UHF
User Manual

600 MHz Lock-in Amplifier

Zurich
Instruments

UHF User Manual

Zurich Instruments AG

Revision 24.04

Table of Contents

Table of Contents

CE Declaration of Conformity

The manufacturer

Zurich Instruments
Technoparkstrasse 1
8005 Zurich
Switzerland

declares that the product

UHF Series (UHFLI, UHFAWG, UHFQA), 600 MHz, 1.8 GSamples/s

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, CTO

UKCA Declaration of Conformity

The manufacturer

Zurich Instruments
Technoparkstrasse 1
8005 Zurich
Switzerland

declares that the product

UHF Series (UHFLI, UHFAWG, UHFQA), 600 MHz, 1.8 GSamples/s

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

Flavio Heer, CTO

1. Change Log

1.1. Release 24.04

Release date: 30-Apr-2024

1.2. Release 24.01

Release date: 31-Jan-2024

- Sweeper: The sweeper grid now includes the exact start and stop sweeping points.
- AM/FM MOD Option: Phase-adjust capability to nullify the measured phase of demodulated signals.
- AWG Option: SeqC command `setDIO` now has constant latency, no matter its arguments.

1.3. Release 23.10

Release date: 31-Oct-2023

- Sweeper: Setting the start and stop points of the sweep parameter from the x-axis cursors in the Sweeper tab.
- Scope: Recording of scope streaming samples in HDF5 format from the Config tab.
- Connectivity: Possibility to connect and use the device from a different network.

1.4. Release 23.06

Release date: 30-Jun-2023

1.5. Release 23.02

Release date: 28-Feb-2023

- Aux Outputs: Changed the default scale factor from 1 to 10.
- LabOne API: Added support of Python 3.11.
- Sweeper Module: Improved phase unwrap feature.
- LabOne: Dropped support for Windows 7 and Windows 8.1 as they have reached their end of life.

1.6. Release 22.08

Release date: 31-Aug-2022

- LabOne UI: Enabled Sweeper tab to handle Average Power and Standard Deviation of Arithmetic Unit (AU) signals.
- LabOne UI: Improved the Spectral Density feature of Sweeper tab.

1.7. Release 22.02

Release date: 28-Feb-2022

- LabOne Software: 'Flat Top' window function for FFT mode of Scope, DAQ, and Spectrum.
- LabOne API: Support of Python 3.10.

1.8. Release 21.08

Release date: 31-Aug-2021

- User Manual: HTML version available in LabOne user interface and online documentation.
- LabOne Software: Support for GNU/Linux and macOS on ARM64 and Apple M1 processors.

1.9. Release 21.02

Release date: 28-Feb-2021

- LabOne API: Added online Programming Manual and Documentation.
- Sweeper: Simultaneous display of standard and X-Y plots to visualize Nyquist and Bode plots at the same time.
- Sweeper: Improved rendering of sweeps with more than 2000 points.

1.10. Release 20.07

Release date: 20-Aug-2020

- LabOne: Trends plots to track readings from the Math sub-tab over time.
- LabOne: Device Information report in Device tab.
- LabOne: Improved colormaps available for 2D plots.
- PID Advisor: clamp maximum advised bandwidth to hardware limits to guard against instability.

1.11. Release 20.01

Release date: 28-Feb-2020

- LabOne: added linear fit to the Math sub-tab of Sweeper and DAQ tabs.
- LabOne: histogram data can be saved in CSV format.
- LabOne: added option to display a normal or Rice distribution fit in Plotter tab.
- LabOne: improved importing of saved SVG figures to main vector graphics editors.
- PID option: added control button to keep or reset the integrator value on stop and restart of PID controllers.
- AWG: improved compilation stability.

2. Getting Started

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

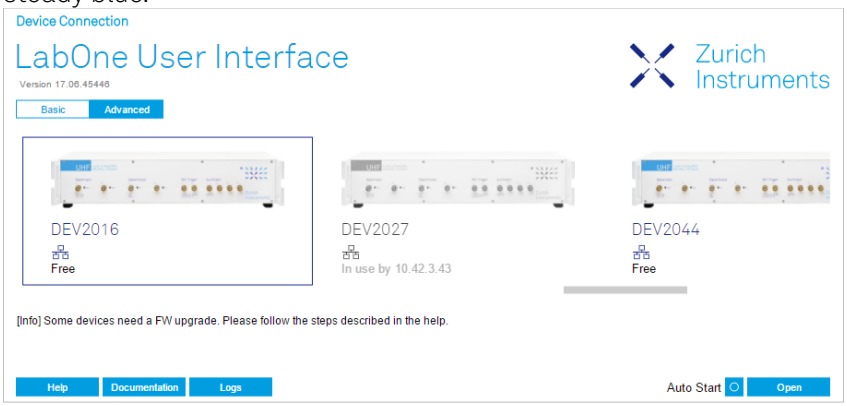
- Quick Start Guide for the impatient
- Inspecting the package content and accessories
- List of essential handling and safety instructions
- Installing LabOne, the UHF Instrument software, on your host computer
- Powering-on the device and connecting the device to a host computer
- Performing basic operation checks on the instrument

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

2.1. Quick Start Guide

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

1. Check 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 then connect in with the measurement PC by using the USB cable. The necessary drivers will now be installed automatically. The front panel LED will blink orange at this stage.
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. The front panel LED turns from blinking orange to a steady blue.



6. Click the **Open** button on the lower right of the start screen. The default configuration is loaded and the first measurements can be taken. In cases the device could not be found or the user interface does not start at all, please be referred to LabOne Software Architecture.
7. The UHF User Manual is included in the LabOne installation. Under Windows 10 it can be found in[1] `Start Menu → Zurich Instruments → User Manual UHF`.

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

Once the Instrument is up and running we recommend to go through some of the tutorials given in Tutorials. Moreover, Functional Description LabOne User Interface provides a general introduction to the various tools and settings tabs with tables in each section providing a detailed description of every UI element as well. 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

It's recommended to enable graphical hardware acceleration to ensure a high responsiveness of the web browser user interface. On most computers, hardware acceleration can be enabled by one of the following methods.

- Control panel: Control Panel\Appearance and Personalization\Display\Screen Resolution. Advanced settings. Trouble shoot. Change settings.
- NVIDIA control panel: select graphic processor. Apply.

Some computers have two graphic chip sets installed, an Intel and a NVIDIA chip set. Activating the NVIDIA along with the acceleration is recommended to achieve best possible performance. The only drawback changing these settings is a slightly increased power consumption.

---

1. Under Windows 7 and 8, the User Manual can be found in `Start Menu` → `All programs / All apps` → `Zurich Instruments` → `User Manual UHF` ↩

2.2. Inspect the Package Contents

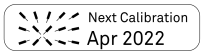
If the shipping container appears to be damaged, keep the container until you have inspected the contents of the shipment and have performed basic functional tests.

Please verify:

- You have received 1 Zurich Instruments UHFLI or UHFAWG Instrument
- You have received 1 power cord with a power plug suited to your country
- You have received 1 USB cable and/or 1 LAN cable (category 5/6 required)
- A printed version of the "Getting Started" section
- The "Next Calibration" sticker on the back panel of the Instrument indicates approximately 2 years ahead in time. Zurich Instruments recommends calibration intervals of 2 years
- The MAC address and serial number of the instrument are displayed on a sticker on the back panel

Table 2.1: Package contents for the UHFLI or UHFAWG

 or 	
	the power cord (e.g. EU norm)
	USB cable

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

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

Carefully inspect your Instrument. If there is mechanical damage or the amplifier 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 UHFLI 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 Specification for a comprehensive list of ratings.

Do not service or adjust anything yourself	There are no serviceable parts inside the instrument.
Software updates	Frequent software updates provide the user with many important improvements as well as new features. Only the last released software version is supported by Zurich Instruments.
Warnings	Instructions contained in any warning issued by the instrument, either by the software, the graphical user interface, the notes on the instrument or mentioned in this manual, must be followed.
Notes	Instructions contained in the notes of this user manual are of essential importance for correctly interpreting the acquired measurement data.
High voltage transients due to inductive loads	When measuring devices with high inductance, take adequate measures to protect the Signal Input connectors against the high voltages of inductive load switching transients. These voltages can exceed the maximum voltage ratings of the Signal Inputs and lead to damage.
Location and ventilation	This instrument or system is intended for indoor use in an installation category II and pollution degree 2 environment as per IEC 61010-1. Do not operate or store the instrument outside the ambient conditions specified in the Specification 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 and mains line fuse	For continued protection against fire, replace the line fuse only with a fuse of the specified type and rating. Use only the power cord specified for this product and certified for the country of use. Always position the device so that its power switch and the power cord are easily accessible during operation.
Main power disconnect	Unplug product from wall outlet and remove power cord before servicing. Only qualified, service-trained personnel should remove the cover from the instrument.
RJ45 sockets labeled ZCtrl	The two RJ45 sockets on the back panel labeled "Peripheral ZCtrl" 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 Specification 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 UHF Series 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.

Windows LabOne Installation

1. The UHF Series Instrument should not be connected to your computer during the LabOne software installation process.
2. Start the LabOne installer program with a name of the form `LabOne64-XX.XX.XXXXX.msi` by a double click and follow the instructions. Windows Administrator rights are required for installation. The installation proceeds as follows:
   - On the welcome screen click the **Next** button.

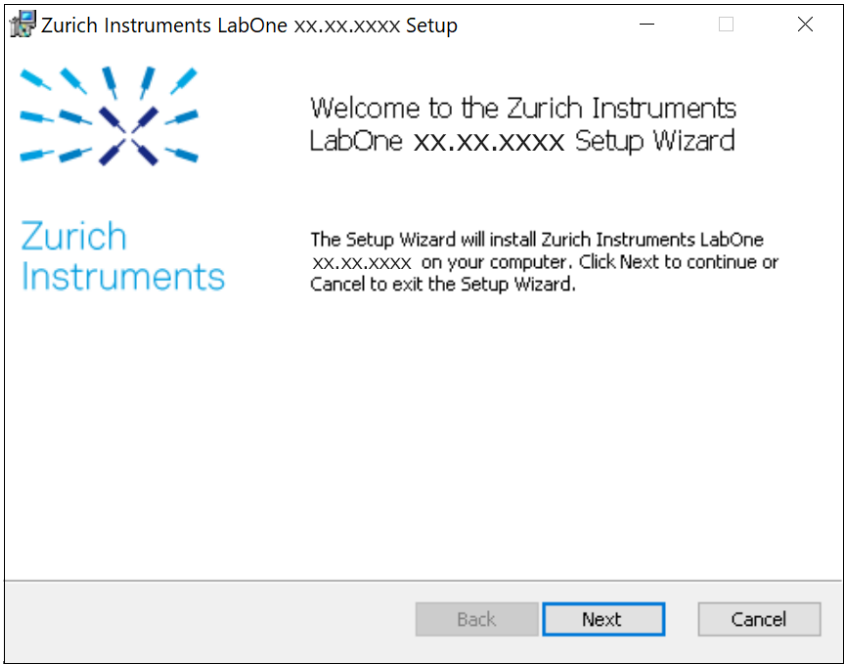


Figure 2.1: Installation welcome screen

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

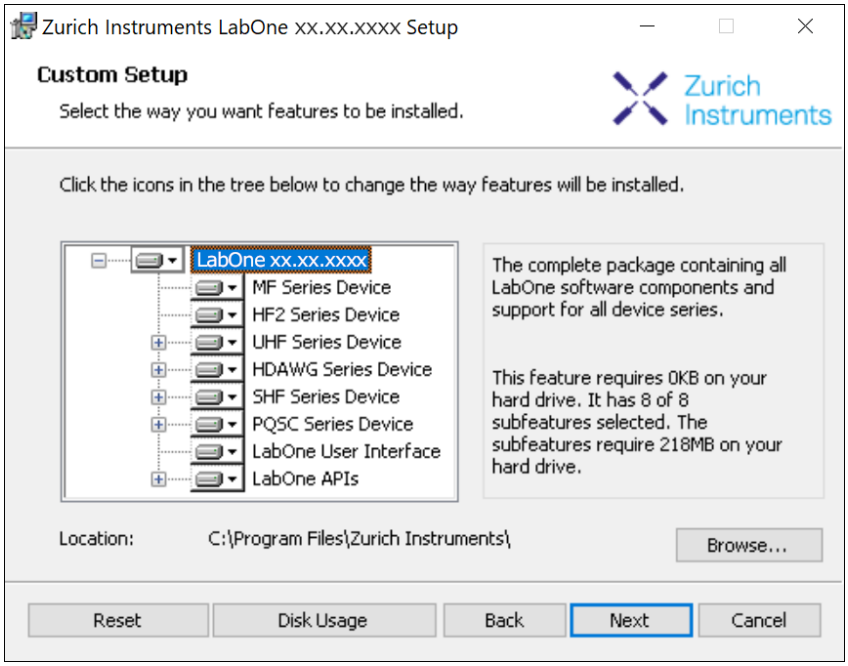


Figure 2.2: Custom setup screen

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

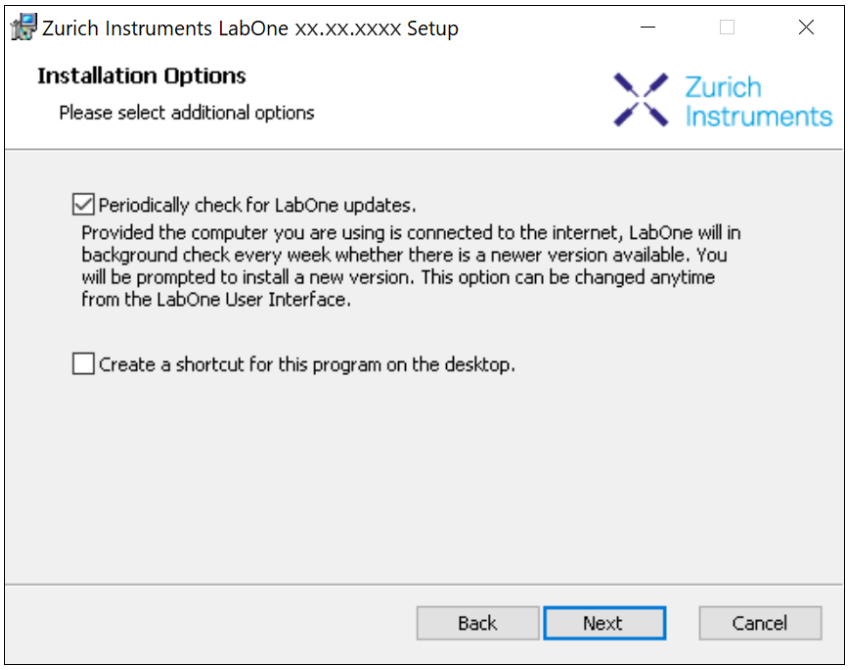


Figure 2.3: Automatic update check

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

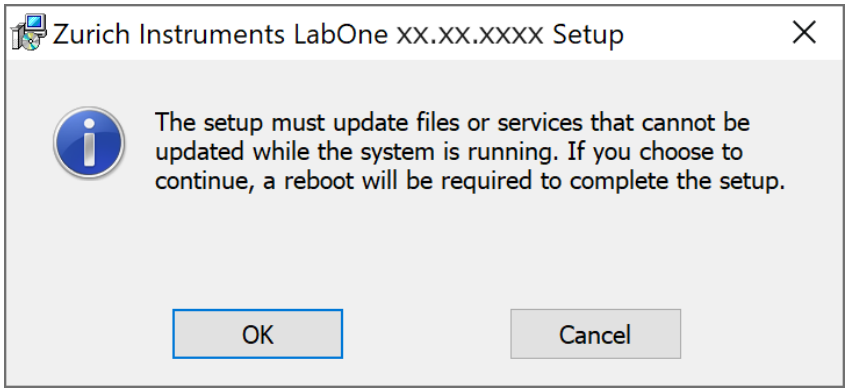


Figure 2.4: Installation reboot request

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

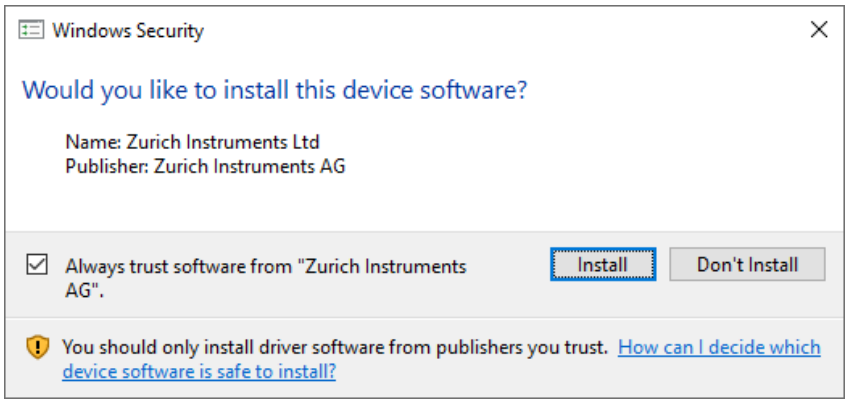


Figure 2.5: Installation driver acceptance

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

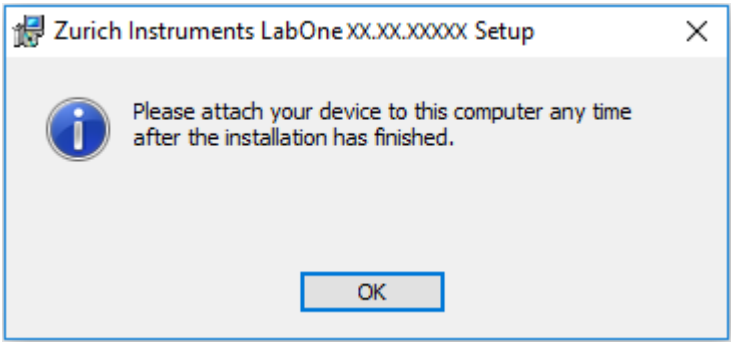


Figure 2.6: Installation completion screen

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

Warning

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

Start LabOne Manually on the Command Line

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

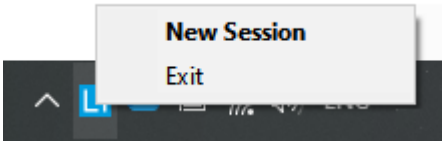
terminal (Command Prompt, PowerShell (Windows) or Bash (Linux)). For Windows, the current working directory needs to be the installation directory of the Web Server and Data Server. They are installed in the Program Files folder (usually: C:\Program Files) under \Zurich Instruments\LabOne in the WebServer and DataServer folders, respectively. The Web Server and Data Server (ziDataServer) are started by running the respective executable in each folder. Please be aware that only one instance of the Web Server can run at a time per computer. The behavior of the Servers can be changed by providing command line arguments. For a detailed list of all arguments see the command line help text:

```
$ ziWebServer --help
```

For the Data Server:

```
$ ziDataServer --help
```

One useful application of running the Webserver manually from a terminal window is to change the data directory from its default path in the user home directory. The data directory is a folder in which the LabOne Webserver saves all the measured data in the format specified by the user. Before running the Webserver from the terminal, the user needs to ensure there is no other instance of Webserver running in the background. This can be checked using the Tray Icon as shown below.



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

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

Windows LabOne Uninstallation

To uninstall the LabOne software package from a Windows computer, one can open the "Apps & features" 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 2.7.

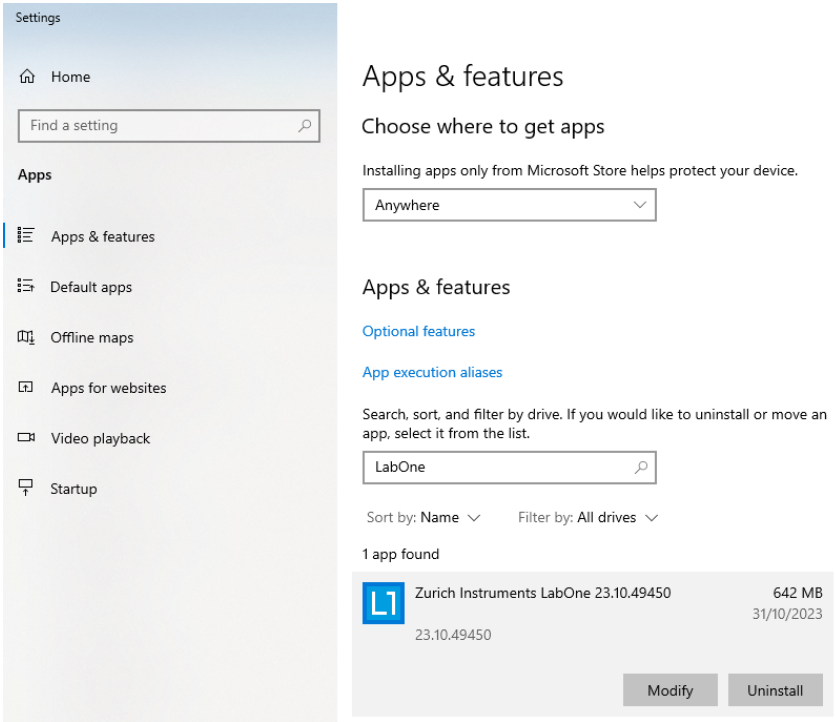


Figure 2.7: Uninstallation of LabOne on Windows computers

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 2.7. This will open the LabOne installation wizard with the option "Repair" as shown in Figure 2.8.

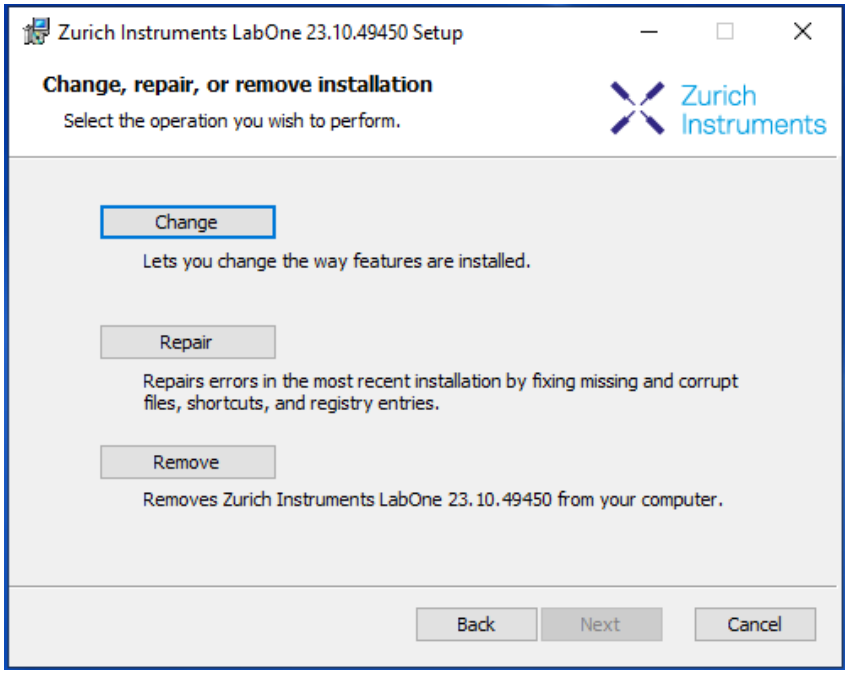


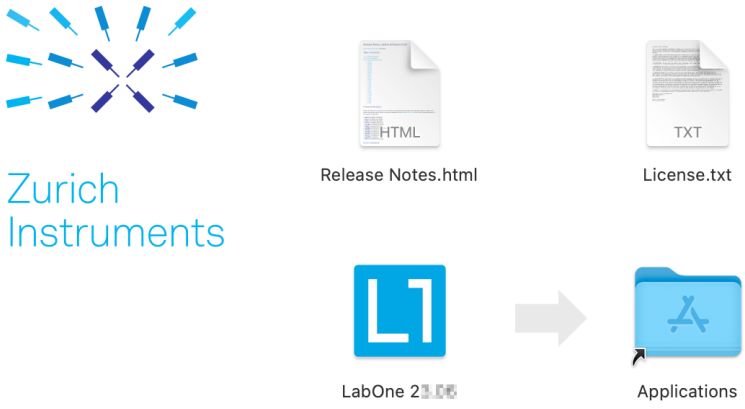
Figure 2.8: Repair of LabOne on Windows computers

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

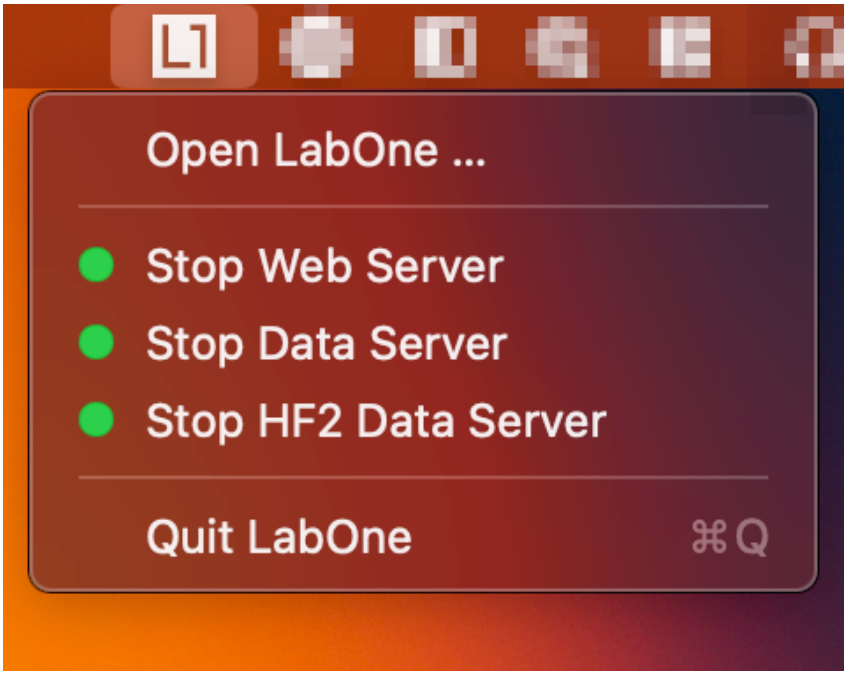
2.4.2. Installing LabOne on macOS

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.



Uninstalling LabOne on macOS

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

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\ 2X.XX.app/Contents/Resources`

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\ 2X.XX.app/Contents/Resources/DataServer/`.

The web server binary is called `ziWebServer` and is located at `Applications/LabOne\ 2X.XX.app/Contents/Resources/WebServer/`.

Note

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

2.4.3. Installing LabOne on Linux

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.

Linux LabOne Installation

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

1. Extract the LabOne tarball in a temporary directory:

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

2. Navigate into the extracted directory.

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

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

   ```
   sudo bash install.sh
   ```

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

Running the Software on Linux

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

1. Start the LabOne Data Server program at a command prompt:

   ```
   $ ziDataServer
   ```

   You should be able to access your instrument. In case of problems please consult the Troubleshooting at the end of this chapter.
2. Start the Web Server program at a command prompt:

   ```
   $ ziWebServer
   ```

3. 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.
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.

Important

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

Uninstalling LabOne on Linux

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

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

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

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

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

2.5. Connecting to the Instrument

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

Note

The following web browsers are supported (latest versions)

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

2.5.1. LabOne Software Architecture

The Zurich Instruments LabOne software gives quick and easy access to the instrument from a host PC. LabOne also supports advanced configurations with simultaneous access by multiple software clients (i.e., LabOne User Interface clients and/or API clients), and even simultaneous access by

several users working on different computers. Here we give a brief overview of the architecture of the LabOne software. This will help to better understand the following chapters.

The software of Zurich Instruments equipment is server-based. The servers and other software components are organized in layers as shown in Figure 2.9.

- 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.

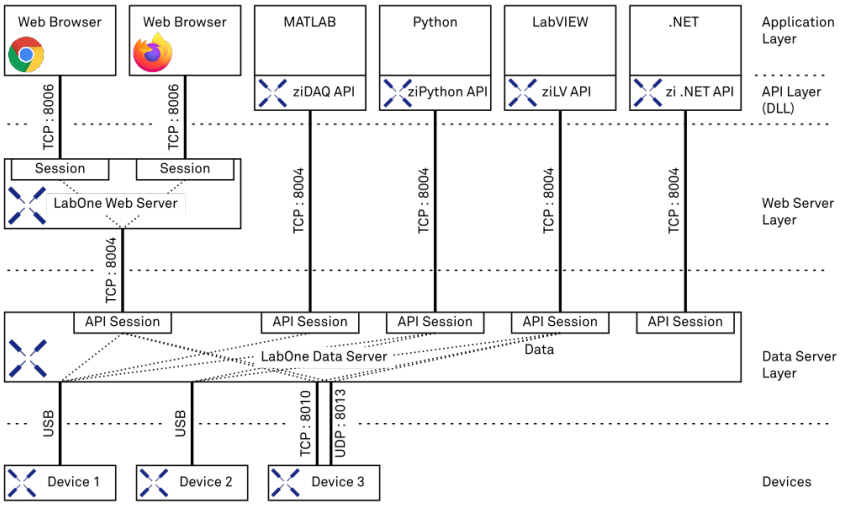


Figure 2.9: LabOne Software architecture

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. The Data Server is started automatically by a service when the PC is started. This service can be disabled if necessary, though the Data Server consumes only little resources when there is no active session. 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 UHF Series 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 Start Menu (Under Windows 7 and 8, the LabOne User Interface start-up link can be found in `Start Menu → all programs / all apps → Zurich Instruments LabOne`). As shown in Figure 2.10, 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.

