unit: None

A fallback clock frequency that can be used by clients for calculating time bases when no other is available.

/ZI/CONFIG/OPEN

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

Enable communication with the LabOne Data Server from other computers in the network.

Allowed Values:

0 "local": Communication only possible with the local machine.

1 "network": Communication possible with other machines in the network.

/ZI/CONFIG/PORT

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

The IP port on which the LabOne Data Server listens.

/ZI/DEBUG/LEVEL

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

Set the logging level (amount of detail) of the LabOne Data Server.

Allowed Values:

0 "trace": Trace. Messages designated as traces are logged.

1 "debug": Debug. Messages designated as debugging info are logged.

2 "info": Info. Messages designated as informational are logged.

3 "status": Status. Messages designated as status info are logged.

4 "warning": Warning. Messages designated as warnings are logged.

5 "error": Error. Messages designated as errors are logged.

6 "fatal": Fatal. Messages designated as fatal errors are logged.

/ZI/DEBUG/LOG

Properties: Read	Type: String	Unit: None

Returns the logfile text of the LabOne Data Server.

/ZI/DEBUG/LOGPATH

Properties: Read	Type: String	Unit: None

Returns the path of the log directory.

/ZI/DEVICES/CONNECTED

Properties: Read	Type: String	Unit: None

Contains a list of devices connected to the LabOne Data Server.

/ZI/DEVICES/DISCOVER

Properties: Read, Write	Type: String	Unit: None

Not used.

/ZI/DEVICES/VISIBLE

Properties: Read	Type: String	Unit: None

Contains a list of devices in the network visible to the LabOne Data Server.

/ZI/MDS/GROUPS/n/DEVICES

Properties: Read, Write, Setting	Type: String	Unit: None

Contains a list of devices in this synchronization group.

/ZI/MDS/GROUPS/n/KEEPALIVE

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

Set by the MDS module to indicate control over this synchronization group.

/ZI/MDS/GROUPS/n/LOCKED

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

Indicates whether the device group is locked by a MDS module.

/ZI/MDS/GROUPS/n/STATUS

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

Indicates the status the synchronization group.

Allowed Values:

- -1 Error. An error occurred in the synchronization process.
- 0 New
- 1 Sync
- 2 Alive

/ZI/SYSTEM/USAGEDATA

Properties: Read | Type: String | Unit: None

Contains a JSON formatted string giving usage information about the LabOne Software (which tabs/modules used, error conditions).

Glossary

This glossary provides easy to understand descriptions for many terms related to measurement instrumentation including the abbreviations used inside this user manual.

A

A/D	Analog to Digital See Also ADC.
AC	Alternate Current
ADC	Analog to Digital Converter
AM	Amplitude Modulation
Amplitude Modulated AFM (AM-AFM)	AFM mode where the amplitude change between drive and measured signal encodes the topography or the measured AFM variable. See Also Atomic Force Microscope.
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
Atomic Force Microscope (AFM)	Microscope that scans surfaces by means an oscillating mechanical structure (e.g. cantilever, tuning fork) whose oscillating tip gets so close to the surface to enter in interaction because of electrostatic, chemical, magnetic or other forces. With an AFM it is possible to produce images with atomic resolution. See Also Amplitude Modulated AFM, Frequency Modulated AFM, Phase modulation AFM.
AVAR	Allen Variance

B

Bandwidth (BW)	The signal bandwidth represents the highest frequency components of interest in a signal. For filters the signal bandwidth is the cut-off point, where the transfer function of a system shows 3 dB attenuation versus DC. In this context the bandwidth is a synonym of cut-off frequency $f_{cut-off}$ or 3dB frequency f_{-3dB}. The concept of bandwidth is used when the dynamic behavior of a signal is important or separation of different signals is required. In the context of a open-loop or closed-loop system, the bandwidth can be used to indicate the fastest speed of the system, or the highest signal update change rate that is possible with the system. Sometimes the term bandwidth is erroneously used as synonym of frequency range. See Also Noise Equivalent Power Bandwidth.
BNC	Bayonet Neill-Concelman Connector

C

CF	Clock Fail (internal processor clock missing)

Common Mode Rejection Ratio (CMRR)	Specification of a differential amplifier (or other device) indicating the ability of an amplifier to obtain the difference between two inputs while rejecting the components that do not differ from the signal (common mode). A high CMRR is important in applications where the signal of interest is represented by a small voltage fluctuation superimposed on a (possibly large) voltage offset, or when relevant information is contained in the voltage difference between two signals. The simplest mathematical definition of common-mode rejection ratio is: CMRR = 20 * log(differential gain / common mode gain).
CSV	Comma Separated Values

D

D/A	Digital to Analog
DAC	Digital to Analog Converter
DC	Direct Current
DDS	Direct Digital Synthesis
DHCP	Dynamic Host Configuration Protocol
DIO	Digital Input/Output
DNS	Domain Name Server
DSP	Digital Signal Processor
DUT	Device Under Test
Dynamic Reserve (DR)	The measure of a lock-in amplifier's capability to withstand the disturbing signals and noise at non-reference frequencies, while maintaining the specified measurement accuracy within the signal bandwidth.

E

XML	Extensible Markup Language. See Also XML.

F

FFT	Fast Fourier Transform
FIFO	First In First Out
FM	Frequency Modulation
Frequency Accuracy (FA)	Measure of an instrument's ability to faithfully indicate the correct frequency versus a traceable standard.
Frequency Modulated AFM (FM-AFM)	AFM mode where the frequency change between drive and measured signal encodes the topography or the measured AFM variable. See Also Atomic Force Microscope.
Frequency Response Analyzer (FRA)	Instrument capable to stimulate a device under test and plot the frequency response over a selectable frequency range with a fine granularity.

