PQSC User Manual

Programmable Quantum System Controller

Zurich Instruments

PQSC User Manual

Zurich Instruments AG

Revision 25.04

Table of Contents

Table of Contents

CE Declaration of Conformity

The manufacturer

Zurich Instruments
Technoparkstrasse 1
8005 Zurich
Switzerland

declares that the product

PQSC, Programmable Quantum System Controller

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

Directive / Regulation	Conformity proven by compliance with the standards
2014/30/EU (Electromagnetic compatibility [EMC])	EN 61326-1:2013, EN 55011:2016, EN 55011:2016/A1:2017, EN 55011:2016/A11:2020 (Group 1, Class A and B equipment)
2014/35/EU (Low voltage equipment [LVD])	EN 61010-1:2010, EN 61010-1:2010/A1:2019, EN 61010-1:2010/A1:2019/AC:2019-04
2011/65/EU, as amended by 2015/863 and 2017/2102 (Restriction of the use of certain hazardous substances [RoHS])	EN IEC 63000:2018
(EC) 1907/2006 (Registration, Evaluation, Authorisation, and Restrictions of Chemicals [REACH])	-

Zurich, October 20th, 2022

Flavio Heer

Flavio Heer, CTO

UKCA Declaration of Conformity

The manufacturer

Zurich Instruments
Technoparkstrasse 1
8005 Zurich
Switzerland

declares that the product

PQSC, Programmable Quantum System Controller

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

Statutory Instruments	Conformity proven by compliance with the standards
S.I. 2016/1091 (Electromagnetic Compatibility Regulations)	EN 61326-1:2013, EN 55011:2016, EN 55011:2016/A1:2017, EN 55011:2016/A11:2020 (Group 1, Class A and B equipment)
S.I. 2016/1101 (Electrical Equipment (Safety) Regulations)	EN 61010-1:2010, EN 61010-1:2010/A1:2019, EN 61010-1:2010/A1:2019/AC:2019-04
S.I. 2012/3032 (Restriction of the Use of Certain Hazardous Substances Regulations)	EN IEC 63000:2018

Zurich, October 20th, 2022

[illegible]

Flavio Heer, CTO

1. Change Log

Info

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

1.1. Release 25.04

Release date: 30-April-2025

See Release Notes 25.04 for a detailed list of all changes.

1.2. Release 25.01

Release date: 31-January-2025

See Release Notes 25.01 for a detailed list of all changes.

1.3. Release 24.10

Release date: 31-October-2024

See Release Notes 24.10 for a detailed list of all changes.

1.4. Release 24.07

Release date: 31-July-2024

See Release Notes 24.07 for a detailed list of all changes.

1.5. Release 24.04

Release date: 30-April-2024

See Release Notes 24.04 for a detailed list of all changes.

1.6. Release 24.01

Release date: 31-January-2024

See Release Notes 24.01 for a detailed list of all changes.

1.7. Release 23.10

Release date: 31-Oct-2023

- Added automatic fallback to a link-local IP address in case no DHCP server could be found.
- Introduced a new high-performance data-server kernel. It improves reliability and performances of communication with the instrument.

1.8. Release 23.06

Release date: 30-Jun-2023

- Added an optional synchronization check that waits until all ZSync workers have reported their readiness before ZSync start triggers are executed.
- The register forwarding unit can now forward up to eight readout results, or 16 bits, to each ZSync port. See Extended register forwarding for more details.
- Feedback events don't generate a ZSync trigger anymore. See New ZSync trigger behavior for more details.
- Readout instruments, either SHFQC, SHFQA or UHFQA (via HDAWG bridge), can now be connected to all ZSync ports, not only to the first eight.

1.9. Release 23.02

Release date: 28-Feb-2023

1.10. Release 22.08

Release date: 31-Aug-2022

- Aligned ZSync start trigger between HDAWG, SHFQA, SHFSG and SHFQC instruments.
- Aligned ZSync start trigger to external reference clock for either 10 MHz or 100 MHz PQSC output clock.

1.11. Release 22.02

Release date: 28-Feb-2022

- Added new indicator that advises when the PQSC is ready after warmup
- Improved ZSync link stability
- Improved locking to external clock

1.12. Release 21.08

Release date: 30-Aug-2021

- Full support for SHFQA instrument, expanded readout registry bank
- Stability improvements in ZSync link and synchronization for all instruments
- Add graphical interface to control the feedback processing
- Each bit in the register forwarding can be enabled separately

1.13. Release 21.02

Release date: 28-Feb-2021

- LabOne API: Added online Programming Manual and Documentation
- Enabled ZSync support of SHF instruments
- Improved locking to external clock

1.14. Release 20.07

Release date: 20-Aug-2020

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

1.15. Release 20.01

Release date: 28-Feb-2020

First version of PQSC User Manual.

1.16. Release 23.06 Additional Information

1.16.1. Extended register forwarding

Status until LabOne 23.02

The PQSC has two feedback units: the register forwarding and the error decoder (see Feedback Tab for more details). The register forwarding unit is intended for simple feedback experiments, where a qubit readout maps directly to a feedback action, like active qubit reset. It can send up to four arbitrary qubit readout results per ZSync port, each one bit in size. Such results are selected from the readout register bank, a temporary storage of readout results populated by the QA units. The user can select the results to forward without constraint. This enable the system to perform a conditional action on a worker device, regardless of the origin of the readout. The number of results that can be forwarded is designed to match the capability of the instruments connected to the PQSC. Typically, an HDAWG8 is used to control up to four qubits, hence the choice of allowing the forwarding of four qubit readout results. The output of the register forwarding unit is multiplexed with the output of the error decoder unit and both are available for the worker instrument in the same message.

Example

In this example, there are two QA channels writing their readout results in register 1 and 29 respectively. The PQSC is configured to forward eight results to two different instruments connected to the first two ZSync ports, as shown in Figure 1.1. When a QA performs a readout, its results are automatically written in the readout register specified in the fourth argument of the **startQA** command, and the PQSC automatically forwards the chosen results.

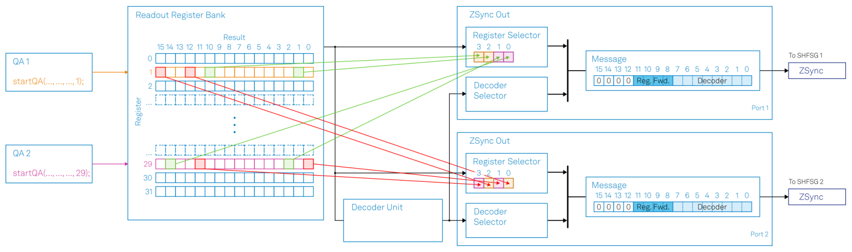


Figure 1.1: An example of register forwarding in the PQSC until LabOne 23.02

To configure the PQSC as in Figure 1.1, its nodes should be programmed as in Table 1.1.

Table 1.1: Settings: PQSC register forwarding settings

Node	Value
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/ENABLE`	True
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/*/ENABLE`	True
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/0/REGISTER`	29
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/0/INDEX`	14
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/1/REGISTER`	29
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/1/INDEX`	2
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/2/REGISTER`	1
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/2/INDEX`	1

Node	Value
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/3/REGISTER`	1
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/3/INDEX`	10
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/ENABLE`	True
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/*/ENABLE`	True
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/0/REGISTER`	1
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/0/INDEX`	12
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/1/REGISTER`	29
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/1/INDEX`	0
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/2/REGISTER`	1
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/2/INDEX`	15
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/3/REGISTER`	29
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/3/INDEX`	11

The instruments receive a word from the data sources configured in the PQSC. Such words contain the output of both the register forwarding and the error decoder unit. The conditional instruction, either `getFeedback` or `executeTableEntry`, should specify the source of feedback with the constant `ZSYNC_DATA_PQSC_REGISTER` or `ZSYNC_DATA_PQSC_DECODER` to select the relevant portion of the message. The constant `ZSYNC_DATA_RAW` provides the full message without any processing. Each sequencer in the instrument can reduce the message to the portion that is relevant for it; for example the state of a qubit to reset. This is done by binary shift and masking. An optional additive offset could be added. In Figure 1.2 an example of the flow of data to reset four qubits connected to a SHFSG4 is provided.

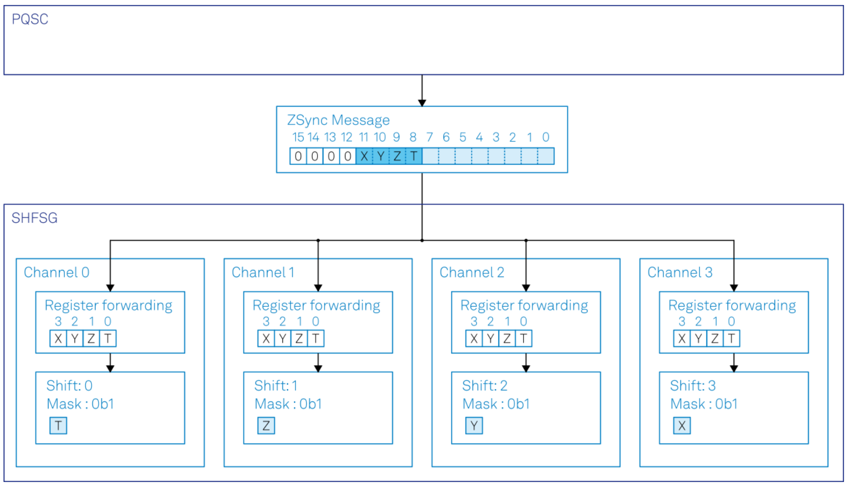


Figure 1.2: Feedback data flow in a SHFSG4 doing active qubit reset until LabOne 23.02

In order to enable such processing, the instrument nodes should be configured as in Table 1.2.

Table 1.2: Settings: SHFSG4 feedback processing settings until LabOne 23.02

Node	Value
`/DEV.../SGCHANNELS/0/AWG/ZSYNC/REGISTER/SHIFT`	0
`/DEV.../SGCHANNELS/0/AWG/ZSYNC/REGISTER/MASK`	0b1
`/DEV.../SGCHANNELS/1/AWG/ZSYNC/REGISTER/SHIFT`	1
`/DEV.../SGCHANNELS/1/AWG/ZSYNC/REGISTER/MASK`	0b1
`/DEV.../SGCHANNELS/2/AWG/ZSYNC/REGISTER/SHIFT`	2
`/DEV.../SGCHANNELS/2/AWG/ZSYNC/REGISTER/MASK`	0b1
`/DEV.../SGCHANNELS/3/AWG/ZSYNC/REGISTER/SHIFT`	3

Node	Value
`/DEV.../SGCHANNELS/3/AWG/ZSYNC/REGISTER/MASK`	0b1

With such settings, each sequencer can get its processed feedback in a variable with `var feedback = getFeedback(ZSYNC_DATA_PQSC_REGISTER)` or directly play a conditional pulse with `executeTableEntry(ZSYNC_DATA_PQSC_REGISTER)`.

New behavior since LabOne 23.06

The register forwarding unit has been extended and it now supports forwarding of up to eight readout results, each made of up to two bits. This allows the system to perform active qubit/qutrit/ququad reset for each control channel on larger instruments like the SHFSG8 or SHFQC6. To enable that, three main aspects have been changed:

- The register forward selector picks couples of bits instead of single bits. The index now refers to such couples and it's equivalent to `2n` bits. Allowed values are therefore from 0 to 7 instead of 0 to 15.
- There are eight selectors instead of four.
- The data from the register forwarding unit and the decoder unit are not multiplexed anymore. Therefore a port can exclusively send data from register forwarding or the error decoder.

The nodes

- `/DEV.../ZSYNCS/n/OUTPUT/DECODER/ENABLE`
- `/DEV.../ZSYNCS/n/OUTPUT/REGISTERBANK/ENABLE`

on the PQSC have been removed, and their functionality replaced by the nodes

- `/DEV.../ZSYNCS/n/OUTPUT/ENABLE`
- `/DEV.../ZSYNCS/n/OUTPUT/SOURCE`

The `ENABLE` node is used to enable feedback output on a given port, while the `SOURCE` node is used to select if a port should send data from the register forwarding unit or from the decoder unit.

Example

Like in the previous example, there are two QA channels writing their readout results in register 1 and 29 respectively. QA1 is configured such that two results (X and Y) are the output of Multi State Discrimination (MSD), so each is composed by two bits, while another (Z) is the result of regular discrimination, so it uses only one bit. The QA2 produces three results (I,J and K) from regular discrimination, so a total of three bits. The PQSC is configured to forward these results to two different instruments connected to the first two ZSync ports, as shown in Figure 1.1. A third instrument is configured to receive the output of the decoder unit. For clarity in this example, the register forwarding is sending only three results per port instead of the maximum of eight and the configuration of the decoder unit is omitted.

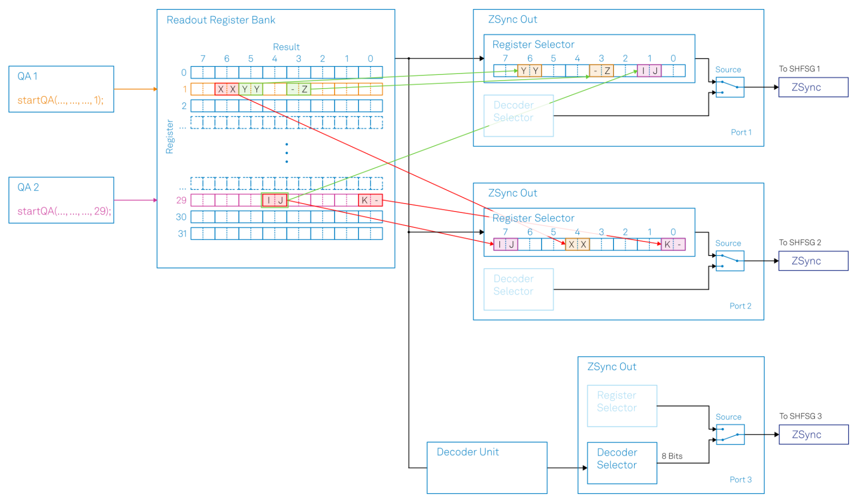


Figure 1.3: An example of register forwarding in the PQSC of LabOne 23.06 and above

To configure the PQSC as in Figure 1.3, its nodes should be programmed as in Table 1.3.

Table 1.3: Settings: PQSC register forwarding settings

Node	Value
`/DEV.../ZSYNCS/0/OUTPUT/ENABLE`	True
`/DEV.../ZSYNCS/0/OUTPUT/SOURCE`	"reg"
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/1/ENABLE`	True
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/1/REGISTER`	29
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/1/INDEX`	4
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/3/ENABLE`	True
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/3/REGISTER`	1
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/3/INDEX`	3
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/6/ENABLE`	True
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/6/REGISTER`	1
`/DEV.../ZSYNCS/0/OUTPUT/REGISTERBANK/SOURCES/6/INDEX`	5
`/DEV.../ZSYNCS/1/OUTPUT/ENABLE`	True
`/DEV.../ZSYNCS/1/OUTPUT/SOURCE`	"reg"
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/0/ENABLE`	True
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/0/REGISTER`	29
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/0/INDEX`	0
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/4/ENABLE`	True
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/4/REGISTER`	1
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/4/INDEX`	6
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/7/ENABLE`	True
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/7/REGISTER`	29
`/DEV.../ZSYNCS/1/OUTPUT/REGISTERBANK/SOURCES/7/INDEX`	4
`/DEV.../ZSYNCS/2/OUTPUT/ENABLE`	True
`/DEV.../ZSYNCS/2/OUTPUT/SOURCE`	"decoder"

Like in the previous example, each sequencer in the instrument can reduce the message to the portion that is relevant for it by binary shift and masking. In Figure 1.4, there is an example of the flow of data of the SHFSG8 connected to the first ZSync port of the PQSC; it reset three qubits (channels 2 and 3) and one qutrit. Similarly, in Figure 1.5, the same goes for the SHFSG8 connected to the second ZSync port of the PQSC.

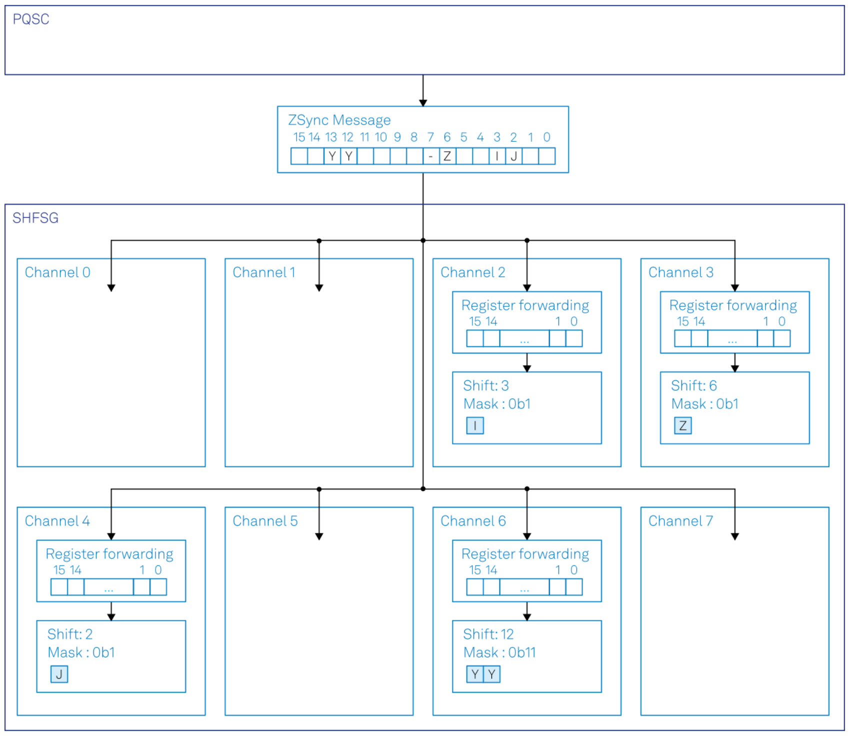


Figure 1.4: Feedback data flow in the first SHFSG8 doing active qubit and qutrit reset with LabOne 23.06 and above

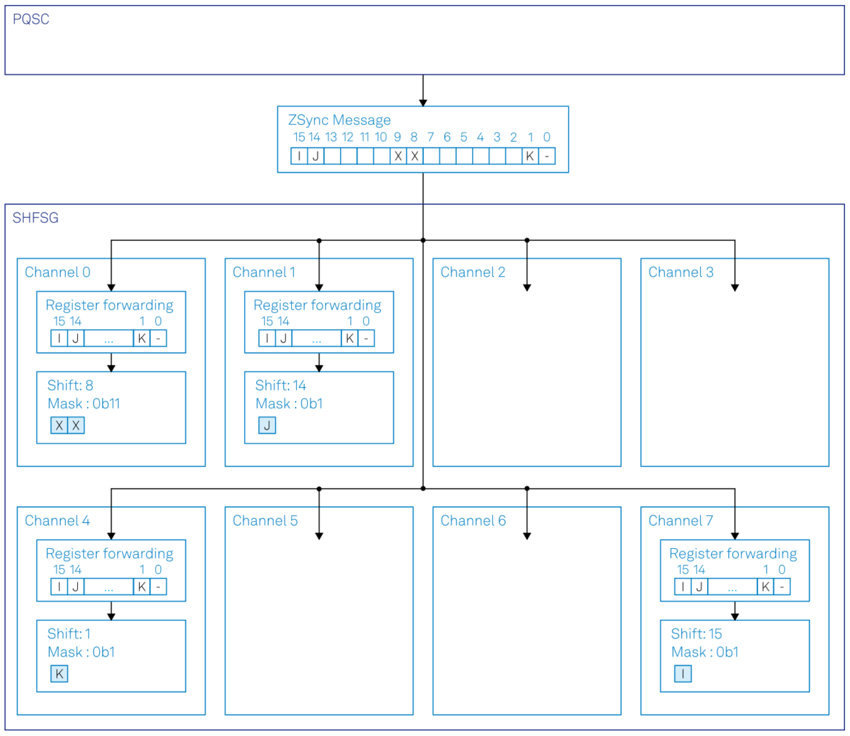


Figure 1.5: Feedback data flow in the second SHFSG8 doing active qubit and qutrit reset with LabOne 23.06 and above

In order to enable such processing, the instrument nodes should be configured as in Table 1.4 and Table 1.5. It should be noted that the shift is still expressed in bits, not couples of bits like in the register selector of the PQSC. This allows for finer granularity in the selection of the interesting portion of the message. It also allows for the execution of feedback on a single bit, even if the PQSC register forwarding unit sent two bits, like in channel 2 and 3 of the first SHFSG.