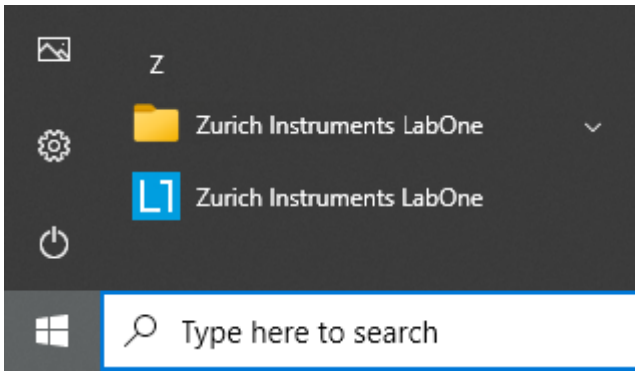


Figure 2.10: Link to the LabOne User Interface in the Windows 10 Start Menu

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

Note

By creating a shortcut to Google Chrome on your desktop with the Target `path\to\chrome.exe -app=http://127.0.0.1:8006` set in Properties you 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 2.11 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.

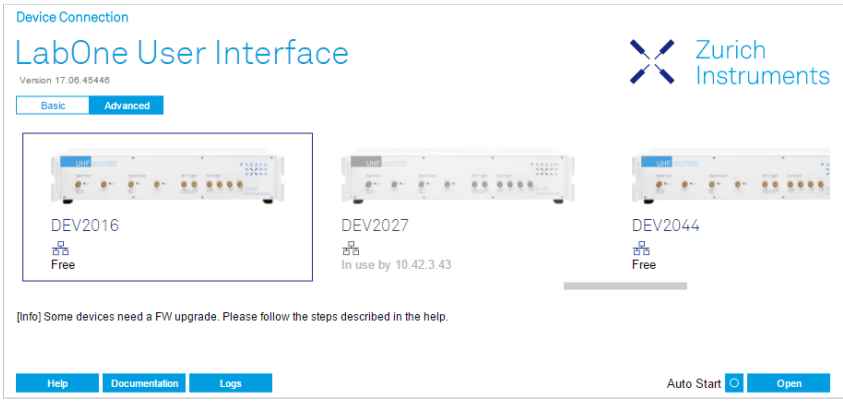


Figure 2.11: Device Connection dialog

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

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

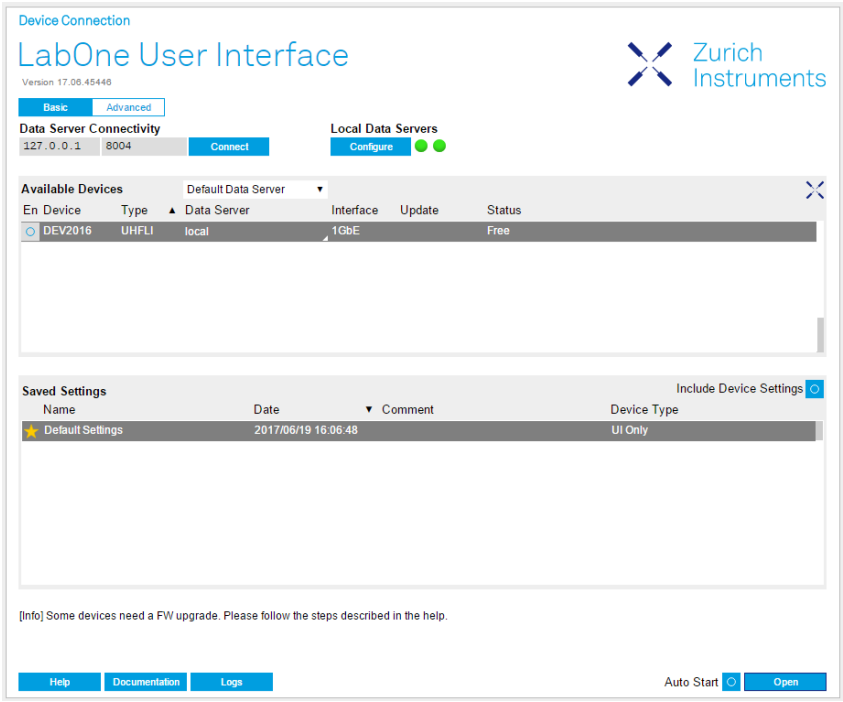


Figure 2.12: Device Connection dialog (Advanced view)

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 UHF Series 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 UHF 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 an interface is physically connected but not visible please read Visibility and Connection.

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

Table 2.5: Device Status Information

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

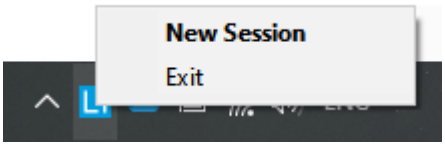
Note

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.

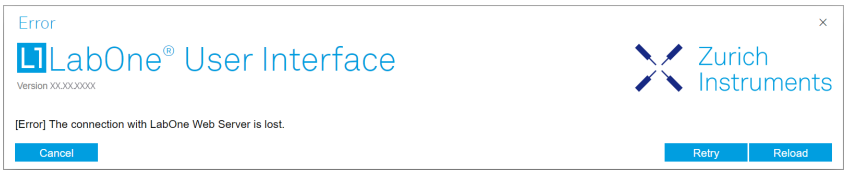


Messages

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.

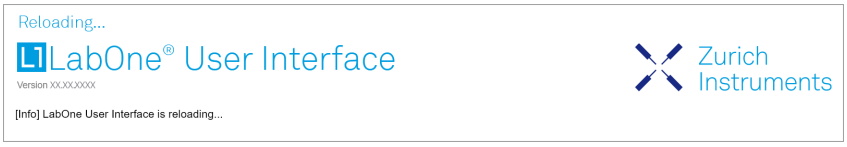
Lost Connection to the LabOne Web Server

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 2.11. The figure below shows an example of the Connection Lost dialog.



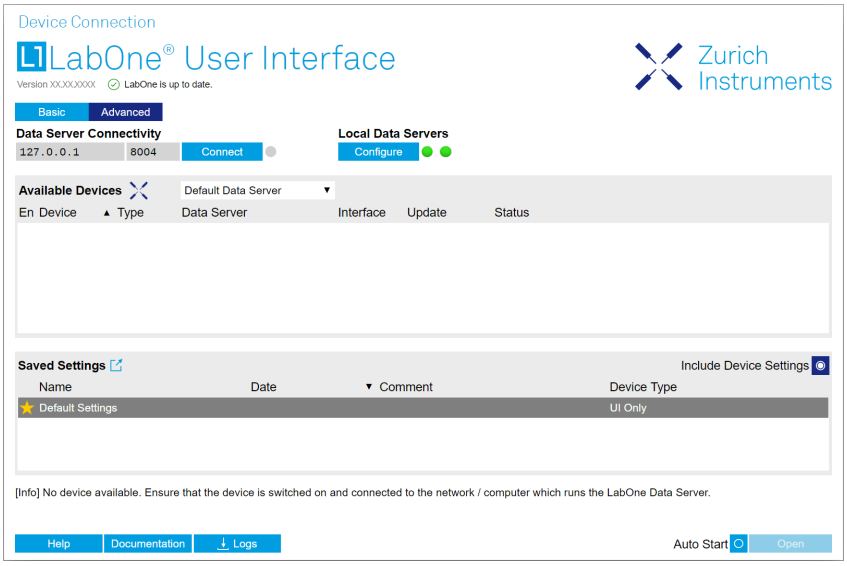
Reloading...

If a session error cannot be handled, the LabOne Web Server will restart to show a new Device Connection dialog as shown in Figure 2.11. During the restart a window is displayed indicating that the LabOne User Interface will reload. If reloading does not happen the same effect can be triggered by pressing F5 on the keyboard. The figure below shows an example of this dialog.



No Device Discovered

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



No Device Available

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

2.5.3. Visibility and Connection

There are several ways to connect the Zurich Instruments lock-in amplifier 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 a device is connected to a network, it can be accessed from multiple PCs. To manage the access to the device, there are two different connectivity states: visible and connected. It is important to distinguish if a device is just physically connected over USB or 1GbE or actively controlled by the LabOne Data Server. In the first case the device is visible to the LabOne Data Server. In the second case the device is logically connected.

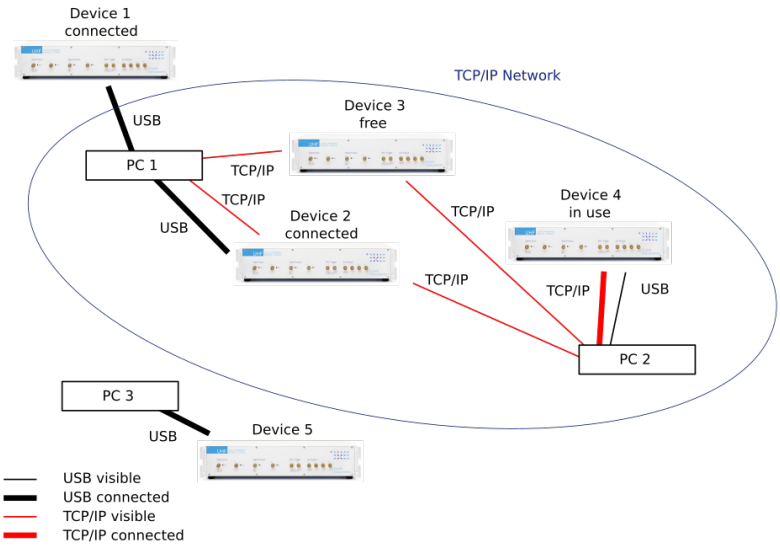


Figure 2.13: Connectivity

Figure 2.13 shows some examples of possible configurations of PC-to-device 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 Devices

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

Connected Device

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

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

2.5.4. USB Connectivity

To control the device over USB, connect the instrument with the supplied USB cable to the PC on which the LabOne Software is installed. The USB driver needed for controlling the device is included in the LabOne Installer package. Ensure that the device uses the latest firmware. The software will automatically use the USB interface for controlling the device if available. If the USB connection is not available, the 1GbE connection may be selected. It is possible to enforce or exclude a specific interface connection.

Note

To use the device exclusively over the USB interface, modify the shortcut of the LabOne User Interface and LabOne Data Server in the Windows Start menu. Right-click and go to Properties, then add the following command line argument to the Target LabOne User Interface: `--interface-usb true --interface-ip false`

A device connected over USB can be automatically connected to by the Data Server because there is only a single host PC to which the device interface is physically connected.

auto-connect = on

If a device is attached via a USB cable, a connection will be established automatically by the Data Server. This is the default behavior.

auto-connect = off

To disable automatic connection via USB, add the following command line argument when starting the Data Server: `--auto-connect=off`

This is achieved by right clicking the LabOne Data Server shortcut in the Start menu, selecting "Properties" and adding the text to the Target field as shown in Figure 2.14.

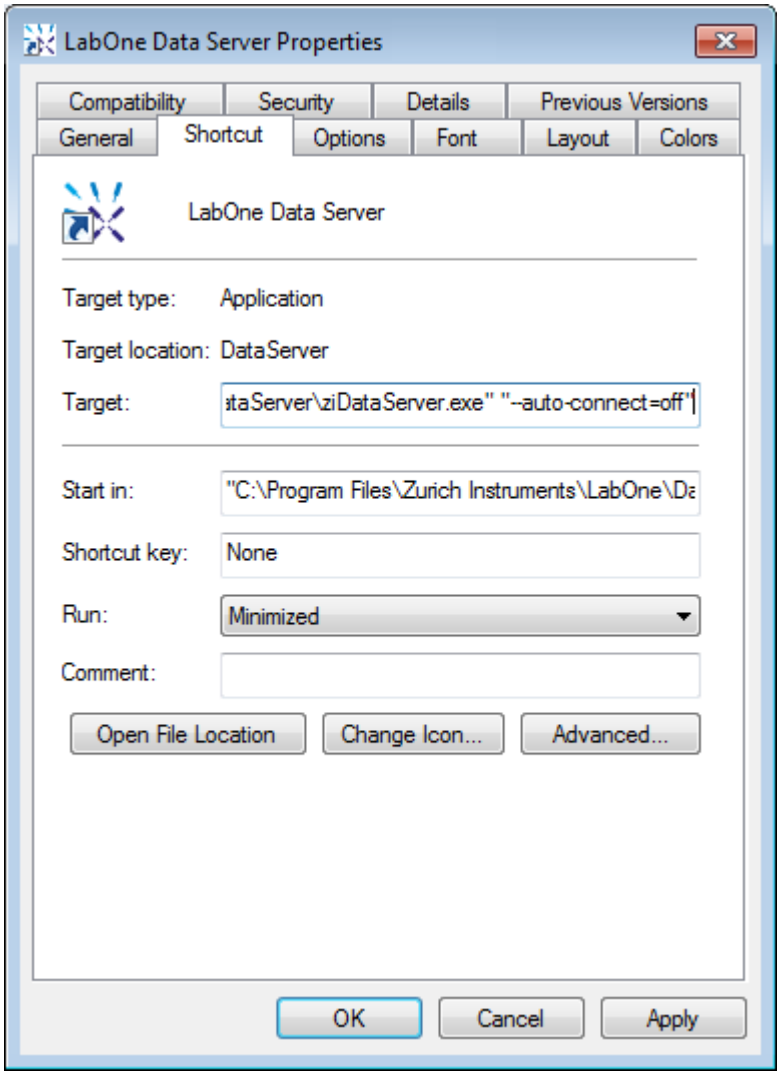


Figure 2.14: auto-connect

2.5.5. 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. This particularly concerns the enabling of Jumbo frames, which is an essential setting for good performance when using high data transfer rates.

Note

To use the device exclusively over the Ethernet interface, modify the shortcut of the LabOne User Interface UHF and LabOne Data Server UHF in the Windows Start menu. Right-click and go to Properties, then add the following command line argument to the Target field: `--interface-usb false --interface-ip true`

Multicast DHCP

The most straightforward TCP/IP connection method is to rely on a network configuration to recognize the UHF 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 UHF 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 UHF 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 network connectivity (e.g. internet), the other can be used for connecting to the Instrument. Notebooks can generally profit from wireless LAN for internet connection.

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 device connectivity directly to the 1GbE port of the UHF 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 below.

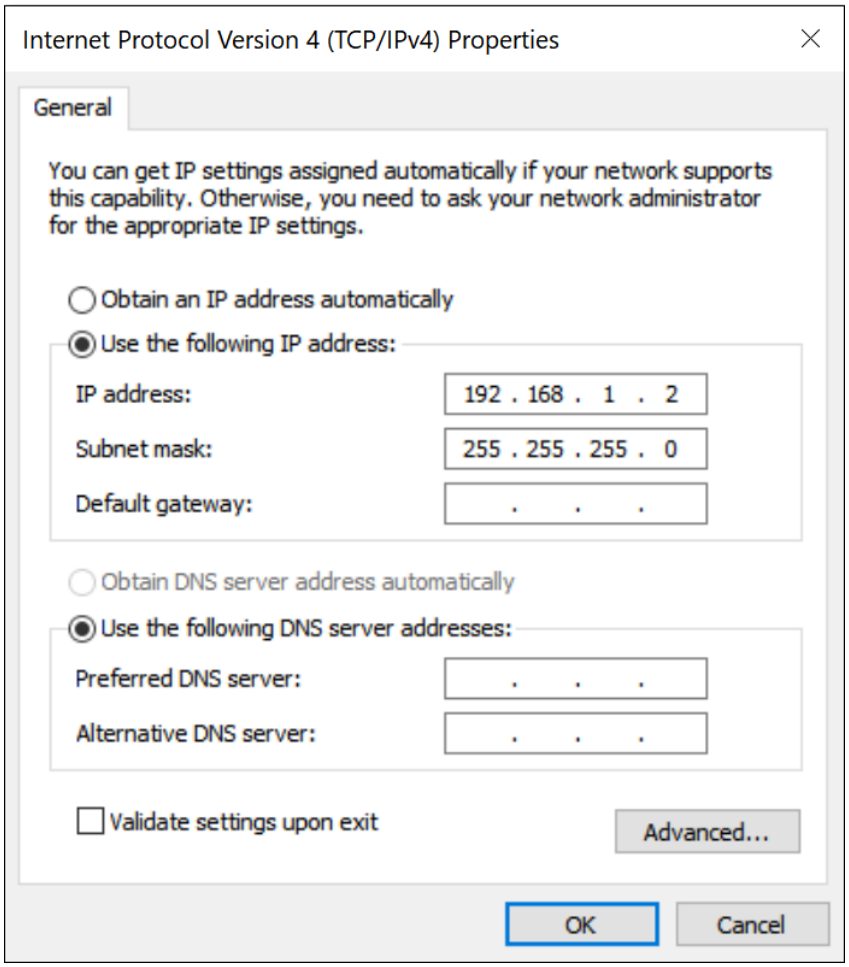


Figure 2.15: Static IP configuration for the host computer

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 UHF 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. This is achieved by turning on network discovery in the Private tab of the network's advanced sharing settings as shown in the figure below.

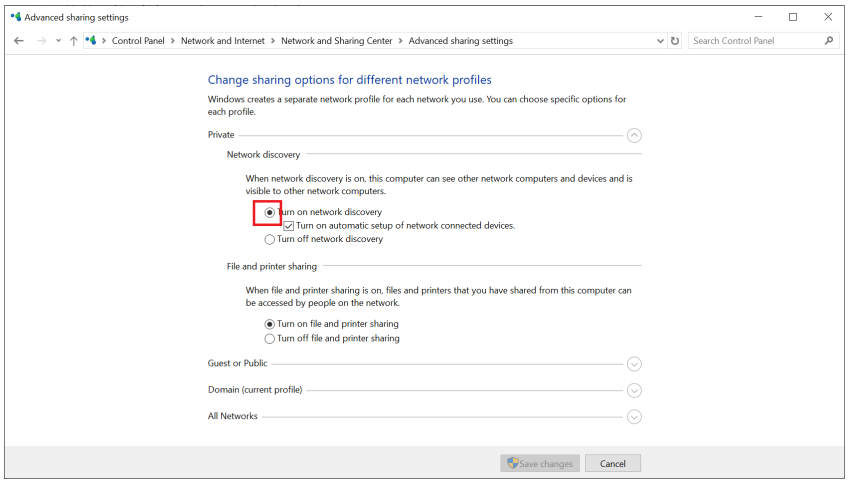


Figure 2.16: Turn on network discovery for Private P2P connection

Warning

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

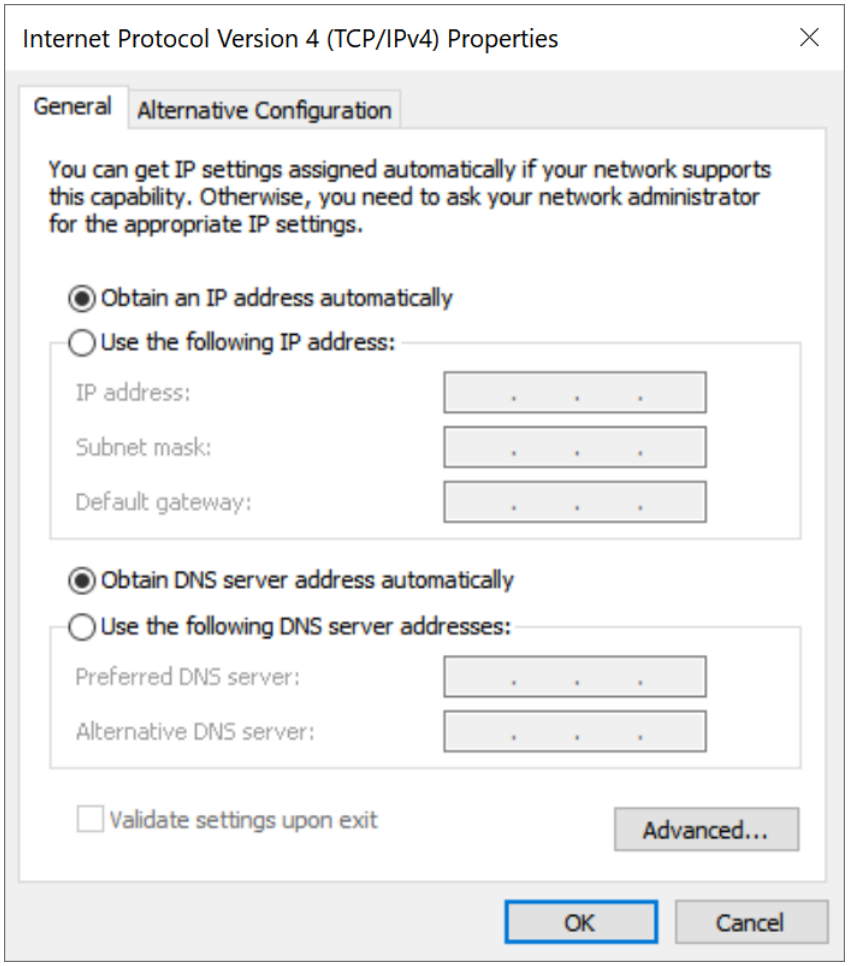
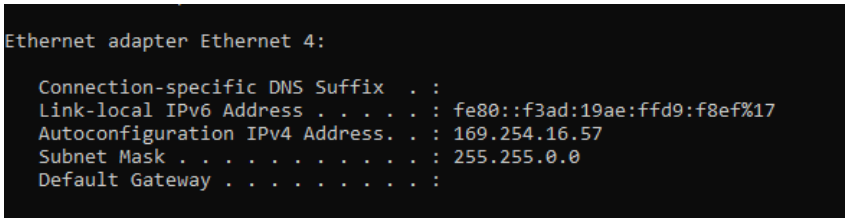


Figure 2.17: Dynamic IP configuration for the host computer

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 2.18 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 2.18: IP and subnet mask of host computer

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

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

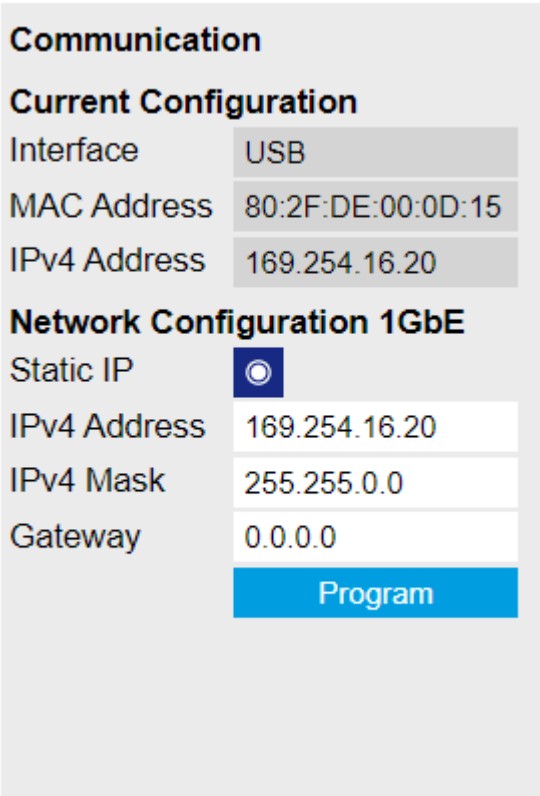
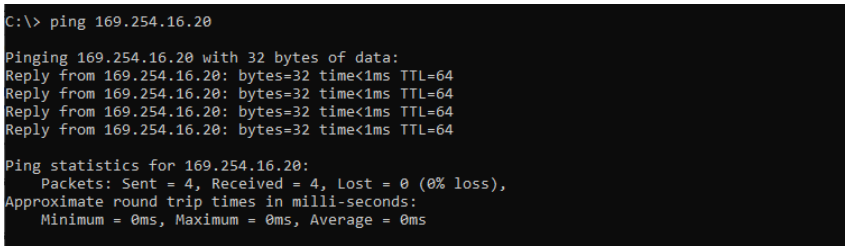


Figure 2.19: Configuration of static IP in LabOne UI

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 2.20 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 2.20: 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.

Recommended Network Cards

Especially when working at high data transfer rates, it's recommended to use one of the network card models that have been tested by Zurich Instruments. In comparison, some older network cards either don't support performance features such as receive side scaling or jumbo packets, or have performance limitations that can lead to sample loss. In addition to the choice of the network card, a powerful processor on the host PC is essential for preventing data loss.

lists the network cards recommended by Zurich Instruments. The rightmost column lists the recommended settings where they differ from the default. Under Windows, the network card settings can be accessed from the Device Manager. In the Network Adapters group, right-click on the entry for the network card and select Properties to open a dialog such as the one shown in Figure 2.21

Recommended network cards

Model	Requirements	Settings
Intel I210-T1	PCIe x1	▬ Energy Efficient Ethernet: off ▬ Jumbo Packet: 9014 bytes ▬ Receive Side Scaling Queues: 4 queues
Intel I350-T2	PCIe x16	▬ Jumbo Packet: 9014 bytes ▬ Receive Side Scaling: enabled ▬ Maximum Number of RSS Queues: 4
Intel PRO/1000 PT	PCIe x4	▬ Jumbo Packet: 9014 bytes

Recommended network cards

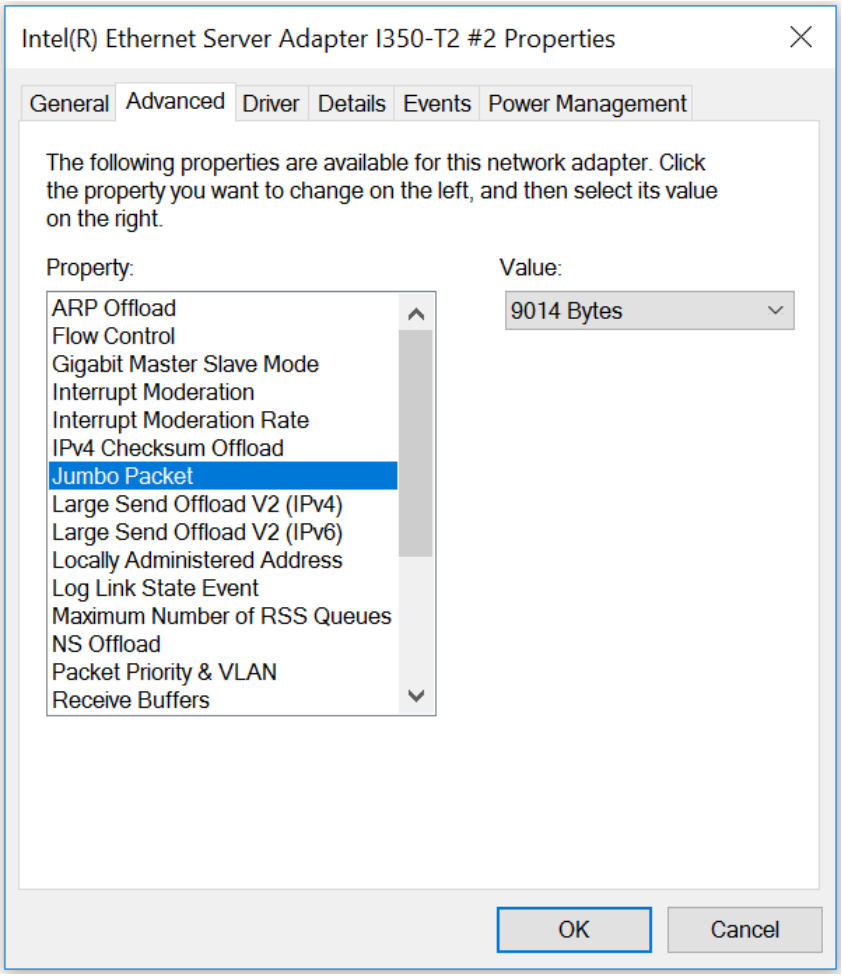


Figure 2.21: Network card properties dialog for the I350-T2 model. The dialog is accessible via the Windows Device Manager.

2.6. Software Update

2.6.1. Overview

It is recommended to regularly update the LabOne software on the UHF 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 2.22 will start a download the latest LabOne installer for Windows or Linux, see Figure 2.23. After download, proceed as explained in Software Installation to update LabOne.

Figure 2.22: Device Connection dialog: LabOne update available

Figure 2.23: 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 2.24. 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 2.25. The firmware upgrade takes approximately 2 minutes.