Frequency Sweeper	See Also Frequency Response Analyzer.

G

Gain Phase Meter	See Also Vector Network Analyzer.
GPIB	General Purpose Interface Bus
GUI	Graphical User Interface

I

I/O	Input / Output
Impedance Spectroscope (IS)	Instrument suited to stimulate a device under test and to measure the impedance (by means of a current measurement) at a selectable frequency and its amplitude and phase change over time. The output is both amplitude and phase information referred to the stimulus signal.
Input Amplitude Accuracy (IAA)	Measure of instrument's capability to faithfully indicate the signal amplitude at the input channel versus a traceable standard.
Input voltage noise (IVN)	Total noise generated by the instrument and referred to the signal input, thus expressed as additional source of noise for the measured signal.
IP	Internet Protocol

L

LAN	Local Area Network
LED	Light Emitting Diode
Lock-in Amplifier (LI, LIA)	Instrument suited for the acquisition of small signals in noisy environments, or quickly changing signal with good signal to noise ratio - lock-in amplifiers recover the signal of interest knowing the frequency of the signal by demodulation with the suited reference frequency - the result of the demodulation are amplitude and phase of the signal compared to the reference: these are value pairs in the complex plane (X, Y), (R, Θ).

M

Media Access Control address (MAC address)	Refers to the unique identifier assigned to network adapters for physical network communication.
Multi-frequency (MF)	Refers to the simultaneous measurement of signals modulated at arbitrary frequencies. The objective of multi-frequency is to increase the information that can be derived from a measurement which is particularly important for one-time, non-repeating events, and to increase the speed of a measurement since different frequencies do not have to be applied one after the other. See Also Multi-harmonic.
Multi-harmonic (MH)	Refers to the simultaneous measurement of modulated signals at various harmonic frequencies. The objective of multi-frequency is to increase the

information that can be derived from a measurement which is particularly important for one-time, non-repeating events, and to increase the speed of a measurement since different frequencies do not have to be applied one after the other.
See Also Multi-frequency.

N

Noise Equivalent Power Bandwidth (NEPBW)	Effective bandwidth considering the area below the transfer function of a low-pass filter in the frequency spectrum. NEPBW is used when the amount of power within a certain bandwidth is important, such as noise measurements. This unit corresponds to a perfect filter with infinite steepness at the equivalent frequency. See Also Bandwidth.
Nyquist Frequency (NF)	For sampled analog signals, the Nyquist frequency corresponds to two times the highest frequency component that is being correctly represented after the signal conversion.

O

Output Amplitude Accuracy (OAA)	Measure of an instrument's ability to faithfully output a set voltage at a given frequency versus a traceable standard.
OV	Over Volt (signal input saturation and clipping of signal)

P

PC	Personal Computer
PD	Phase Detector
Phase-locked Loop (PLL)	Electronic circuit that serves to track and control a defined frequency. For this purpose a copy of the external signal is generated such that it is in phase with the original signal, but with usually better spectral characteristics. It can act as frequency stabilization, frequency multiplication, or as frequency recovery. In both analog and digital implementations it consists of a phase detector, a loop filter, a controller, and an oscillator.
Phase modulation AFM (PM-AFM)	AFM mode where the phase between drive and measured signal encodes the topography or the measured AFM variable. See Also Atomic Force Microscope.
PID	Proportional-Integral-Derivative
PL	Packet Loss (loss of packets of data between the instruments and the host computer)

R

RISC	Reduced Instruction Set Computer
Root Mean Square (RMS)	Statistical measure of the magnitude of a varying quantity. It is especially useful when variates are positive and negative, e.g., sinusoids, sawtooth, square waves. For a sine wave the following relation holds between the

amplitude and the RMS value: $U_{RMS} = U_{PK} / \sqrt{2} = U_{PK} / 1.41$. The RMS is also called quadratic mean.

RT — Real-time

S

Scalar Network Analyzer (SNA) — Instrument that measures the voltage of an analog input signal providing just the amplitude (gain) information.
See Also Spectrum Analyzer, Vector Network Analyzer.

SL — Sample Loss (loss of samples between the instrument and the host computer)

Spectrum Analyzer (SA) — Instrument that measures the voltage of an analog input signal providing just the amplitude (gain) information over a defined spectrum.
See Also Scalar Network Analyzer.

SSH — Secure Shell

T

TC — Time Constant

TCP/IP — Transmission Control Protocol / Internet Protocol

Thread — An independent sequence of instructions to be executed by a processor.

Total Harmonic Distortion (THD) — Measure of the non-linearity of signal channels (input and output)

TTL — Transistor to Transistor Logic level

U

UHF — Ultra-High Frequency

UHS — Ultra-High Stability

USB — Universal Serial Bus

V

VCO — Voltage Controlled Oscillator

Vector Network Analyzer (VNA) — Instrument that measures the network parameters of electrical networks, commonly expressed as s-parameters. For this purpose it measures the voltage of an input signal providing both amplitude (gain) and phase information. For this characteristic an older name was gain phase meter.
See Also Gain Phase Meter, Scalar Network Analyzer.

X

XML — Extensible Markup Language: Markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

Z

ZCtrl	Zurich Instruments Control bus
ZoomFFT	This technique performs FFT processing on demodulated samples, for instance after a lock-in amplifier. Since the resolution of an FFT depends on the number of point acquired and the spanned time (not the sample rate), it is possible to obtain very highly resolution spectral analysis.
ZSync	Zurich Instruments Synchronization bus

Index