Table 1.4: Settings: First SHFSG8 feedback processing settings in LabOne 23.06 and

Node	Value
`/DEV.../SGCHANNELS/2/AWG/ZSYNC/REGISTER/SHIFT`	3
`/DEV.../SGCHANNELS/2/AWG/ZSYNC/REGISTER/MASK`	0b1

Node	Value
/DEV.../SGCHANNELS/3/AWG/ZSYNC/REGISTER/SHIFT	6
/DEV.../SGCHANNELS/3/AWG/ZSYNC/REGISTER/MASK	0b1
/DEV.../SGCHANNELS/4/AWG/ZSYNC/REGISTER/SHIFT	2
/DEV.../SGCHANNELS/4/AWG/ZSYNC/REGISTER/MASK	0b1
/DEV.../SGCHANNELS/6/AWG/ZSYNC/REGISTER/SHIFT	12
/DEV.../SGCHANNELS/6/AWG/ZSYNC/REGISTER/MASK	0b11

above

Table 1.5: Settings: Second SHFSG8 feedback processing settings in LabOne 23.06 and

Node	Value
/DEV.../SGCHANNELS/0/AWG/ZSYNC/REGISTER/SHIFT	8
/DEV.../SGCHANNELS/0/AWG/ZSYNC/REGISTER/MASK	0b11
/DEV.../SGCHANNELS/1/AWG/ZSYNC/REGISTER/SHIFT	14
/DEV.../SGCHANNELS/1/AWG/ZSYNC/REGISTER/MASK	0b1
/DEV.../SGCHANNELS/4/AWG/ZSYNC/REGISTER/SHIFT	1
/DEV.../SGCHANNELS/4/AWG/ZSYNC/REGISTER/MASK	0b1
/DEV.../SGCHANNELS/7/AWG/ZSYNC/REGISTER/SHIFT	15
/DEV.../SGCHANNELS/7/AWG/ZSYNC/REGISTER/MASK	0b1

above

Like before, the sequencers can acquire their feedback in a variable with `var feedback = getFeedback(ZSYNC_DATA_PQSC_REGISTER)`, or directly play a conditional pulse with `executeTableEntry(ZSYNC_DATA_PQSC_REGISTER)`, so no changes there.

1.16.2. New ZSync trigger behavior

Status until L1 23.02

The PQSC can send messages with a fixed size of 16 bits. The purpose of the message is not carried with the data itself, and the sequencer must know a priori which kind of message it is. A message always generates a trigger in the sequencer that can be waited for with the instruction `waitZSyncTrigger`. The payload of the message can be acquired in the sequencer with the instruction `getFeedback` (or alternatively `getZSyncData`), or a conditional playback can be executed with the instruction `executeTableEntry` (or alternatively `playWaveZSync`).

The PQSC sends two kind of messages. Start triggers are used to synchronize the execution of the sequencers across instruments. Each sequencer should wait for it at the beginning of the sequence, and its data payload is not used.

Feedback data is used to perform conditional operations depending on their payload. This operation can be timed in two ways. The recommended one is to specify the time of execution with respect to the last start trigger as a second parameter of `getFeedback` or `executeTableEntry`. Such time can be deduced by looking at the RTLogger, or by using the timing model provided by the package `zhinst-utils`. This is called time-based feedback. Another method is to wait for the trigger associated with the feedback data and only then perform the conditional action. This is called trigger-based feedback. This method, while simpler on the surface, doesn't give the user explicit information about the timing, nor does it guarantee a stable feedback time across experiments. Moreover, it is prone to errors when multiple feedback events are generated.

Example

Time-based feedback sequence

A sample sequence using time-based feedback. The constant `FEEDBACK_DELAY` holds the feedback delay in sequencer clock cycles. It has to be provided by the user using the timing model.

```
while(true) {
  //Wait for a start trigger
  waitZSyncTrigger();
  //Play a conditional pulse, only when feedback is received
  executeTableEntry(ZSYNC_DATA_PQSC_REGISTER, FEEDBACK_DELAY);
}
```

Trigger-based feedback sequence

When trigger-based feedback is used, a trigger is used to detect the arrival of the feedback message

```
while(true) {
  //Wait for a start trigger
  waitZSyncTrigger();
  //Wait for the feedback trigger
  waitZSyncTrigger();
  //Play a conditional pulse
  executeTableEntry(ZSYNC_DATA_PQSC_REGISTER);
}
```

New behavior since L1 23.06

The start triggers and the feedback data are now two semantically different messages sent over ZSync. The command `waitZSyncTrigger` will now only wait for start triggers. If a feedback message is received while the sequencer is waiting for a start trigger, it will continue to wait. This avoids the issue of the sequencer getting confused and starting a new portion of the sequence at the wrong point in time. Similarly, a new start trigger doesn't clear the last message received.

As a side effect, it is no longer possible to use the trigger-based feedback. All the sequences using such a method must be migrated to time-based feedback to keep working.

The instructions `getZSyncData` and `playWaveZSync` are deprecated. The more generic instructions `getFeedback` and `executeTableEntry` should be used instead with the same arguments respectively.

2. 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:

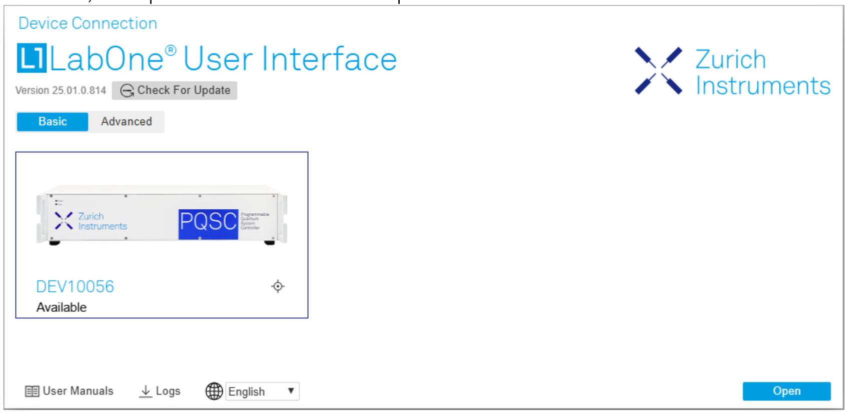
- A Quick Start Guide for the impatient
- Inspecting the package content and accessories
- List of essential handling and safety instructions
- Connecting to the PQSC Instrument
- Handy list of troubleshooting guidelines

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

2.1. Quick Start Guide

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

1. Inspect the package content. Besides the Instrument there should be a country-specific power cable, a USB cable, an Ethernet cable and a hard copy of the user manual Getting Started
2. Check the Handling and Safety Instructions in Handling and Safety Instructions.
3. Download and install the latest LabOne software from the Zurich Instruments Download Center. Choose the download file that fits your computer (e.g. Windows with 64-bit addressing). For more detailed information see Software Installation.
4. Connect the Instrument to the power line. Turn it on and connect it to a switch in the LAN using the Ethernet cable.
5. Start the LabOne User Interface from the Windows Start Menu. The default web browser will open and display your instrument in a start screen as shown below. Use Chrome, Edge, Firefox, or Opera for best user experience.



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

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

The functional description of the LabOne User Interface Functional Overview provides a general introduction to the various tools and settings tabs with tables in each section describing every UI element. For specific application know-how, the blog section of the Zurich Instruments website will serve as a valuable resource that is constantly updated and expanded.

Note

The PQSC needs to warm up for 30 minutes after power-up. Do not lock to external reference clock or start triggering before it’s ready. The CLK LED on the bottom right of the LabOne user interface will turn green when the instrument is ready to use.

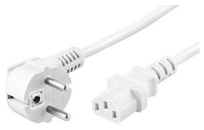
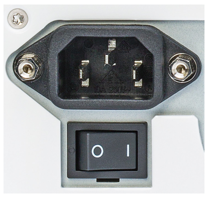
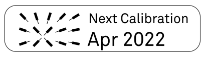
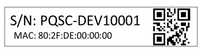
2.2. Inspect the Package Contents

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

Please verify:

- 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)
- 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 2.1: Package contents for the PQSC

Zurich Instruments PQSC Programmable Quantum System Controller	PQSC instrument
	power cord (e.g. EU norm)
	USB 3.0 cable
	power inlet, with power switch
	LAN / Ethernet cable (category 5/6 required)
 Next Calibration Apr 2022	"Next Calibration" sticker on the back panel of the instrument
 S/N: PQSC-DEV10001 MAC: 80:2F:DE:00:00:00	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.

2.3. Handling and Safety Instructions

The PQSC Instrument is a sensitive piece of electronic equipment, and under no circumstances should its casing be opened, as there are high-voltage parts inside which may be harmful to human beings. There are no serviceable parts inside the instrument. Do not install substitute parts or perform any unauthorized modification to the product. Opening the instrument immediately voids the warranty provided by Zurich Instruments.

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

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

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

Table 2.2: Safety Instructions

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

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

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

Table 2.3: Unusual Conditions

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

Table 2.4: Symbols

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

2.4. Software Installation

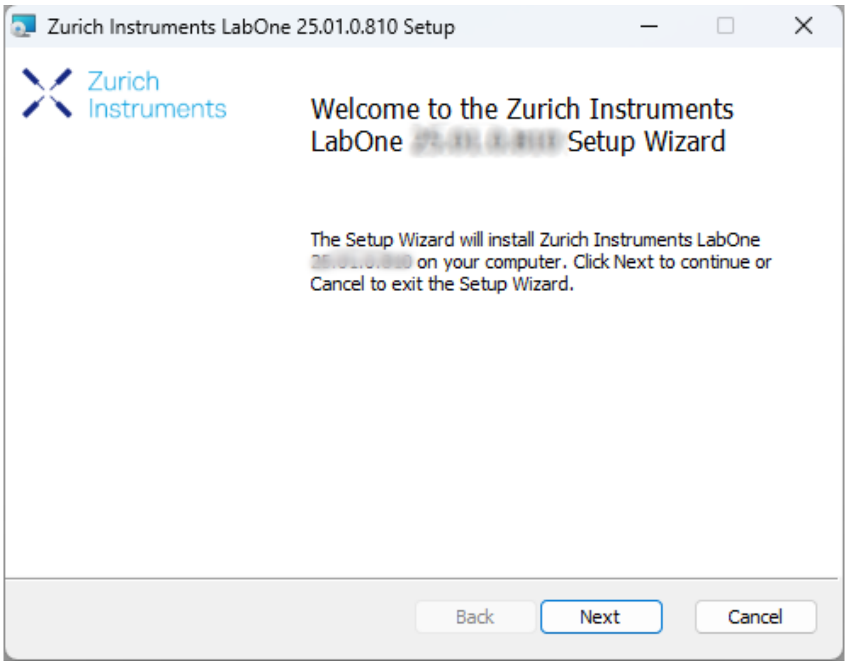
The PQSC Instrument is operated from a host computer with the LabOne software. To install the LabOne software on a computer, administrator rights may be required. In order to simply run the software later, a regular user account is sufficient. Instructions for downloading the correct version of the software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

2.4.1. Installing LabOne on Windows

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

2.4.2. Windows LabOne Installation

1. The 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 `LabOne64-XX.XX.XXXXX.msi` by a double click and follow the instructions. Windows Administrator rights are required for installation. The installation proceeds as follows:
   - On the welcome screen click the **Next** button.

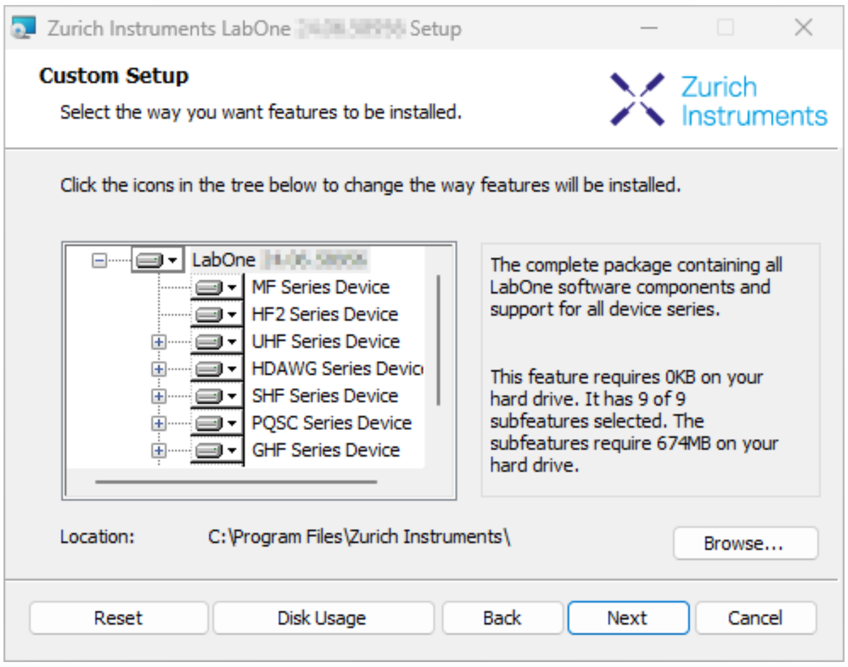

Zurich Instruments LabOne 25.01.0.810 Setup

Welcome to the Zurich Instruments LabOne [illegible] Setup Wizard

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

Figure 2.1: Installation welcome screen

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


Zurich Instruments LabOne [illegible] Setup

Custom Setup

Select the way you want features to be installed.

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

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

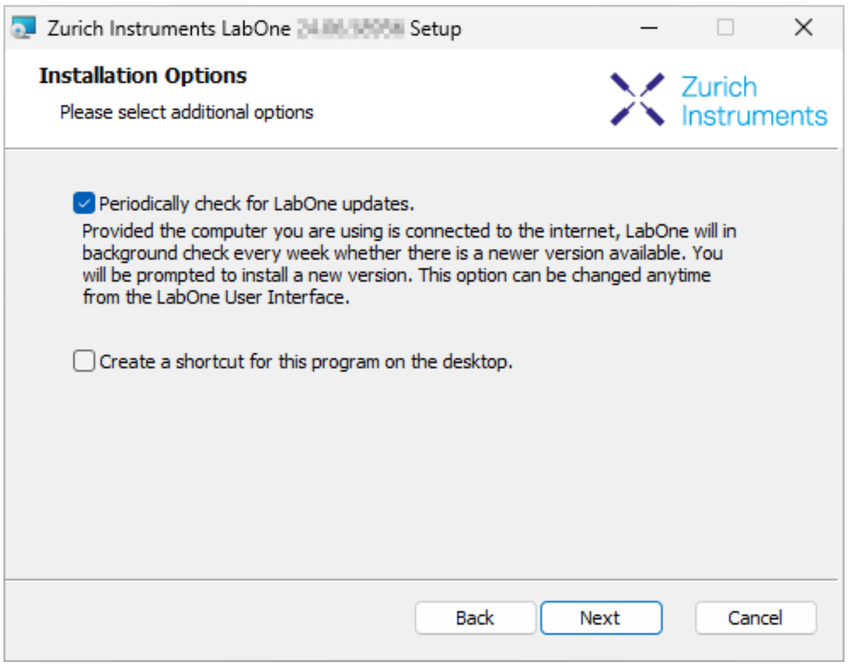
The complete package containing all LabOne software components and support for all device series.

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

Location: C:\Program Files\Zurich Instruments\

Figure 2.2: Custom setup screen

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


Zurich Instruments LabOne Setup

Installation Options

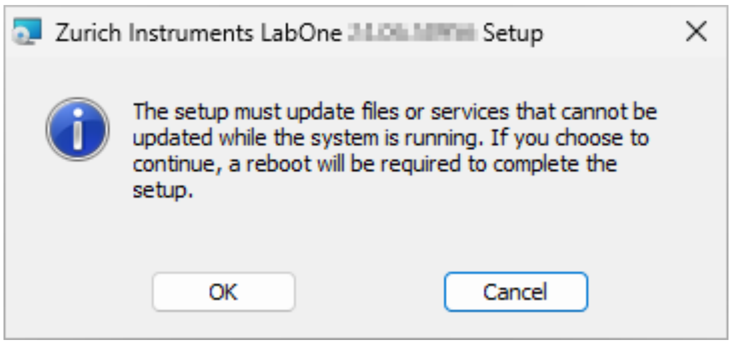
Please select additional options

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

[ ] Create a shortcut for this program on the desktop.

Figure 2.3: Automatic update check

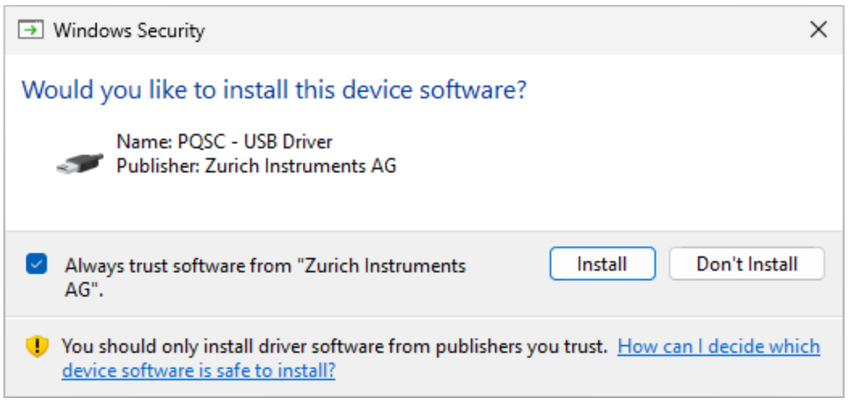
- Click the **Install** button to start the installation process.
- Windows may ask up to two times to reboot the computer if you are upgrading. Make sure you have no unsaved work on your computer.