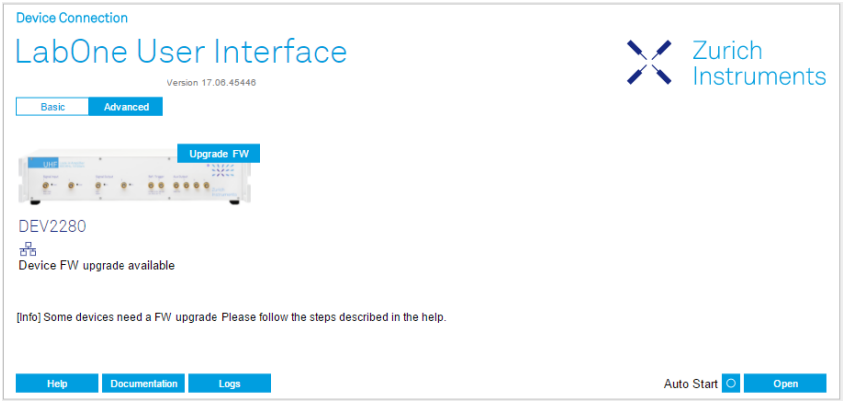


Figure 2.24: Device Connection dialog with available firmware update

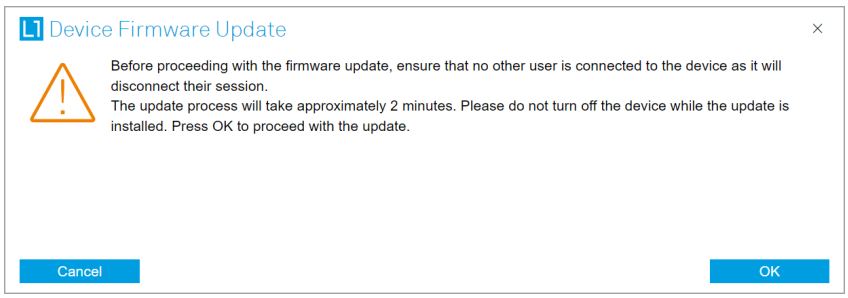


Figure 2.25: Device Firmware Update start-up dialog

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

Table 2.8: Common Problems

Problem	Check item
The software cannot be installed or uninstalled	Please verify you have administrator/root rights.
The software cannot be updated	Please use the Modify option in Windows Apps & Features functionality. In the software installer select Repair, then uninstall the old software version, and install the new version.

Problem	Check item
The Instrument does not turn on	Please verify the power supply connection and inspect the fuse. The fuse holder is integrated in the power connector on the back panel of the instrument.
The Instrument has a high input noise floor (when connected to host computer by USB)	the USB cable connects the Instrument ground to computer ground, which might inject some unwanted noise to the measurements results. In this case it is recommended to use the Ethernet connection which is galvanically isolated using a UTP Cat 5 or 6 cable (UTP stands for "unshielded twisted pair").
The Instrument performs poorly at low frequencies (below 160 kHz with 50 Ω or below 100 Hz with 1 MΩ coupling) :	the signal inputs of the instrument might be set to AC operation. Please verify to turn off the AC switch in the Lock-in Tab or In / Out Tab.
The Instrument performs poorly during operation	the demodulator filters might be set too wide (too much noise) or too narrow (slow response) for your application. Please verify if the demodulator filter settings match your frequency versus noise plan.
The Instrument performs poorly during operation	clipping of the input signal may be occurring. This is detectable by monitoring the red LEDs on the front panel of the instrument or the Input Overflow (OVI) flags on the STATUS_TAB of the user interface. It can be avoided by adding enough margin on the input range setting (for instance 50% to 70% of the maximum signal peak).
The Instrument performs strangely when working with the UHF-MF Multi-frequency Option	it is easily possible to turn on more signal generators than intended. Check the generated Signal Output with the integrated oscilloscope and check the number of simultaneously activated oscillator voltages.
The Instrument performs close to specification, but higher performance is expected	After 2 years since the last calibration, a few analog parameters are subject to drift. This may cause inaccurate measurements. Zurich Instruments recommends re-calibration of the Instrument every 2 years.
The Instrument measurements are unpredictable	Please check the Status Tab to see if there is any active warning (red flag), or if one has occurred in the past (yellow flag).
The Instrument does not generate any output signal	verify that signal output switch has been activated in the Lock-in Tab or in the In / Out Tab.
The Instrument locks poorly using the digital I/O as reference	make sure that the digital input signal has a high slew rate and clean level crossings.
The Instrument locks poorly using the auxiliary analog inputs as reference	the input signal amplitude might be too small. Use proper gain setting of the input channel.
The sample stream from the Instrument to the host computer is not continuous	Check the communication (COM) flags in the status bar. The three flags indicate occasional sample loss, packet loss, or stall. Sample loss occurs when a sampling rate is set too high (the instrument sends more samples than the interface and the host computer can absorb). The packet loss indicates an important failure of the communications to the host computer and compromises the behavior of the instrument. Both problems are prevented by reducing the sample rate settings. The stall flag indicates that a setting was actively changed by the system to prevent UI crash.

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.

2.8. UHFLI Firmware Upgrade Guide

Zurich Instruments LabOne consists of both software that runs on your PC and software (firmware) that runs on the UHFLI instrument. In order for the PC software to operate correctly it is necessary to update the UHFLI instrument's firmware when installing LabOne. This step only has to be performed once. The firmware upgrade process is integrated into the LabOne user interface. Alternatively, the firmware upgrade utility program can be used for this purpose. This guide explain how to upgrade the instrument firmware using this program.

2.8.1. Preparation

In order to upgrade the instrument firmware, you must first take the following steps:

1. Download and install the latest version of LabOne on your PC. Administrator rights are necessary for the installation. Please see the UHF User Manual.
2. Either start the UHF instrument or, if it was already running, switch off and restart the UHF instrument.
3. Connect the UHF to the PC with the LabOne installation via USB cable.

2.8.2. Starting the Firmware Upgrade UHFLI utility

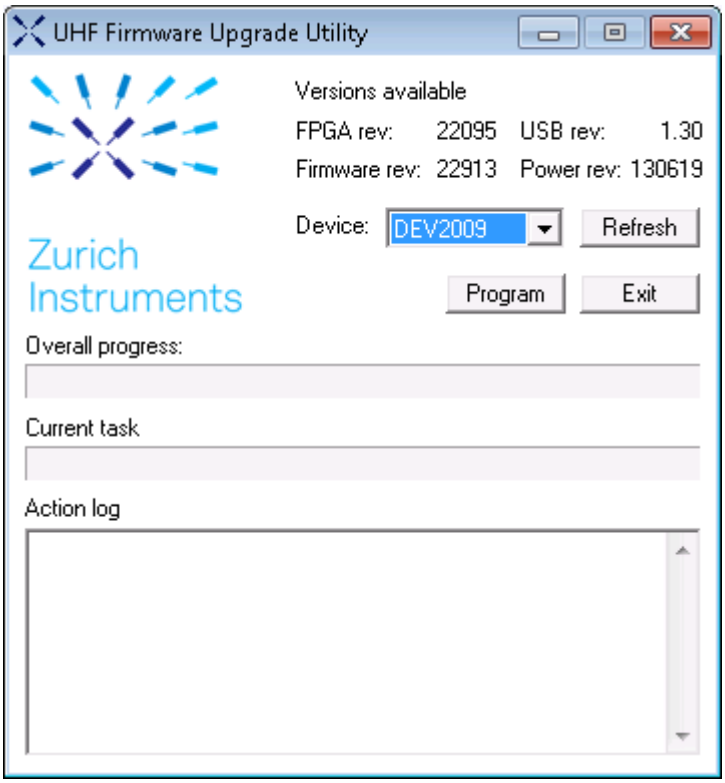
The Firmware Upgrade UHFLI utility is a program with graphical user interface used to perform a firmware upgrade and is included in the LabOne installation. Execute the file `uhfli_flash.exe` in the LabOne installation folder (usually `C:\Program Files\Zurich Instruments\LabOne\DataServer\UHF_Flash`).

2.8.3. Using the Firmware Upgrade UHFLI utility

Important

Do not disconnect the USB cable to the UHF instrument or power-cycle the instrument whilst performing any of the following steps.

Upon starting the Firmware Upgrade UHFLI utility it should detect the device that is connected to the PC via USB. The device serial number is displayed next to `Device:`.



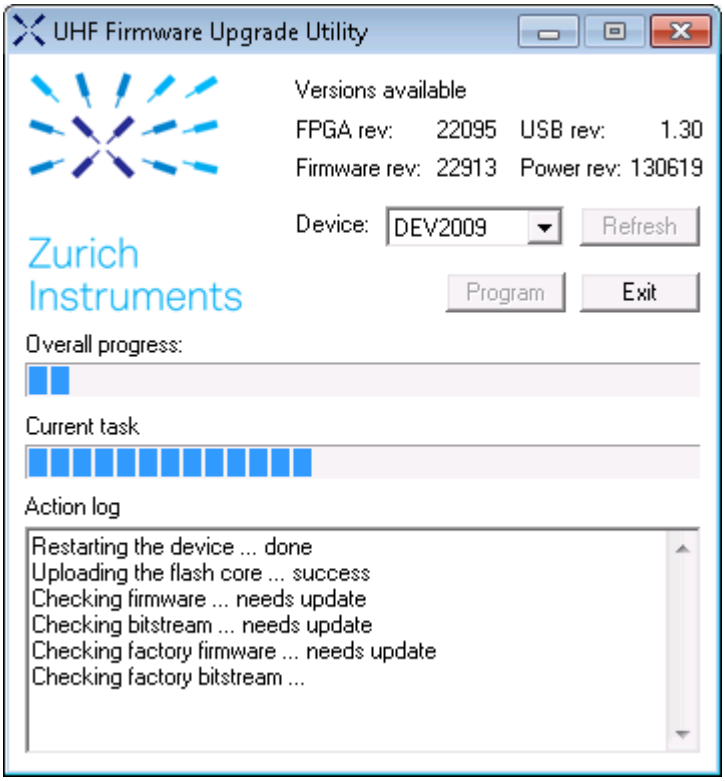
2.8.4. Select the device you would like to upgrade

Select which device you would like to upgrade via the pull-down menu. If no device is listed, please try the following steps:

1. Ensure that the USB cable is properly connected.
2. Try power-cycling the device.
3. Click the "Refresh" button.

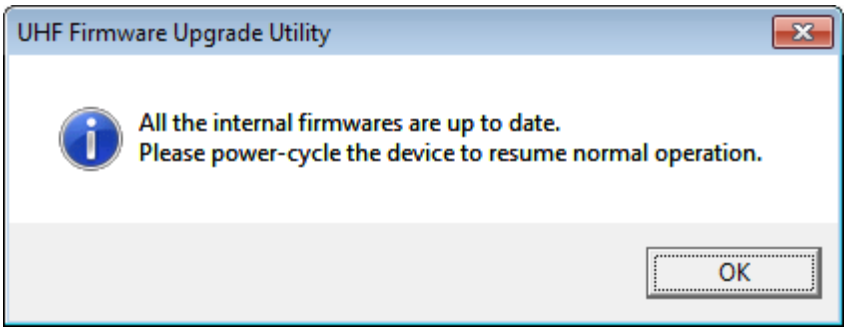
2.8.5. Program the firmware of the connected device

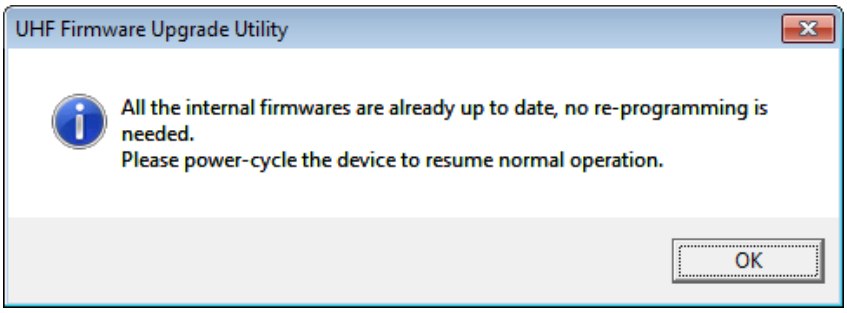
Click the "Program" button to check the version of the current firmware and install the new firmware on the device.



Important

After clicking "Program" and the update is finished it is always necessary to power-cycle the instrument to resume normal operation, even if the firmware was previously up-to-date.





2.8.6. Close the Firmware Upgrade UHFLI utility

Click the `Exit` button to close the Firmware Upgrade UHFLI utility.

2.8.7. Support

If you encounter any issues whilst upgrading the instrument firmware, please contact: support@zhinst.com.

3. Functional Overview

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

3.1. Features

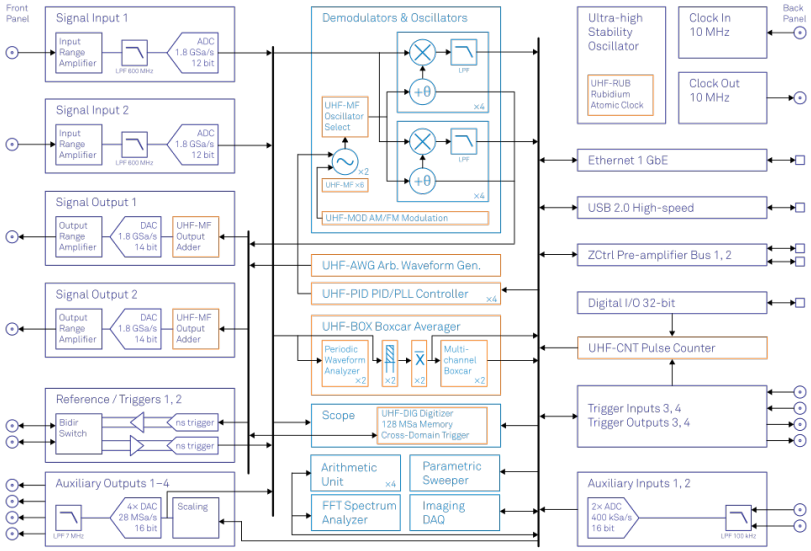


Figure 3.1: UHF instrument functional diagram

The UHF Instrument according to Figure 3.1 consists of several internal units (light blue color) surrounded by several interface units (dark blue color) and the front panel on the left-hand side and the back panel on the right-hand side. The arrows between the panels and the interface units indicates selected physical connections and the data direction flow.

The orange blocks are optional units that can be either ordered at the beginning or upgraded later. The UHF Series comprises two instrument types (UHFLI Lock-in Amplifier and UHFAWG Arbitrary Waveform Generator) based on the same internal hardware. Ordering Guide details the available upgrade options for each instrument type and whether the corresponding option can be upgraded directly in the field.

The signal to be measured is usually connected to one of the two UHF signal inputs where it is amplified to a defined range and digitized at very high speed. The resulting samples are fed into the digital signal processor consisting of 8 dual-phase demodulators . The output of the digital signal processor flow into a digital interface to be transferred to a host computer (LAN and USB interfaces) or are available on the auxiliary outputs on the front panel of the UHF Instrument . Two low-distortion UHF signal outputs provide the signal generator functionality for lock-in operation, or arbitrary waveform generator output.

The numerical oscillators generate sine and cosine signal pairs that are used for the demodulation of the input samples and also for the generation of the UHF output signals. For this purpose, the Output Adder can generate a linear combination of the oscillator outputs to generate a multi-frequency output signal: digital to analog conversion and signal scaling (range) are supported.

Hardware trigger and reference signals are used for various purposes inside the instrument, such as triggering demodulation, triggering oscilloscope data acquisition, or to generate external reference clocks or triggering signals to other equipment.

Lock-in Operating Modes

- Internal reference mode
- External reference mode
- Auto reference mode
- Dual-lock-in operation (two independent lock-in amplifiers in the same box)

- Triple-harmonic mode (simultaneous measurement at three harmonic frequencies)
- Arbitrary frequency mode (optional, simultaneous measurement at six arbitrary frequencies)

Ultra-high-frequency Signal Inputs

- 2 low-noise UHF inputs, single-ended, 600 MHz bandwidth
- Variable input range
- Switchable input impedance
- Selectable AC/DC coupling

Ultra-high-frequency Signal Outputs

- 2 low-distortion UHF outputs, single-ended, 600 MHz bandwidth
- Variable output range

Demodulators & Reference

- Up to 8 dual-phase demodulators
- Up to 8 programmable numerical oscillators
- Up to 2 external reference signals
- Up to 4 input and up to 4 output trigger signals
- Individually programmable demodulator filters
- 128-bit internal processing
- 64-bit resolution demodulator sample
- 48-bit internal reference resolution

AWG Features

- Sequencing and conditional branching
- Amplitude Modulation mode
- Parametric Sweeper
- High-level programming with LabOne AWG Sequencer
- 4-channel output mode

Auxiliary Input and Outputs

- 4 auxiliary outputs, user defined signals
- 2 auxiliary inputs, general purpose

High-speed Connectivity

- USB 2.0 high-speed 480 Mbit/s host interface
- LAN 1 Gbit/s controller interface
- DIO: 32-bit digital input-output port
- ZCtrl: 2 ports peripheral control
- Clock input connector (10 MHz)
- Clock output connector (10 MHz)

Extensive Time and Frequency Domain Analysis Tools

- Numeric tool
- Oscilloscope
- Frequency response analyzer
- FFT spectrum analyzer
- ZoomFFT spectrum analyzer
- Spectroscope
- Data Acquisition tool

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 BNC connectors and control LEDs are arranged as shown in Figure 3.2 and Figure 3.3 and listed in Table 3.1.

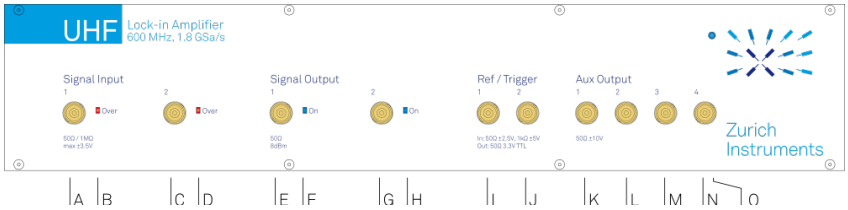


Figure 3.2: UHFLI Lock-in Amplifier front panel

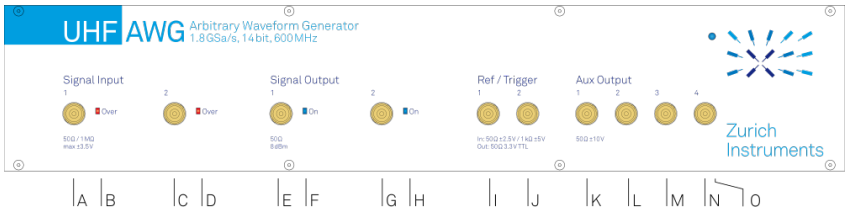


Figure 3.3: UHFAWG Arbitrary Waveform Generator front panel

Table 3.1: UHF Instrument front panel description

Position	Label / Name	Description
A	Signal Input 1	single-ended UHF input
B	Signal Input 1 Over	this red LED indicates that the input signal saturates the A/D converter and therefore the input range must be increased or the signal must be attenuated
C	Signal Input 2	single-ended UHF input
D	Signal Input 2 Over	this red LED indicates that the input signal saturates the A/D converter and therefore the input range must be increased or the signal must be attenuated
E	Signal Output 1	single-ended UHF output
F	Signal Output 1 ON	this blue LED indicates that the signal output is actively driven by the instrument
G	Signal Output 2	single-ended UHF output
H	Signal Output 2 ON	this blue LED indicates that the signal output is actively driven by the instrument
I	Ref / Trigger 1	analog reference input, TTL reference output, AWG marker output, or bidirectional digital TTL trigger
J	Ref / Trigger 2	analog reference input, TTL reference output, AWG marker output, or bidirectional digital TTL trigger
K	Aux Output 1	this connector provides a user defined signal, often used to output lock-in signals X, Y, R, Θ, or to output AWG signals
L	Aux Output 2	this connector provides a user defined signal, often used to output lock-in signals X, Y, R, Θ, or to output AWG signals
M	Aux Output 3	this connector provides a user defined signal, often used to output lock-in signals X, Y, R, Θ, or to output AWG signals
N	Aux Output 4	this connector provides a user defined signal, often used to output lock-in signals X, Y, R, Θ, or to output AWG signals
O	Power	this LED indicates that the instrument is powered
		color blue: the device has an active connection over USB or Ethernet
		color orange: indicates ready to connect. The device is ready for connection over USB or Ethernet. The internal auto calibration process is also indicated by an orange LED

Position	Label / Name	Description
		color orange blinking: device is in start-up mode and waiting for an IP address. As long as the device does not have a dynamic IP address or does use its static default address a connection attempt over Ethernet will fail

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 3.4 and Table 3.2 for the detailed description of the items.

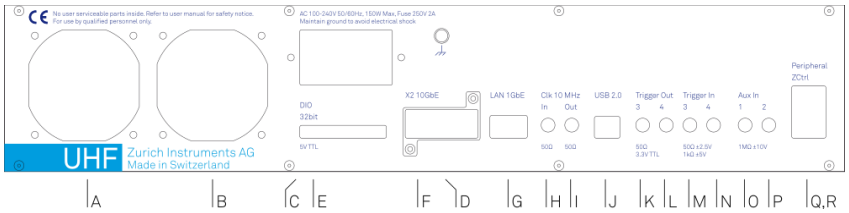


Figure 3.4: UHF Instrument back panel

Table 3.2: UHF Instrument back panel description

Position	Label / Name	Description
A	-	ventilator (important: keep clear from obstruction)
B	-	ventilator (important: keep clear from obstruction)
C	Power inlet	power inlet with ON/OFF switch
D	Earth ground	4 mm banana jack connector for earth ground, electrically connected to the chassis and the earth pin of the power inlet
E	DIO	32-bit digital input/output connector
F	X2 10GbE	10 Gbit LAN connector
G	LAN 1GbE	1 Gbit LAN connector
H	Clk 10 MHz In	clock input (10 MHz) for synchronization with other instruments
I	Clk 10 MHz Out	clock output (10 MHz) for synchronization with other instruments
J	USB	universal serial bus host computer connection
K	Trigger Out 3	digital TTL trigger and AWG marker output - note: some UHF Instruments indicate Trigger 1 on the back panel instead of Trigger 3
L	Trigger Out 4	digital TTL trigger and AWG marker output - note: some UHF Instruments indicate Trigger 2 on the back panel instead of Trigger 4
M	Trigger In 3	digital trigger input - note: some UHF Instruments indicate Trigger 1 on the back panel instead of Trigger 3
N	Trigger In 4	digital trigger input - note: some UHF Instruments indicate Trigger 2 on the back panel instead of Trigger 4
O	Aux In 1	auxiliary input
P	Aux In 2	auxiliary input
Q	ZCtrl 1	peripheral pre-amplifier power & control bus. Attention: this is not an Ethernet plug, connection to an Ethernet network might damage the instrument
R	ZCtrl 2	peripheral pre-amplifier power & control bus. Attention: this is not an Ethernet plug, connection to an Ethernet network might damage the instrument

3.4. Signalling Pathways Diagram

The following diagram illustrates the UHFLI's various signal inputs, signal outputs, functional blocks along with a selection of signalling pathways inside the instrument and towards the host computer.

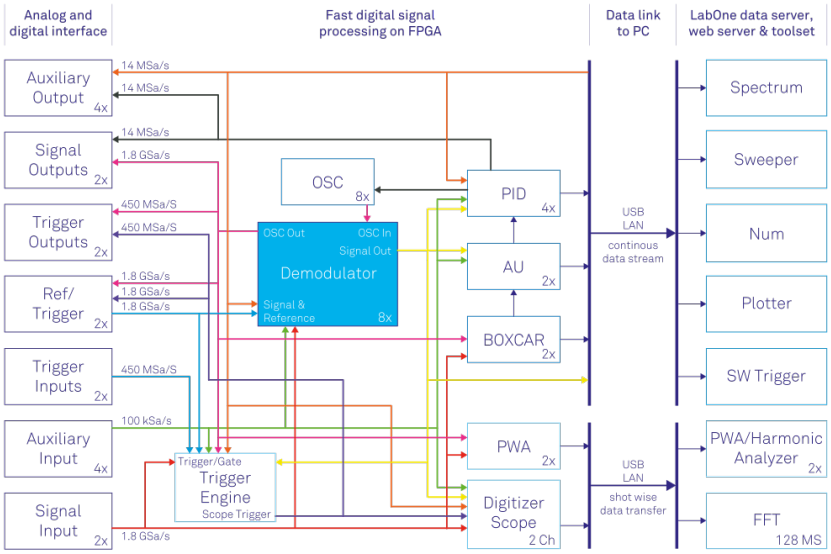


Figure 3.5: UHF instrument main functional blocks and associated signal pathways

3.5. Ordering Guide

The UHF Series is a product line comprising a digital lock-in amplifier UHFLI and an arbitrary waveform generator UHFAWG. Table 3.3 provides an overview of the available UHF products. Upgradeable features are options that can be purchased anytime without need to send the Instrument to Zurich Instruments.

Table 3.3: UHF Instrument product codes for ordering

Product code	Product name	Description	Field upgrade possible
UHFLI	UHFLI Lock-in Amplifier	base lock-in amplifier	-
UHF-PID	UHF-PID Quad PID/PLL Controller	option	yes
UHF-DIG	UHF-DIG Digitizer	option	yes
UHF-MF	UHF-MF Multi-frequency	option	yes
UHF-MOD	UHF-MOD AM/FM Modulation	option	yes[1]
UHF-BOX	UHF-BOX Boxcar Averager	option	yes
UHF-AWG	UHF-AWG Arbitrary Waveform Generator	option	yes
UHF-CNT	UHF-CNT Pulse Counter	option	yes
UHF-RUB	UHF-RUB Rubidium Atomic Clock	option	no
-	-	-	-
UHFAWG	UHFAWG Arbitrary Waveform Generator	base arbitrary waveform generator	yes
UHF-PID	UHF-PID Quad PID/PLL Controller	option	yes[2]
UHF-DIG	UHF-DIG Digitizer	option	yes
UHF-MF	UHF-MF Multi-frequency	option	yes

Product code	Product name	Description	Field upgrade possible
UHF-MOD	UHF-MOD AM/FM Modulation	option	yes[1,2]
UHF-BOX	UHF-BOX Boxcar Averager	option	yes
UHF-LIA	UHF-LIA Lock-in Amplifier	option	yes
UHF-CNT	UHF-CNT Pulse Counter	option	yes
UHF-RUB	UHF-RUB Rubidium Atomic Clock	option	no

[1] Requires UHF-MF Multi-frequency option

[2] Requires UHF-LIA Lock-in Amplifier option

Table 3.4: Product selector UHFLI

Feature	UHFLI	UHFLI + UHF-MF	UHFLI + UHF-PID	UHFLI + UHF-MF + UHF-PID
Internal reference mode	yes	yes	yes	yes
External reference mode	yes	yes	yes	yes
Auto reference mode	yes	yes	yes	yes
Dual-channel operation (2 independent measurement units)	yes	yes	yes	yes
Signal generators	2	2	2	2
Superposed output sinusoidals per generator	1	up to 8	1	up to 8
Quad-harmonic mode	yes	yes	yes	yes
Multi-frequency mode	-	yes	-	yes
Arbitrary frequency mode	-	yes	-	yes
Number of demodulators	8	8	8	8
Simultaneous frequencies	2	8	2	8
Simultaneous harmonics	4+4	-	4+4	-
External references	2	2	2	2
PID controllers	-	-	4	4
600 MHz, 1.8 GSa/s	yes	yes	yes	yes
Dynamic reserve	100 dB	100 dB	100 dB	100 dB
Lock-in range	600 MHz	600 MHz	600 MHz	600 MHz
USB 2.0 480 Mbit/s	yes	yes	yes	yes
LAN 1 Gbit/s	yes	yes	yes	yes

Table 3.5: Product selector UHFAWG

Feature	UHFAWG	UHFAWG + UHF-MF	UHFAWG + UHF-LIA	UHFAWG + UHF-DIG
Dual-channel AWG operation	yes	yes	yes	yes
Sequencing	yes	yes	yes	yes
Amplitude Modulation mode	yes	yes	yes	yes
Direct Output mode	yes	yes	yes	yes
4-channel output mode	yes	yes	yes	yes
External reference mode	yes	yes	yes	yes

Feature	UHFAWG	UHFAWG + UHF-MF	UHFAWG + UHF-LIA	UHFAWG + UHF-DIG
Signal generators	2	2	2	2
Superposed sinusoidals per signal output (Amplitude Modulation mode)	1	up to 4	1	1
Waveform memory	128 MSa	128 MSa	128 MSa	128 MSa
Scope channels	1	1	1	2
Scope memory	65 kSa	65 kSa	65 kSa	128 MSa
Number of demodulators	-	-	8	-
600 MHz, 1.8 GSa/s	yes	yes	yes	yes
Frequency range	600 MHz	600 MHz	600 MHz	600 MHz
USB 2.0 480 Mbit/s	yes	yes	yes	yes
LAN 1 Gbit/s	yes	yes	yes	yes

4. Tutorials

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

Note

For all tutorials, you must have LabOne installed as described in the Getting Started.

- 1 USB 2.0 cable, 1 LAN cable (supplied with your UHF Instrument)
- 3 BNC cables
- SMA cable and adaptors
- 1 male BNC shorting cap (optional)
- 1 oscilloscope (optional)
- 1 BNC T-piece (optional)
- 1 resonator (for the PLL tutorial)

4.1. Simple Loop

Note

This lock-in amplifier tutorial is applicable to all UHFLI instruments and to UHFAWG instruments with the UHF-LI option installed. No other options are required. Some settings depend on whether the UHF-MF Multi-frequency option is installed, and the differences are pointed out where necessary.

4.1.1. Goals and Requirements

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

There are no special requirements for this tutorial.

4.1.2. Preparation

In this tutorial, you are asked to generate a signal with the UHF Instrument and measure that generated signal with the same instrument. This is done by connecting Signal Output 1 to Signal Input 1 with a short BNC cable (ideally < 30 cm). Alternatively, it is possible to connect the generated signal at Signal Output 1 to an oscilloscope by using a T-piece and an additional BNC cable. Figure 4.1 displays a sketch of the hardware setup.

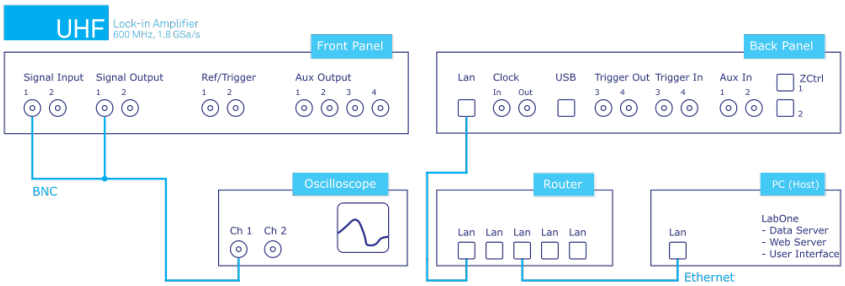


Figure 4.1: Tutorial simple loop setup (LAN connection shown)

Connect the cables as described above. Make sure that the UHF unit is powered on and connected by USB to your host computer or by Ethernet to your local area network (LAN) where the host computer resides. Start the LabOne User Interface from the Windows start menu. The LabOne Data Server and the LabOne Web Server are automatically started and run in the background.

4.1.3. Generate the Test Signal

Perform the following steps in order to generate a 30 MHz signal of 0.5 V peak amplitude on Signal Output 1.

1. Change the frequency value of oscillator 1 (Lock-in tab, Oscillators section) to 30 MHz: click on the field, enter 30000000 or 30 M in short and press either <TAB> or <ENTER> on your keyboard to activate the setting.
2. (Without UHF-MF option) In the Signal Outputs section of the Lock-in tab, set the Range pull-down to 1.5 V, the Offset to 0 V and the amplitude to 500 mV for Output 1.
   (With UHF-MF option) In the Output Amplitudes section of the Lock-in tab, set Amp 1 to 500 mV for demodulator 4 (4th row and 1st column of the Output Amplitudes section) and enable the button next to this field. In the Signal Outputs 1 section set the Range selector to 1.5 V and the Offset to 0 V.
3. By default all physical outputs of the UHF instrument are inactive to prevent damage to connected circuits. Turn on the main output switch by clicking on the button labeled "On" in the Signal Outputs 1 section. The switch turns to dark blue indicating it's enabled.
4. If you have an oscilloscope connected to the setup, you should now be able to see the generated signal.

Table 4.1 and Table 4.2 quickly summarize the instrument settings to be made without and with installed UHF-MF Multi-frequency option.

Table 4.1: Settings: generate the test input signal (without UHF-MF Multi-frequency option)

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Oscillators	1	Frequency	30 MHz
Lock-in	All	Signal Outputs	1	Amplitude	500 mVpk
Lock-in	All	Signal Outputs	1	Offset	0 V
Lock-in	All	Signal Outputs	1	On	ON

Table 4.2: Settings: generate the test input signal (with UHF-MF Multi-frequency option)

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Oscillators	1	Frequency	30 MHz
Lock-in	All	Output Amplitudes	4	Amp 1	500 mVpk
Lock-in	All	Output Amplitudes	4	Amp 1 Enable	ON
Lock-in	All	Signal Outputs	1	Offset	0 V
Lock-in	All	Signal Outputs	1	On	ON

Oscillators and Demodulators are both represented as rows in the Lock-in tab (parameter table, or "All" side tab), but need to be distinguished for a good understanding of the user interface. This is particularly important for users of the UHF-MF Multi-frequency. By default, oscillator 1 is assigned to demodulators 1-4, and oscillator 2 is assigned to demodulators 5-8. This means for example that when generating a signal using row 2 of the Output Amplitudes section, the frequency of this signal depends on row 1 of the Oscillators section (and not row 2) by default.

4.1.4. Check the Test Input Signal

Next, adjust the input parameters range, impedance and coupling to match the values in the following table.

Table 4.3: Settings: configure the Signal Input

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Signal Inputs	1	Range	1 V
Lock-in	All	Signal Inputs	1	Scaling	1 V / V

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Signal Inputs	1	AC	ON
Lock-in	All	Signal Inputs	1	50 Ω	ON

The range setting ensures that the analog amplification on the Signal Input 1 is set such that the dynamic range of the input high-speed analog-digital converter is used optimally without clipping the signal. The graphical range indicator next to the numerical range setting shows about 50% usage of the possible dynamic range.

The incoming signal can now be observed over time in the Scope tab. A Scope view can be placed in the web browser by clicking on the icon in the left sidebar or by dragging the Scope icon to one of the open tab rows. Choose the following settings on the Scope tab to display the signal entering Signal Input 1:

Table 4.4: Settings: configure the Scope

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Scope	Control	Horizontal		Sampling Rate	1.8 GHz
Scope	Control	Horizontal		Length	4 k
Scope	Control	Vertical		Channel 1	Signal Input 1
Scope	Trig	Trigger		Enable	ON
Scope	Trig	Trigger		Level	0 V
Scope				Run / Stop	ON

The Scope now displays single shots of Signal Input 1 with a temporal distance given by the Hold off Time. The scale on top of the graphs indicates the zoom level for orientation. The icons on the left and below the figure give access to the main scaling properties and allow one to store the measurement data as a SVG image file or plain data text file. Moreover, the view can be panned by clicking and holding the left mouse button inside the graph while moving the mouse.

Note

The mouse wheel can be used to zooming in and out horizontally. To zoom vertically, the shift key needs to be pressed while using the mouse wheel.

Having set the Input Range to 1 V ensures that no signal clipping occurs. If you set the Input Range to 0.2 V, clipping can be seen immediately on the scope window accompanied by a red error flag on the status bar in the lower right corner of the LabOne User Interface. At the same time, the LED next to the Signal Input 1 BNC connector on the instrument's front panel will turn red. The error flag can be cleared by pressing the clear button marked with the letter C on the right side of the status bar after setting the Input Range back to 1 V.

The Scope is a handy tool for checking quickly the properties of the input signal in the time and frequency domain. The Scope window can display up to 64 kSa in the basic version, or up to 128 MSa with installed UHF-DIG option. For the full description of the Scope tool please refer to the functional description in Scope Tab.

4.1.5. Measure the Test Input Signal

Now, you are ready to use the UHF instrument to demodulate the input signal and measure its amplitude and phase. You will use two tools of the LabOne User Interface: Numerical and the Plotter.

First, adjust the following parameters on the Lock-in tab for demodulator 1 (or choose another demodulator if desired):

Table 4.5: Settings: measure the test input signal

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Frequencies	1	Harm	1

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Frequencies	1	Phase	0
Lock-in	All	Input	1	Signal	Sig In 1
Lock-in	All	Low-Pass Filters	1	Sinc	OFF
Lock-in	All	Low-Pass Filters	1	Order	3 (18 dB/Oct)
Lock-in	All	Low-Pass Filters	1	TC / BW 3dB	9.3 ms / 8.7 Hz
Lock-in	All	Data Transfer	1	Rate	100 Sample/s (automatically adjusted to 107 Sample/s)
Lock-in	All	Data Transfer	1	Trigger	Continuous
Lock-in	All	Data Transfer	1	Enable	ON

These above settings configure the demodulation filter to the third-order low-pass operation with a 9 ms integration time constant. Alternatively, the corresponding bandwidths BW NEP or BW 3 dB can be displayed and entered. The output of the demodulator filter is read out at a rate of 107 Hz, implying that 107 data samples are sent to the host PC per second with equidistant spacing. These samples can be viewed in the Numerical and the Plotter tool which we will examine now.

The Numerical tool provides the space for 16 or more measurement panels. Each of the panels has the option to display the samples in the Cartesian (X,Y) or in the polar format (R, Θ) plus other quantities such as the Demodulation Frequencies and Auxiliary Inputs. The unit of the (X,Y,R) values are by default given in V_{RMS}. The scaling and the displayed unit can be altered in the Signal Input section of the Lock-in tab. The numerical values are supported by graphical bar scale indicators to achieve better readability, e.g. for alignment procedures. Display zoom is also available by holding the control key pressed while scrolling with the mouse wheel. Certain users may observe rapidly changing digits. This is due to the fact that you are measuring thermal noise that maybe in the µV or even nV range depending on the filter settings. This provides a first glimpse of the level of precision of your UHF instrument.

If you wish to play around with the settings, you can now change the amplitude of the generated signal, and observe the effect on the demodulator output.

Next, we will have a look at the Plotter tool that allows users to observe the demodulator signals as a function of time. It is possible to adjust the scaling of the graph in both directions, or make detailed measurements with 2 cursors for each direction. Signals of the same signal property are automatically added to the same default y-axis group. This ensures that the axis scaling is identical. Signals can be moved between groups. More information on y-axis groups can be found in the section called "Plot Area Elements".

Try zooming in along the time dimension using the mouse wheel or the icons below the graph to display about one second of the data stream. While zooming in, the mode in which the data are displayed will change from a min-max envelope plot to linear point interpolation depending on the density of points along the x axis as compared to the number of pixels available on the screen.

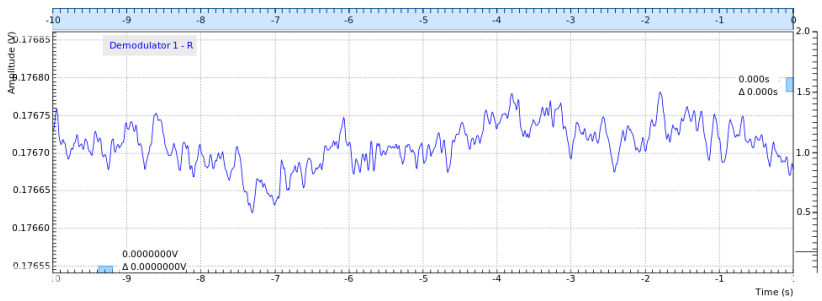


Figure 4.2: LabOne User Interface Plotter displaying demodulator results continuously over time (roll mode)

Data displayed in the Plotter can also be saved continuously to the computer memory. Please have a look at User Interface Overview for a detailed description of the data saving and recording functionality. Instrument and user interface settings can be saved and loaded using the Config Tab (Settings section).

4.1.6. Different Filter Settings

As next step in this tutorial you will learn to change the filter settings and see their effect on the measurement results. For this exercise, use the second demodulator with the same settings as the first except in changing the time constant of the integration to 1 ms which corresponds to a 3 dB bandwidth of 83 Hz.

Table 4.6: Settings: change the demodulator filter settings

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Low-Pass Filters	1	Order	3 (18 dB/Oct)
Lock-in	All	Low-Pass Filters	1	TC / BW 3dB	1 ms / 83 Hz

Increasing the time constant increases the filter integration time of the demodulators. This will in turn "smooth out" the demodulator outputs and hence decrease available time resolution. It is recommended to keep the sample rate 7 to 10 times the filter 3 dB bandwidth. The sample rate will be rounded off to the next available sampling frequency. For example, typing 1 k in the Rate field will result in 1.7 kSa/s which is sufficient to not only properly resolve the signal, but also to avoid aliasing effects. Figure 4.3 shows data samples displayed for the two demodulators with different filter settings described above.

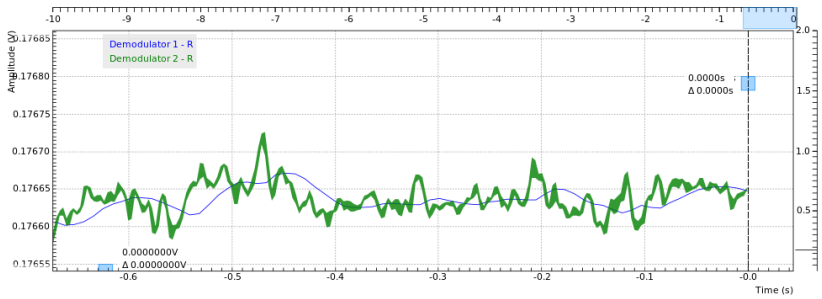


Figure 4.3: LabOne User Interface Plotter: Demodulator 1 (TC = 9.3 ms, blue), Demodulator 2 (TC = 1 ms, green)

Moreover, you may for instance "disturb" the demodulator with a change of test signal amplitude, for example from 0.5 V to 0.7 V and vice-versa. The green plot will go out of the display range which can be re-adjusted by clicking the Auto Scale button , cf. Plot Functionality. With a large time constant, the demodulated data change slower in reaction to the change in the input signal compared to a small time constant. In addition, the number of stable significant digits in the Numerical tab will also be higher with a high time constant.

4.2. External Reference

Note

This tutorial is applicable to all UHFLI instruments and to UHFAWG instruments with the UHF-LI option installed. No other options are required. Some settings depend on whether the UHF-MF Multi-frequency option is installed, and the differences are pointed out where necessary.

4.2.1. Preparation

This tutorial explains how to perform demodulation using an external reference frequency. An external reference will be simulated by using one of the internal oscillators. The signal from this internal oscillator will be fed to one of the signal outputs and then fed back in using various connections in order to reference another internal oscillator used for demodulation.

First of all, connect the Signal Output 2 connector to both Signal Input 1 and to the Ref/Trigger Input 1 connector using two BNC cables and a BNC T-junction. The measurement setup is shown in the following figure.

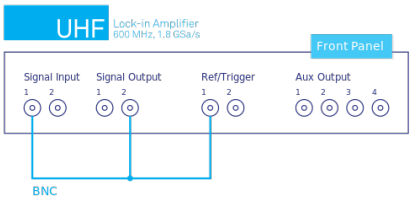


Figure 4.4: External reference on Signal Input 2

Connect the cables as described above. Make sure that the UHF unit is powered on and connected by USB to your host computer or by Ethernet to your local area network (LAN) where the host computer resides. After starting LabOne the default web browser opens with the LabOne graphical user interface.

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

4.2.2. Generate the Reference Signal

In this section you generate a 30.0 MHz signal oscillating between 0 V and +/-0.5 V on Output 2 for use as the external reference. Table 4.7 and Table 4.8 summarize the instrument settings for generating and measuring the reference signal without and with installed UHF-MF Multi-frequency option.

Table 4.7: Settings: generate and measure the reference signal (without UHF-MF Multi-frequency option)

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Signal Outputs	2	Range	1.5 V
Lock-in	All	Signal Outputs	2	Amplitude	1.0 V
Lock-in	All	Signal Outputs	2	Offset	0.0
Lock-in	All	Signal Outputs	2	On	ON
Lock-in	All	Oscillators	2	Frequency	30 MHz
Lock-in	All	Data Transfer	5	Enable	ON
Lock-in	All	Input	2	Range	1.5 V
Lock-in	All	Input	2	AC	ON
Lock-in	All	Input	2	50 Ω	ON

Table 4.8: Settings: generate and measure the reference signal (with UHF-MF Multi-frequency option)

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Output Amplitudes	5	Amp 2	1.0 Vpk
Lock-in	All	Output Amplitudes	5	Amp 2 Enable	ON
Lock-in	All	Signal Outputs	2	Range	1.5 V
Lock-in	All	Signal Outputs	2	Offset	0.0
Lock-in	All	Signal Outputs	2	On	ON
Lock-in	All	Oscillators	2	Frequency	30 MHz
Lock-in	All	Data Transfer	5	Enable	ON
Lock-in	All	Input	2	Range	1.5 V
Lock-in	All	Input	2	AC	ON
Lock-in	All	Input	2	50 Ω	ON

In case the UHF-MF option is installed, note that we use the 5th row of the Output Amplitudes in order to ensure that the generated signal is linked to Oscillator 2. The Oscillator assignment may be changed with the Osc setting in the Demodulators section.

To visualize the signal, we can reconnect the Signal Output 2 with Signal Input 2 and check the signal shape on the Scope using the following settings.

Table 4.9: Settings: configure the Scope

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Scope	Control	Vertical		Channel 1	Signal Input 2
Scope	Trig	Trigger		Trigger	ON
Scope	Trig	Trigger		Signal	Signal Input 2
Scope	Trig	Trigger		Level	0 mV
Scope				Run / Stop	ON

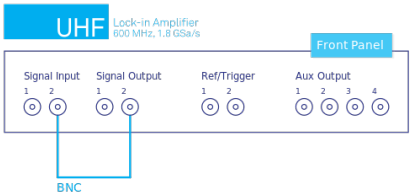


Figure 4.5: External reference on Signal Input 2

The resulting scope trace should look similar as indicated in the following screen capture.

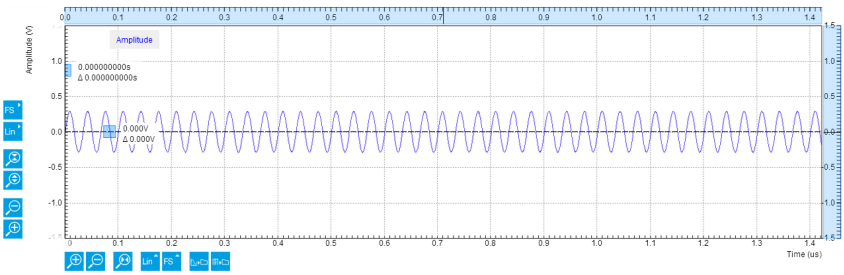


Figure 4.6: Reference signal viewed with the LabOne Scope

Note

Alternatively, the Scope mode Frequency Domain FFT (instead of Time Domain) can be used to check the frequency content of the signal. Set the scale settings automatic for the X axis and logarithmic scale (dB) for the Y axis for convenient viewing. The averaging filter can be set Exp Moving Avg to reduce the noise floor on the display.

4.2.3. Activate the External Reference Mode

After reconnecting the cable as shown in Figure 4.4 the external reference mode can be activated and output the regenerated signal of interest. The following additional settings have to be adjusted:

Table 4.10: Settings: acquire the reference signal

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Data Transfer	1	Enable	ON
Lock-in	All	Signal Input	1	Range	1.2 V
Lock-in	All	Signal Input	1	AC	OFF
Lock-in	All	Signal Input	1	50 Ω	OFF

Demodulator 4 and Demodulator 8 can be set to the external reference mode to track the external reference. The external reference can come from the Signal Input 1 and 2, Ref/Trigger 1 and 2 (in the front), Trigger 3 and 4 (on the back), or Aux In 3 and 4 (on the back). The 4 Auxiliary Outputs can also be chosen in the external reference mode although they are not external references. They are useful in the case of tandem demodulation where the result of a first lock-in operation is fed into a second lock-in, typically at a lower frequency. For this tutorial, Sig In 1 is selected as the external reference for Demodulator 4 (i.e. under the Signal column) and activated by selecting ExtRef in the Mode column.

Table 4.11: Settings: activate external reference mode

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Demodulators Input	4	Signal	Sig In 1
Lock-in	All	Reference	4	Mode	ExtRef

As a result the oscillator 1 frequency indicator in the Oscillator section almost immediately changes from 10 MHz to 30 MHz. Once the external reference mode has been enabled, the frequency of oscillator 1 changes continuously, adapting to the frequency of the external reference signal. This can be verified by changing the frequency of oscillator 2 and noting how the frequency of oscillator 1 follows. A green indicator appears next to the reference selection for channel 1 indicating that the instrument has locked to an external reference. Graphically, this can be visualized in the Plotter by displaying the frequency of Demodulator 1 and then changing the frequency of the oscillator 2 in steps of, say, 1 kHz:

Table 4.12: Settings: displaying demodulator reference frequency over time

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Plotter	Control	Vertical Axis Groups		Tree Selector	/0/sample/Frequency
Plotter				Run / Stop	ON

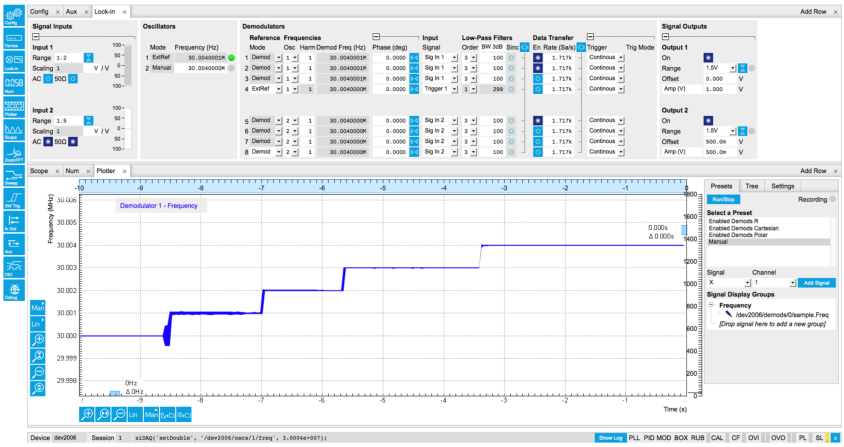


Figure 4.7: LabOne enabling external reference mode

Note that the external reference signal is never used directly for demodulation. Instead, the frequency and phase of the external reference signal is mapped to one of the internal oscillators first through an internal phase-locked loop. This internal oscillator can then serve as a reference for any of the demodulators. This mapping procedure is implemented with an automatic bandwidth adjustment that assures optimum operation over the whole frequency range for a broad variety of signal qualities in terms of frequency stability as well as the signal-to-noise ratio. Over the course of automatic adjustment, the Low-Pass Filter bandwidth of the associated demodulators 4 or 8 usually ramps down until a final value is reached after a few seconds. The indicated bandwidth also marks an upper limit to the bandwidth of the phase-locked loop that does the mapping of the external signal to the internal oscillator. The following figure shows a typical result in the plotter for the frequency tracking immediately after it is turned on.

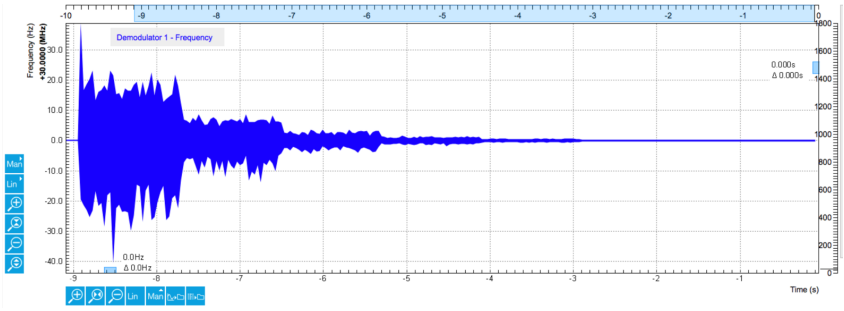


Figure 4.8: Frequency tracking of an external reference signal over time with automatic bandwidth adjustment

4.2.4. Using Ref / Trigger Input and Output for Referencing

In this section you will slightly modify the setup to use Ref / Trigger Input 1 (instrument front panel) as an entry port for the external reference instead of Signal Input 1. A sketch of the modified setup is shown in Figure 4.9.

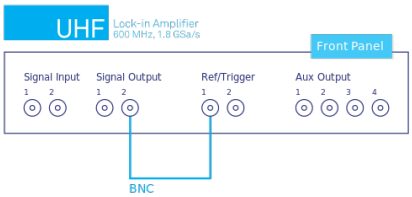


Figure 4.9: External reference using Ref/Trigger Input 1 setup

There are 2 Ref / Trigger inputs on the front panel of the instrument and two more Trigger inputs on the rear panel. By using the dedicated trigger inputs, both Signal Inputs remain available for measurement. The drawback is that one cannot observe the external reference signal on the Scope tool when a Ref / Trigger inputs are used.

Ref/Trigger Inputs are comparator-based digital channels where the input impedance can be set to either 50 Ω or 1 kΩ in the Ref / Trigger section in the DIO Tab. Moreover, a suitable Trigger threshold can be defined there by adjusting the Input Level setting.

Note

To discriminate the two logical states, the Ref/Trigger input operates with a hysteresis of about 100 mV. Consequently, a peak-to-peak signal amplitude of 200 mV or more should be provided to ensure reliable switching.

Note

For signal frequencies larger than 10 MHz, the 50 Ω input termination is recommended to avoid signal reflections that can lead to false switching events.

When the signal is applied with a proper discrimination threshold chosen, the two rectangular control LEDs will both turn on to indicate that the corresponding channel alternates quickly between high and low logical states. Once this is happening, one can then select Trigger 1 as a Signal Input for demodulator 4 in order to reference oscillator 1. The following settings are used for this example:

Table 4.13: Settings: configure the Ref / Trigger input

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DIO		Ref / Trigger	1	Input Level	250 mV
DIO		Ref / Trigger	1	50 Ω	ON
DIO		Ref / Trigger	1	Drive	OFF
Lock-in	All	Demodulators Input	4	Signal	Trigger 1

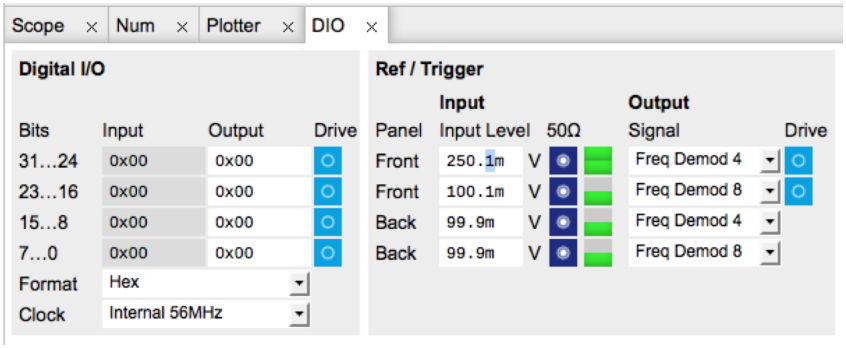


Figure 4.10: Configuring Ref / Trigger 1 as reference input in the DIO tab

The default settings are chosen such that a standard 3.3 V TTL signal can be directly attached without further adjustments. This can be easily tested by generating a TTL reference signal on the Trigger outputs on the back panel. A sketch of the modified setup is shown on Figure 4.11 and the necessary settings are shown in the table below. You should now see as well that the oscillator 1 now tracks the frequency generated from oscillator 2.

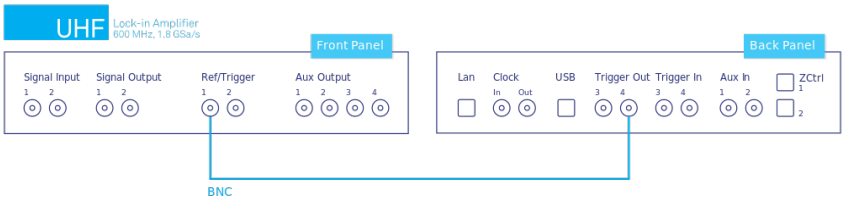


Figure 4.11: Referencing to a TTL signal generated on Trigger Output 3 using Ref/Trigger Input 1

Table 4.14: Settings: generate a TTL reference signal

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DIO		Ref / Trigger	4	Signal	Osc φ Demod 8
DIO		Ref / Trigger	4	Drive	ON

4.3. Amplitude Modulation

Note

This tutorial is applicable to UHF Series Instruments with the UHF-MF Multi-frequency and the UHF-MOD AM/FM Modulation options installed.

4.3.1. Goals and Requirements

This tutorial explains how to generate an amplitude modulated (AM) signal as well as how to demodulate an AM signal by reading out amplitude and phase of the carrier and the two sidebands simultaneously. The tutorial can be done using a simple loop back connection.

4.3.2. Preparation

To perform this tutorial, one simply needs to connect a BNC cable from Signal Output 1 to Signal Input 1 as shown in Figure 4.12. This will allow the user to perform the AM modulation and demodulation in this tutorial without needing an external source.

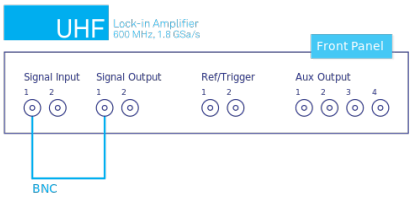


Figure 4.12: Internally generated AM signal measured on Signal Input 1

Connect the cables as described above. Make sure that the UHF unit is powered on and connected by USB to your host computer or by Ethernet to your local area network (LAN) where the host computer resides. After starting LabOne, the default web browser opens with the LabOne graphical user interface.

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

4.3.3. Generate the Test Signal

In this section you will learn how to generate an AM signal with a 10.0 MHz, 1.0 Vpk sinusoidal carrier modulated by a 100 kHz, 200 mVpk sinusoid. The Lock-in tab and the MOD tab settings are shown in the following table.

Table 4.15: Settings: generate the AM signal

Tab	Sub-tab	Section	#	Label	Setting / Value / State
MOD	MOD 1	Oscillators	1	Enable	ON
MOD	MOD 1	Oscillators	1	Carrier	AM / 10.0 M
MOD	MOD 1	Oscillators	1	Sideband 1	100.0 k
MOD	MOD 1	Input	1	Signal	Sig In 1
MOD	MOD 1	Generation	1	Signal Output	1
MOD	MOD 1	Generation	1	Carrier (V)	1.0 / ON
MOD	MOD 1	Generation	1	Modulation (V)	200.0 m / ON
Lock-in	All	Signal Outputs	1	Range	1.5 V
Lock-in	All	Signal Outputs	1	On	ON
Lock-in	All	Signal Outputs	1	50 Ω	OFF
Lock-in	All	Data Transfer	1- 3	Enable	ON
Lock-in	All	Data Transfer	4- 8	Enable	OFF
Lock-in	All	Output Amplitudes	4- 8	Amp 1	OFF
Lock-in	All	Signal Inputs	1	Range	1.5 V
Lock-in	All	Signal Inputs	1	50 Ω	OFF

To quickly verify that the AM signal is generated correctly, we can check the spectrum of the AM signal on Signal Input 1 using the Scope tool with the following settings. The Scope basically displays the FFT spectrum of Signal Input 1. With a sampling rate of 56 MHz, it satisfies sufficiently the Nyquist rate to see the 10 MHz carrier. The 64000 points samples correspond to a shot length of about 1.1 ms. This is enough to capture the frequency spectrum at kHz resolution.