Zurich Instruments LabOne Setup

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

Figure 2.4: Installation reboot request

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


Windows Security

Would you like to install this device software?

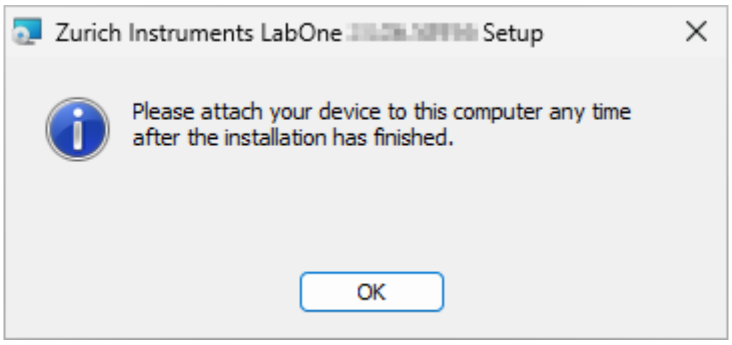
Name: PQSC - USB Driver
Publisher: Zurich Instruments AG

[✓] Always trust software from "Zurich Instruments AG".

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

Figure 2.5: Installation driver acceptance

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


Zurich Instruments LabOne [redacted] Setup

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

Figure 2.6: Installation completion screen

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

Warning

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

2.4.3. Running LabOne manually from the Command Line

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

Running the Web Server from the Command Line

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

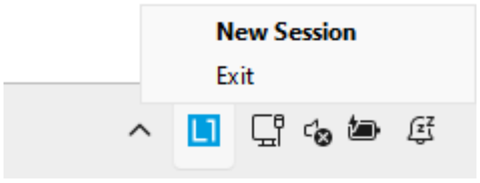


Figure 2.7: LabOne Tray Icon in Windows 11

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

```
$ ziWebServer --help
```

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

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

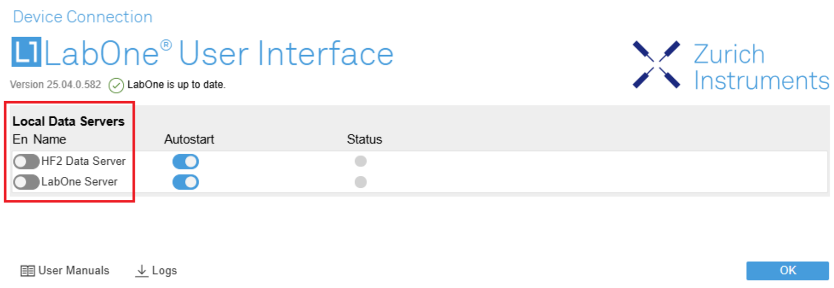
```
C:\Program Files\Zurich Instruments\LabOne\WebServer> ziWebServer --data-path "C:
\data"
```

Running the Data Server from the Command Line

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

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

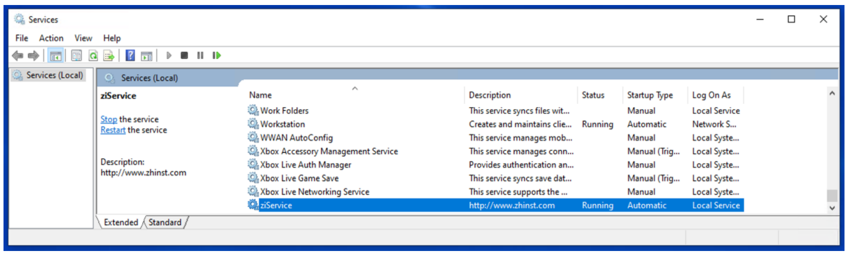
In the "Advanced" mode of LabOne Session Manager, press the "Configure" button to open the following window for switching on/off the data servers.



Local Data Servers

En	Name	Autostart	Status
	HF2 Data Server		
	LabOne Server		

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



ziService

Stop the service
Restart the service

Description:
http://www.zhinst.com

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

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

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

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

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

2.4.4. Windows LabOne Uninstallation

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

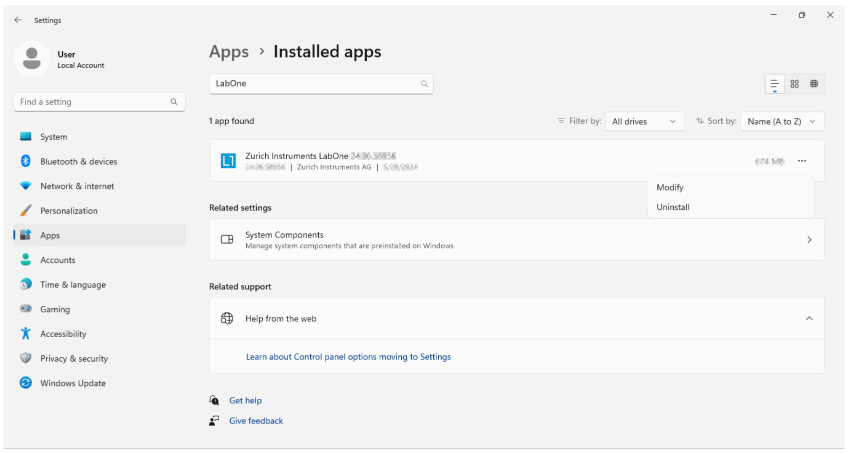
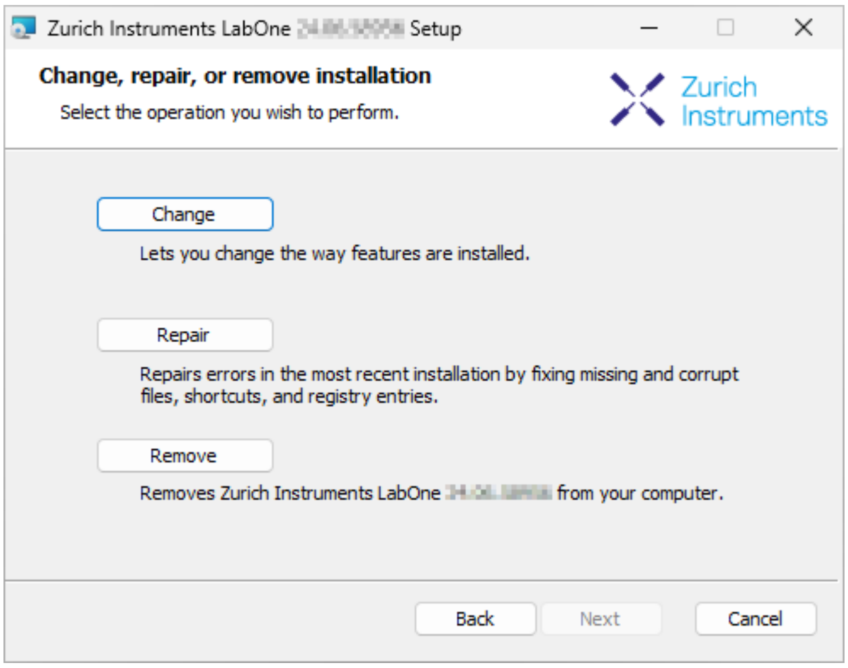


Figure 2.8: Uninstallation of LabOne on Windows computers

Warning

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

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


Zurich Instruments LabOne [redacted] Setup

Change, repair, or remove installation

Select the operation you wish to perform.

Change

Lets you change the way features are installed.

Repair

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

Remove

Removes Zurich Instruments LabOne [redacted] from your computer.

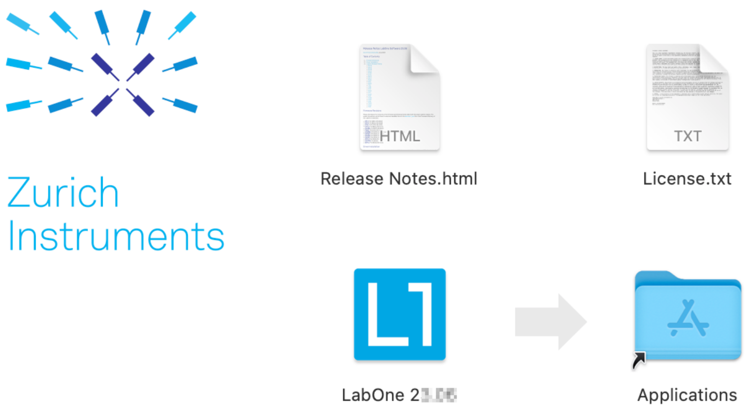
Figure 2.9: Repair of LabOne on Windows computers

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

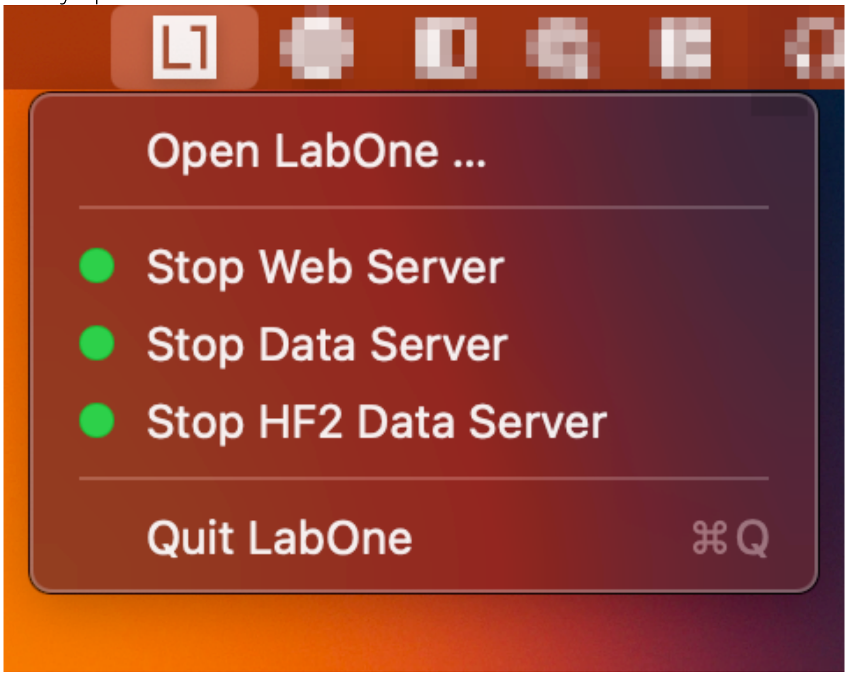
2.4.5. Installing LabOne on macOS

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

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



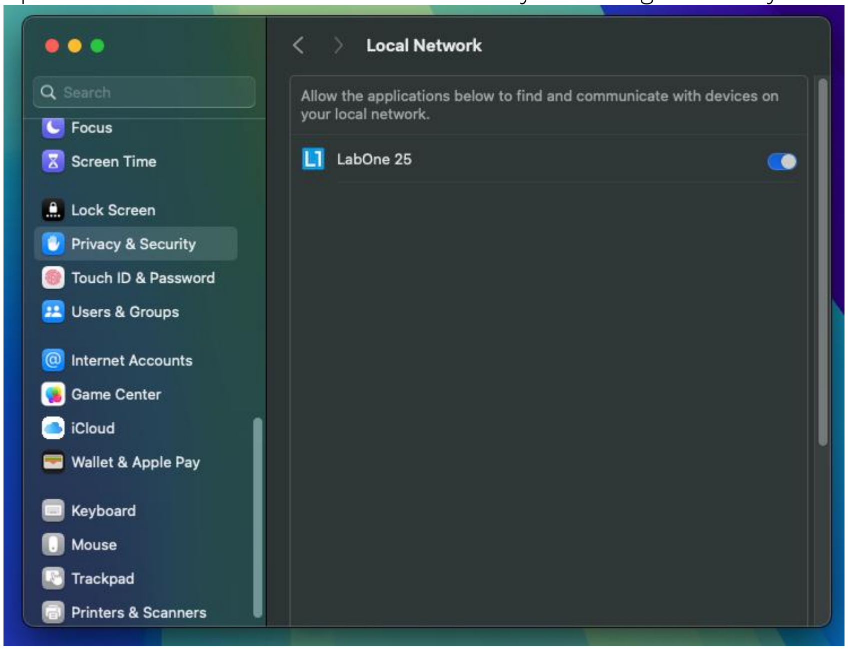
- The image contains a single LabOne application with all services needed.
- Once the application is started, a labone icon will appear in the menu bar. It allows the user to easily open a new session and shows the status of all services.



Note

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

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



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

LabOne 25

2.4.6. Uninstalling LabOne on macOS

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

2.4.7. Application Content

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

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

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

Note

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

2.4.8. Start LabOne Manually on the Command Line

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

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

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

Note

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

2.4.9. Installing LabOne on Linux

2.4.10. Requirements

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

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

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

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

```
uname -m
```

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

2.4.11. Linux LabOne Installation

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

1. Extract the LabOne tarball in a temporary directory:

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

2. Navigate into the extracted directory.

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

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

   ```
   sudo bash install.sh
   ```

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

2.4.12. Running the Software on Linux

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

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

   ```
   $ ziWebServer
   ```

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

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

3. You can now start the LabOne User Interface as described in LabOne Software Start-up and choose an instrument to connect to via the Device Connection dialog shown in Device Connection dialog.

Important

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

2.4.13. Uninstalling LabOne on Linux

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

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

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

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

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

2.5. Connecting to the Instrument

The Zurich Instruments 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

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

2.5.1. LabOne Software Architecture

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

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

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

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

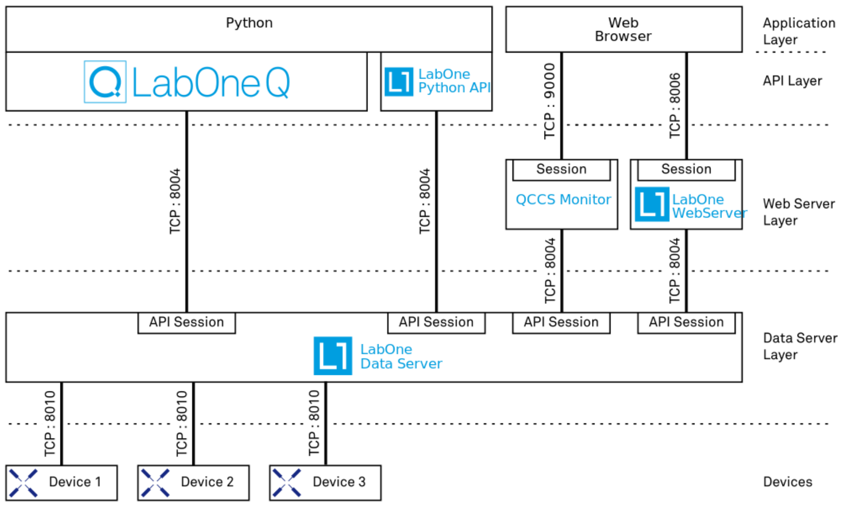


Figure 2.10: LabOne Software architecture

LabOne Data Server

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

LabOne Web Server

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

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

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

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

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

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

LabOne API Layer

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

- MATLAB
- Python
- LabVIEW
- .NET
- C

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

2.5.2. LabOne Software Start-up

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

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

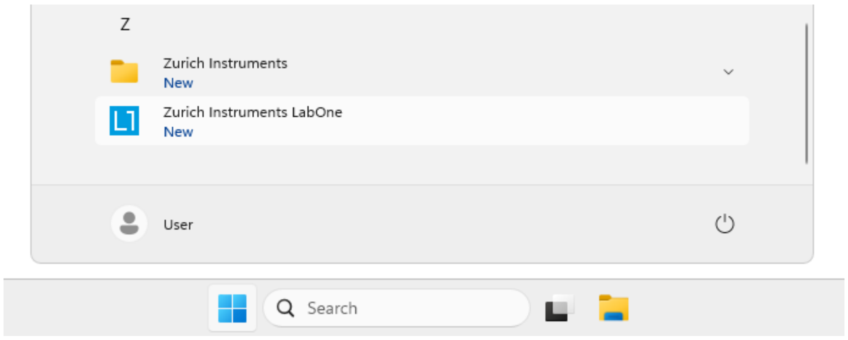


Figure 2.11: Link to the LabOne User Interface in the Windows 11 Start Menu

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

Note

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

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

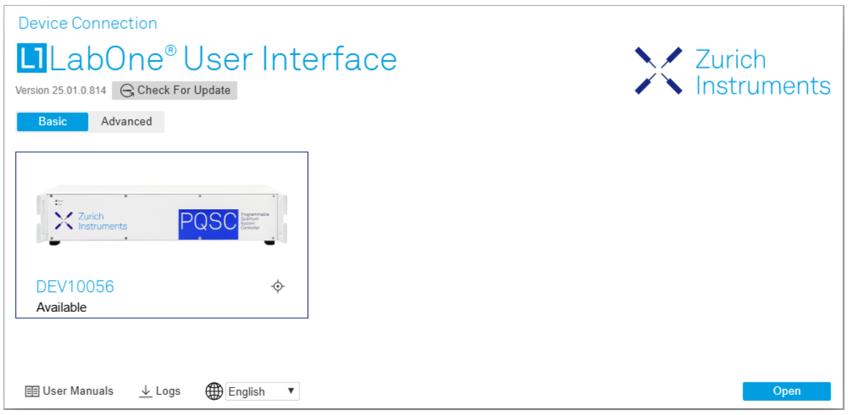
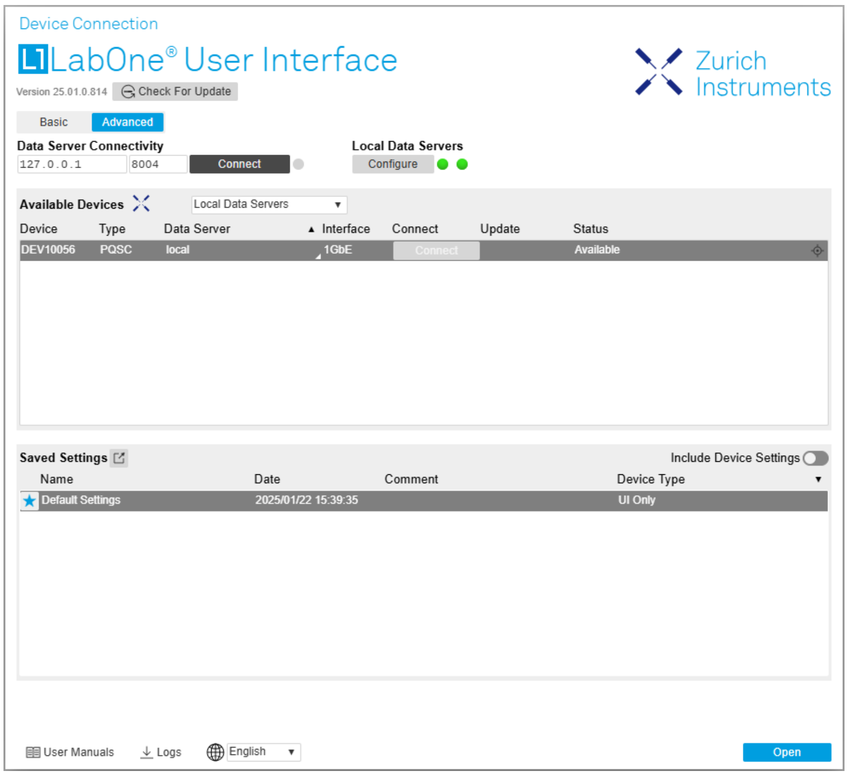


Figure 2.12: Device Connection dialog

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

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



Device	Type	Data Server	Interface	Connect	Update	Status
DEV10056	PQSC	local	1GbE	Connect		Available

Name	Date	Comment	Device Type
Default Settings	2025/01/22 15:39:35		UI Only

Figure 2.13: Device Connection dialog (Advanced view)

The Advanced view consists of three parts:

- Data Server Connectivity
- Available Devices
- Saved Settings

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

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

Note

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

Note

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

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

Data Server Connectivity

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

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

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

Note

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

Available Devices

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

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

If a firmware update of the instrument is available, the second to last column will show a button, and a simple click on it will update the instrument. The last column indicates the status of the device. Table 2.5 explains the meaning of some of the possible device statuses.