Table 4.16: Settings: acquire the reference signal

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Scope	Control	Horizontal		Mode	Freq Domain FFT
Scope	Control	Horizontal		Sampling Rate	28 MHz
Scope	Control	Horizontal		Length (pts)	64000
Scope				Run/Stop	ON

You should now observe a spectrum like the one shown in the screen capture below. All amplitudes are measured in peak values. The peak at the carrier frequency has an amplitude of about 1.0 V. The two sidebands have amplitudes of 100 mV, i.e. half of the Modulation amplitude 200 mV set in the MOD tab. The factor 0.5 is due to the fact that the original AM modulation signal power is shared between two sidebands. Note that, if we had not disabled the 50 Ω input impedance in the Lock-in tab, we would observe an additional factor 0.5 due to the voltage divider effect from the combination of the Signal Output impedance and the 50 Ω input impedance.

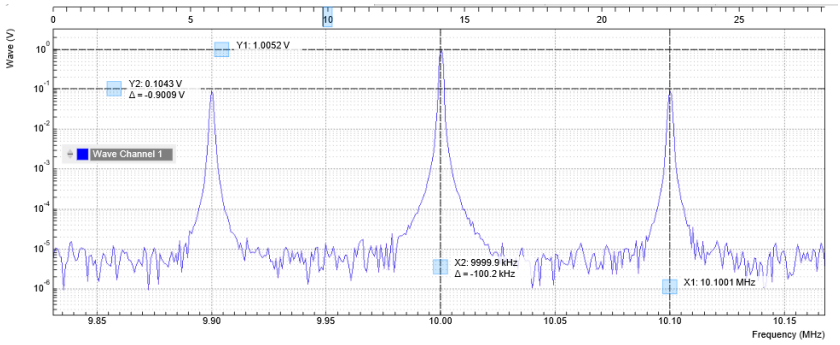


Figure 4.13: Generated AM signal measured with the LabOne Scope in the frequency domain

4.3.4. AM Demodulation Result

If you look at the Demod Freq column in the Lock-in tab, you will see demodulation frequencies of 10 MHz on demodulator 1, 10.1 MHz on demodulator 2 and 9.9 MHz on demodulator 3. You can now read out simultaneously the magnitude and the phase (R, Θ) or (X, Y) of the carrier component on demodulator 1, and the upper and lower sideband components on demodulator 2 and 3, respectively. The measurement result is available in the Numeric tab as shown in Figure 4.14

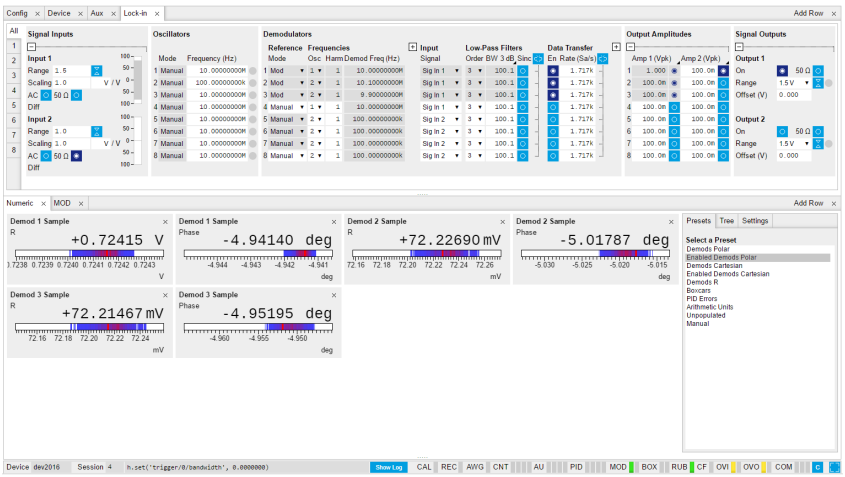


Figure 4.14: Numerical results of AM demodulation in the Numeric tab

Note

By selecting "Enable Demod Polar" in the Numeric tab, only the enabled demodulator outputs will show.

If we take the sum of the double sideband's amplitude (i.e. demodulator 2 and 3) and divide it by the amplitude of the carrier (demodulator 1), we will get an AM modulation index of $h=A_{sidebands}/A_{carrier}=0.2$.

4.4. Phase-locked Loop

Note

This tutorial is applicable to UHF Instruments with the UHF-PID Quad PID/PLL Controller option installed.

4.4.1. Goals and Requirements

This tutorial explains how to track the resonance frequency shift of a resonator using a phase-locked loop (PLL). To follow this tutorial, one needs to connect a resonator between Signal Output 1 and Signal Input 1.

4.4.2. Preparation

Connect the cables as shown in the figure below. Make sure that the UHF Instrument is powered on and connected by USB to your host computer or by Ethernet to your local area network (LAN) where the host computer resides. After starting LabOne the default web browser opens with the LabOne graphical user interface.

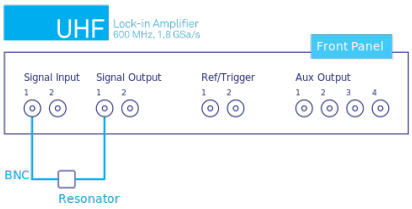


Figure 4.15: PLL connection with UHF Instrument

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

4.4.3. Determine the Resonance of the Quartz

In this section you will learn first how to find the resonance of your resonator with the Sweeper Tab tool. In the Sweeper tab, one can start by defining a frequency sweep across the full instrument bandwidth and narrow down the range using multiple sweeps in order to find the resonance peak of interest. In our case, we know already that the resonance lies at around 1.8 MHz which saves us some time in finding the peak, knowing that its Q factor is rather high. The Sweeper tab and Lock-in tab settings are shown in the table below.

Note

The table below applies to instruments without the UHF-MF Multi-frequency option installed. With the option installed, the output amplitude needs to be configured in the Output Amplitudes section of the Lock-in tab.

Table 4.17: Settings: sweep the measurement frequency

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Signal Outputs	1	Amp (V)	100.0 m / ON
Lock-in	All	Signal Outputs	1	Output 1	ON
Lock-in	All	Signal Inputs	1	50 Ω	ON
Lock-in	All	Signal Inputs	1	Diff	OFF
Lock-in	All	Demodulators	1	Osc	1
Lock-in	All	Demodulators	1	Input	Sig In 1
Lock-in	All	Data Transfer	1	Enable	ON
Sweeper	Control	Horizontal		Sweep Param.	Osc 1 Frequency
Sweeper	Control	Vertical Axis Groups		Signal Type / Channel	Demod Θ / 1
Sweeper	Control	Vertical Axis Groups		Add Signal	click
Sweeper	Control	Vertical Axis Groups		Signal Type / Channel	Demod R / 1
Sweeper	Control	Vertical Axis Groups		Add Signal	click
Sweeper	Control	Horizontal		Start (Hz)	1 M

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Sweeper	Control	Horizontal		Stop (Hz)	3 M
Sweeper	History			Length	2
Sweeper	Control	Settings		Dual Plot	ON
Sweeper	Control	Settings		Run/Stop	ON

We use demodulator 1 to generate the sweep signal and to demodulate the signal transmitted through the resonator. The Lock-in settings ensure that the oscillator used both for the generation and the measurement is the same (oscillator 1). In addition, the input must be set to Signal Input 1 in accordance with the connection diagram.

Once the Sweeper Run/Stop button is clicked, the Sweeper will repeatedly sweep the frequency response of the quartz oscillator. The History Length of 2 allows you to keep one previous sweep on the screen while adjusting the sweep range. You can use the zoom tools to get a higher resolution on the resonance peak. To redefine the start and stop frequencies for a finer sweeper range, just click the Copy From Range button. This will automatically paste the plot frequency range into the Start and Stop fields of the Sweeper frequency range.

Note

The sweep frequency resolution will get finer when zooming in horizontally using the Copy From Range button even without changing the number of points.

When a resonance peak has been found, you should get a measurement similar to the solid lines in the two figures below. The resonance fitting tool allows us to easily determine resonance parameters such as Q factor, center frequency, or peak amplitude. To use the tool, place the two X cursors to the left and right of the resonance, open the Math sub-tab of the Sweeper tab, select "Resonance" from the left drop-down menu, and click on Add. Repeat this operation, once with the demodulator amplitude as the active trace in the plot, and once with the demodulator phase (see Vertical Axis Groups). The tool will perform a least-squares fit to the response function of an LCR circuit. In the limit of large Q factors, this corresponds to a fit to the square root of a Lorentzian function for the amplitude, and to an inverse tangent for the phase. The exact fitting functions are documented in the section called "Cursors and Math".

The fitting curves are added as dashed lines to the plot as shown in Figure 4.16 and Figure Figure 4.17. Since the two fits are independent, they may lead to different results if the resonance significantly deviates from a simple LCR circuit model, which often is the case if there is capacitive coupling between the leads. In this case, the fit to the phase curve which is clearly better than that to the amplitude curve yields a Q factor of about 12,800, and a center frequency of 1.8428 MHz.

The phase in Figure 4.17 follows a typical resonator response going from +90° to −90° when passing through the resonance on a 50 Ω input. Directly at the resonance, the measured phase is close to 0°. We will use this value as a phase setpoint for the PLL. After having completed the Sweeper measurements, turn off sweeping by clicking on Run/Stop. This will release the oscillator frequency from the control by the Sweeper.

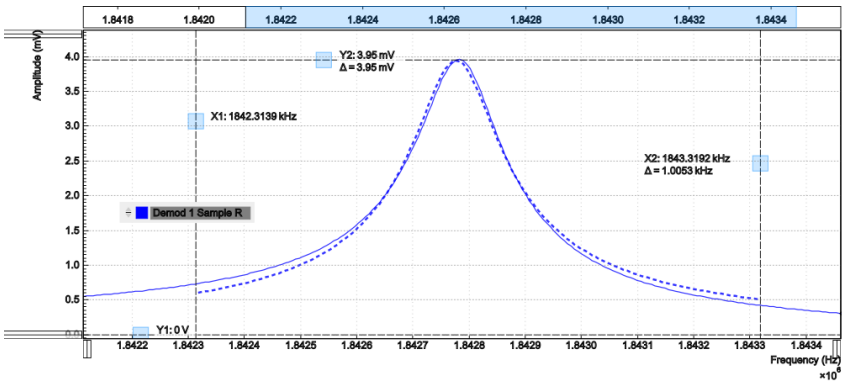


Figure 4.16: Amplitude of the resonator's frequency response measured with the LabOne Sweeper. Solid line are measurement data, dashed line is a fit to the response function of an LCR circuit model using the resonance fitting tool.

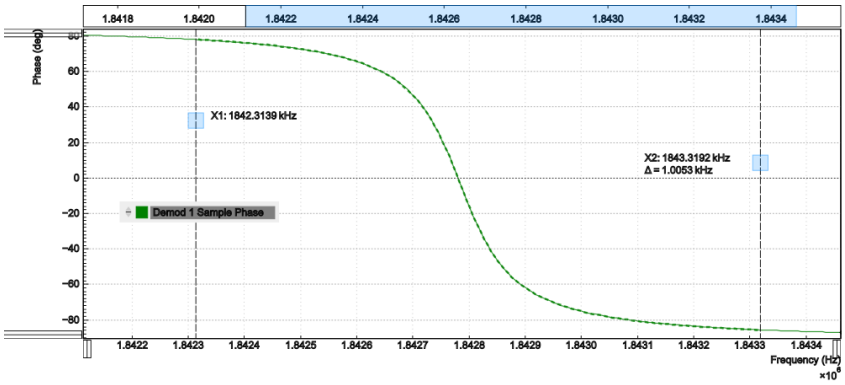


Figure 4.17: Phase of the resonator's frequency response measured with the LabOne Sweeper. Solid line are measurement data, dashed line is a fit to the response function of an LCR circuit model using the resonance fitting tool.

4.4.4. Resonance Tracking with the PLL

Now we know the resonance frequency and the phase measured at this frequency. We can track the drift in resonance frequency by locking on to this phase, hence the name phase-locked loop (PLL). The phase-locked loop is available in the PLL tab. There are four PID (proportional-integral-derivative) controllers in each UHF Series instrument, and the first two have dual use as PLL controllers. For this tutorial, we will use PLL 1. We first set up the basic PLL 1 fields as shown in the table below, using the values from the previous measurement.

Table 4.18: Settings: set up the phase-locked loop

Tab	Sub-tab	Section	#	Label	Setting / Value / State
PLL	PLL		1	Mode	PLL
PLL	PLL		1	Auto Mode	PID Coeff
PLL	PLL	Input	1	Setpoint (deg)	0.0
PLL	PLL	Output	1	Output	Oscillator Frequency / 1
PLL	PLL	Output	1	Center Freq (Hz)	1.8428 M
PLL	PLL	Output	1	Lower / Upper Limit (Hz)	–10k / +10 k

The upper and lower frequency (or range) relative to the Center Frequency should be chosen narrow enough so that the phase of the device follows a monotonous curve with a single crossing at the setpoint, else the feedback controller will fail to lock correctly. We select the 1st oscillator and demodulator 1 for the phase-locked loop operation. Now, we need to find suitable feedback gain parameters (P, I, D) which we do using the Advisor. Set the DUT Model to Resonator Frequency, the Target BW (Hz) to 1.0 kHz, and click on To Advisor to copy the Center Frequency to the resonance frequency field of the Advisor. The target bandwidth should be at least as large as the expected bandwidth of the frequency variations. In the present case, the resonator frequency is practically stable, so 1 kHz bandwidth is largely enough. Click on the Advise button to have the Advisor find a set of feedback gain parameters using a numerical optimization algorithm. Figure 4.18 shows a typical view of the PLL tab after the Advisor has finished. The Advisor tries to match or exceed the target bandwidth in its simulation. The achieved bandwidth can be read from the BW (Hz) field, or directly from the 3 dB point of the simulated Bode plot on the right. The Phase Margin value of the simulation is displayed in the PM (deg) field and should exceed 45° to ensure stable feedback operation without oscillations. Once you are satisfied with the Advisor results, click on the To PLL button to transfer the feedback gain parameters to the physical PLL controller. To start PLL operation, click on the Enable button at the top of the PLL tab.

Table 4.19: Settings: set up and run the PID Advisor

Tab	Sub-tab	Section	#	Label	Setting / Value / State
PLL	Advisor	Advisor	1	Target BW (Hz)	1 k
PLL	Advisor	DUT Model	1	DUT Model	Resonator Frequency
PLL	Advisor	DUT Model	1	Res Frequency (Hz)	1.8 M
PLL	Advisor	DUT Model	1	Q	12.8 k

Tab	Sub-tab	Section	#	Label	Setting / Value / State
PLL	Advisor	Advisor	1	Advise	click

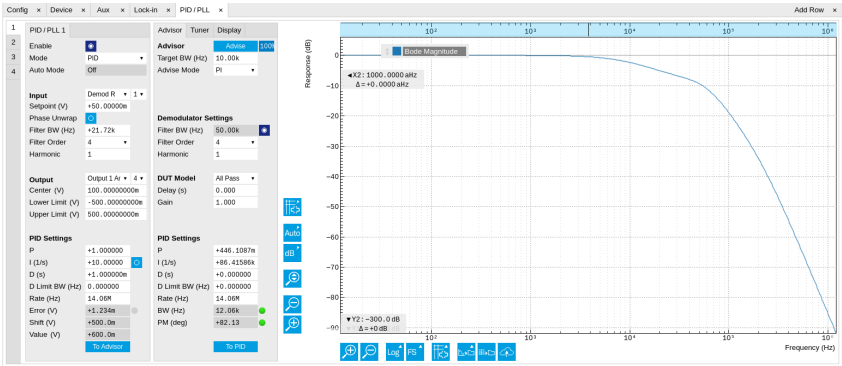


Figure 4.18: Settings and Advisor simulation in the PLL tab (typical – parameters may differ from the example)

When the PLL is locked, the green indicator next to the label Error/PLL Lock will be switched on. The actual frequency shift is shown in the field Freq Shift (Hz).

Note

At this point, it is recommended to adjust the signal input range by clicking the Auto Range button in the Lock-in tab. This often increases the signal-to-noise ratio which helps the PLL to lock to an input signal.

The easiest way to visualize the frequency drift is to use the Plotter tool. The frequency can be added to the display by using the Tree Selector to navigate to Demodulator 1 → Sample and selecting Frequency. The frequency noise increases with the PLL bandwidth, so for optimum noise performance the bandwidth should not be higher than what is required by the experiment. The frequency noise also scales inversely with the drive amplitude of the resonator.

4.5. Automatic Gain Control

Note

This tutorial is applicable to UHF Instruments with the UHF-PID Quad PID/PLL Controller option installed.

4.5.1. Goals and Requirements

This tutorial explains how to set up a PID controller for automatic gain control. We use the PID Advisor to simulate the step response of a feedback loop and the Data Acquisition tool to capture the physical step response of the loop. We perform the test using a quartz resonator between Signal Output 1 and Signal Input 1.

4.5.2. Preparation

Connect the cables as illustrated below. Make sure the UHF Instrument is powered on, and then connect the UHF Instrument through the USB to your PC, or to your local area network (LAN) where the host computer resides. After starting LabOne the default web browser opens with the LabOne graphical user interface.

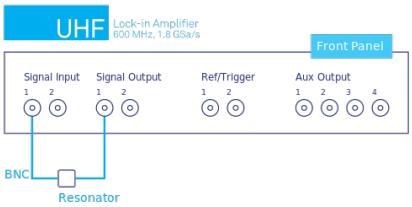


Figure 4.19: PID connection with UHF instrument

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

4.5.3. Automatic Gain Control

In this section you will learn how to control the output amplitude of your device under test with a PID controller. We will use a quartz resonator driven at its resonance frequency by the signal generator of the instrument, and measured with a demodulator.

If you are continuing from the Phase-locked Loop, then you can just leave the PLL enabled. Otherwise, you should know how to generate an excitation signal at the required frequency and how to measure the signal amplitude that you want to control. The device-under-test does not need to be a resonator.

As shown in the frequency response curve below, we are measuring an amplitude of about 4.0 mV at the peak of the resonance while driving with 100 mV_{pk}. The goal is to have this amplitude programmable by the user on the fly.

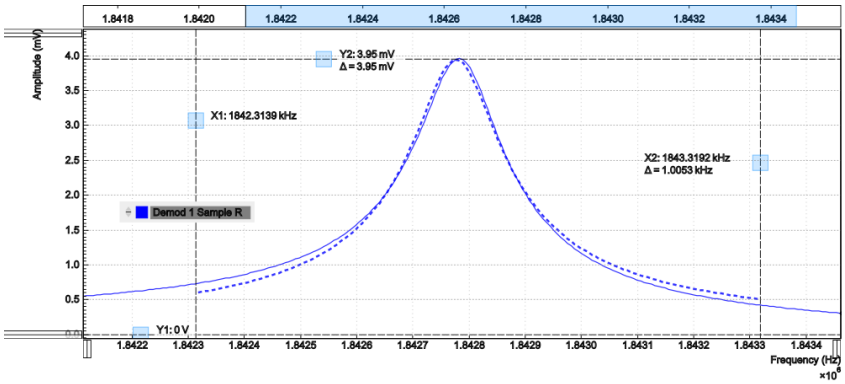


Figure 4.20: Amplitude of the resonator's frequency response measured with the LabOne Sweeper. Solid line are measurement data, dashed line is a fit to the response function of an LCR circuit model using the resonance fitting tool.

For setting up automatic gain control, open the PID / PLL tab in which the four available PID controllers are represented in different side-tabs. Since the first two controllers have a dual use as PLLs, we'll use PID 3 for this tutorial. We'll define the Input of the controller as the measured lock-in R signal, and the Output as the drive amplitude. The settings are shown in the table below.

Note

The table below applies to instruments without the UHF-MF Multi-frequency option installed. With the option installed, the Output 1 Amplitude channel needs to be set to the number of the demodulator used to generate the signal in the Output Amplitudes section of the Lock-in tab.

Table 4.20: Settings: Set up the PID controller

Tab	Sub-tab	Section	#	Label	Setting / Value / State
PID / PLL	PID / PLL		3	Mode	PID
PID / PLL	PID / PLL	Input	3		Demod R / 1
PID / PLL	PID / PLL	Input	3	Setpoint (V)	10 m
PID / PLL	PID / PLL	Output	3		Output 1 Amplitude / 1

Tab	Sub-tab	Section	#	Label	Setting / Value / State
PID / PLL	PID / PLL	Output	3	Center (V)	0.5
PID / PLL	PID / PLL	Output	3	Lower/Upper Limit (V)	−0.5/+0.5
PID / PLL	PID / PLL	Output	3	Range	0.5

The next step is to select the proper feedback gain parameters (P, I, D). On the UHF instrument we can do this with the help of the PID Advisor. Based on a set of mathematical models for the device under test (DUT), it can simulate the step response for a certain set of feedback gain values. The PID Advisor numerically optimizes the feedback gain parameters to obtain a step response that matches or exceeds a user-specified target bandwidth.

The list of available DUT models is found in PID / PLL Tab. In case your DUT is not well described by one of the models, the methods presented here are nonetheless useful to implement certain heuristic tuning method such as the Good Gain method (Finn Haugen, Telemark University College, Norway, 2010), as they enable measurement of the closed-loop step response.

The PID Advisor offers an efficient graphical tool for setting the feedback gain parameters manually. To access it, enable the Advanced Mode in the Display sub-tab and select PID from the Transfer Function menu. Three cursor lines will be added to the display section which represent the frequency dependence of the P, I, and D part of the PID controller transfer function. The cursors can be dragged, allowing you to define a target Bode plot. If you enable the Advisor Link button, the feedback gain parameters derived from the cursors are linked with the simulation parameters from the Advisor from where they can be transferred to the instrument.

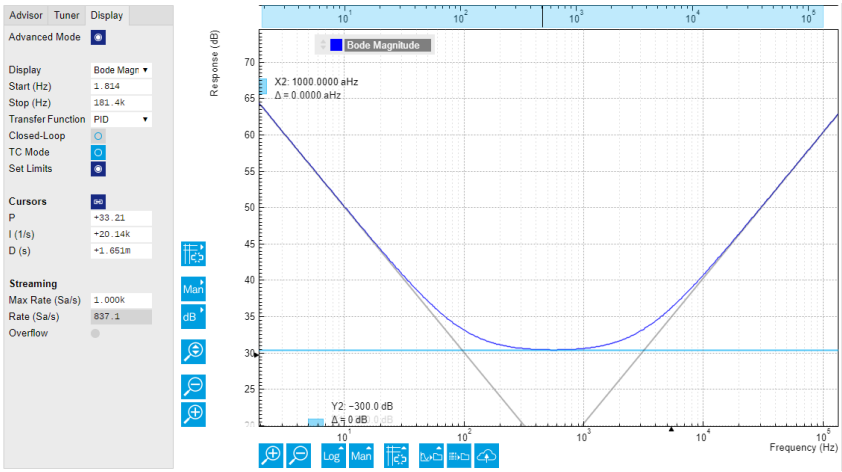


Figure 4.21: Graphical setting of the PID parameters using the cursors. The three cursor lines with negative, zero, and positive slope correspond to the frequency dependence of the P, I, and D parts of the controller, respectively.

4.5.4. Simulating the Device Under Test

In the Advisor sub-tab, select "Resonator Amplitude" as the model of the DUT. This model is characterized by four parameters: delay, gain, center frequency, and Q. The latter two can easily be determined from a frequency response measurement in the Sweeper tab using the resonance fitting tool available in the Math sub-tab as described in Determine the Resonance of the Quartz. We obtain a Q factor of \~12,800 and a center frequency of 1.8428 MHz. The delay value represents extra delays such as those coming from cables (typically 4 to 5 ns per meter). Since we use short cables these are negligible and we can leave the delay parameter at 0 s. The gain value parametrizes overall signal gain or attenuation between PID controller output and input, including unit conversion. In our case, measuring an R amplitude of 4.0 mV_{rms} on resonance while the drive amplitude is set to 100 mV_{pk}, we have a gain of 0.040.

With the Mode selector in the Advisor sub-tab, you can define which of the feedback gain parameters the Advisor uses for his optimization. E.g., when you select PI advise mode, P and I parameters are varied but D is fixed at the value presently set. In this way you can choose the most efficient way of using the Advisor: you can have everything be done by the Advisor, you can control some of the parameters manually and have the Advisor deal with the rest, or you do all the adjustments manually and use the Advisor only to simulate the outcome.

We leave the D parameter at 0 and let the Advisor run in PI mode. Enter a target BW of 1 kHz and click on the Advise button. The Advisor will suggest some values for P and I. The BW field indicates the bandwidth of the simulated loop, with a green lamp showing that the target bandwidth was reached or exceeded. The PM field shows the phase margin, with a green lamp indicating a stable feedback loop.

In the given example, the resonator has a bandwidth of about 140 Hz, so the target bandwidth of 1 kHz is just about within reach. However, in order to reach this value, the corresponding demodulator filter bandwidth may need adjustment. It should be larger than the target bandwidth, but not larger than necessary in order to avoid excessive noise. When enabling Auto Bandwidth (the checkbox next to the Filter BW field in the Demodulator Settings), the PID Advisor selects a suitable demodulator bandwidth which later will be transferred automatically to the demodulator.

The Bode plot on the right-hand side of the tab corresponds to the simulated closed-loop frequency response based on the P, I, and D gain values and the DUT model presently set in the Advisor sub-tab. In order to show the simulated closed-loop step response for our example as in Figure 4.22, set Display to Step Response in the Display sub-tab.

Note

In case a demodulator measurement is selected as the PID input, the Advisor will control the corresponding demodulator filter bandwidth, but not the filter order. If you encounter problems with oscillating feedback, bear in mind that low-order filters often lead to more stable feedback loop behavior because of their smaller delay.

Table 4.21: Settings: set up and run the PID Advisor

Tab	Sub-tab	Section	#	Label	Setting / Value / State
PID / PLL	Advisor	Advisor	3	Target BW (Hz)	1 k
PID / PLL	Advisor	Advisor	3	Advise Mode	PI
PID / PLL	Advisor	Demodulator Settings	3	Filter BW / Auto Bandwidth	ON
PID / PLL	Advisor	DUT Model	3	DUT Model	Resonator Amplitude
PID / PLL	Advisor	DUT Model	3	Delay	0.0 s
PID / PLL	Advisor	DUT Model	3	Gain	0.040
PID / PLL	Advisor	DUT Model	3	Center Frequency	1.8 M
PID / PLL	Advisor	DUT Model	3	Q	12.8 k
PID / PLL	Display		3	Display	Step Response
PID / PLL	Advisor	Advisor	3	Advise	ON

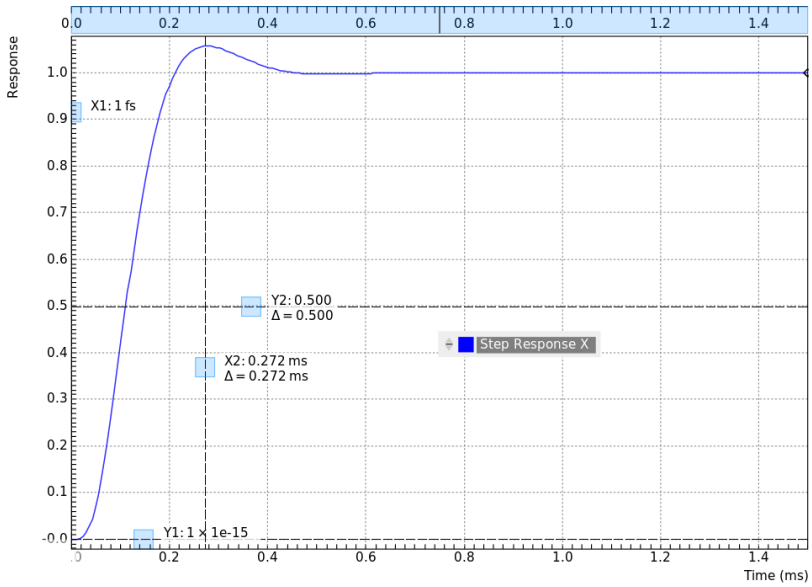


Figure 4.22: Closed-loop step response simulated with the PID Advisor

4.5.5. Measuring the Step Response

Once you are satisfied with the Advisor results, click on the To PID button to transfer the feedback gain parameters to the physical PID / PLL controller represented on the left. Enable the PID / PLL controller and check, e.g. using the Plotter Tab, whether demodulator 1 R has settled at the setpoint of 10 mV. Toggling the setpoint in the PID / PLL tab will then immediately be visible as a step in the Plotter. To capture the step response, the Data Acquisition Tab is the tool of choice. Open the DAQ tab and configure the trigger in the Settings and Grid sub-tabs according to the table below.
.Settings: set up the Data Acquisition tool

Table 4.22: Settings: set up the Data Acquisition tool

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DAQ	Settings	Trigger Settings		Trigger Signal	Demod 1 R
DAQ	Settings	Trigger Settings		Level (V)	11 m
DAQ	Settings	Trigger Settings		Hysteresis (V)	0
DAQ	Settings	Horizontal		Delay (s)	–1 m
DAQ	Grid	Grid Settings		Mode	Linear
DAQ	Grid	Grid Settings		Duration (s)	5 m
Lock-in	All	Data Transfer	1	Rate (Hz) / Enable	100 k / ON

We also increased the demodulator data transfer rate to get a high time resolution for this measurement. Start the Data Acquisition tool by clicking on Run/Stop Any time you toggle the setpoint across the Trigger Level (e.g. from 10 mV to 12 mV), a single trace will be recorded and displayed in the DAQ tab as shown in the figure below.