Table 2.5: Device Status Information

Connected	The device is connected to a LabOne Data Server, either on the same PC (indicated as local) or on a remote PC (indicated by its IP address). The user can start a session to work with that device.
Free	The device is not in use by any LabOne Data Server and can be connected by clicking the **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 Software Update.
Device not yet ready	The device is visible and starting up.

Saved Settings

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

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

Table 2.6: Column Descriptions

☆ ★	Allows favorite settings files to be grouped together. By activating the stars adjacent to a settings file and clicking on the column heading, the chosen files will be grouped together at the top or bottom of the list accordingly. The favorite marker is saved to the settings file. When the LabOne user interface is started next time, the row will be marked as favorite again.
Name	The name of the settings file. In the file system, the file name has the extension .md.
Date	The date and time the settings file was last written.
Comment	Allows a comment to be stored in the settings file. By clicking on the comment field a text can be typed in which is subsequently stored in the settings file. This comment is useful to describe the specific conditions of a measurement.
Device Type	The instrument type with which this settings file was saved.

Special Settings Files

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

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

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

Table 2.7: Button Descriptions

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

Note

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

Note

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

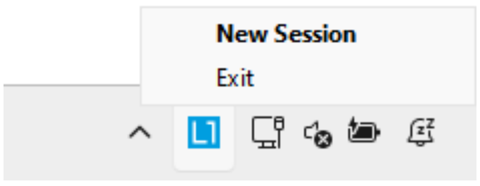
Note

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

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

Tray Icon

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

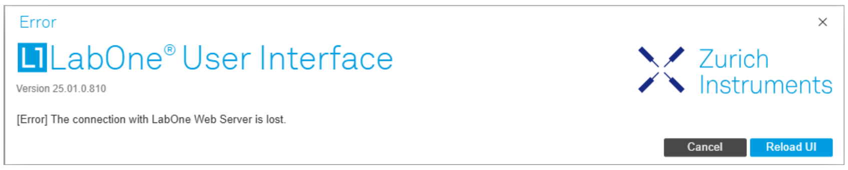


Messages

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

Lost Connection to the LabOne Web Server

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

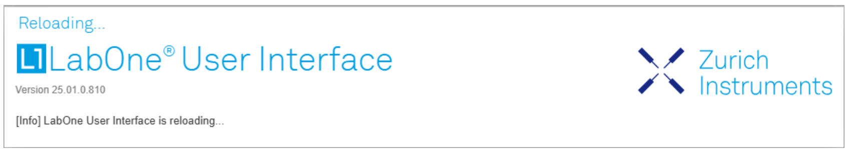


Version 25.01.0.810

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

Reloading...

If a session error cannot be handled, the LabOne Web Server will restart to show a new Device Connection dialog as shown in Figure 2.12. 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.

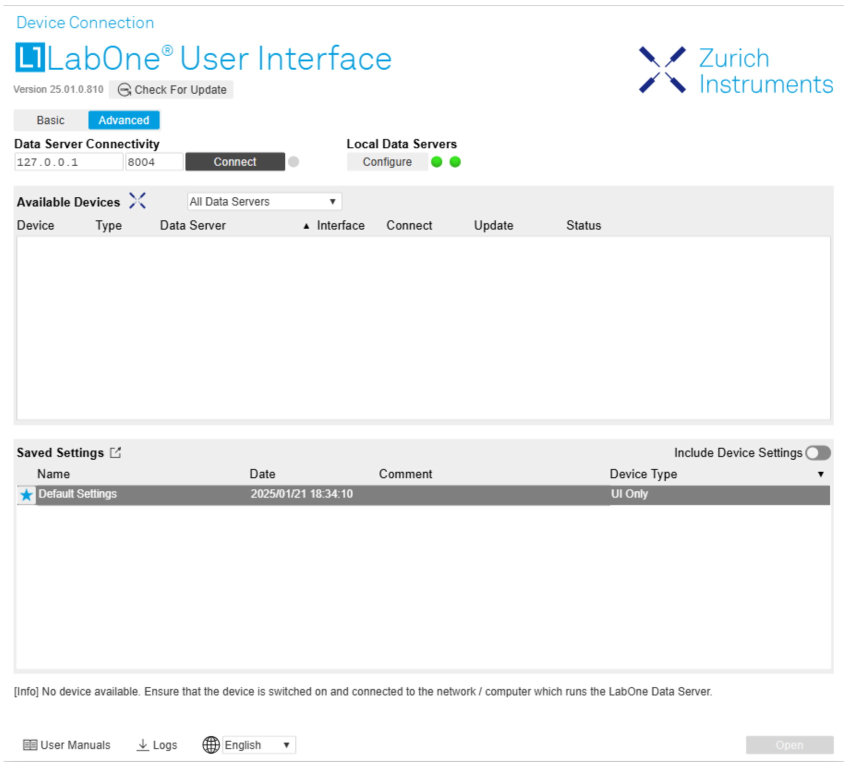


Version 25.01.0.810

[Info] LabOne User Interface is 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 Visibility and Connection. The figure below shows an example of this dialog.



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

No Device Available

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

2.5.3. Visibility and Connection

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

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

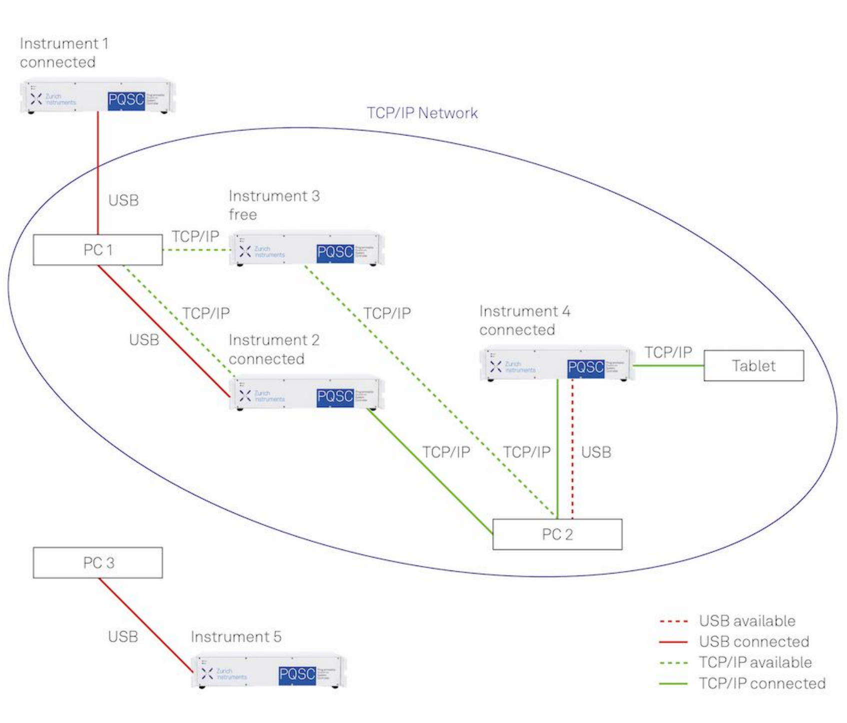


Figure 2.14: Connectivity

Figure 2.14 shows some examples of possible configurations of computer-to-instrument connectivity.

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

Visible Instruments

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

Connected Instrument

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

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

2.5.4. 1GbE Connectivity

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

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

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

Multicast DHCP

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

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

Requirements

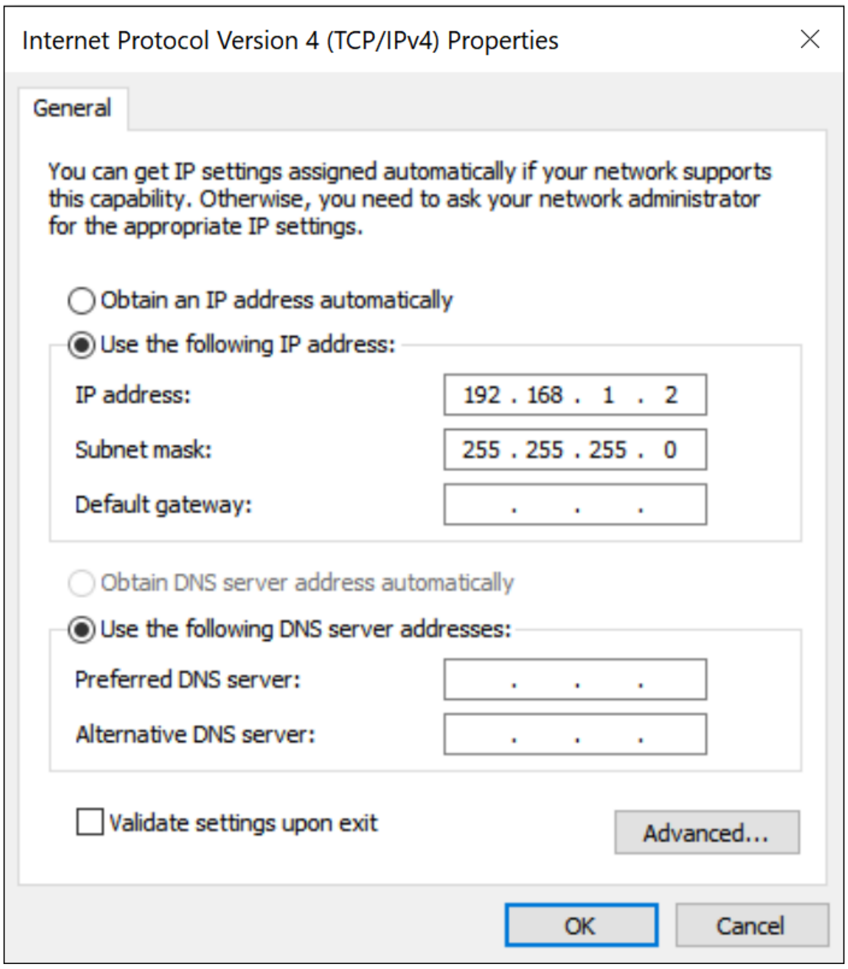
- Network supports multicast

Multicast Point-to-Point

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

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

1. Connect the 1GbE port of the network card that is dedicated for instrument connectivity directly to the 1GbE port of the instrument
2. Set this network card to static IP in TCP/IPv4 using the address `192.168.1.n`, where n=[2..9] and the mask `255.255.255.0`, see Static IP configuration for the host computer (go to `Control Panel → Internet Options → Network and Internet → Network and Sharing Center → Local Area Connection → Properties`).
3. Start up the LabOne User Interface normally. If your instrument does not show in the list of Available Devices, the reason may be that your network card does not support multicast. In that case use a static device IP as described in the following section.


Internet Protocol Version 4 (TCP/IPv4) Properties

General

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

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

IP address:	192 . 168 . 1 . 2
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	. . .

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

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

☐ Validate settings upon exit

Figure 2.15: 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

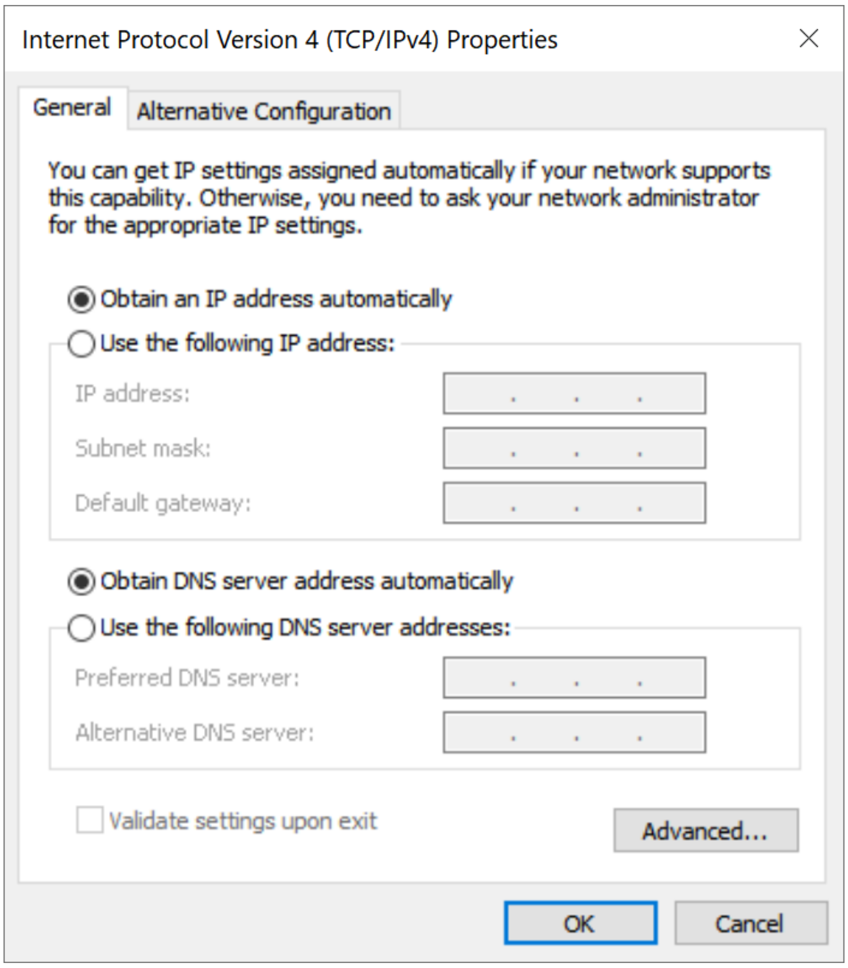
Only IP v4 is currently supported. There is no support for IP v6.

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.



Internet Protocol Version 4 (TCP/IPv4) Properties

General | Alternative Configuration

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

[✓ Obtain an IP address automatically]
Use the following IP address:

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

[✓ Obtain DNS server address automatically]
Use the following DNS server addresses:

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

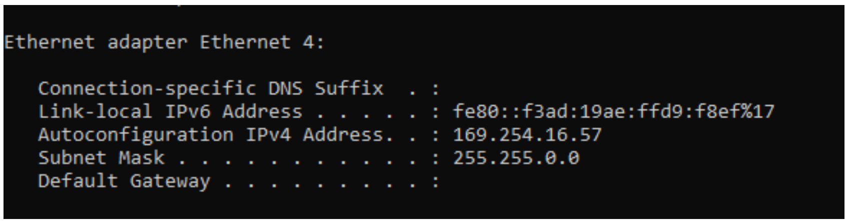
☐ Validate settings upon exit

Figure 2.16: Dynamic IP configuration for the host computer

Static Device IP

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

Note that the static IP assigned to the instrument must be within the same range of the IP assigned to the host computer. Whether the host computer's IP is assigned statically or by a fallback mechanism, one can find this IP by running the command `ipconfig` or `ipconfig/all` in the operating system's terminal. As an example, Figure 2.17 shows the outcome of running `ipconfig` in the terminal.



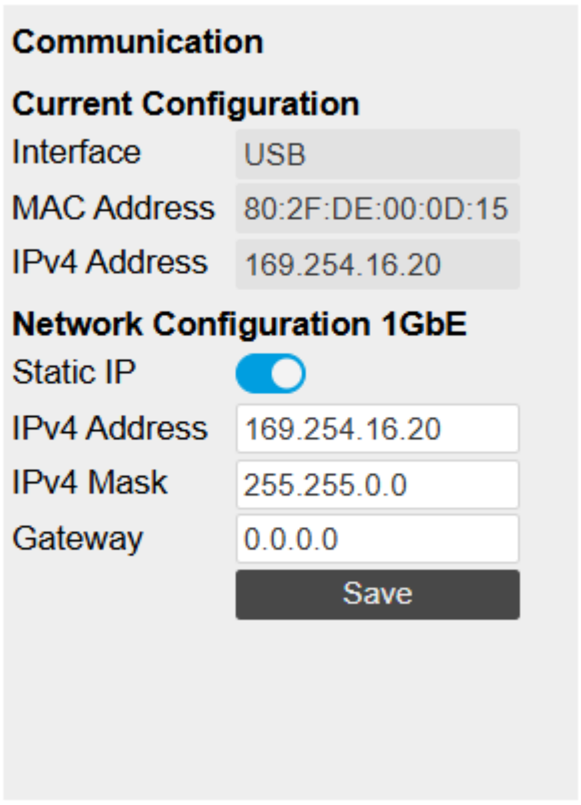
```
Ethernet adapter Ethernet 4:

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

Figure 2.17: IP and subnet mask of host computer

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

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



Communication

Current Configuration

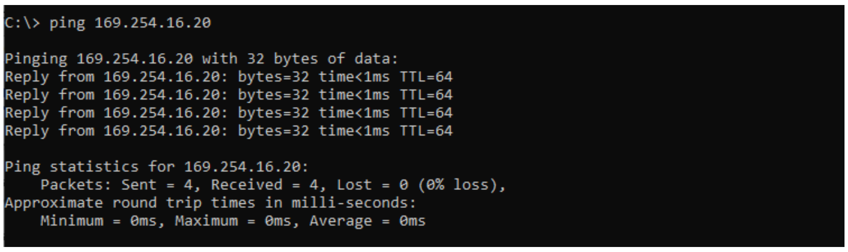
Interface	USB
MAC Address	80:2F:DE:00:0D:15
IPv4 Address	169.254.16.20

Network Configuration 1GbE

Static IP	●
IPv4 Address	169.254.16.20
IPv4 Mask	255.255.0.0
Gateway	0.0.0.0

Figure 2.18: Configuration of static IP in LabOne UI

To make sure the IP assignment is done properly, one can use the command `ping` to check if the instrument can be reached through the network using its IP address. Figure 2.19 shows the outcome of `ping` when the instrument is visible via the IP `169.254.16.20`.



```
C:\> ping 169.254.16.20

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

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

Figure 2.19: Instrument visible through pinging

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

Fallback Device IP

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

2.6. Software Update

2.6.1. Overview

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

2.6.2. Updating LabOne using Automatic Update Check

Updating the software is done in two steps. First, LabOne is updated on the PC by downloading and installing the LabOne software from the Zurich Instruments downloads page, as shown in Software Installation. Second, the instrument firmware needs to be updated from the Device Connection dialog after starting up LabOne. This is shown in Updating the Instrument Firmware . In case "Periodically check for updates" has been enabled during the LabOne installation and LabOne has access to the internet, a notification will appear on the Device Connection dialog whenever a new version of the software is available for download. This setting can later be changed in the Config tab of the LabOne user interface. In case automatic update check is disabled, the user can manually check for updates at any time by clicking on the button Check For Update in the Device Connection dialog. In case an update is found, clicking on the button "Update Available" shown in Figure 2.20 will start a download of the latest LabOne installer for Windows or Linux, see Figure 2.21. After download, proceed as explained in Software Installation to update LabOne.

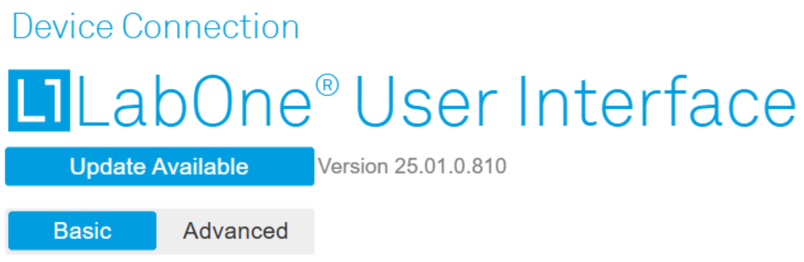
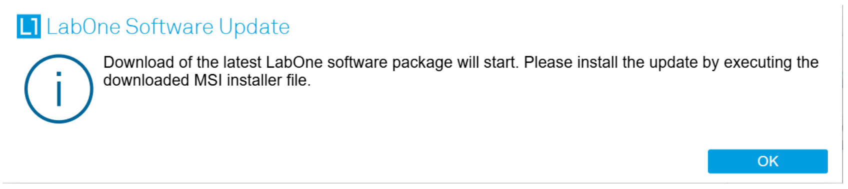


Figure 2.20: Device Connection dialog: LabOne update available



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

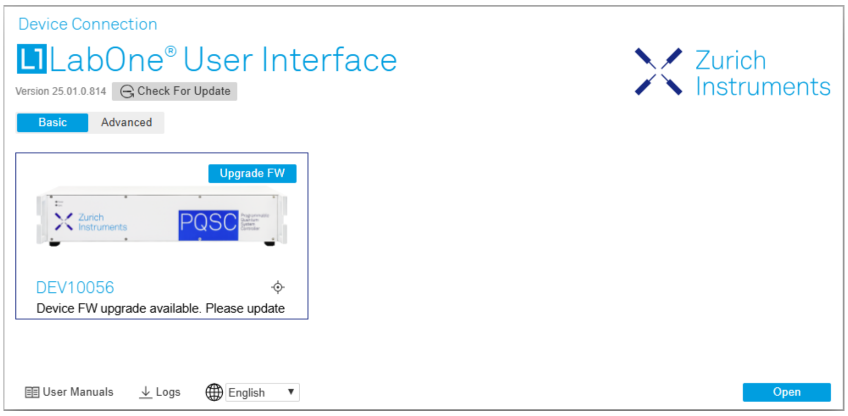
Figure 2.21: Download LabOne MSI using Automatic Update Check feature

2.6.3. Updating the Instrument Firmware

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

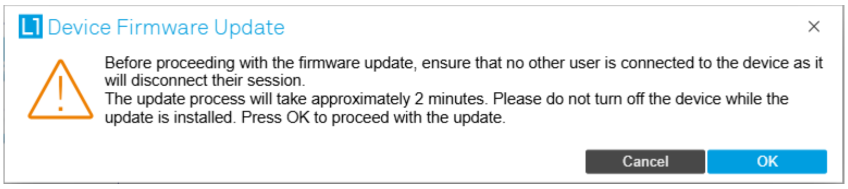
If the firmware needs an update, this is indicated in the Device Connection dialog of the LabOne user interface under Windows.

In the Basic view of the dialog, there will be a button "Upgrade FW" appearing together with the instrument icon as shown in Figure 2.22. In the Advanced view, there will be a link "Upgrade FW" in the Update column of the Available Devices table. Click on **Upgrade FW** to open the firmware update start-up dialog shown in Figure 2.23. The firmware upgrade takes approximately 2 minutes.



Device FW upgrade available. Please update

Figure 2.22: Device Connection dialog with available firmware update



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

Figure 2.23: Device Firmware Update start-up dialog

Important

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

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

2.7. Troubleshooting

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

2.7.1. Common Problems

Your PQSC Instrument is an advanced piece of laboratory equipment which has many features and capabilities. In order to benefit from these, the user needs access to a large number of settings in the API or the LabOne User Interface. The complexity of the settings might overwhelm a first-time user, and even expert users can get surprised by certain combinations of settings. This section provides an easy-to-follow checklist to solve the most common mishaps.

Table 2.8: Common Problems

Problem	Check item
The software cannot be installed or uninstalled	Please verify you have administrator/root rights.
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.

Problem	Check item
The LabOne User Interface does not start	Verify that the LabOne Data Server (**`ziDataServer.exe`**) and the LabOne Web Server (**`ziWebServer.exe`**) are running via the Windows Task Manager. The Data Server should be started automatically by **`ziService.exe`** and the Web Server should be started upon clicking "Zurich Instruments LabOne" in the Windows Start Menu. If both are running, but clicking the Start Menu does not open a new User Interface session in a new tab of your default browser then try to create a new session manually by entering **`127.0.0.1:8006`** in the address bar of your browser.
The user interface does not start or starts but remains idle	Verify that the Data Server has been started and is running on your host computer.
The user interface is slow and the web browser process consumes a lot of CPU power	Make sure that the hardware acceleration is enabled for the web browser that is used for LabOne. For the Windows operating system, the hardware acceleration can be enabled in **`Control Panel → Display → Screen Resolution`**. Go to Advanced Settings and then Trouble Shoot. In case you use a NVIDIA graphics card, you have to use the NVIDIA control panel. Go to Manage 3D Settings, then Program Settings and select the program that you want to customize.

2.7.2. Location of the Log Files

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

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

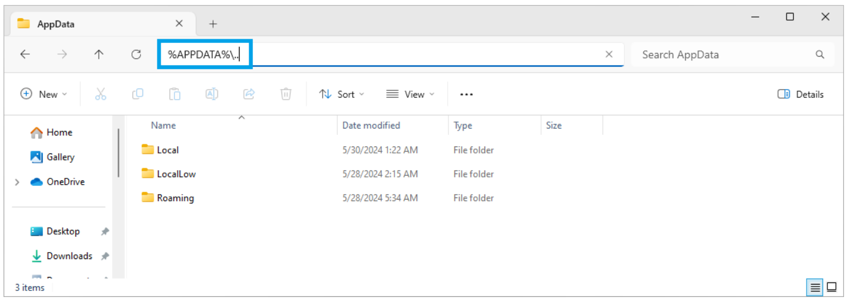
Windows

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

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

Note

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



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

Figure 2.24: Using the

Linux and macOS

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

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

2.7.3. Prevent web browsers from sleep mode

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

Edge

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

Chrome

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

Firefox

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

Opera

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

Safari

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

3. Functional Overview

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

3.1. Features

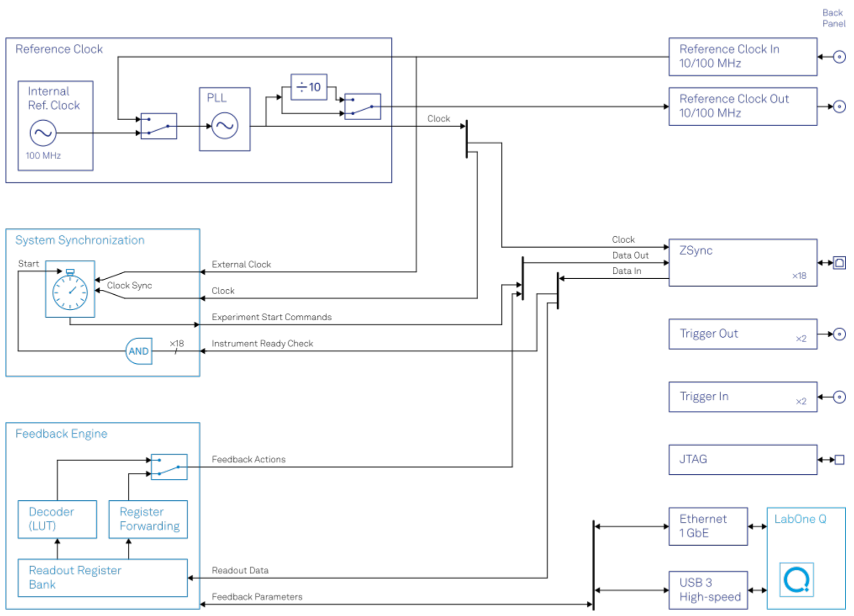


Figure 3.1: PQSC instrument functional diagram

The PQSC Instrument according to Figure 3.1 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 arrows between the panels and the interface units indicate selected physical connections and the data flow.

The PQSC comes with 18 ZSync ports to connect with the Zurich Instruments SHFQC, SHFSG and HDAWG for qubit control and with the Zurich Instruments SHFQC, SHFQA and 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 with sub-nanosecond precision. 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 connected 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.

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
- Device connection: 18 ZSync ports
- 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

3.2. Front Panel Tour

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

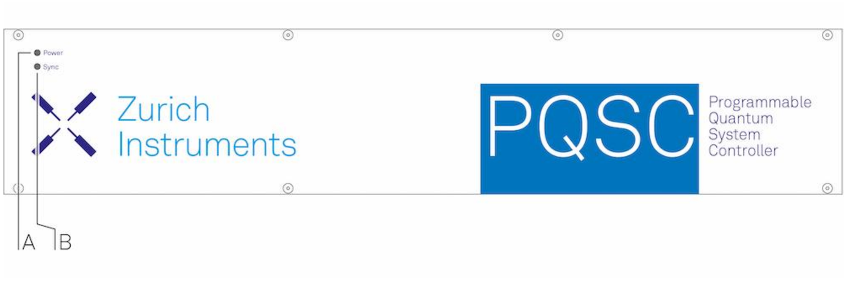


Figure 3.2: PQSC Programmable Quantum System Controller front panel

Table 3.1: PQSC Instrument front panel description

Position	Label / Name	Description
A	Power	device status LED off Instrument off and disconnected from mains power blue the instrument is ready to connect or has an active connection over USB or Ethernet steady glow, yellow the instrument is starting blinking, red the instrument is starting or is ready to be switched off
B	Sync	Unused. It might flash during instrument start

3.3. Back Panel Tour

The back panel is the main interface for power, control, service and connectivity to other ZI instruments. Please refer to Figure 3.3 and Table 3.2 for the detailed description of the items.

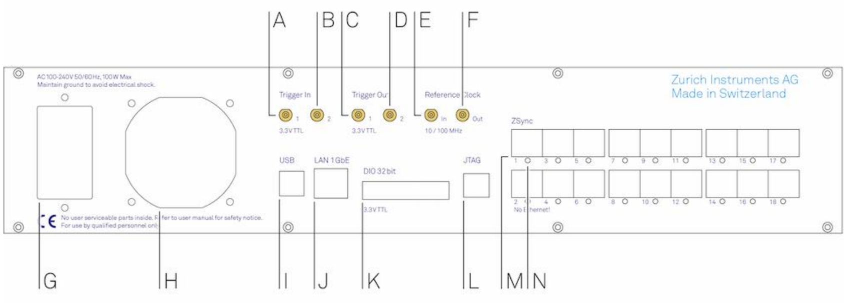

A
B
C
D
E
F
AC 100-240V 50/60Hz, 100W Max
Maintain ground to avoid electrical shock.
Trigger In
3.3V TTL
Trigger Out
3.3V TTL
Reference Clock
In
Out
10 / 100 MHz
Zurich Instruments AG
Made in Switzerland
ZSync
USB
LAN 1GbE
DIO 32 bit
3.3V TTL
JTAG
No Ethernet!
No user serviceable parts inside. Refer to user manual for safety notice.
For use by qualified personnel only
G
H
I
J
K
L
M
N

Figure 3.3: PQSC Instrument back panel

Table 3.2: PQSC Instrument back panel description

Position	Label / Name	Description
A	Trigger In 1	digital trigger input

Position	Label / Name	Description
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 V TTL	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 link off No instrument has been detected. blue Link with the instrument is established yellow Busy, establishing a connection red Error, connection not successfully established

4. 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 Software Installation. If you upgraded from a previous LabOne version, please be sure that all the devices run the same version of the firmware.

4.1. Synchronization of multiple HDAWGs

Note

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

4.1.1. 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

4.1.2. 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.

For best performance, the system should be placed in a climate-controlled environment with stable temperature and humidity. Avoid exposing the instruments to direct sunlight.

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.

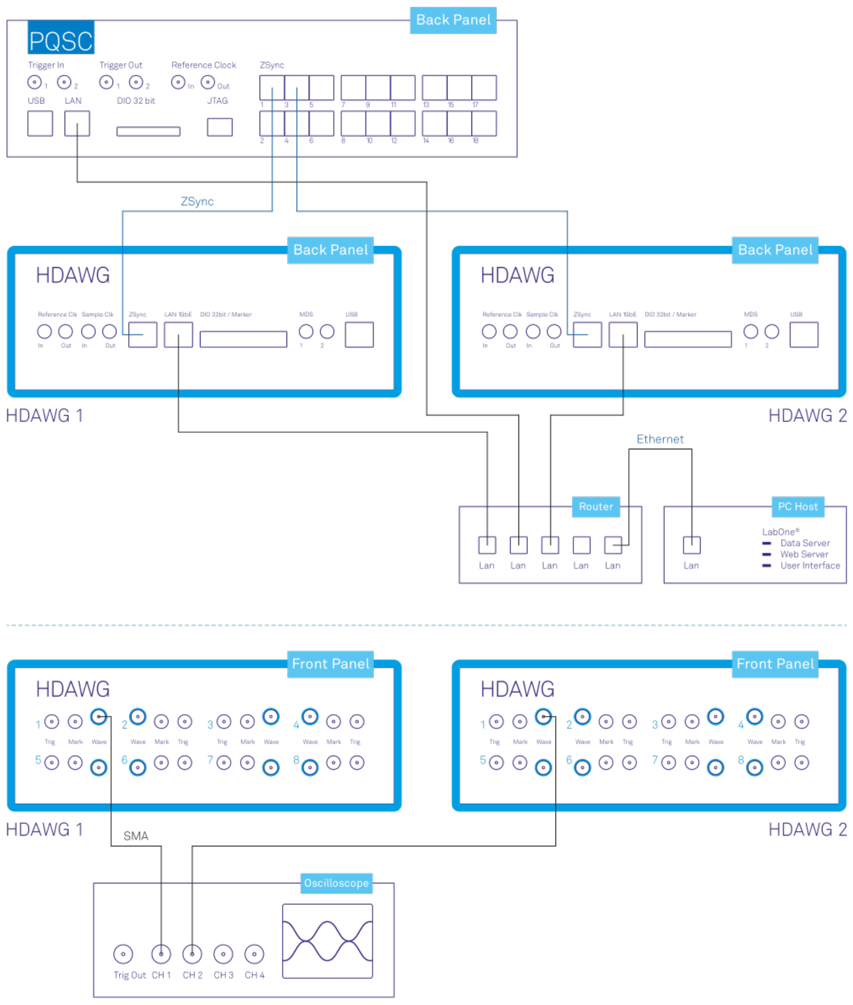


Figure 4.1: Connections for the multiple HDAWG synchronization tutorial

The tutorial can be started with the default instrument configuration (e.g. after a power cycle or after loading the Factory Default preset from the 'Device' tab) and the default user interface settings (e.g. as is after pressing F5 in the browser).

Note

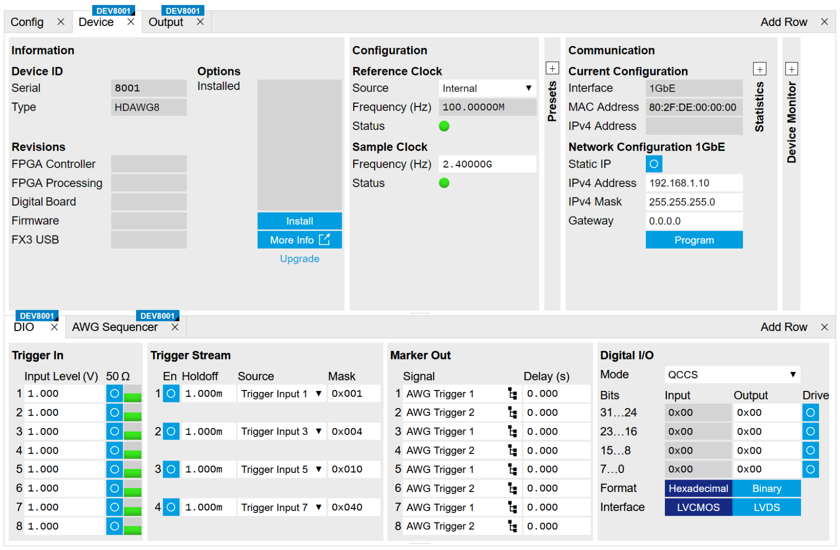
The PQSC needs to warm up for 30 minutes after power-up. Do not lock to external reference clock or start triggering before it’s ready. The CLK LED on the bottom right of the LabOne user interface will turn green when the instrument is ready to use.