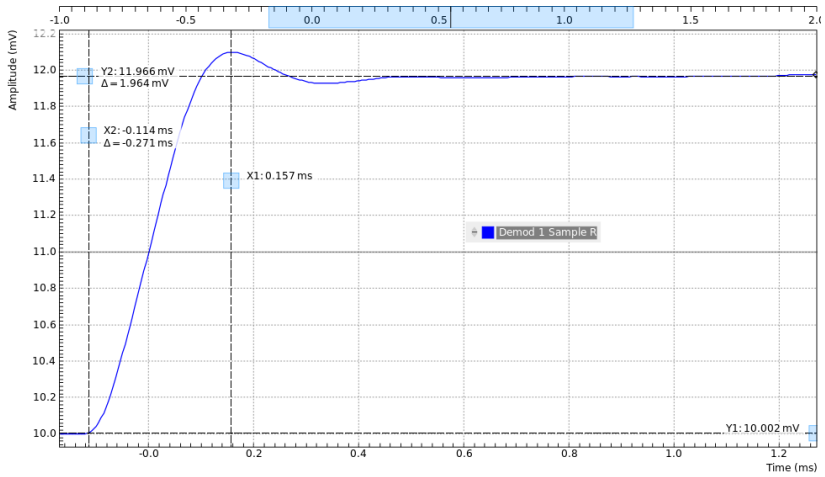


Figure 4.23: Closed-loop step response measured with the Data Acquisition tool

Comparing Figure 4.23 with Figure 4.22 demonstrates the excellent quantitative match between simulation and measurement.

4.6. Imaging

Note

This tutorial is applicable to all UHF Instruments.

4.6.1. Goals and Requirements

This tutorial explains how to capture and display an imaging signal, i.e., a signal structured in lines and frames that can be built up to a 2-dimensional data set. To follow this tutorial, one will require a 3rd-party programmable arbitrary waveform generator to generate a realistic imaging signal with line triggers, or access to a real imaging signal including line triggers or EOL triggers e.g. from an atomic force microscope.

4.6.2. Preparation

Connect the cables as shown in the figure below. Make sure that the UHF Instrument is powered on and connected by USB to your host computer or by Ethernet to your local area network (LAN) where the host computer resides . After starting LabOne the default web browser opens with the LabOne graphical user interface.

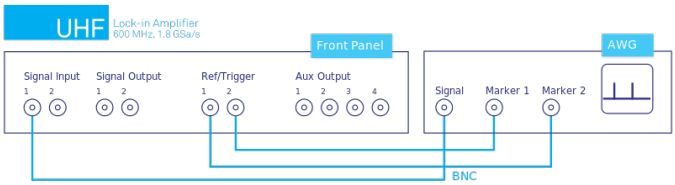


Figure 4.23: Setup for the imaging tutorial with UHF Instrument

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

4.6.3. Imaging Signal Properties

In this section we discuss the properties of the external signal used in this tutorial. It is most illustrative to discuss the imaging functionality based on a realistic signal generated by an arbitrary waveform generator (AWG), or even a real imaging signal. The imaging functionality of the instrument can also be tried out without external equipment, but it's not easily possible to generate a nicely structured imaging signal with the UHF instrument alone. In order to facilitate the phase-locking between AWG and lock-in, an AWG with digital modulation capability and a possibility to output the

phase reference signal separately from the AWG signal is helpful. Examples are the UHFAWG and the HDAWG from Zurich Instruments.

We will assume the following scanning parameters: a line scanning frequency of about 200 Hz and a line number of 256. We will furthermore assume that the imaging signal on the AWG signal output is an amplitude-modulated signal at a fixed carrier frequency of 300 kHz. This signal is wired to the Signal Input 1 connector of the UHF instrument. The carrier phase reference, a square wave at 300 kHz with about 1 V_{pk} amplitude, is generated on the AWG marker output 1 and is connected to the lock-in reference input Ref/Trigger 2. At the start of each line, the AWG generates a rising edge of a TTL signal generated on its marker output 2. This line trigger signal is connected to the Ref/Trigger 1 connector of the UHF instrument. The minimum trigger signal width required to correctly trigger the data acquisition is equal to the inverse demodulator sample rate used. The reason is that the state of the is Ref/Trigger 1 connector is transferred to the host computer together with the demodulator data which limits the time resolution and therefore the minimum trigger pulse width.

4.6.4. Measure the Imaging Signal

For this example, we programmed the AWG to generate a signal with an amplitude varying between 0 and about 0.6 V_{rms} which builds up to an image of the Zurich Instruments logo. We let the AWG run continuously, which means it will permanently generate this signal, the line trigger, and the phase reference signal. Here we will set up the lock-in amplifier with sufficiently high demodulator bandwidth and sampling rate in order to faithfully measure the imaging signal in external reference mode.

For locking to the external reference lock-in reference input, we need to select the reference input signal and change the lock-in amplifier to external reference mode. You can check in the DIO Tab whether the corresponding input connector shows a toggling signal. It may be necessary to adjust the threshold level of the input to match the TTL signal level generated by the AWG. Setting up a measurement in external reference mode is more generally described in External Reference.

Table 4.23: Settings: enable external reference mode

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Demodulators	4	Signal	Trigger 2
Lock-in	All	Demodulators	4	Mode	ExtRef

We choose demodulator filter settings and sampling rate sufficiently high to measure the fast components in the signal up to several 10 kHz. You can find a more general description on selecting filter constants in Simple Loop. The table below shows the settings to be made.

Table 4.24: Settings: configure the demodulator

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Signal Input	1	Range	1.2 V
Lock-in	All	Low-pass Filters	1	BW 3 dB	30 kHz
Lock-in	All	Low-pass Filters	1	Order	8
Lock-in	All	Data Transfer	1	Rate	220 kSa/s
Lock-in	All	Data Transfer	1	Enable	ON

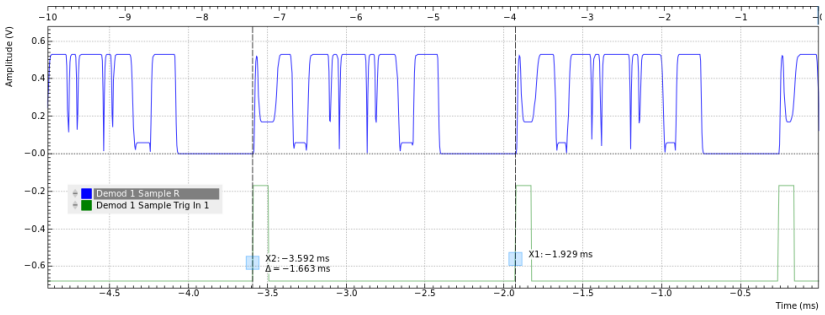
Now we can monitor the imaging signal as well as the line triggers in the Plotter Tab. Open the Plotter tab and add the demodulator R signal as well as Ref/Trigger 1 to the plot.

Table 4.25: Settings: measure imaging signal and line trigger in the Plotter

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Plotter	Control	Vertical Axis Groups		Tree Selector	Demodulators/1/ Sample/R
Plotter	Control	Vertical Axis Groups		Tree Selector	Demodulators/1/ Sample/Trig In 1
Plotter	Control			Run / Stop	ON

The Plotter should now display the continuously streamed imaging data. The figure below shows in blue the demodulator R signal, and in green the line trigger signal marking the beginning of each line. The cursors indicate a line repetition period of about 1.66 ms, and In the following, instead of

displaying these data in a continuous stream in the Plotter, we would like to capture a full image frame.



4.6.5. Set up the Grid Mode

The Data Acquisition Tab with its Grid Mode is the suitable tool to capture images. In this section we go through the configuration of this tool.

The Data Acquisition tool in grid mode acquires 2-dimensional data sets with pre-defined rows and columns that are defined by a trigger timing for each line, a well-defined line number, and a well-defined line duration. The acquired data stream can be linearly interpolated to a well-defined number of data points (e.g. pixels) for each line, or it can be acquired exactly with the transfer rate of the demodulator in exact mode. It furthermore supports averaging over multiple frames.

Here we select the Ref/Trigger 1 signal as trigger source in the Settings sub-tab. We set the hold-off time to 0 s to ensure that no triggers are lost in between successive lines. By changing the delay, we can compensate for a possible misalignment between trigger timing and line start, or to configure the Data Acquisition tool for a line end trigger, rather than line start trigger.

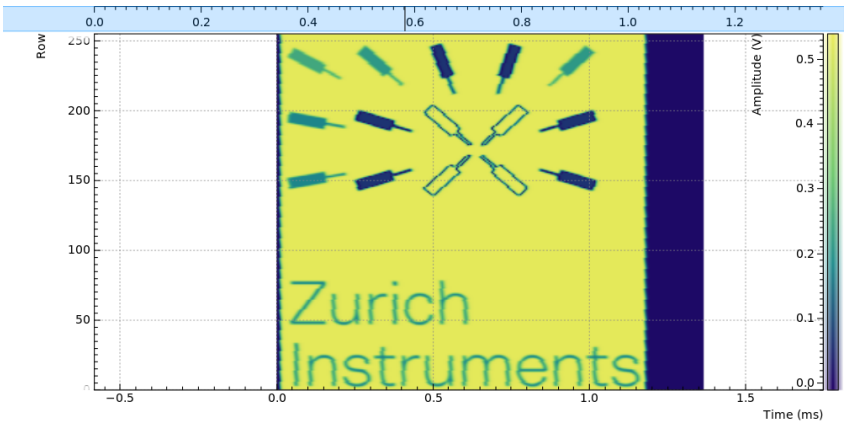
In the Grid sub-tab, we select a number of rows corresponding to what we have programmed on the AWG. In Exact (on-grid) mode, we select the number of columns such that the duration is sufficiently long to capture one line, but shorter than 1 line trigger period so the DAQ tool can re-arm for every new line. Here we select N = 300 columns, corresponding to a duration of T = 1.36 ms. The two numbers are related to the demodulator sampling rate f_s = 220 kSa/s by $T = N/f_s$.

Finally, we select the 2D display in the Control sub-tab, and we make sure to add demodulator 1 R as a displayed signal in the Vertical Axis Groups section. The DAQ tab also supports multi-channel acquisition as more signals, e.g. the phase or other demodulators can be recorded. The table below summarizes the settings.

Table 4.26: Settings: set up the grid mode

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DAQ	Settings	Trigger Settings		Trigger Signal	Demod 1 Trig In 1
DAQ	Settings	Horizontal		Hold off time	0 s
DAQ	Settings	Horizontal		Delay	0 s
DAQ	Grid	Grid Settings		Mode	Exact (on-grid)
DAQ	Grid	Grid Settings		Columns	300
DAQ	Grid	Grid Settings		Duration	1.36 ms (read-only in Exact mode)
DAQ	Grid	Grid Settings		Rows	256
DAQ	Control	Time Domain		Plot Type	2D

In order to capture one fresh frame, we shortly disable the AWG. We arm the Data Acquisition tool by clicking on Single to acquire a single frame with the exact number of rows specified before, and then restart the AWG. The figure below shows the captured image. The acquired data appear as an entry in the History sub-tab and can easily be saved from there.



4.7. PWA and Boxcar Averager

Note

This tutorial is applicable to UHF Instruments with the UHF-BOX Boxcar Averager option installed.

4.7.1. Goals and Requirements

This tutorial explains how to set up a periodic waveform analyzer (PWA) and a boxcar averager for measuring periodic signals with low duty cycles. The advantages of using the PWA and the boxcar averager over a digital scope or a lock-in amplification technique will be explained and demonstrated as follows.

The duty cycle and the signal energy that is available in the fundamental frequency scale almost linearly. For example, a rectangular signal pulse with 50% duty cycle has only 1/3 of the signal amplitude in the fundamental frequency. And if the duty cycle is further halved, then the signal in the fundamental is also halved. Hence, lock-in amplification, which normally references to the fundamental frequency, may not always be the best way to recover a signal if the pulse waveform has a duty cycle smaller than 50%. In this case, boxcar averaging may be the more efficient measurement method. If the signal spreads out over many harmonic components without any prominent peak, a boxcar detection scheme might be the wiser choice to achieve the best possible signal-to-noise ratio.

To perform the measurements in this tutorial, one will require a 3rd-party programmable arbitrary waveform/function generator for narrow pulse generation.

4.7.2. Preparation

Connect the cables as described below. Make sure that the UHF unit is powered on and connected by USB to your host computer or by Ethernet to your local area network (LAN) where the host computer resides. After starting LabOne the default web browser opens with the LabOne graphical user interface.

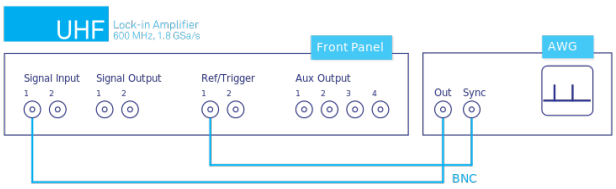


Figure 4.24: UHF connections to an external arbitrary waveform generator

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

4.7.3. Low Duty Cycle Signal Measurement

There are a couple of ways to measure a low duty cycle signal with the UHF instrument. The obvious method is to use the Scope function inside the LabOne interface to observe the sampled signal in the time domain. The other method is to use the PWA and the boxcar averager. Both methods will be shown. The first task is to generate a test signal.

Narrow Pulse Signal Generation

Using the external arbitrary waveform generator, generate a pulse with the following specifications.

Table 4.27: Narrow pulse signal specifications

Pulse Specification	Section
Pulse Type	Square
Amplitude	100 mVpp
Frequency	9.7 MHz
Duty Cycle	< 16%

Note

For this exercise, an Agilent 33500B Truefrom waveform generator is used. The minimum duty cycle for a 9.7 MHz signal on this instrument is about 16%.

The LabOne Scope can be used to observe the generated pulse waveform. Connect the output of the AWG directly to Signal Input 1 of the UHF instrument. The Scope settings in LabOne are given in the table below. Also, the AWG should also be able to provide a TTL synchronization signal to be connected to the Ref / Trigger input. This trigger signal will be used later on for the PWA.

Table 4.28: Settings: observe the pulse waveform

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Signal Inputs	1	AC	OFF
Lock-in	All	Signal Inputs	1	50 Ω	ON
Lock-in	All	Signal Inputs	1	Range	200.0 m
Scope	Control	Vertical		Channel 1	Signal Input 1/ON
Scope	Trig	Trigger		Signal	Signal Input 1/ON
Scope	Trig	Trigger		Enable	ON
Scope	Trig	Trigger		Level	30.0 m
Scope	Trig	Trigger		Hysteresis	10.0 m
Scope				Run/Stop	ON

One should now be able to observe Signal Input 1 similar to the following waveform in the Scope window. The Scope is set to self trigger on the pulse edges. Use the horizontal zoom to focus on a single period. This can be done by rolling the mouse wheel forward to zoom in the horizontal axis. To zoom in on the vertical axis, press down the Shift key and roll the mouse wheel. One can also recenter the waveform by pressing on the left mouse button and dragging the Scope plot area.

One can observe that the shape of the supposedly square pulse does not have sharp edges as one would expect. This is due to the effect of the 600 MHz low pass filter at the input of the UHF instrument. In fact, the signal input bandwidth of 600 MHz corresponds to about 1.5 ns rise time (20% - 80%). Here, the sampled pulse width shown in the Scope is measured to be about 29 ns or 30% duty cycle. The smeared out waveform has a duty cycle bigger than the 16% that was originally set.

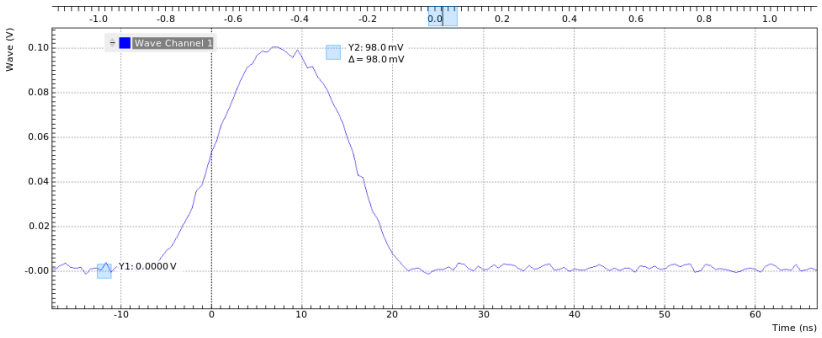


Figure 4.25: Pulse waveform in Scope

Low Duty Cycle Analysis with Period Waveform Analyzer

To analyze the pulse waveform using the PWA, the UHF instrument first has to lock to the trigger signal of the pulses. This is done using the Ext Ref mode. The trigger signal is fed to the Ref / Trigger connector on the front panel which can be an analog signal or a TTL signal. The trigger level can be adjusted in the DIO tab as shown in Using Ref / Trigger Input and Output for Referencing. To lock to the trigger signal, use the Lock-in tab settings in the table below. The goal is to lock the internal oscillator 1 to the external trigger from the AWG. The frequency of oscillator 1 in the Lock-in tab should now display 9.7 MHz, with the green light indicating a lock condition.

Table 4.29: Settings: lock oscillator 1 to external trigger 1

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Reference	4	Mode	ExtRef
Lock-in	All	Demodulators Input	4	Signal	Trigger 1

To activate the PWA function, place one instance of the Boxcar tab in the LabOne web interface. To display the 9.7 MHz pulse over a single period, the following parameters need to be set.

Table 4.30: Settings: activate PWA

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Boxcar	PWA	Signal Input	1	Input Signal	Sig In 1/ON
Boxcar	PWA		1	Run/Stop	ON

Immediately, one can see in the PWA a very stable and smooth peak in one pulse period. The horizontal axis is shown in phase over 360 degrees to represent one period of the pulse waveform. The position of the peak also indicates the precise phase delay with respect to the trigger signal. In this phase representation, the PWA divides the full 360 degrees into 1024 bins. The phase resolution is therefore about 0.35 deg; for a signal of 9.7 MHz this corresponds to a time resolution of about 100 ps.

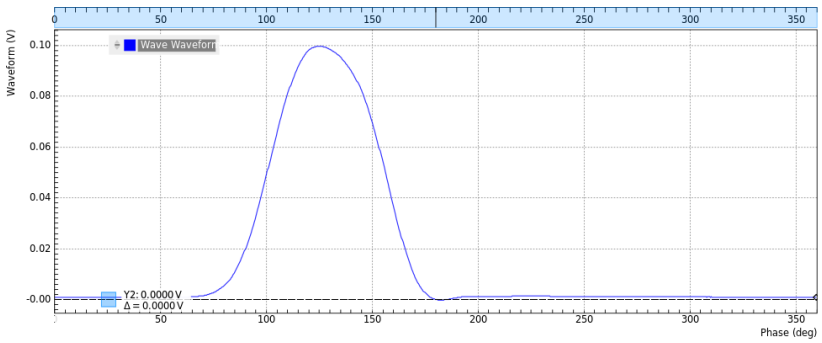


Figure 4.26: Pulse waveform in PWA in the Boxcar tab

If this resolution is not sufficient, one can use the Zoom mode. By reducing the Width (deg), one can get more details of the characteristics of the pulse. The redefined phase range will then again be

divided into 1024 bins. To acquire the same number of samples for a smaller Width requires a longer acquisition time.

Note

The Zoom mode references the input signal to a higher harmonic of the reference frequency which allows zooming into the region of interest, and hence increasing the temporal resolution down to millidegrees. This gives a precise analysis for pulsed signals with low duty cycles or any other periodically repeating transient. Of course the real resolution is still limited by the signal input bandwidth, as in the case of the Scope.

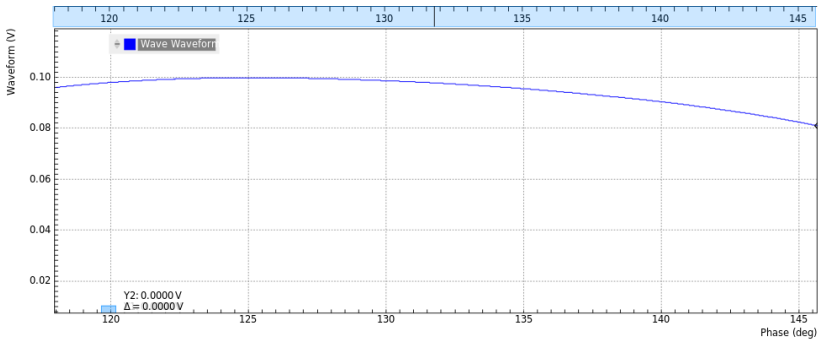


Figure 4.27: Pulse waveform in PWA with a zoom width of 27 degrees

Beside the phase domain display, one can also choose the horizontal display axis in the unit of time or frequency. The harmonics of the pulse waveform can also be analyzed by setting Mode to Harmonics. These options are all part of the multi-channel, multi-domain PWA for peak analysis.

The frequency of 9.7 MHz is not chosen accidentally. In general, one should avoid choosing a modulation frequency that shares the same divisor as the maximum UHF-BOX repetition rate of 450 MHz i.e. the two numbers should not be commensurable. For example, 10 MHz and 450 MHz are commensurable since they can be both divided by 10. This commensurability issue arises from the internal digital signal processing which may cause certain bins to get filled constantly but not others. Such an example is shown in the figure below. A red warning indicator will be switched on when a potential commensurability problem is detected.

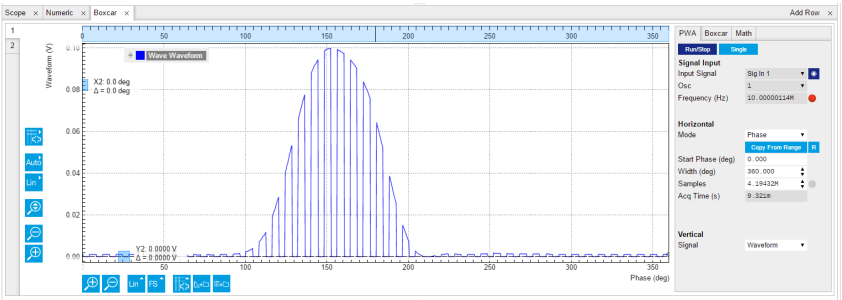


Figure 4.28: Problem of commensurability with the choice of the modulation frequency

Low Duty Cycle Analysis with Scope

The digitized waveform in the Scope can be jittery and noisy. One must remember that the pulse is sampled at 1.8 GSa/s which corresponds to a minimum resolution of 555 ps. This resolution implies that in the zero-crossing triggering, the triggered point on the waveform will not be the same for every pulse. This is indeed one major source of jitter observed.

The Scope comes with averaging and the persistence function which can in theory help to minimize jitter and noise. To use the averaging mode, one simply has to set Avg Filter field under the Scope Control tab to Exp Moving Avg. Then one can choose the number of Averages desired. Below is the averaged pulse waveform at 10 points. Compared to the previous non-averaged waveform, it can be seen that now the spikes are smoothed out.

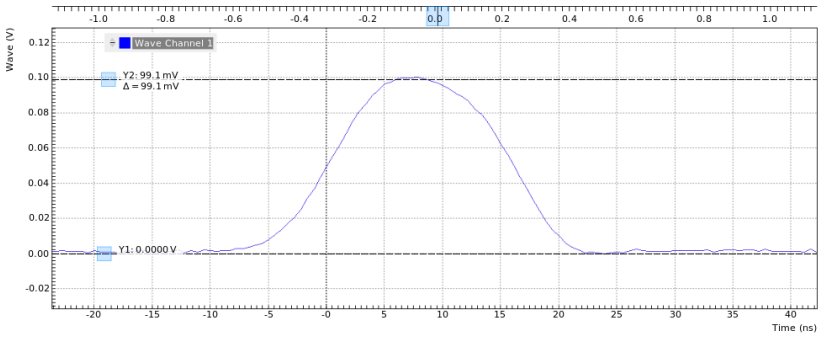


Figure 4.29: Scope waveform with 10 exponential moving averages

In order to observe the extent of jitter and noise, one can use the Persistence mode. Persistence can be enabled in the Advanced sub-tab. Enabling persistence causes each triggered waveform to be superimposed on top of the previous ones. The result of the persistence is shown in the graph below where the superimposed traces are in red. Under this condition, the Scope method can be said to be not an ideal tool to analyze a narrow peak, especially when the peak width would be below a nanosecond.

Note

Persistence cannot be used simultaneously with averaging.

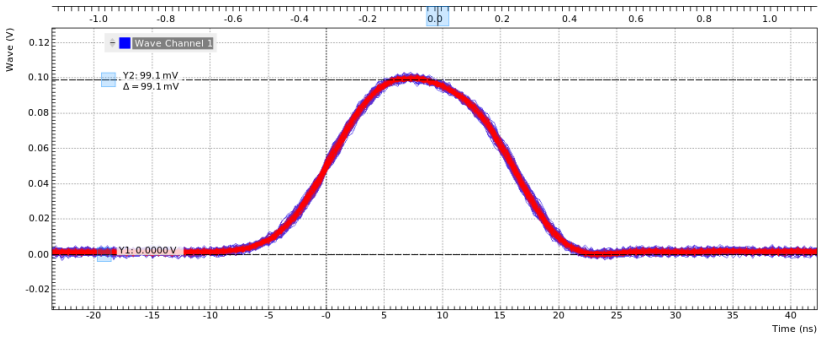


Figure 4.30: Scope waveform with persistence display

Comparison shows that the PWA tool is a more precise and elegant way to analyze this type of narrow pulse waveform.

Boxcar Integration

The boxcar averager can be configured in the Boxcar sub-tab. The boxcar averager integrates and normalizes a section of the signal and generates an output signal in units of volt. The integrated gate can be set either manually in the Start Phase (deg) and Width (deg) fields, or by positioning to vertical cursors and then clicking Copy From Cursor. The integrated value is updated in the Value (V) field. An example boxcar setting is shown below. Choosing a boxcar window roughly equal to the pulse width usually yields the best signal-to-noise ratio.

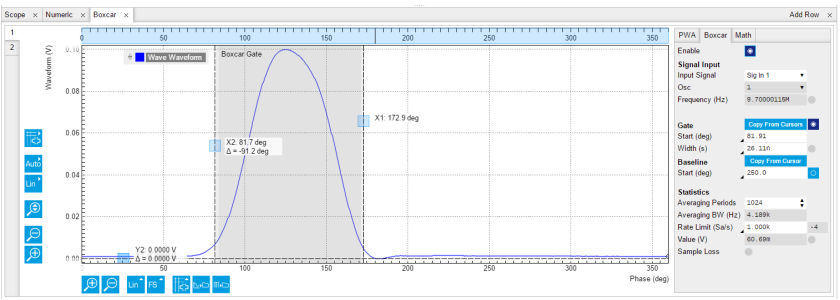


Figure 4.31: Boxcar window setup for measuring the pulse waveform

The result of the integration can also be shown graphically using the Plotter tool as shown below.

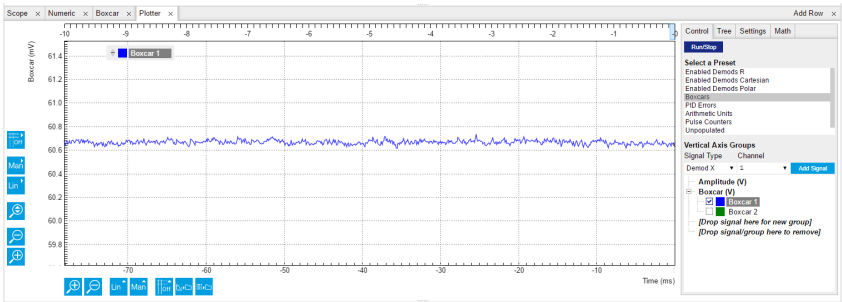


Figure 4.32: Boxcar output signal shown in the Plotter tab

Baseline Suppression

It may happen that a low-frequency noise signal is visible on the boxcar output signal. This noise can come from the power supply, emf noise coupled through the external wirings or even from the experiment itself. In this case, the baseline suppression function can be applied to remove the undesired noise found in the Boxcar integration. To show the benefits of the baseline suppression, the following connections can be made to simulate an undesired period noise injection. In this example, the UHF Signal Output 1 is used to generated a 100 Hz sine wave superimposed on top of the AWG waveform through a T-connector.

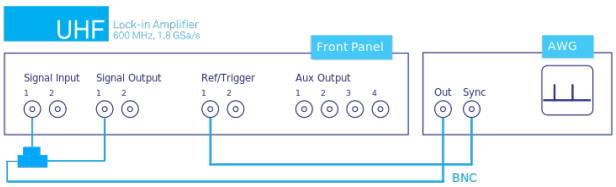


Figure 4.33: UHF instrument connection for baseline subtraction test

Table 4.31: Settings: superpose a sine wave on top of the pulse waveform

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Oscillators	2	Frequency	100 Hz
Lock-in	All	Output Amplitudes	2	Amp 1(Vpk)	0.3
Lock-in	All	Signal Outputs		Output 1	ON

When this is done, the Plotter tool will display an integrated value with the 100 Hz sine component instead of the flat line shown previously.

This undesired sine variation can be eliminated using the Baseline settings in the Boxcar sub-tab. The baseline subtraction window has the same width as the Boxcar integration window. When enabling baseline subtraction, the window position is shown in the PWA waveform just like the integration window. The baseline window should be chosen outside of the pulse in order to subtract only the superimposed sine, and not the signal.