4.1.3. 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 4.1: 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



Config | Device (DEV8001) | Output (DEV8001) | Add Row

Information

Device ID — Serial: 8001; Type: HDAWG8

Options — Installed

Revisions — FPGA Controller; FPGA Processing; Digital Board; Firmware; FX3 USB

Install | More Info | Upgrade

Configuration

Reference Clock — Source: Internal; Frequency (Hz): 100.00000M; Status: 🟢

Sample Clock — Frequency (Hz): 2.40000G; Status: 🟢

Communication

Current Configuration — Interface: 1GbE; MAC Address: 80:2F:DE:00:00:00; IPv4 Address

Network Configuration 1GbE — Static IP; IPv4 Address: 192.168.1.10; IPv4 Mask: 255.255.255.0; Gateway: 0.0.0.0

Program

Presets | Statistics | Device Monitor

DIO (DEV8001) | AWG Sequencer (DEV8001) | Add Row

Trigger In

	Input Level (V)	50 Ω
1	1.000	
2	1.000	
3	1.000	
4	1.000	
5	1.000	
6	1.000	
7	1.000	
8	1.000	

Trigger Stream

En	Holdoff	Source	Mask
1	1.000m	Trigger Input 1	0x001
2	1.000m	Trigger Input 3	0x004
3	1.000m	Trigger Input 5	0x010
4	1.000m	Trigger Input 7	0x040

Marker Out

	Signal	Delay (s)
1	AWG Trigger 1	0.000
2	AWG Trigger 2	0.000
3	AWG Trigger 1	0.000
4	AWG Trigger 2	0.000
5	AWG Trigger 1	0.000
6	AWG Trigger 2	0.000
7	AWG Trigger 1	0.000
8	AWG Trigger 2	0.000

Digital I/O

Mode: QCCS

Bits	Input	Output	Drive
31...24	0x00	0x00	
23...16	0x00	0x00	
15...8	0x00	0x00	
7...0	0x00	0x00	

Format: Hexadecimal | Binary

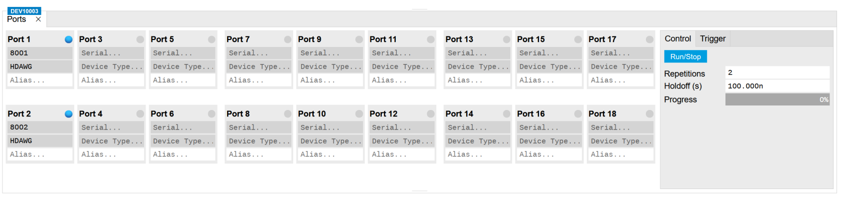
Interface: LVCMOS | LVDS

Figure 4.2: 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.

The sample rate of the HDAWGs is variable. When used with the PQSC, two values are supported: 2.0 GSa/s or 2.4 GSa/s. The sample rate 2.0 GSa/s is useful when the HDAWG is used in conjunction with instruments from the SHF series which operate at the same sample rate. It’s not possible to mix the two sample rates: all HDAWGs connected to a PQSC must the same sample rate.



Ports (DEV10003)

Port 1	Port 3	Port 5	Port 7	Port 9	Port 11	Port 13	Port 15	Port 17
8001	Serial...	Serial...	Serial...	Serial...	Serial...	Serial...	Serial...	Serial...
HDAWG	Device Type...	Device Type...	Device Type...	Device Type...	Device Type...	Device Type...	Device Type...	Device Type...
Alias...	Alias...	Alias...	Alias...	Alias...	Alias...	Alias...	Alias...	Alias...
Port 2	Port 4	Port 6	Port 8	Port 10	Port 12	Port 14	Port 16	Port 18
8002	Serial...	Serial...	Serial...	Serial...	Serial...	Serial...	Serial...	Serial...
HDAWG	Device Type...	Device Type...	Device Type...	Device Type...	Device Type...	Device Type...	Device Type...	Device Type...
Alias...	Alias...	Alias...	Alias...	Alias...	Alias...	Alias...	Alias...	Alias...

Control | Trigger

Run/Stop

Repetitions: 2

Holdoff (s): 100.000n

Progress: 0%

Figure 4.3: LabOne UI PQSC: Ports tab

4.1.4. 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 4.2: 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	1 V
Output		Wave Outputs	1	Enable	ON

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG Sequencer	Trigger	DIO Trigger	1	Strobe Slope	None
AWG Sequencer	Trigger	DIO Trigger	1	Valid Polarity	None

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 4.3: 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 4.4: 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 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.

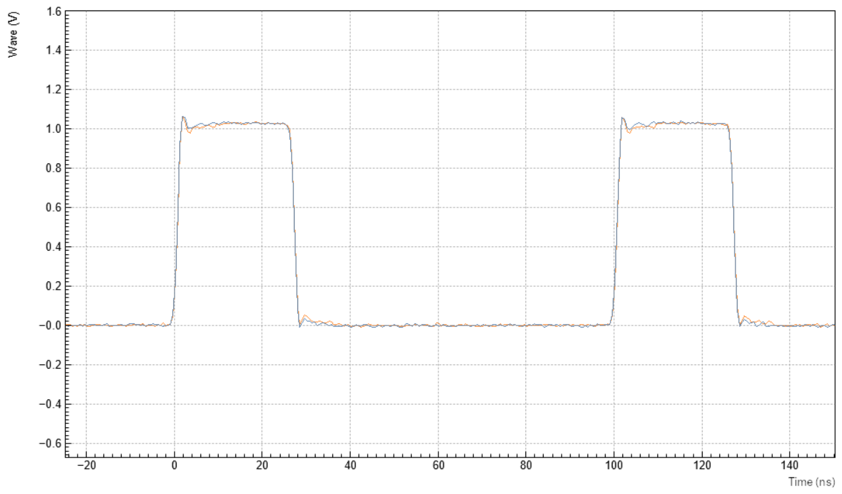


Figure 4.4: Pulses as generated by the two HDAWG and captured by the scope

4.2. Synchronization of HDAWGs and UHFQA

Note

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

4.2.1. 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 Synchronization of multiple HDAWGs. 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

4.2.2. 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.

For best performance, the system should be placed in a climate-controlled environment with stable temperature and humidity. Avoid exposing the instruments to direct sunlight.

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.

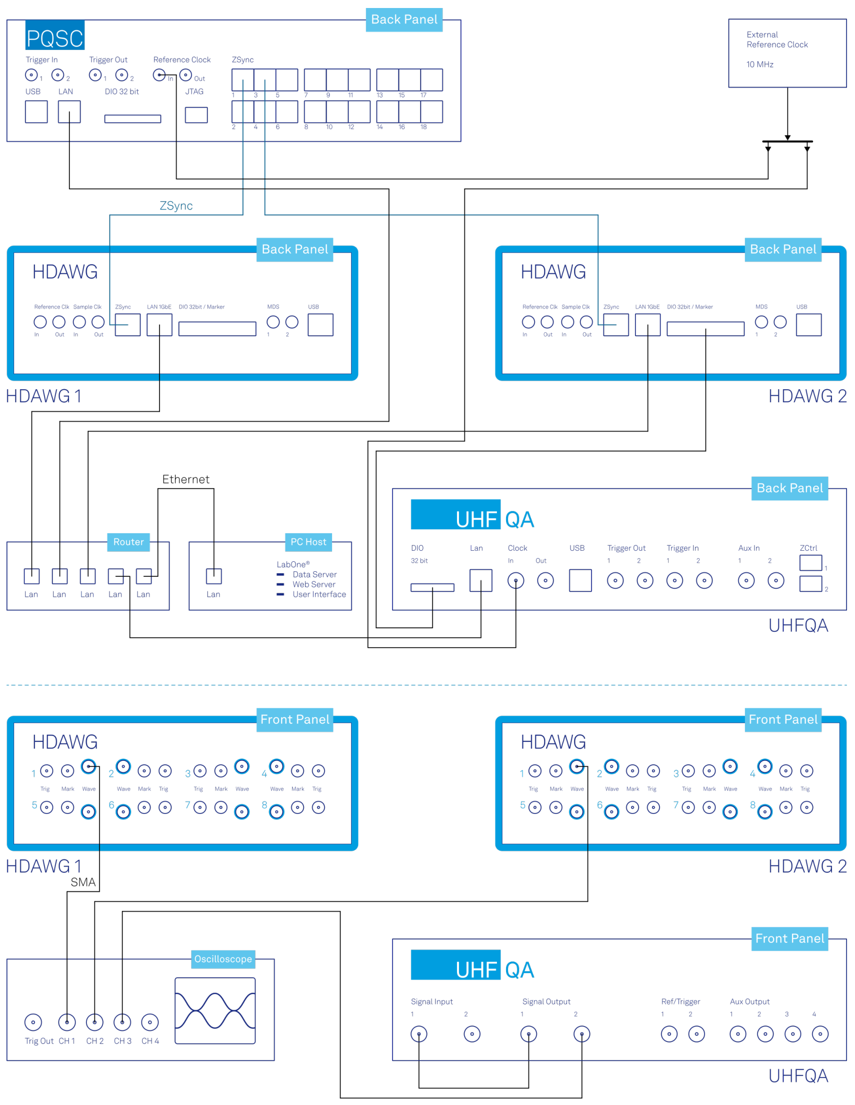


Figure 4.5: Connections for the multiple HDAWG/UHFQA synchronization tutorial

The tutorial can be started with the default instrument configuration (e.g. after a power cycle or after loading the Factory Default preset from the 'Device' tab) and the default user interface settings (e.g. as is after pressing F5 in the browser).

Note

The PQSC needs to warm-up for 30 minutes after power-up. Do not lock to external reference clock or start triggering before it's ready. The CLK LED on the bottom right of the LabOne interface will turn green when the instrument is warmed up and ready to use.

4.2.3. 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 4.5: 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 4.6: 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 4.7: 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.

Note

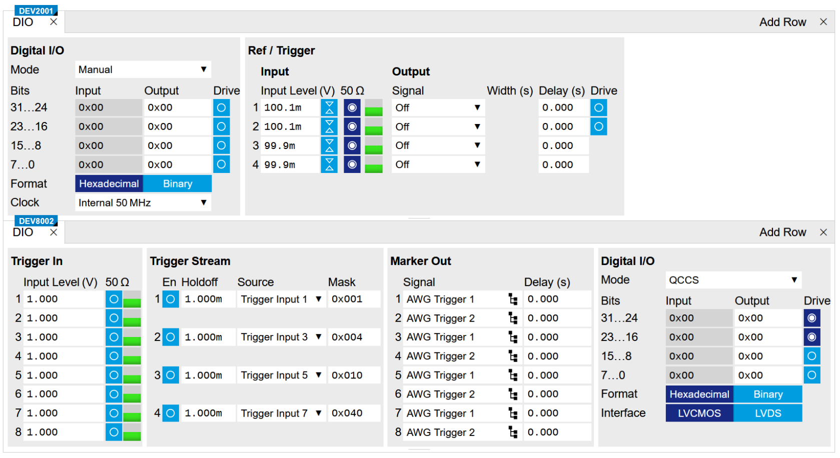
When an HDAWG is used as bridge for the UHFQA, its sample rate must be 2.4 GSa/s. As consequence, also the other HDAWGs in the system must use this sample rate, since the sample rates 2.0 GSa/s and 2.4 GSa/s can't be mixed.

Table 4.8: 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 4.9: Settings: configure the DIO interface on the HDAWG 2

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



DEV2001 DIO — Add Row

Digital I/O

Mode: Manual

Bits	Input	Output	Drive
31…24	0x00	0x00	○
23…16	0x00	0x00	○
15…8	0x00	0x00	○
7…0	0x00	0x00	○

Format: Hexadecimal | Binary

Clock: Internal 50 MHz

Ref / Trigger

	Input Level (V)	50 Ω	Output Signal	Width (s)	Delay (s)	Drive
1	100.1m	●	Off		0.000	○
2	100.1m	●	Off		0.000	○
3	99.9m	●	Off		0.000	
4	99.9m	●	Off		0.000	

DEV8002 DIO — Add Row

Trigger In

	Input Level (V)	50 Ω
1	1.000	○
2	1.000	○
3	1.000	○
4	1.000	○
5	1.000	○
6	1.000	○
7	1.000	○
8	1.000	○

Trigger Stream

	En	Holdoff	Source	Mask
1	○	1.000m	Trigger Input 1	0x001
2	○	1.000m	Trigger Input 3	0x004
3	○	1.000m	Trigger Input 5	0x010
4	○	1.000m	Trigger Input 7	0x040

Marker Out

	Signal	Delay (s)
1	AWG Trigger 1	0.000
2	AWG Trigger 2	0.000
3	AWG Trigger 1	0.000
4	AWG Trigger 2	0.000
5	AWG Trigger 1	0.000
6	AWG Trigger 2	0.000
7	AWG Trigger 1	0.000
8	AWG Trigger 2	0.000

Digital I/O

Mode: QCCS

Bits	Input	Output	Drive
31…24	0x00	0x00	●
23…16	0x00	0x00	●
15…8	0x00	0x00	○
7…0	0x00	0x00	○

Format: Hexadecimal | Binary

Interface: LVCMOS | LVDS

Figure 4.6: LabOne UI HDAWG and UHFQA: DIO tabs

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

Table 4.10: Settings: configure the DIO triggers on the sequencer of the HDAWG 2

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG Sequencer	Trigger	DIO Trigger	1	Strobe Slope	None
AWG Sequencer	Trigger	DIO Trigger	1	Valid Polarity	None

Table 4.11: Settings: configure the DIO triggers on the sequencer of the UHFQA

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG	Trigger	DIO Trigger		Strobe Slope	None
AWG	Trigger	DIO Trigger		Valid Index	16
AWG	Trigger	DIO Trigger		Valid Polarity	High

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

4.2.4. 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 4.12: 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	2 V
Output		Wave Outputs	1	On	ON

Table 4.13: 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

Tab	Sub-tab	Section	#	Label	Setting / Value / State
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 Synchronization of multiple HDAWGs. 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 4.14: 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)".

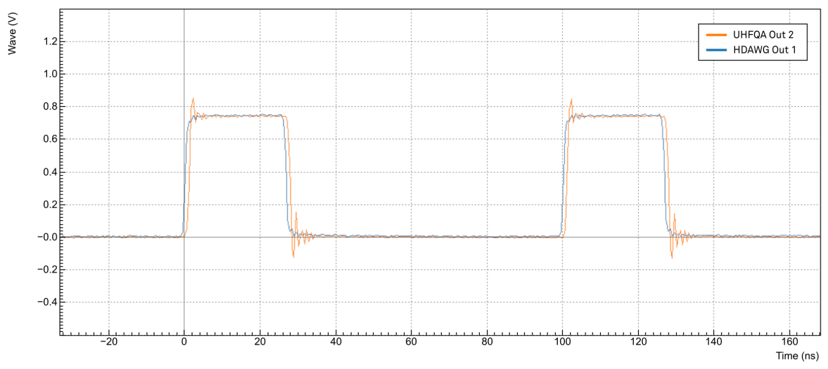
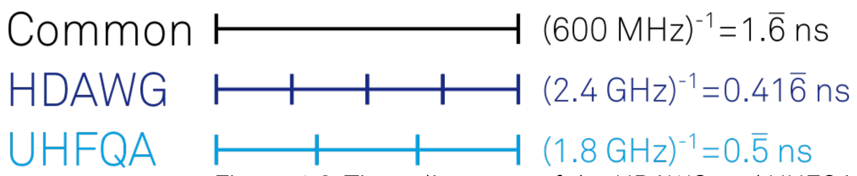


Figure 4.7: Pulses as generated by the HDAWG 2 and the UHFQA and captured by the scope

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


Common
$(600\text{ MHz})^{-1}=1.\overline{6}\text{ ns}$
HDAWG
$(2.4\text{ GHz})^{-1}=0.41\overline{6}\text{ ns}$
UHFQA
$(1.8\text{ GHz})^{-1}=0.\overline{5}\text{ ns}$
Figure 4.8: 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);
}
```

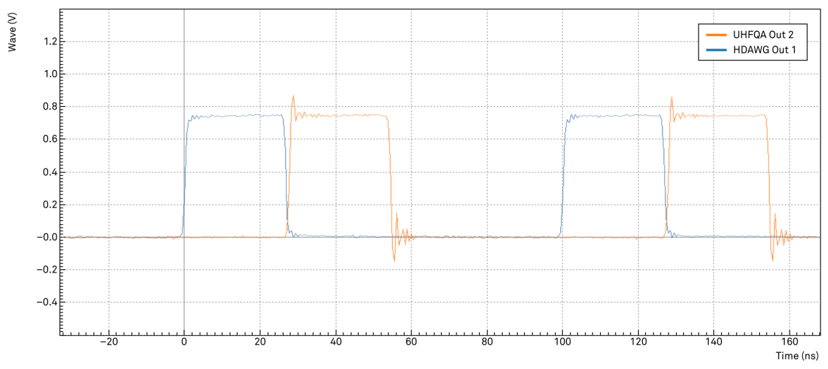


Figure 4.9: Shifted pulses with

4.2.5. 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. The holdoff time is increased, so the PQSC will be running during the play of all the waveforms. Configure the PQSC as follows:

Table 4.15: Settings: configure the start trigger generation on the PQSC

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Ports	Control			Repetitions	1
Ports	Control			Holdoff (s)	1m

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

```
// Constants declaration
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*50;
const READOUT_FREQ = 50e6;
const AVERAGES = 1024;

// Waveform declaration
wave w_sine = sine(READOUT_LEN, 0, READOUT_FREQ*READOUT_LEN/DEVICE_SAMPLE_RATE);

// Execution
var i;
while(true) {
  waitZSyncTrigger();
  for (i = 0; i < AVERAGES; i++) {
    playZero(DRIVE_PULSE_LEN);

    playWave(1, w_sine, 2, w_sine);
    // Trigger the readout
    startQA(QA_INT_ALL, true);

    playZero(PAD_LEN);
  }
}
```

And for the HDAWG:

```
// Constants declaration
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*50;
const DRIVE_FREQ = 150e6;
const AVERAGES = 1024;

// Waveform declaration
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);

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

    playZero(READOUT_LEN);
    // Trigger the readout (on the UHFQA)
    //startQA(QA_INT_ALL, true);

    playZero(PAD_LEN);
  }
}
```

The UHFQA sequence generates a trigger signal to start the QA Weighted Integration unit and the QA Input Monitor. While the command to generate 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. In order to trigger correctly the readout, we have to configure the integration unit in accordance with the generated signal:

Table 4.16: Settings: configure the UHFQA for the readout

Tab	Sub-tab	Section	#	Label	Setting / Value / State
QA Setup		Integration		Mode	Standard
QA Setup		Integration		Length	240
QA Setup		Integration		Trigger Signal	AWG Integration Trigger

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

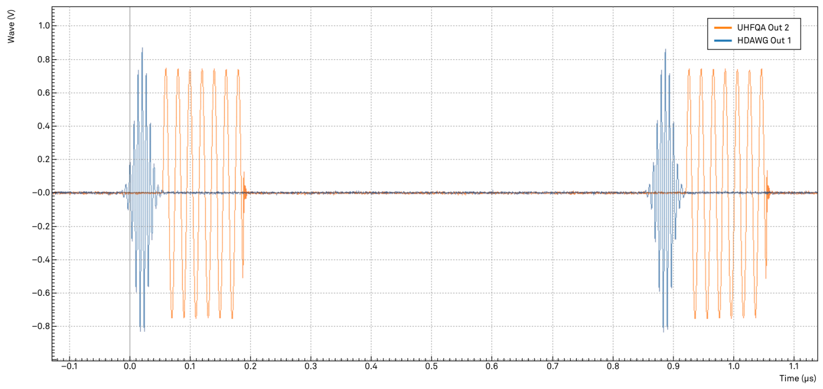


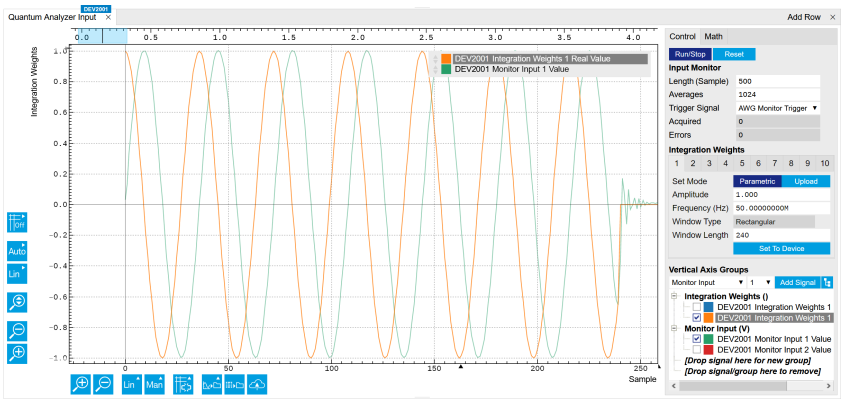
Figure 4.10: Control pulse and readout pulse generated by the HDAWG and the UHFQA

To acquire the time trace of the signal on the channel 1 we have to configure the UHFQA as follows:

Table 4.17: Settings: configure the UHFQA Input Monitor

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		Delay (Sample)	192
QA Input	Control	Input Monitor		Length (Sample)	500
QA Input	Control	Input Monitor		Averages	1024
QA Input	Control			Run/Stop	click
QA Input	Control			Reset	click

When the PQSC trigger is sent again, we can observe that the QA Input Monitor gets triggered by the sequencer. It acquires and averages 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, the value given in this tutorial is for a very short loopback cable.



Control | Math

Run/Stop | Reset

Input Monitor

Length (Sample)	500
Averages	1024
Trigger Signal	AWG Monitor Trigger
Acquired	0
Errors	0

Integration Weights

1 2 3 4 5 6 7 8 9 10

Set Mode	Parametric / Upload
Amplitude	1.000
Frequency (Hz)	50.00000000M
Window Type	Rectangular
Window Length	240

Set To Device

Vertical Axis Groups

Monitor Input | 1 | Add Signal

- **Integration Weights ()**
  - ☐ DEV2001 Integration Weights 1
  - ✓ DEV2001 Integration Weights 1
- **Monitor Input (V)**
  - ✓ DEV2001 Monitor Input 1 Value
  - ☐ DEV2001 Monitor Input 2 Value
- *[Drop signal here for new group]*
- *[Drop signal/group here to remove]*

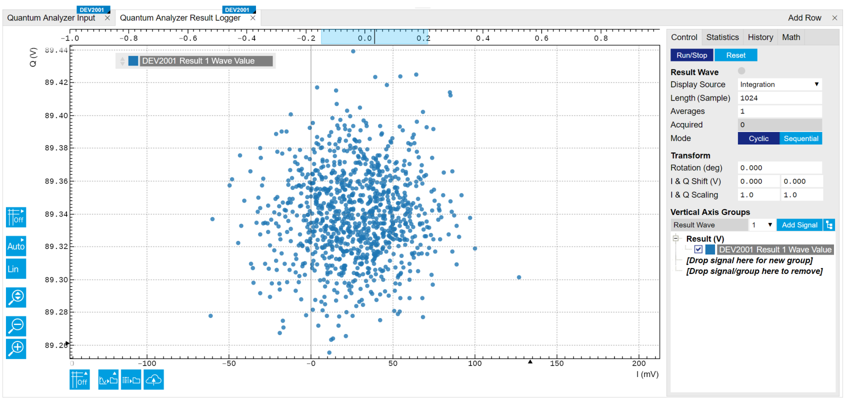
Figure 4.11: Signal as captured by the QA Monitor Input

The Monitor is good to visualize the pulses, but for the real acquisition we need to configure the integration unit to demodulate our signal. We will demodulate the signal only on channel 1, like in a typical lock-in measurement. Configure the QA as follows and start the PQSC trigger again.

Table 4.18: Settings: configure the UHFQA for the weighted integration readout

Tab	Sub-tab	Section	#	Label	Setting / Value / State
QA Setup		Integration	1	Signal Input Mapping	1→ Real, 1→ Imag
QA Input	Control	Weights / Generate		Amplitude	1
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 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
QA Result	Control			Reset	click

All 1024 results are identical up to the noise, as is evident from the scatter plot in the complex plane shown in Figure 4.12.



Control | Statistics | History | Math

Run/Stop | Reset

Result Wave

Display Source	Integration	
Length (Sample)	1024	
Averages	1	
Acquired	0	
Mode	Cyclic / Sequential	

Transform

Rotation (deg)	0.000	
I & Q Shift (V)	0.000	0.000
I & Q Scaling	1.0	1.0

Vertical Axis Groups

Result Wave | 1 | Add Signal

- **Result (V)**
  - ✓ DEV2001 Result 1 Wave Value
- *[Drop signal here for new group]*
- *[Drop signal/group here to remove]*

Figure 4.12: Signal as displayed in the QA Results tab

In this case, the acquired value is approximately $0.021 + 89.34i$, which is compatible with the expected value $0 + 90i = A/2 \cdot N$, where A is the peak amplitude of the acquired signal (750 mV) and N is the number of acquired points (240). The signal is all in quadrature because we are generating a signal all in quadrature (sine instead of cosine). For more details, please refer to the UHFQA User Manual

5. 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.

5.1. User Interface Overview

5.1.1. 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.

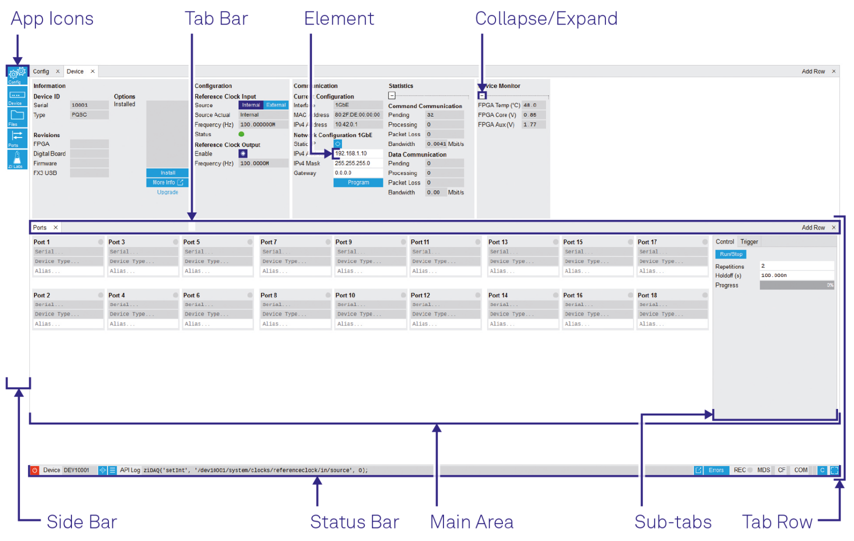


Figure 5.1: LabOne User Interface (default view)

The LabOne User Interface automatically opens some tabs by default after a new UI session has been started. At start-up, the UI is 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 moved by drag-and-drop within a row or across rows.

Table 5.1 gives a brief descriptions and naming conventions for the most important UI items.

Table 5.1: 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 tabs - a click on an icon adds or activates the corresponding tab in the active tab row	app icons

Item name	Position	Description	Contains
status bar	bottom of the UI	contains important status and warning indicators, 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

5.1.2. 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 may depend on the upgrade options installed on a given instrument.

Table 5.2: 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 ZSync ports.
Feedback		Control and display of the feedback pipeline.
Device		Provides instrument specific settings.
ZI Labs		Experimental settings and controls.

Table 5.3 provides a quick overview over the different status bar elements along with a short description.

Table 5.3: 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
Shutdown		Shuts down the instrument.
CLK	yellow/ green/red	State of internal clocks. Yellow: Device is warming up and will be ready 30 minutes after power-up. Do not lock to external reference clock or start triggering yet. Green: Device is warmed up and ready to use. Red: Clocks have become unstable. Expect potential phase errors. This might be caused by temperature changes.
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.

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

5.2. Ports Tab

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

5.2.1. Features

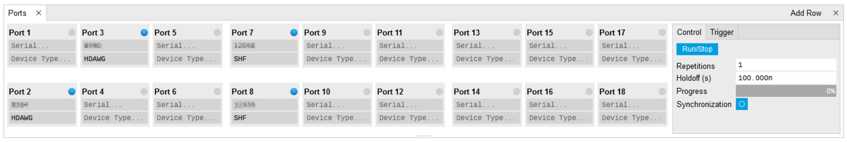
- Display the state of the complete system at first glance
- Display port synchronization of all 18 ZSync ports
- Provide user-given names to connected instruments
- Control of triggers that are sent to the connected instruments

5.2.2. Description

Table 5.4: App icon and short description

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

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



Repetitions 1
Holdoff (s) 100.000n
Progress 0%

Figure 5.2: 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.

Feedback processing is active only when trigger generation is enabled. This means that even if a single trigger is configured, a sufficiently large holdoff time must also be set to ensure that feedback processing is active throughout the experiment.

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 Synchronization of multiple HDAWGs for more details.

5.2.3. Functional Elements

Table 5.5: Ports tab

Control/Tool	Option/ Range	Description
Connection Status	off/blue/ yellow/red	Indicates the availability of the instrument connected to 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.
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.
Synchronization		Enable synchronization. Trigger generation will only start once all workers have reported a ready status. Synchronization checks will be repeated with the same trigger generation settings (holdoff and repetitions) until synchronization is disabled.
Enable	ON / OFF	Enable Trigger Out connector.
Source		Select the source for the Trigger Out connector.
	Start Triggers	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.
Source 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.

5.3. Feedback Tab

The Feedback tab provides control and displays the flow of the feedback pipeline. It is available on all PQSC instruments.

5.3.1. Features

- Display the full feedback pipeline and how the elements are connected
- Control the register forwarding parameters

5.3.2. Description

Table 5.6: App icon and short description

Control/Tool	Option/Range	Description
Feedback		Control and display of the feedback pipeline.

The Feedback tab (see LabOne UI: Feedback tab) shows the flow of data from the input ZSync ports on the left, to the output ZSync ports on the right. The readout register forwarding (see Register Forwarding for more details) can be configured from the output tab.

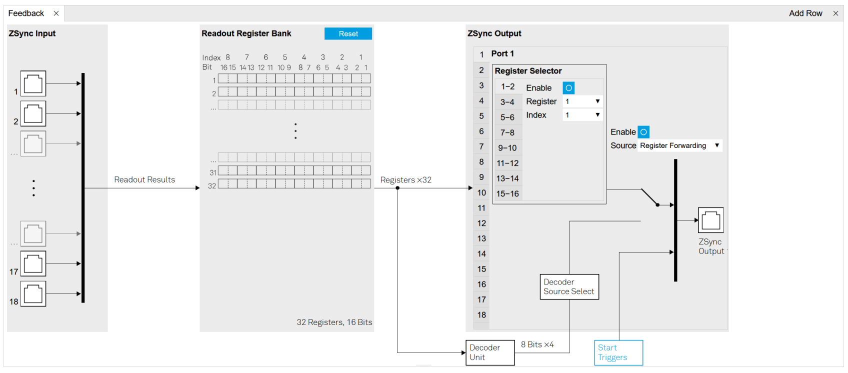


Figure 5.3: LabOne UI: Feedback tab

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 SHFQA, The QAChannel of the SHFQC or the UHFQA, while the feedback data are received by Signal Generators, like the SHFSG, the SGChannels of the SHFQC or the HDAWG. All the communications are done over ZSync.

The PQSC feedback architecture according to Figure 5.4 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.

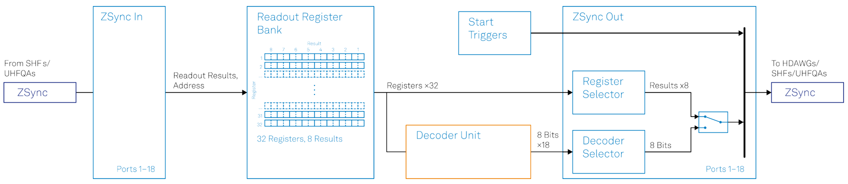


Figure 5.4: Block diagram of the PQSC feedback architecture

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 16 qubit readout results from one QA readout channel (10 qubit readout results for the UHFQA). 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 eight freely chosen readout results, corresponding to up to sixteen qubit readouts or eight qutrit/ququad readouts. Every port can be configured with its own set of forwarded results. The forwarded results are specified by the register number and index of the desired result inside of the register. Whenever one of the linked registers is updated, its content is automatically forwarded to the receiving instrument. This functionality is intended for feedback which requires minimal latency and depends on the state of very few qubits. The typical example is the active qubit/qutrit/ququad reset, an initialization technique that relies on fast readout and conditional reset. The forwarded results can be decoded on the receiving instrument with the function `getFeedback(ZSYNC_DATA_PROCESSED_A, ...)` or to immediately trigger a specific waveform play with `executeTableEntry(ZSYNC_DATA_PROCESSED_A, ...)`. Please refer to the user manual of the receiving instrument for more details.

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 decoder output can be read on the receiving instrument with the function `getFeedback(ZSYNC_DATA_PROCESSED_B)` or trigger immediately a specific waveform play with `executeTableEntry(ZSYNC_DATA_PROCESSED_B)`. Please refer to the user manual of the receiving instrument for more details. 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 5.5 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.

Note

The configuration of the Lookup table decoder is currently only accessible through the API.

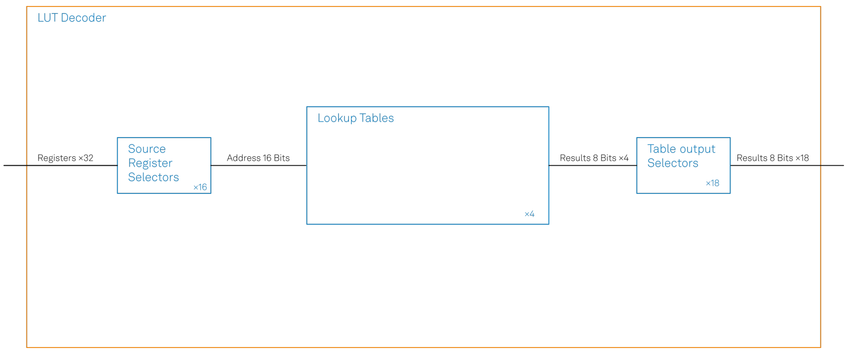


Figure 5.5: Block diagram of the Lookup Table decoder

5.3.3. Functional Elements

Table 5.7: Feedback tab

Control/Tool	Option/ Range	Description
Input Port		The input ZSync port where the QA instrument is connected.
Readout Register Bank		The Readout Register Bank where the results coming from the QAs are stored. The destination coordinates are controlled by the QA sequencer.
Reset		Clear all the readout registers.
Enable		Enable feedback output for the current port.
Source		Select the source of feedback data for the current port.
	Register Forwarding	Forward selected readout results to the given port.
	Decoder	Send the output of the Decoder unit to the given port.

Control/Tool	Option/ Range	Description
Output Port		The output ZSync port where triggers and feedback events are directed.
Decoder Unit		The quantum error decoder unit.
Start Triggers		The start synchronization triggers.
Decoder Source Select		The source of the error decoder that is forwarded.
Enable		Enable readout register forwarding of bits xx-yy to the current port.
Register		The readout register to be forwarded.
Index		The index of the results in the readout register to be forwarded.
Bits		The pair of bits in the forwarded message.

5.4. Config Tab

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

5.4.1. 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

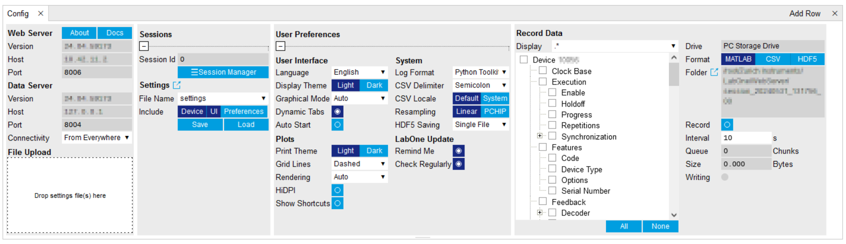
5.4.2. Description

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 5.8: App icon and short description

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

The Config tab (see LabOne UI: Config tab) is divided into four sections to control connections, sessions, settings, user interface appearance and data recording.



Web Server
Version [redacted]
Host [redacted]
Port 8006

Data Server
Version [redacted]
Host [redacted]
Port 8004
Connectivity From Everywhere

File Upload

Sessions
Session Id 0

Settings
File Name settings
Include Device UI Preferences

User Preferences

User Interface
Language English
Display Theme Light Dark
Graphical Mode Auto
Dynamic Tabs ◉
Auto Start ○

Plots
Print Theme Light Dark
Grid Lines Dashed
Rendering Auto
HiDPI ○
Show Shortcuts ○

System
Log Format Python Toolki
CSV Delimiter Semicolon
CSV Locale Default System
Resampling Linear PCHIP
HDF5 Saving Single File

LabOne Update
Remind Me ◉
Check Regularly ◉

Record Data
Display .*
Device [redacted]
Clock Base
Execution
Enable
Holdoff
Progress
Repetitions
Synchronization
Features
Code
Device Type
Options
Serial Number
Feedback
Decoder

Drive PC Storage Drive
Format MATLAB CSV HDF5
Folder [redacted]
Record ○
Interval 10 s
Queue 0 Chunks
Size 0.000 Bytes
Writing

Figure 5.6: LabOne UI: Config tab

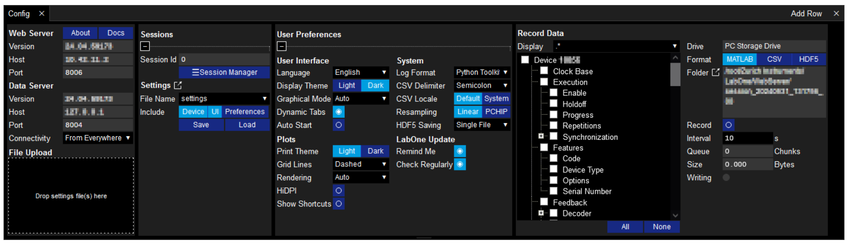
The **Connection** section provides information about connection and server versions. Access from remote locations can be restricted with the connectivity setting.

The **Session** section provides the session number which is also displayed in the 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 dialog.

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 5.7).



Config

Web Server — About, Docs
Version: [redacted]
Host: [redacted]
Port: 8006
Data Server
Version: [redacted]
Host: [redacted]
Port: 8004
Connectivity: From Everywhere
File Upload
Drop settings file(s) here

Sessions
Session Id: 0
Session Manager
Settings
File Name: settings
Include: Device, UI, Preferences
Save, Load

User Preferences
User Interface
Language: English
Display Theme: Light [✓ Dark]
Graphical Mode: Auto
Dynamic Tabs: ●
Auto Start: ○
Plots
Print Theme: [✓ Light] Dark
Grid Lines: Dashed
Rendering: Auto
HiDPI: ○
Show Shortcuts: ○
System
Log Format: Python Toolkit
CSV Delimiter: Semicolon
CSV Locale: [✓ Default] System
Resampling: [✓ Linear] PCHIP
HDF5 Saving: Single File
LabOne Update
Remind Me: ●
Check Regularly: ●

Record Data
Display: .*
Device [redacted]
Clock Base
Execution
Enable
Holdoff
Progress
Repetitions
Synchronization
Features
Code
Device Type
Options
Serial Number
Feedback
Decoder
All, None
Drive: PC Storage Drive
Format: [✓ MATLAB] CSV HDF5
Folder: [redacted]
Record: ○
Interval: 10 s
Queue: 0 Chunks
Size: 0.000 Bytes
Writing

Add Row

Figure 5.7: LabOne UI: Config tab - dark theme

5.4.3. Functional Elements

Table 5.9: Config tab

Control/ Tool	Option/Range	Description
About	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. Red: error while connecting.
Connectivity	From Everywhere	Forbid/Allow to connect to this Data Server from other computers.
	Localhost Only	
File Upload	drop area	Drag and drop files in this box to upload files. Clicking on the 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 Device		Enable Save/Load of Device settings.
Include UI		Enable Save/Load of User Interface settings.

Control/ Tool	Option/Range	Description
Include Preferences		Enable loading of User Preferences from settings file.
Save	Save	Save the user interface and device setting to a file.
Load	Load	Load the user interface and device setting from a file.
Display Theme	Dark	Choose theme of the user interface.
	Light	
Plot Print Theme	Dark	Choose theme for printing SVG plots.
	Light	
Plot Grid	None	Select active grid setting for all SVG plots.
	Dashed	
	Solid	
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.
Dynamic Tabs	ON / OFF	If enabled, sections inside the application tabs are collapsed automatically depending on the window width.
Graphical Mode	Collapsed	Select the display mode for the graphical elements. Auto format will select the format which fits best the current window width.
	Auto	
	Expanded	
Log Format	.NET	Choose the command log format. See status bar and [User] \Documents\Zurich Instruments\LabOne\WebServer\Log
	MATLAB	
	Python	
CSV Delimiter	Tab	Select which delimiter to insert for CSV files.
	Comma	
	Semicolon	

Control/ Tool	Option/Range	Description
CSV Locale	System locale. Use the symbols set in the language and region settings of the computer	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.
	Default locale. Dot for the decimal point and no digit grouping, e.g. 1005.07	
HDF5 Saving	Multiple files. Each measurement goes in a separate 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.
	Single file. All measurements go in one 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.
Format	HDF5	File format of recorded and saved data.
	MATLAB	
	CSV	
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.

5.5. Device Tab

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

5.5.1. Features

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

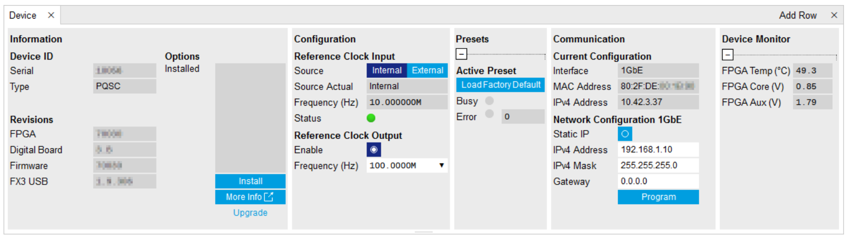
5.5.2. Description

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 5.10: App icon and short description

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

The Device tab (see LabOne UI: Device tab) is divided into five sections: general instrument information, configuration, device presets, communication parameters and a device monitor.



Information
Device ID
Serial [redacted]
Type PQSC
Options
Installed
Revisions
FPGA [redacted]
Digital Board [redacted]
Firmware [redacted]
FX3 USB [redacted]

Configuration
Reference Clock Input
Source Internal External
Source Actual Internal
Frequency (Hz) 10.000000M
Status ●
Reference Clock Output
Enable
Frequency (Hz) 100.0000M

Presets
Active Preset
Busy
Error 0

Communication
Current Configuration
Interface 1GbE
MAC Address 80:2F:DE:[redacted]
IPv4 Address 10.42.3.37
Network Configuration 1GbE
Static IP
IPv4 Address 192.168.1.10
IPv4 Mask 255.255.255.0
Gateway 0.0.0.0

Device Monitor
FPGA Temp (°C) 49.3
FPGA Core (V) 0.85
FPGA Aux (V) 1.79

Figure 5.8: 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 Reference Clock In 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 Reference Clock Out 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 **Presets** section allows one to reset the instrument to the factory defaults.

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

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.

5.5.3. Functional Elements

Table 5.11: 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.

Control/Tool	Option/Range	Description
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 to the appropriate back panel connector.
Actual Input Reference Clock Source		Currently active clock source. This might differ from the Set Source choice if the set clock is not available.
	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.
Load Factory Default	Load	Load the factory default settings.
Busy	grey/red	Indicates that the device is busy with either loading, saving or erasing a preset.
Error		Returns a 0 if the last preset operation was successfully completed or 1 if the last preset operation was illegal.
	0	Last preset operation was successfully completed.
	1	Last preset operation was illegal.
Error LED	grey/red	Turns red if the last operation was illegal.
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.

Control/Tool	Option/Range	Description
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.
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
Save	Program	Click to save the specified IPv4 address, IPv4 Mask and Gateway to the device. Otherwise, the settings will be lost after power cycling the device.

5.6. File Manager Tab

5.6.1. Features

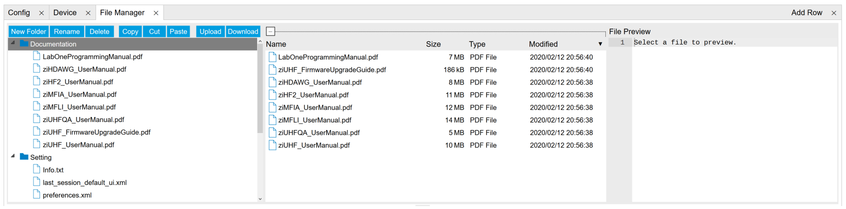
- File preview for settings files and log files

5.6.2. Description

Table 5.12: 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 LabOne UI: File Manager tab) 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.



Documentation
- LabOneProgrammingManual.pdf
- ziHDAWG_UserManual.pdf
- ziHF2_UserManual.pdf
- ziMFIA_UserManual.pdf
- ziMFLI_UserManual.pdf
- ziUHFQA_UserManual.pdf
- ziUHF_FirmwareUpgradeGuide.pdf
- ziUHF_UserManual.pdf

Setting
- Info.txt
- last_session_default_ui.xml
- preferences.xml

Name	Size	Type	Modified
LabOneProgrammingManual.pdf	7 MB	PDF File	2020/02/12 20:56:40
ziUHF_FirmwareUpgradeGuide.pdf	186 kB	PDF File	2020/02/12 20:56:40
ziHDAWG_UserManual.pdf	8 MB	PDF File	2020/02/12 20:56:38
ziHF2_UserManual.pdf	11 MB	PDF File	2020/02/12 20:56:38
ziMFIA_UserManual.pdf	12 MB	PDF File	2020/02/12 20:56:38
ziMFLI_UserManual.pdf	14 MB	PDF File	2020/02/12 20:56:38
ziUHFQA_UserManual.pdf	5 MB	PDF File	2020/02/12 20:56:38
ziUHF_UserManual.pdf	10 MB	PDF File	2020/02/12 20:56:38