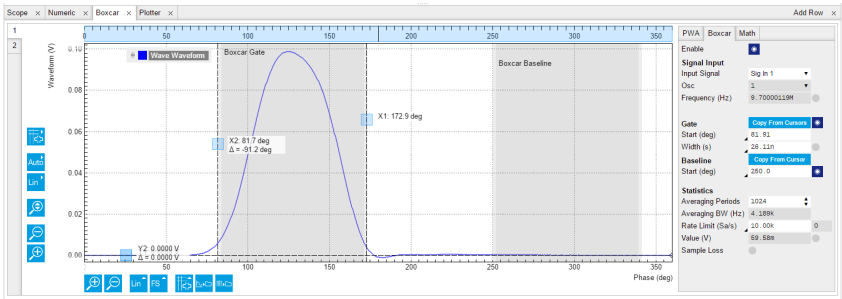
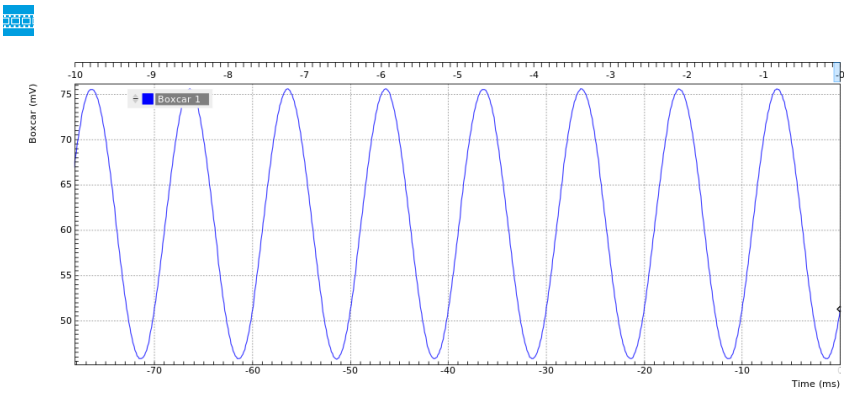
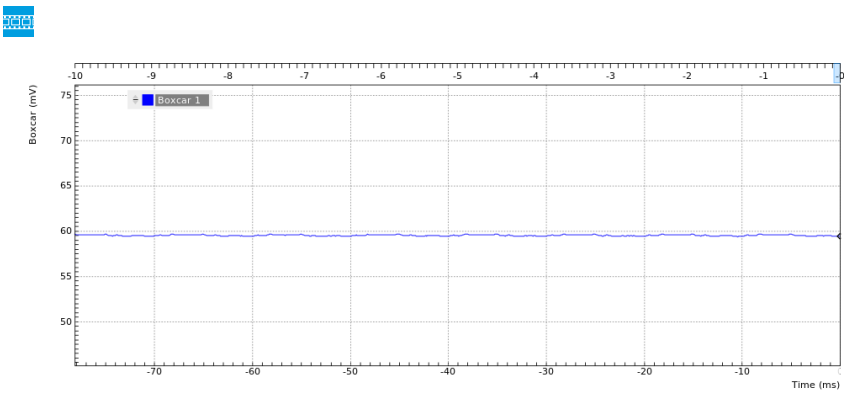


Figure 4.34: Baseline subtraction window setup for measuring the pulse waveform

Once baseline subtraction is enabled and the baseline window is defined suitably, the Plotter will show that the sine component has disappeared. The trace that is left is equal to the original Boxcar averager value.





With enabled Baseline suppression, the operating frequency of the Boxcar Averager is limited to 450 MHz. There is, however, a trick to use it at higher frequencies by combining both Boxcar Averager units. The idea is to measure at exactly half the frequency. By setting the Harmonic of the demodulator used to lock to an external reference to 2, it will generate a sub-harmonic of the external reference on it's associated oscillator (e.g. oscillator 1). Both boxcar averagers and PWAs are then configured to measure with this oscillator reference. The two PWA will each show two periods (i.e. peaks) of the signal. The first Boxcar Averager gate should be centered around the first peak, the second Boxcar Averager gate around the second peak. The baseline suppression gates should be individually selected. The two Boxcar Averager output signals can then be added up in the Arithmetic Unit (AU) tab. This setup corresponds then to a full Boxcar Averager operating above 450 MHz.

4.8. Multi-channel Boxcar Averager

Note

This tutorial is applicable to UHF Instruments with the UHF-BOX Boxcar Averager option installed.

4.8.1. Goals and Requirements

This tutorial explains how to extract the envelope of an amplitude-modulated pulse waveform with the Output PWA tool or multi-channel boxcar averager. More generally, the Output PWA enables measurements of signals that are modulated with two time bases: the fast time base produces the pulses as measured by the boxcar averager, and the slow time base corresponds to a change of the pulse envelope. A typical application would be an amplitude modulated narrow laser pulse waveform.

To follow this tutorial, you need an external arbitrary waveform generator with an external AM modulation capability.

4.8.2. Preparation

Connect the cables as illustrated below. Make sure that the UHF unit is powered on and connected by USB to your host computer or by Ethernet to your local area network (LAN) where the host computer resides. After starting LabOne the default web browser opens with the LabOne graphical user interface.

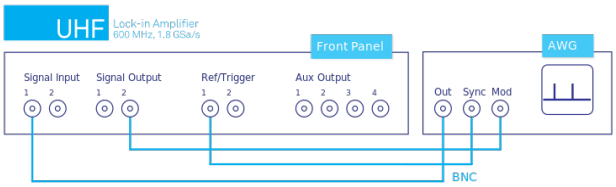


Figure 4.35: UHF connections to an external arbitrary function/waveform generator

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

4.8.3. Amplitude-Modulated Pulse Test Signal Generation

Using the external arbitrary waveform generator, a pulse waveform with the following specification should be generated.

Table 4.32: Narrow pulse signal specifications

Pulse Specification	Value
Pulse Type	Square
Amplitude	100 mVpp
Frequency	9.7 MHz
Duty Cycle	< 16%

Note

An Agilent 33500B Truefrom waveform generator is used in this example. The minimum duty cycle for a 10 MHz signal for this instrument is about 16%. An external amplitude modulation scheme is activated with 100% AM depth.

Furthermore, a sine wave should be generated from the UHF instrument to amplitude modulate the AWG output. The output settings of the UHF instrument are given below.

Table 4.33: Settings: observe the pulse waveform

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Oscillators		Frequency (Hz)	10.0 kHz
Lock-in	All	Signal Outputs	2	Amp (Vpk)	1.5 V
Lock-in	All	Signal Outputs	2	On	ON
Scope	Control	Horizontal		Sampling Rate	28.1 MHz
Scope	Trig	Trigger		Signal	Signal Input 1/ON
Scope	Trig	Trigger		Enable	ON
Scope	Trig	Trigger		Run/Stop	ON

Now, one should be able to see a waveform in Scope that is similar to the one shown below.

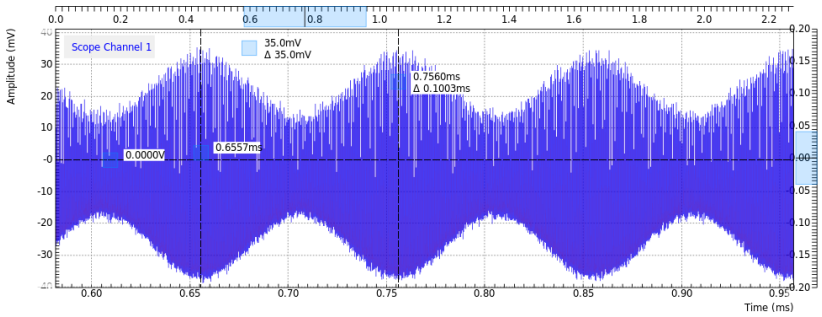


Figure 4.36: Amplitude-modulated pulse waveform measured with the Scope

4.8.4. Envelope Recovery with Output PWA

Just like the previous tutorial in PWA Averager, the PWA can be used to observe the pulse train. Although the measured result is similar to the previous tutorial, one can see in the PWA screen shot below that the peak-to-peak amplitude is no longer 100 mV peak but rather around 50 mV. One has to remember that we have now an amplitude modulated pulse, and the PWA is showing the average amplitude of these pulses over time. If one decreases the number of averages in PWA then the pulse amplitude will start fluctuating.

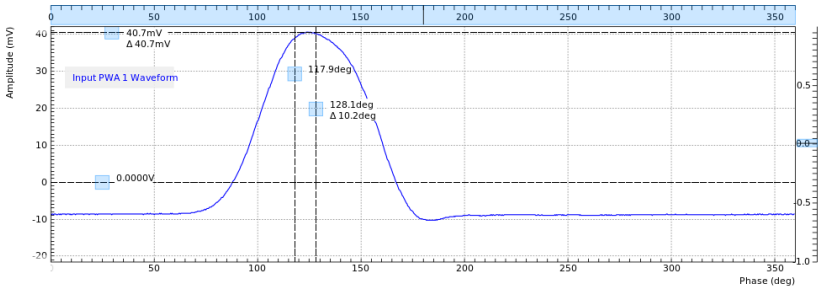


Figure 4.37: Averaged carrier pulse waveform in PWA in the Boxcar tab

As shown previously, the Boxcar averager can be used to obtain the integrated pulse energy over a pre-defined gate width. This integrated value will of course be amplitude modulated as well. The Output PWA is able to recover this envelope of the integrated value. To do this, one now has to place an instance of the Out PWA tab on the LabOne user interface. The settings of the Output PWA are given below.

Table 4.34: Settings: observe the pulse waveform

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Out PWA	Settings	Signal Input	2	Input Signal	Boxcar 1
Out PWA	Settings	Signal Input	2	Osc Select	2
Out PWA			2	Run / Stop	ON

One should be able to observe a sine wave similar to the one shown below. The V magnitude is proportional to the AM modulation depth. One can verify this by changing the AM depth to 50% (see second screen shot). The envelope magnitude indeed decreased by a factor of 2. The Output PWA acts like a multi-channel boxcar. In combination with the UHF-MF option, the Output PWA enables analysis at multiple modulation frequencies.

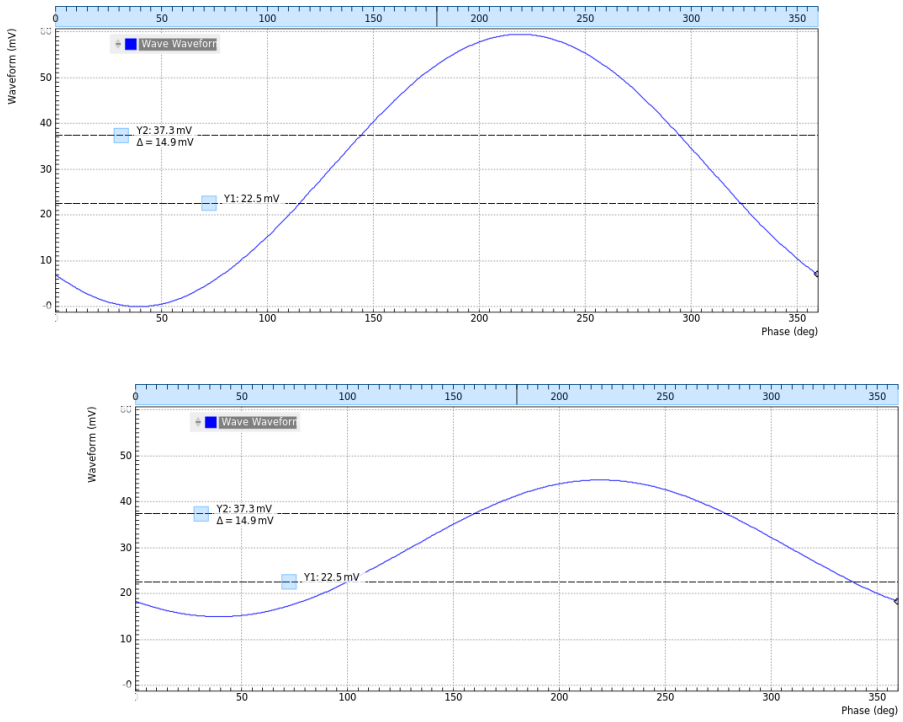


Figure 4.38: AM envelope in Out PWA with 100% and 50% AM depth

4.9. Arbitrary Waveform Generator

Note

This tutorial is applicable to UHFLI Instruments with the UHF-AWG Arbitrary Waveform Generator option installed and to UHFAWG Instruments. Where indicated, additional options such as UHF-DIG, UHF-BOX, UHF-CNT, UHF-MF, or UHF-LIA are required.

4.9.1. Goals and Requirements

The goal of this tutorial is to demonstrate the basic use of the AWG. We demonstrate waveform generation and playback, triggering and synchronization, carrier modulation, and sequence branching. We conclude with a list of tips for operating the AWG. The measurements in this tutorial can be performed using simple loop back connections.

4.9.2. Preparation

Connect the cables as illustrated below. Make sure that the UHF unit is powered on and connected by USB to your host computer or by Ethernet to your local area network (LAN) where the host computer resides. After starting LabOne, the default web browser opens with the LabOne graphical user interface.

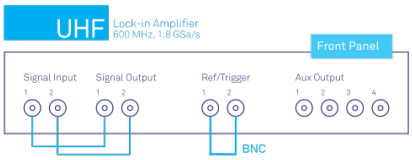


Figure 4.39: UHF connections for the arbitrary waveform generator tutorial

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

4.9.3. Waveform Generation and Playback

In this tutorial we generate arbitrary signals with the AWG and visualize them with the Scope. In a first step we enable the Signal Outputs, but disable all sinusoidal signals generated by the lock-in unit by default. We also configure the Scope signal input and triggering and arm it by clicking on Run/Stop in the Scope. The following table summarizes the necessary settings.

Table 4.35: Settings: enable the output and configure the Scope

Tab	Sub-tab	Section	#	Label	Setting / Value / State
In/Out		Signal Outputs	1	Enable	ON
In/Out		Signal Outputs	2	Enable	ON
Lock-in	All	Output Amplitudes	1-8	Amp 1 Enable	OFF
Lock-in	All	Output Amplitudes	1-8	Amp 2 Enable	OFF
Scope	Control	Vertical		Channel 1	Signal Input 1
Scope	Trigger	Trigger		Enable	ON
Scope	Trigger	Trigger		Signal	Signal Input 1
Scope	Trigger	Trigger		Level	0.1 V
Scope	Control			Run/Stop	ON

Figure 4.40: LabOne UI: AWG tab

In the AWG tab, we configure both channels to output signals at the full scale (FS) in plain output mode as summarized in the following table.

Table 4.36: Settings: configure the AWG output

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG	Control			Rate (Sa/s)	1.8 GHz
AWG	Control			Rerun	OFF
AWG	Control	Output 1		Amplitude (FS)	1.0
AWG	Control	Output 1		Mode	Plain
AWG	Control	Output 2		Amplitude (FS)	1.0
AWG	Control	Output 2		Mode	Plain

Operating the AWG means first of all to specify a sequence program. This can be done interactively by typing the program in the Sequence Editor window. Let's start by typing the following code into the Sequence Editor.

```
wave w_gauss = 1.0*gauss(8000, 4000, 1000);
playWave(1, w_gauss);
```

In the first line of the program, we generate a waveform with a Gaussian shape with a length of 8000 samples and store the waveform under the name `w_gauss`. The peak center position 4000 and the standard deviation 1000 are both defined in units of samples. You can convert them into time by dividing by the chosen Rate (1.8 GSa/s by default). The waveform generated by the `gauss` function has a peak amplitude of 1. This amplitude is dimensionless and the physical signal amplitude is given by this number multiplied with the signal output range (e.g. 1.5 V). We put a scaling factor of 1.0 in

place which can be replaced by any other value below 1. The code line is terminated by a semicolon according to C conventions. In the second line, the generated waveform `w_gauss` is played on AWG Output 1.

Note

For this tutorial, we will keep the description of the Sequencer commands short. You can find the full specification of the LabOne Sequencer language in LabOne Sequence Programming

Note

The AWG has a waveform granularity of 16 samples. It's recommended to use waveform lengths that are multiples of 16, e.g. 8000 like in this example, to avoid having ill-defined samples between successively played waveforms. Other waveform lengths are allowed, though.

If we now click on Save, the program gets compiled. This means the program is translated into instructions for the LabOne Sequencer on the UHF instrument, see AWG Tab. If no error occurs (due to wrong program syntax, for example), the Status LED lights up green, and the resulting program as well as the waveform data is written to the instrument memory. If an error or warning occurs, messages in the Status field will help in debugging the program. If we now have a look at the Waveform sub-tab, we see that our Gaussian waveform appeared in the list. The Memory Usage field at the bottom of the Waveform sub-tab shows what fraction of the instrument memory is filled by the waveform data. The Waveform viewer sub-tab allows you to graphically display the currently marked waveform in the list.

By clicking on Start with Rerun disabled, we have the AWG execute our program once. Since we have armed the Scope previously with a suitable trigger level, it has captured our Gaussian pulse with a FWHM of about 1.33 µs as shown in Figure 4.41.

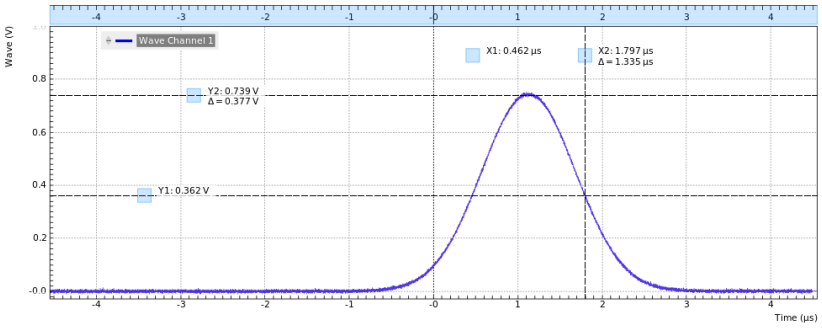


Figure 4.41: Gaussian pulse as generated by the AWG and captured by the LabOne Scope

The LabOne Sequencer language contains various control structures. The basic functionality is to play back a waveform several times. In the following example, all the code within the curly brackets '{...}' is repeated 5 times. Upon clicking Save and Start you should observe 5 short Gaussian pulses in a new scope shot, see Figure 4.42 .

```
wave w_gauss = 1.0 * gauss(640, 320, 50);

repeat (5) {
  playWave(1, w_gauss);
}
```

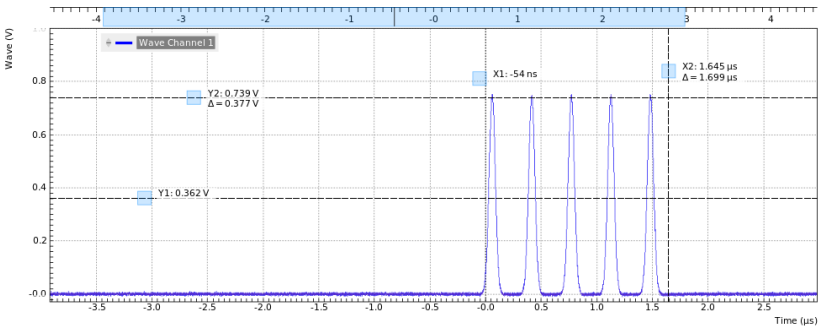


Figure 4.42: Burst of Gaussian pulses generated by the AWG and captured by the LabOne Scope

In order to generate more complex waveforms, the LabOne Sequencer programming language offers a rich toolset for waveform editing. On the basis of a selection of standard waveform generation functions, waveforms can be added, multiplied, scaled, concatenated, and truncated. It's also possible to use compile-time evaluated loops to generate pulse series with systematic parameter variations – see LabOne Sequence Programming for more precise information. In the following code example, we make use of these tools to generate a pulse with a smooth rising edge, a flat plateau, and a smooth falling edge. We use the `cut` function to cut a waveform at defined sample indices, the `rect` function to generate a waveform with constant level 1.0 and length 320, and the `join` function to concatenate three (or arbitrarily many) waveforms.

```
wave w_gauss = gauss(640, 320, 50);
wave w_rise = cut(w_gauss, 0, 319);
wave w_fall = cut(w_gauss, 320, 639);
wave w_flat = rect(320, 1.0);

wave w_pulse = join(w_rise, w_flat, w_fall);

while (true) {
  playWave(1, w_pulse);
}
```

Note that we replaced the finite repetition by an infinite repetition by using a `while` loop. Loops can be nested in order to generate complex playback routines. The Rerun button in the Control sub-tab will simply cause the entire AWG program to be restarted automatically. This is a simple alternative for creating loops, however, unlike the `while` loop the Rerun method does not allow back-to-back waveform playback and comes with a certain timing jitter. The output generated by the program above is shown in Figure 4.43.

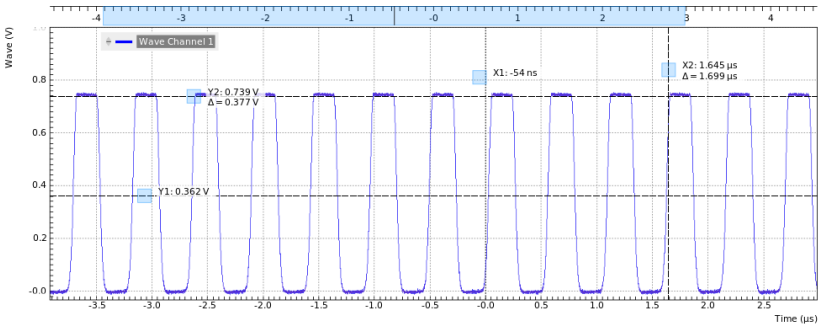


Figure 4.43: Infinite pulse series generated by the AWG and captured by the LabOne Scope

One pitfall when using loops has to do with the nature of the `playWave` and related commands. This command initiates the waveform playback, but during the playback the sequencer will start to execute the next command on his list. This is useful as it allows you to execute commands during playback. In a loop, it means the sequencer can jump back to the beginning of the loop while the waveform is still being played. You can easily change this behavior by adding `waitWave` as the last command in the loop.

As programs get longer, it becomes useful to store and recall them. Clicking on allows Save as... you to store the present program under a newly chosen file name. Clicking on Save then saves your program to the file name displayed at the top of the editor. As you begin to work on sequence programs more regularly, it's worth expanding your repertoire by some of the editor keyboard shortcuts listed in Sequence Editor Keyboard Shortcuts.

It's also possible to iterate over the samples of a waveform array and calculate each one of them in a loop over a compile-time variable `cvar`. This often allows to go beyond the possibilities of using the predefined waveform generation function, particularly when using nested formulas of elementary functions like in the following example. The waveform array needs to be pre-allocated e.g. using the instruction `zeros`.

```
const N = 1024;
const width = 100;
const position = N/2;
const f_start = 0.1;
const f_stop = 0.2;
```

```
cvar i;
wave w_array = zeros(N);
for (i = 0; i < N; i++) {
  w_array[i] = sin(10/(cosh((i-position)/width)));
}

playWave(w_array);
```

Should you require more customization than what is offered by the LabOne AWG Sequencer language, you can import any waveform from a comma-separated value (CSV) file. The CSV file should contain floating-point values in the range from –1.0 to +1.0 and contain one or several columns, corresponding to the number of channels. As an example, the following could be the contents of a file `wave_file.csv` specifying a dual-channel wave with a length of 16 samples:

```
  -1.0   0.0
  -0.8   0.0
  -0.7   0.1
  -0.5   0.2
  -0.2   0.3
  -0.1   0.2
 0.1   0.0
 0.2    -0.1
 0.7    -0.3
 1.0    -0.2
 0.9    -0.3
 0.8    -0.2
 0.4    -0.1
 0.0    -0.1
  -0.5    -0.1
  -0.8   0.0
```

Store the file in the location of `C:\Users\<user name>\Documents\Zurich Instruments\LabOne\WebServer\awg\waves\wave_file.csv` under Windows or `~/Zurich Instruments/LabOne/WebServer/awg/waves/wave_file.csv` under Linux. In the sequence program you can then play back the wave by referring to the file name without extension:

```
playWave("wave_file");
```

If you prefer, you can also store it in a `wave` data type first and give it a new name:

```
wave w = "wave_file";
playWave(w);
```

The external wave file can have arbitrary content, but consider that the final signal will pass through the 600 MHz low-pass filter of the instrument. This means that signal components exceeding the filter bandwidth are not reproduced exactly as suggested for example by looking at a plot of the waveform data. In particular, this concerns sharp transitions from one sample to the next.

In order to obtain digital marker data (see below) from a file, specify a second wave file with integer instead of floating-point values. The marker bits are encoded in the binary representation of the integer (i.e., integer 1 corresponds to the first marker high, 2 corresponds to the second marker high, and 3 corresponds to both bits high). Later in the program add up the analog and the marker waveforms. For instance, if the floating-point analog data are contained in `wave_file_analog.csv` and the integer marker data in `wave_file_digital.csv`, the following code can be used to combine and play them.

```
wave w_analog  = "wave_file_analog";
wave w_digital = "wave_file_digital";
wave w = w_analog + w_digital;
playWave(w);
```

As an alternative to specifying analog data as floating-point values in one file, and marker data as integer values in a second file, both may be combined into one file containing integer values that correspond to the raw data format of the instrument. The integer values in that file should be 16-bit unsigned integers with the two least significant bits being the markers. The values are mapped to 0 ⇒ -FS, 65535 ⇒ +FS, with FS equal to the full scale.

4.9.4. Triggering and Synchronization

Now we have a look at the triggering functionality of the AWG. In this section we will explain how to deal with the most important use cases:

- Triggering the AWG with an external TTL signal
- Generating a TTL signal with the AWG to trigger an external device
- Control the AWG repetition rate by an internal oscillator

We will simulate these situations with on-board means of the UHF instrument for the sake of simplicity, but the inclusion of external equipment is straightforward in practice.

The AWG's trigger channels can be freely linked to a variety of connectors, such as the bidirectional Ref/ Trigger connectors on the front panel, and other functional units inside the instrument, such as the Scope or the Demodulators . This freedom of configuration is enabled by the Cross-Domain Trigger feature and enables triggering. In Combining Signal Generation and Detection we will discuss how to use this possibility to synchronize the detection tools of the UHF platform with the AWG.

Triggering the AWG

In this section we show how to trigger the AWG with an external TTL signal. We start by generating a TTL signal on the (bidirectional) Ref / Trigger 2 connector on the front panel. This simulates a trigger coming from an external device and is entirely independent of the AWG. The TTL signal has the frequency of the internal oscillator 2 which we set to 300 kHz. Apply the settings listed in the following table.

Table 4.37: Settings: generate a 300 kHz TTL signal on Ref / Trigger 2

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DIO		Ref / Trigger	2	Output Signal	Osc φ Demod 8
DIO		Ref / Trigger	2	Drive	ON
Lock-in	All	Oscillators	2	Frequency	300 kHz
Lock-in	All	Demodulators	8	Osc	2

The AWG has 4 trigger input channels. As discussed, these are not directly associated with physical device inputs but can be freely configured to probe a variety of internal or external signals. Here, we link the AWG Digital Trigger 1 to the physical Ref / Trigger 1 connector.

Table 4.38: Settings: configure the AWG analog trigger input

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG	Trigger	Digital Trigger 1		Signal	Trig Input 1
AWG	Trigger	Digital Trigger 1		Slope	Rise

Finally, we modify our last AWG program by including a **`waitDigTrigger`** command just before the **`playWave`** command. The result is that upon every repetition inside the infinite **`while`** loop, the AWG will wait for a rising flank on Ref / Trigger input 1.

```
wave w_gauss = gauss(640, 320, 50);
wave w_rise  = cut(w_gauss, 0, 319);
wave w_fall  = cut(w_gauss, 320, 639);
wave w_flat  = rect(320, 1.0);

wave w_pulse = join(w_rise, w_flat, w_fall);

while (true) {
  waitDigTrigger(1, 1);
  playWave(1, w_pulse);
}
```

Compile and run the above program. Note that this and other programming examples are available directly from a drop-down menu on top of the Sequence Editor. Figure 4.44 shows the pulse series as seen in the Scope: the pulses are now spaced by the oscillator period of about 3.3 µs, unlike

previously when the period was determined by the length of the waveform **w_pulse**. Try changing the oscillator frequency in the Lock-in tab , or unplugging the trigger cable, to observe the immediate effect on the signal.

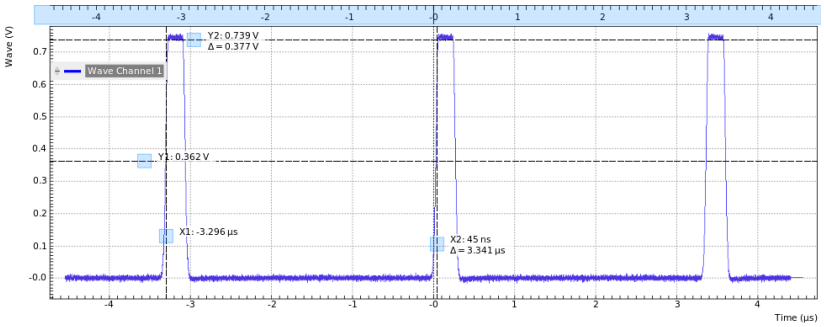


Figure 4.44: Externally triggered pulse series generated by the AWG and captured by the LabOne Scope

Generating Triggers with the AWG

There are two ways of generating trigger output signals with the AWG: as markers, or through sequencer commands.