```
Select a file to preview.
```

Figure 5.9: LabOne UI: File Manager tab

5.6.3. Functional Elements

Table 5.13: 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.

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

5.7. 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 5.14: App Icon and short description

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

6. 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.

6.1. QCCS system specifications (systems based on SHF series and HDAWG)

Parameter	min	typ	max
Supported peripheral devices	SHFQC, HDAWG, SHFSG, SHFQA		
Supported system architecture	Star architecture		
Output channel skew	-	-	1.6 ns
Output channel skew repeatability over restarts	-	-	200 ps
Real-time communication latency	-	-	90 ns
Sample rate	-	2.0 GSa/s	-

6.2. QCCS system specifications (systems based on UHFQA and HDAWG)

Parameter	min	typ	max
Supported peripheral devices	HDAWG, UHFQA		
Supported system architecture	Star architecture		
HDAWG Output channel skew	-	-	1.6 ns
HDAWG Output channel skew repeatability over restarts	-	-	200 ps
Real-time communication latency	-	-	330 ns from UHFQA to PQSC 100 ns from PQSC to HDAWG
Sample rate	-	HDAWG 2.4 GSa/s UHFQA 1.8 GSa/s	-

This configuration requires one HDAWG per UHFQA as bridge over the DIO port of the HDAWG.

6.3. Synchronization and real-time data exchange interface

Parameter	Value
Interface type	ZSync proprietary
Number of ports	18
Cable length	3m

6.4. Analog Interface Specifications

Table 6.1: 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	-	Switchable 10 MHz / 100 MHz		
reference clock input amplitude	-	0 dBm	-	+13 dBm

Table 6.2: 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	-	+4dBm	-
reference clock output frequency	-	Auto-detect 10 MHz / 100 MHz		

6.5. General Specifications

Table 6.3: 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		

Parameter	min	typ	max
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 6.4: 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 0 V)	–	–	+13.5 dBm
damage threshold Reference Clock In (DC)	-4 V	–	+4 V
damage threshold DIO In / Out port	-0.7 V	–	+4 V

Table 6.5: Host computer requirements

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

Table 6.6: Digital Interface Specifications

Parameter	Description
host computer connection	1GbE, LAN / Ethernet
	USB 3.0
Debug interface	JTAG (over USB)

7. Device Node Tree

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

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

Please see:

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

7.1. Introduction

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

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

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

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

```
/dev1000/auxouts/
```

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

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

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

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

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

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

Module Parameters

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

7.1.1. Node Properties and Data Types

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

Property	Description
Read	Data can be read from the node.
Write	Data can be written to the node.
Setting	The node corresponds to a writable instrument configuration. The data of these nodes are persisted in snapshots of the instrument and stored in the LabOne XML settings files.
Streaming	A node with the read attribute that provides instrument data, typically at a user-configured rate. The data is usually a more complex data type, for example demodulator data is returned as `ZIDemodSample`. A full list of streaming nodes is available in the Programming Manual in the Chapter Instrument Communication. Their availability depends on the device class (e.g. MF) and the option set installed on the device.
Pipelined	If the sequence pipeliner mode is off the value set to the node is applied immediately. Otherwise, it goes to the staging area of the sequence pipeliner instead. Multiple pipelined nodes can be programmed as part of a job definition, that is finalized by writing a one to the relevant `commit` node.

A node may contain data of the following types:

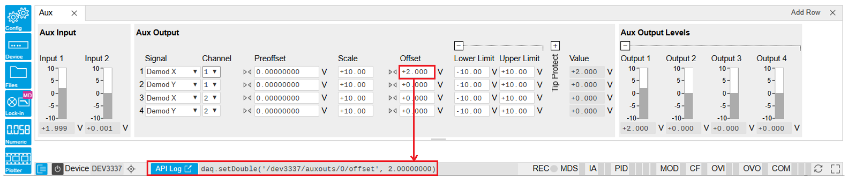
Integer	Integer data.
Double	Double precision floating point data.
String	A string array.
Integer (enumerated)	As for Integer, but the node only allows certain values.
Composite data type	For example, `ZIDemodSample`. These custom data types are structures whose fields contain the instrument output,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

7.1.2. 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 7.1 shows how the equivalent

MATLAB command is displayed after modifying the value of the auxiliary output 1's offset. The format of the LabOne UI's command history can be configured in the Config Tab (MATLAB, Python and .NET are available). The entire history generated in the current UI session can be viewed by clicking the "Show Log" button.



Input 1	Input 2
+1.999 V	+0.001 V

	Signal	Channel	Preoffset	Scale	Offset	Lower Limit	Upper Limit	Value
1	Demod X	1	0.00000000 V	+10.00	+2.000 V	-10.00 V	+10.00 V	+2.000 V
2	Demod Y	1	0.00000000 V	+10.00	+0.000 V	-10.00 V	+10.00 V	+0.000 V
3	Demod X	2	0.00000000 V	+10.00	+0.000 V	-10.00 V	+10.00 V	+0.000 V
4	Demod Y	2	0.00000000 V	+10.00	+0.000 V	-10.00 V	+10.00 V	+0.000 V

Output 1	Output 2	Output 3	Output 4
+2.000 V	+0.000 V	+0.000 V	+0.000 V

API Log
```
daq.setDouble('/dev3337/auxouts/0/offset', 2.00000000)
```

Figure 7.1: When a device's configuration is modified in the LabOne User Interface, the Status Bar displays the equivalent command to perform the same configuration via a LabOne programming interface. Here, the MATLAB code to modify auxiliary output 1's offset value is provided. When "Show Log" is clicked the entire configuration history is displayed in a new browser tab.

In a LabOne Programming Interface

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

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

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

7.1.3. Data Server Nodes

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

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

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

The nodes:

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

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

Nodes that are of particular use to programmers are:

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

The Global nodes of the LabOne Data Server are listed in the Instrument Communication chapter of the LabOne Programming Manual

7.2. Reference Node Documentation

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

7.2.1. CLOCKBASE

/dev..../clockbase

Properties:	Read
Type:	Double
Unit:	Hz

Returns the internal clock frequency of the device.

7.2.2. EXECUTION

/dev..../execution/enable

Properties:	Read, Write
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
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.

/dev..../execution/synchronization/enable

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

Enable synchronization. Trigger generation will only start once all workers have reported a ready status. Synchronization checks will be repeated with the same trigger generation settings (holdoff and repetitions) until synchronization is disabled.

7.2.3. FEATURES

/dev..../features/code

Properties:	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.

7.2.4. FEEDBACK

/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

/dev..../feedback/registerbank/reset

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

Clear all the readout registers.

7.2.5. STATS

/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.

/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.

7.2.6. STATUS

/dev..../status/flags/binary

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

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

/dev..../status/time

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

The current timestamp.

7.2.7. SYSTEM

/dev..../system/activeinterface

Properties:	Read
Type:	String
Unit:	None

Currently active interface of the device.

/dev..../system/boardrevisions/n

Properties:	Read
Type:	String
Unit:	None

Hardware revision of the FPGA base board

/dev..../system/clocks/ready

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

Advises to wait 30 minutes after boot before locking to external reference clock.

/dev..../system/clocks/referenceclock/in/freq

Properties:	Read
Type:	Double
Unit:	Hz

Indicates the frequency of the reference clock.

/dev..../system/clocks/referenceclock/in/source

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

The intended reference clock source. When the source is changed, all the instruments connected with ZSync links will be disconnected. The connection should be re-established manually.

0	"internal": The internal clock is intended to be used as the frequency and time base reference.
1	"external": An external clock is intended to be used as the frequency and time base reference. Provide a clean and stable 10 MHz or 100 MHz reference to the appropriate back panel connector.

/dev..../system/clocks/referenceclock/in/sourceactual

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

The actual reference clock source.

0	"internal": The internal clock is used as the frequency and time base reference.
1	"external": An external clock is used as the frequency and time base reference.

/dev..../system/clocks/referenceclock/in/status

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

Status of the reference clock.

0	"locked": Reference clock has been locked on.
1	"error": There was an error locking onto the reference clock signal.
2	"busy": The device is busy trying to lock onto the reference clock signal.

/dev..../system/clocks/referenceclock/out/enable

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

Enable clock signal on the reference clock output. When the clock output is turned on or off, all the instruments connected with ZSync links will be disconnected. The connection should be re-established manually.

/dev..../system/clocks/referenceclock/out/freq

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

Select the frequency of the output reference clock. Only 10 MHz and 100 MHz are allowed. When the frequency is changed, all the instruments connected with ZSync links will be disconnected. The connection should be re-established manually.

/dev..../system/fpgarevision

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

HDL firmware revision.

/dev..../system/fwlog

Properties:	Read
Type:	String
Unit:	None

Returns log output of the firmware.

/dev..../system/fwlogenable

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

Enables logging to the fwlog node.

/dev..../system/fwrevision

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

Revision of the device-internal controller software.

/dev..../system/fx3revision

Properties:	Read
Type:	String
Unit:	None

USB firmware revision.

/dev..../system/identify

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

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

/dev..../system/kerneltype

Properties:	Read
Type:	String
Unit:	None

Returns the type of the data server kernel (mdk or hpk).

/dev..../system/nics/n/defaultgateway

Properties:	Read, Write
Type:	String
Unit:	None

Default gateway configuration for the network connection.

/dev..../system/nics/n/defaultip4

Properties:	Read, Write
Type:	String
Unit:	None

IPv4 address of the device to use if static IP is enabled.

/dev..../system/nics/n/defaultmask

Properties:	Read, Write
Type:	String
Unit:	None

IPv4 mask in case of static IP.

/dev..../system/nics/n/gateway

Properties:	Read
Type:	String
Unit:	None

Current network gateway.

/dev..../system/nics/n/ip4

Properties:	Read
Type:	String
Unit:	None

Current IPv4 of the device.

/dev..../system/nics/n/mac

Properties:	Read
Type:	String
Unit:	None

Current MAC address of the device network interface.

/dev..../system/nics/n/mask

Properties:	Read
Type:	String
Unit:	None

Current network mask.

/dev..../system/nics/n/saveip

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

If written, this action will program the defined static IP address to the device.

/dev..../system/nics/n/static

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

Enable this flag if the device is used in a network with fixed IP assignment without a DHCP server.

/dev..../system/powerconfigdate

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

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

/dev..../system/preset/busy

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

Indicates if presets are currently loaded.

/dev..../system/preset/error

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

Indicates if the last operation was illegal. Successful: 0, Error: 1.

/dev..../system/preset/load

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

Load the selected preset.

/dev..../system/properties/timebase

Properties:	Read
Type:	Double
Unit:	s

Minimal time difference between two timestamps. The value is equal to 1/(maximum sampling rate).

/dev..../system/shutdown

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

Sending a '1' to this node initiates a shutdown of the operating system on the device. It is recommended to trigger this shutdown before switching the device off with the hardware switch at the back side of the device.

/dev..../system/stall

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

Indicates if the network connection is stalled.

/dev..../system/update

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

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

7.2.8. 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.

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.

7.2.9. 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.

0	No connection
1	Connection in progress
2	Connected
3	Connection error

/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/enable

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

Enable feedback for a given port.

/dev..../zsyncs/n/output/registerbank/sources/n/enable

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

Enable readout register forwarding of bits 2m, 2m+1 to the port n.

/dev..../zsyncs/n/output/registerbank/sources/n/index

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

The index of the results in the readout register to be forwarded.

/dev..../zsyncs/n/output/registerbank/sources/n/register

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

The readout register to be forwarded.

/dev..../zsyncs/n/output/source

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

Select the feedback source for a given port.

0	"reg", "register_forwarding": Register Forwarding
1	"dec", "decoder": Decoder