The method using markers is recommended when precise timing is required, and/or complicated serial bit patterns need to be played on the trigger outputs. Marker bits are part of every waveform which is an array of 16-bit words: 14 bits of each word represent the analog waveform data, and the remaining 2 bits represent two digital marker channels. Upon playback, a digital signal with sample-precise alignment with the analog output is generated.

The method using a sequencer command is simpler, but the timing control is less flexible than when using markers. It is useful for instance to generate a single trigger signal at the start of an AWG program.

Table 4.39: Comparison: AWG markers and triggers

	Marker	Trigger
Implementation	Part of waveform	Sequencer command
Timing control	High	Low
Generation of serial bit patterns	Yes	No
Cross-device synchronization	Yes	Yes

Let us first demonstrate the use of markers. In the following code example we first generate a Gaussian pulse again. The so generated wave does include marker bits – they are simply set to zero by default. We use the **marker** function to assign the desired non-zero marker bits to the wave. The **marker** function takes two arguments, the first is the length of the wave, the second is the marker configuration in binary encoding: the value 0 stands for a both marker bits low, the values 1, 2, and 3 stand for the first, the second, and both marker bits high, respectively. We use this to construct the wave called **w_marker**.

```
const marker_pos = 3000;

wave w_gauss  = gauss(8000, 4000, 1000);
wave w_left   = marker(marker_pos, 0);
wave w_right  = marker(8000-marker_pos, 1);
wave w_marker = join(w_left, w_right);
wave w_gauss_marker = w_gauss + w_marker;

playWave(1, w_gauss_marker);
```

The waveform addition with the '+' operator adds up analog waveform data but also combines marker data. The wave **w_gauss** contains zero marker data, whereas the wave **w_marker** contains zero analog data. Consequentially the wave called **w_gauss_marker** contains the merged analog and marker data. We use the integer constant **marker_pos** to determine the point where the first marker bit flips from 0 to 1 somewhere in the middle of the Gaussian pulse.

Note

The **add** function and the '+' operator combine marker bits by a logical OR operation. This means combining 0 and 1 yields 1, and combining 1 and 1 yields 1 as well.

The following table summarizes the settings to apply in order to output marker 1 on Ref / Trigger 2, and to configure the scope to trigger on Ref / Trigger 1.

Table 4.40: Settings: configure the AWG marker output and scope trigger

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DIO		Output	2	Signal	AWG Marker 1
DIO		Output	2	Drive	ON
Scope	Trigger	Trigger		Signal	Trig Input 1

Figure 4.45 shows the AWG signal captured by the Scope. The green curve shows the second Scope channel (requires UHF-DIG option) configured to display the Trigger Input 1 signal. Try changing the **marker_pos** constant and re-running the sequence program to observe the effect on the temporal alignment of the Gaussian pulse.

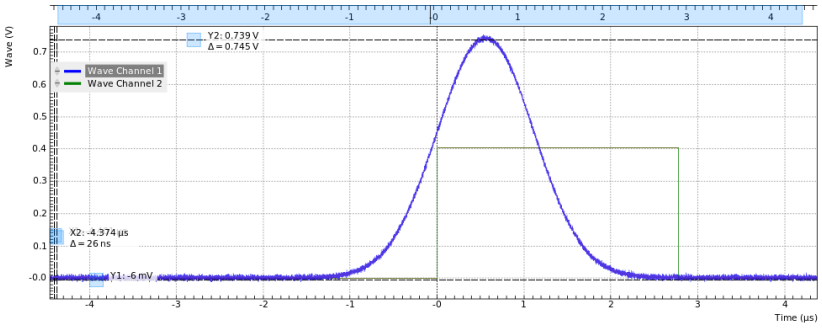


Figure 4.45: Pulse and marker signal generated by the AWG and captured by the LabOne Scope (dual-channel Scope operation requires UHF-DIG option)

Let us now demonstrate the use of sequencer commands to generate a trigger signal. Copy and paste the following code example into the Sequence Editor.

```
wave w_gauss = gauss(8000, 4000, 1000);

setTrigger(1);
playWave(1, w_gauss);
waitWave();
setTrigger(0);
```

The **setTrigger** function takes a single argument encoding the four AWG Trigger output states in binary form – the integer number 1 corresponds to a configuration of 0/0/0/1 for the trigger outputs 4/3/2/1. The binary integer notation of the form 0b0000 is useful for this purpose – e.g. **setTrigger(0b0011)** will set trigger outputs 1 and 2 to 1, and trigger outputs 3 and 4 to 0. We included a **waitWave** command after the **playWave** command. It ensures that the subsequent **setTrigger** command is executed only after the Gaussian wave has finished playing, and not during waveform playback.

We reconfigure the Ref / Trigger 2 connector such that it outputs the AWG Trigger 1, instead of the AWG Marker 1. The rest of the settings can stay unchanged.

Table 4.41: Settings: configure the AWG trigger output

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DIO		Output	2	Signal	AWG Trigger 1

Figure 4.46 shows the AWG signal captured by the Scope. This looks very similar to Figure 4.45 in fact. With this method, we're less flexible in choosing the trigger time, as the rising trigger edge will always be at the beginning of the waveform. But we don't have to bother about assigning the marker bits to the waveform.

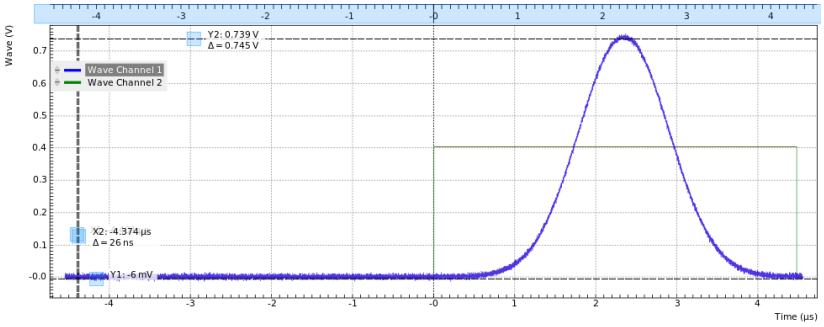


Figure 4.46: Pulse and trigger signal generated by the AWG and captured by the LabOne Scope (dual-channel Scope operation requires UHF-DIG option)

Controlling the AWG Repetition Rate

Finally we show how to synchronize the AWG signal generation with one of the internal oscillators. This enables easy control of the signal repetition rate. It is particularly useful when combining the AWG with synchronous detection methods available on the UHF platform, such as the UHF-LI Lock-in Amplifier, or the UHF-BOX Boxcar Averager.

We achieve this by including a **`waitDemodOscPhase`** command in our Sequencer program. This command works similarly to the **`waitAnaTrigger`** command. In the following example, the AWG will wait in each repetition until the oscillator phase of demodulator 8 passes through zero.

```
wave w_gauss = gauss(640, 320, 50);

while (true) {
  waitDemodOscPhase(8);
  playWave(1, w_gauss);
}
```

The oscillator frequency of demodulator 8 should still be set to 300 kHz from previous examples. Playing the above AWG program produces a signal similar to that shown in Figure 4.44. However, the AWG is now independent of the external trigger signal which simplifies the setup.

4.9.5. Modulation Mode

Generating an I/Q Baseband Signal

One of the key features of the AWG is the ability to work in amplitude modulation mode, where the output of the AWG is multiplied with the amplitude of one or more of the internal oscillator signals of the device. There are numerous advantages to using modulation mode in comparison to simply generating the sinusoidal signal directly using the AWG, such as the ability to change the frequency at will or even control the frequency using the PID/PLL, extremely high frequency resolution independent of AWG waveform length, phase-coherent generation of signals (because the oscillator keeps running even when the AWG is off), ability to analyze input signal at the exact frequency of the generated signal using demodulators, Boxcar and PWA, and more. The goal of this section is to demonstrate how to use the modulation mode.

We design this example around a common use case, which is the generation of dual-channel quadrature (I/Q) modulation signals to feed into a microwave mixer. Such signals require the independent control of two envelope waveforms multiplied by a carrier that is shifted by 90° between the two channels. The program below generates two independent waveforms and plays them repeatedly on both channels. For dual-channel playback we can use the same **`playWave`** function that we used up to now, and simply pass to it two waveforms as arguments. We include the previously used trigger commands for the scope, and include a **`wait`** command whose argument is in units of the sequencer clock period of about 4.44 ns.

```
wave w_gauss = gauss(8000, 4000, 1000);
wave w_drag  = drag(8000, 4000, 1000);

while (true) {
  setTrigger(1);
```

```
  playWave(w_gauss, w_drag);
  waitWave();
  setTrigger(0);
  wait(100);
}
```

For Amplitude Modulation mode, the AWG Output 1 is assigned to the oscillator signal of demodulator 4, and AWG Output 2 is assigned to the oscillator signal of demodulator 8. If the UHF-MF Multi-frequency option is installed, we have the freedom to wire the same oscillator to both demodulators, which is an advantage if we want to control the relative carrier phase of the two AWG Outputs like in this case. Without the UHF-MF option, the two demodulators (and so the two AWG Outputs) are assigned to independent oscillators. In this case, relative phase control is possible but it requires some manual tuning. The following parameter settings apply to the case with installed UHF-MF option.

Table 4.42: Settings: configure the AWG marker output and scope trigger

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DIO		Output	2	Signal	AWG Trigger 1
DIO		Output	2	Drive	ON
Scope	Trigger	Trigger		Signal	Trig Input 1
Lock-in	All	Oscillators	1	Frequency	5 MHz
Lock-in	All	Demodulators	4	Osc	1
Lock-in	All	Demodulators	8	Osc	1
Lock-in	All	Demodulators	8	Phase	0°
Lock-in	All	Demodulators	4	Phase	90°
AWG	Control	Output 1		Mode	Plain
AWG	Control	Output 2		Mode	Plain

Save and play the Sequencer program with the above settings. The upper plot in Figure 4.47 shows the AWG signals captured by the Scope. We see the expected Gaussian pulse on AWG Output 1 (green) and the DRAG pulse, which corresponds to the derivative of a Gaussian function, on AWG Output 2.

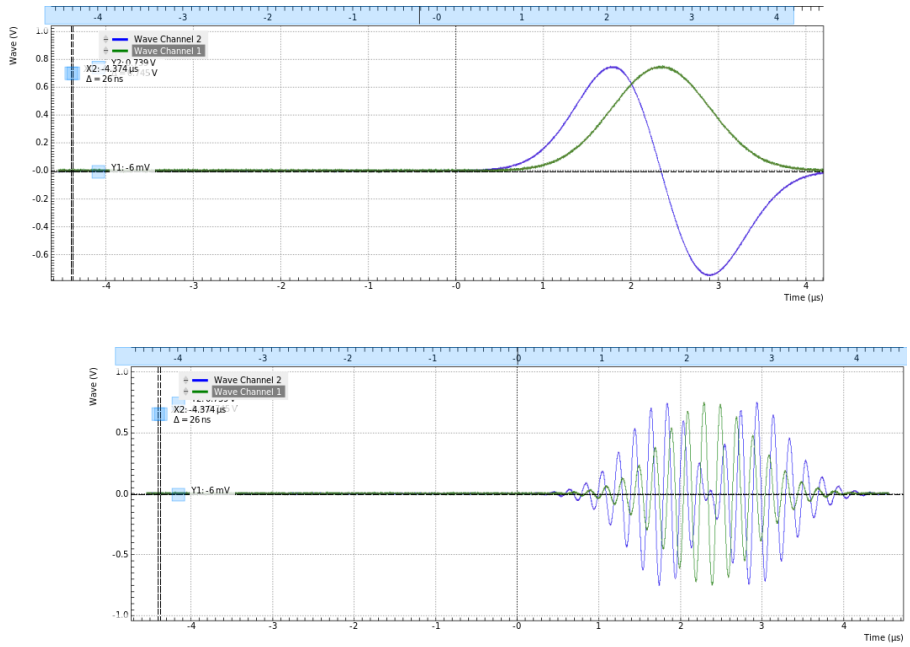


Figure 4.47: Dual-channel signal generated by the AWG and captured by the LabOne Scope (dual-channel Scope operation requires UHF-DIG option). The top figure shows two envelope waveforms played in Plain mode, the bottom figure shows the same envelope waveforms played in Amplitude Modulation mode.

While the AWG is running, you can go ahead now and switch both AWG Outputs to Modulation mode. The lower plot in Figure 4.47 shows the resulting signals, which are the Gaussian and DRAG pulses multiplied by a 5 MHz carrier with phase shift 0° and 90°, respectively.

Table 4.43: Settings: set both AWG Outputs to Modulation mode

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG	Control	Output 1		Mode	Modulation
AWG	Control	Output 2		Mode	Modulation

In this practical case of I/Q modulation, the two AWG Outputs typically require further adjustments of the pulse amplitude, DC offset, and inter-channel phase offset in order to compensate for analog mixer imperfections. All these adjustments can now be done on the fly using the AWG Amplitude, the Signal Output Offset, and the Demodulator Phase settings without having to make any changes to the programmed AWG waveforms.

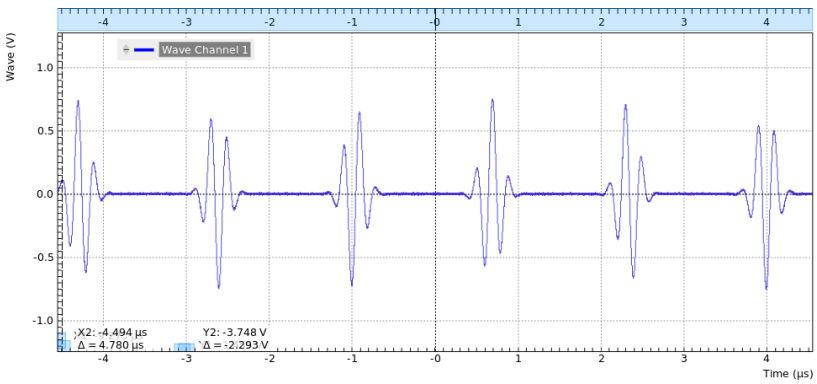
Stabilizing Carrier-Envelope Offset

When playing waveforms in modulation mode, it can sometimes be necessary to synchronize the envelope with the phase of the carrier. This will lead to a final pulse shape that is exactly the same in every repetition. This synchronization is easily achieved with the **`waitDemodOscPhase`** command introduced previously. In the following program, we use this command to align the start of the waveform playback with the oscillator phase of demodulator 4, i.e., the carrier phase.

```
wave w_gauss = gauss(2000, 1000, 200);

while (true) {
  waitDemodOscPhase(4);
  setTrigger(1);
  playWave(1, w_gauss);
  waitWave();
  setTrigger(0);
  wait(100);
}
```

Look at the generated signal once with and once without the **`waitDemodOscPhase`** command. As shown in Figure 4.48, you will see that with this synchronization command, every generated pulse looks exactly the same.



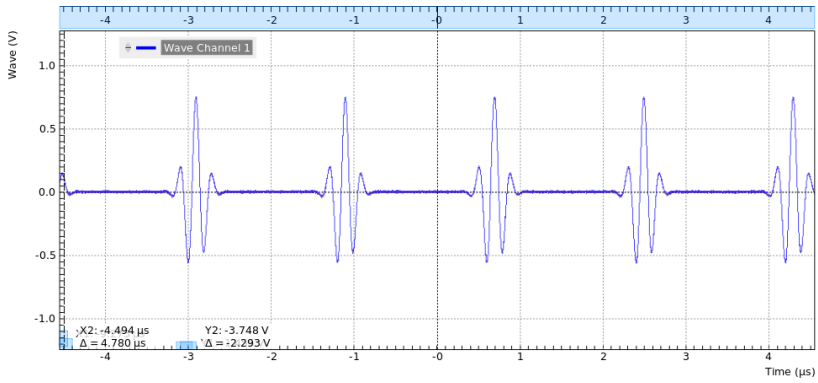


Figure 4.48: Amplitude-modulated signal generated by the AWG and captured by the LabOne Scope. The top figure shows repeated waveform without synchronization between carrier and envelope phase. The bottom figure shows the same signal but with synchronization.

Multi-frequency Modulation

When the UHF-MF Multi-frequency option is installed, the UHF-AWG supports modulation of multiple carriers with individual envelope signals. Typical use cases are phase cycling protocols in NMR spectroscopy, or frequency-multiplexing techniques. The latter requires the UHF-MF option, whereas the former can be used with the base instrument. Multi-carrier modulation is realized by four-fold interleaving of one AWG channel and individual multiplication of the four channel with one of the demodulator's oscillator signal. The envelope sampling rate is therefore reduced by a factor of 4, whereas the carrier signal is still generated at the full sampling rate, therefore giving access to the full bandwidth.

With the following sequence program, we generate a series of pulses with changing carrier. The first four pulses each have a single carrier coming from one of the first four oscillators. In the fifth pulse, all carrier signals are superimposed. The `interleave` command which we use several times allows us to combine four waveforms into one.

```
const n_samples = 512;
wave w_gauss = 0.25*gauss(n_samples, n_samples/2, n_samples/10);
wave w_zeros = zeros(n_samples);

wave w_channel_1 = interleave(w_gauss, w_zeros, w_zeros, w_zeros);
wave w_channel_2 = interleave(w_zeros, w_gauss, w_zeros, w_zeros);
wave w_channel_3 = interleave(w_zeros, w_zeros, w_gauss, w_zeros);
wave w_channel_4 = interleave(w_zeros, w_zeros, w_zeros, w_gauss);
wave w_all_channels = interleave(w_gauss, w_gauss, w_gauss, w_gauss);

while (true) {
  playWave(w_channel_1);
  playWave(w_channel_2);
  playWave(w_channel_3);
  playWave(w_channel_4);
  setTrigger(1);
  setTrigger(0);
  playWave(w_all_channels);
  playZero(n_samples); //spacing between pulses
}
```

We need to make further settings in the user interface in order to distribute the AWG signal on the four oscillators. In the Lock-in tab, the outputs of demodulators 1 to 4 need to be enabled. At the same time, we set their output amplitudes to 0 since otherwise a continuous-wave signal would be added. We route the oscillator signals 1 to 4 to the demodulators 1 to 4 by changing the selectors in the Osc column. We set the frequencies of oscillators 1 to 4 to mutually different values. Finally, in the AWG tab we set Modulation to Advanced.

Table 4.44: Settings: Configure the AWG output and scope trigger

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Scope	Trigger	Trigger		Signal	AWG Trigger 1

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Lock-in	All	Oscillators	1	Frequency	5 MHz
Lock-in	All	Oscillators	2	Frequency	10 MHz
Lock-in	All	Oscillators	3	Frequency	20 MHz
Lock-in	All	Oscillators	4	Frequency	40 MHz
Lock-in	All	Demodulators	1	Osc	1
Lock-in	All	Demodulators	2	Osc	2
Lock-in	All	Demodulators	3	Osc	3
Lock-in	All	Demodulators	4	Osc	4
Lock-in	All	Demodulators	8	Phase	0°
Lock-in	All	Output Amplitudes	1-4	Amp 1	0
Lock-in	All	Output Amplitudes	1-4	Enable	ON
AWG	Control	Output 1		Modulation	Advanced

Figure 4.49 shows the signal captured by the Scope after having made all the settings and uploaded and played the sequence program above.

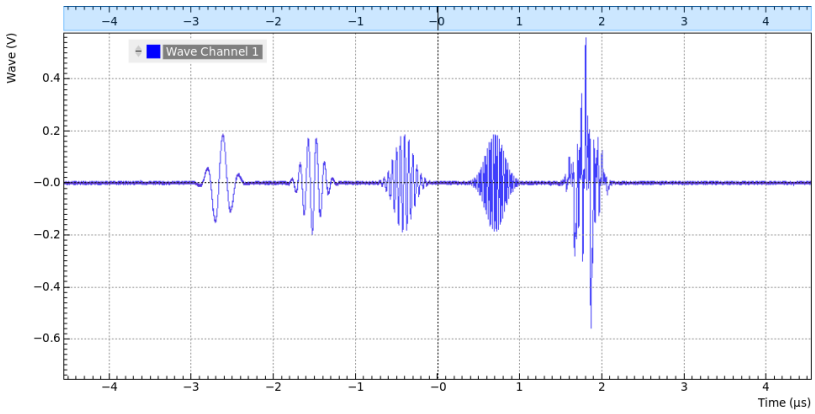


Figure 4.49: Amplitude-modulated signal with multiple carriers generated by the AWG and captured by the LabOne Scope.

4.9.6. Signal Output Assignment

In addition to the single-channel and dual-channel playback used up to now, there are more options for the channel assignment. The `playWave` command can be used with different combinations of arguments: with one `wave` type argument or with two, with a `const` type integer number specifying the signal output or without. These different combinations of arguments allow the user to independently control the AWG outputs (the digital signal sources inside the instrument) and the place where their signal is routed to (the signal outputs on the front panel). The AWG outputs are represented in the AWG tab, whereas the signal outputs are represented in the In / Out tab and in the Lock-in tab .

The `playWave` command always assigns the first `wave` argument to the AWG output 1, and the second one (if it's provided) to the AWG output 2. Each of the `wave` arguments can optionally be preceded by an integer argument of type `const` which specifies the associated signal output. E.g., `playWave(2, w_gauss)` will play the wave `w_gauss` on Signal Output 2.

The systematics of channel assignments in the sequence program even works when using multiple instruments. This is discussed in more detail in LabOne Sequence Programming, whereas here we focus on the case of a single UHF instrument. It's possible to route a single AWG Output to both Signal Outputs at the same time by specifying two integer arguments per wave argument as in `playWave(1, 2, w_gauss)`. This can for example be used to optimize waveform memory. Another option is to add up two AWG Outputs on one Signal Output by using twice the same integer argument as in `playWave(1, w_gauss, 1, w_drag)`. This is e.g. useful in combination with the Modulation Mode as it enables quadrature modulation of an internal oscillator which gives full freedom in controlling the amplitude and phase of a carrier with the AWG. The following sequence

program contains a number of examples for these configurations. Figure 4.50 shows the dual-channel signal generated with this program and measured with the LabOne Scope.

```
wave w_gauss = 0.5*gauss(8000, 4000, 1000);
wave w_drag  = 0.5*drag(8000, 4000, 1000);

while (true) {
  setTrigger(1);
  // play wave on Signal Output 1 with AWG Output 1 (two equivalent commands):
  playWave(w_gauss);
  playWave(1, w_gauss);
  // play wave on Signal Output 2 with AWG Output 1:
  playWave(2, w_gauss);
  // play identical Wave on Signal Output 1 and 2 generated with AWG Output 1:
  playWave(1, 2,  w_gauss);
  // play independent Waves on Signal Output 1 and 2 generated
  // with AWG Output 1 and 2 (two equivalent commands):
  playWave(w_gauss, w_drag);
  playWave(1, w_gauss, 2, w_drag);
  // add up two independent Waves on Signal Output 1
  // generated with AWG Output 1 and 2:
  playWave(1, w_gauss, 1, w_drag);

  waitWave();
  setTrigger(0);
  wait(10000);
}
```

Note

Tricky examples are commands like `playWave(2, w_gauss)` that generate a signal on Signal Output 2, but use the AWG output 1. This means that the relevant Mode and Amplitude (FS) settings are in the section Output 1 of the AWG tab, not in the section Output 2.

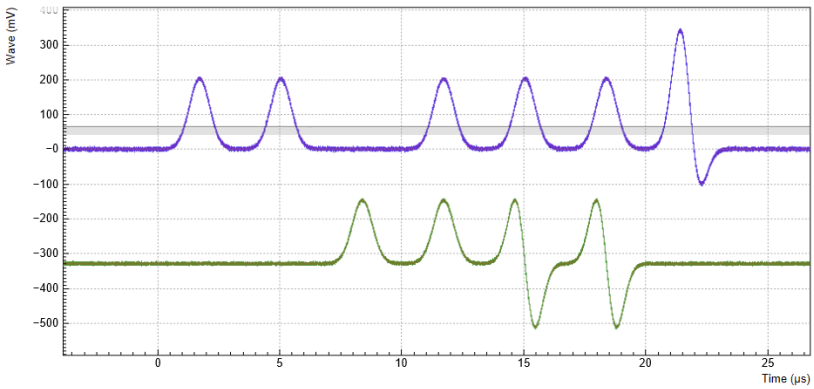


Figure 4.50: Dual-channel signal generated by the AWG and captured by the LabOne Scope (dual-channel Scope operation requires UHF-DIG option). The vertical scales of the two signals have been offset, see

4.9.7. Fast Frequency and Phase Changes

The Sequencer of the UHF-AWG is capable of changing all instrument settings. This allows one to change these settings much more quickly and with much more timing precision than when using the graphical user interface or the API. The majority of the settings are made with a delay of less than 100 µs, but the change of oscillator frequency and phase is implemented in a way that changes are made with a delay of less than 150 ns and, even more importantly, with deterministic timing. This is useful e.g. in application fields like NMR spectroscopy or quantum computing, where it's necessary to make phase or frequency changes rapidly between pulses. Also, this feature allows the user to include special settings that are essential for the correct playback directly in the AWG program.

The instrument settings are made with the **setInt** or the **setDouble** commands. Similarly to the corresponding commands used in the LabOne APIs, their arguments are a settings path and a settings value. The quickest way to find the settings path for a given user interface is to use the Command Log feature. Every time a setting is made in the graphical user interface, the corresponding path and value are displayed in the status bar at the bottom of the browser window. For instance, when enabling the Signal Output 1 the displayed path is **sigouts/0/enable** (zero-based indexing) and the value is 1. A long version of the Command Log can be accessed by clicking on the corresponding button in the status bar. Device Node Tree contains a documentation of all settings paths. In case of using a multi-instrument program, the path has to be prepended with a device number n as in **0/sigouts/0/enable** for n=0. The device number corresponds to the order in the Multi Device Sync Tab (n=0 for Leader, n=1 for Follower 1, and so forth).

The following sequence program contains a few examples of settings that are accessible from the AWG. Here we set the AWG channel 1 amplitude to 0.7, and we prepare the Scope by setting its Trigger Source, by enabling the trigger, and by starting the Scope. These settings are examples of "slow" settings, which take of the order of 100 µs to take effect. Subsequently, we play a series of 4 pulses in Modulation mode, and we change the carrier frequency and phase between pulses. These are examples of "fast" settings with a delay of less than 150 ns and deterministic timing. The list of settings that are available for fast real-time access are given in Table 4.45 along with some additional information.

```
wave w_gauss = gauss(1024, 512, 100);
const degree_to_phaseshift(const degree) {
  return degree*pow(2, 23)/360.0;
}

const freq_1 = 15e6;
const freq_2 = 35e6;
const phase_1 = degree_to_phaseshift(45);
const phase_2 = degree_to_phaseshift(90);

const set_time = 50;

setDouble('awgs/0/outputs/0/amplitude', 0.7);
setInt('scopes/0/trigchannel', 192); // Set Scope trigger source to AWG Trigger 1
setInt('scopes/0/trigenable', 1);    // Enable Scope trigger
setInt('scopes/0/enable', 1);        // Enable Scope

wait(100000);

while (true) {
  setTrigger(1);
  setTrigger(0);
  playWave(w_gauss);
    waitWave();
    setDouble('oscs/0/freq', freq_1);
    wait(set_time);
  playWave(w_gauss);
    waitWave();
    setDouble('oscs/0/freq', freq_2);
    wait(set_time);
  playWave(w_gauss);
    waitWave();
    setDouble('demods/2/phaseshift', phase_1);
    wait(set_time);
  playWave(w_gauss);
    waitWave();
    setDouble('demods/2/phaseshift', phase_2);
    wait(set_time);
}
```

Table 4.45: Settings that are available for fast real-time access from the AWG

Path	Setting	Example
`oscs/0...7/freq`	Oscillator frequency (floating point representation 0-600e6)	`setDouble('oscs/0/freq', 41.23e6);`

Path	Setting	Example
`demods/0...7/phaseshift`	Demodulator phase shift (integer 0-8388608 corresponding to 0-360 degrees)	`setInt('demods/3/phaseshift', 1048576);`
`demods/0...7/oscselect`	Demodulator Oscillator selector (integer number 0-7)	`setInt('demods/3/oscselect', 2);`

4.9.8. Combining Signal Generation and Detection

The base version of the UHF Arbitrary Waveform Generator contains a single-channel Scope for data acquisition. The UHF hardware platform can furthermore be equipped with a Boxcar Averager, Pulse Counter, or Lock-in Amplifier for more advanced measurement tasks. In this section, we will demonstrate how to use Cross-Domain Triggering and other features in order to combine AWG signal generation and detection in an efficient, precise, and easy way.

Scope/Digitizer

Having the Scope/Digitizer in the same housing as the AWG enables internal routing of trigger and marker signals from the AWG to the Scope. The setup of AWG triggers and markers for internal routing is in no way different than for external use and is explained in Generating Triggers with the AWG. Once generated, the generated trigger/marker signals can be selected simply in the Trigger Signal setting of the Scope:

Table 4.46: Settings: configure Scope for internal triggering by AWG

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Scope	Trigger	Trigger		Signal	AWG Trigger 1

Alternatively, AWG Trigger 2–4 or AWG Marker 1–4 can be used.

Boxcar Averager

The Boxcar Tab is an excellent tool for the analysis of signals with low duty cycles. In combination with the AWG, it is well suited whenever the setup response to a short, repeated pulse needs to be measured. Boxcar averager and AWG have to be referenced to the same internal oscillator for proper synchronization. Here, let's consider the case where the AWG generates a signal on Signal Output 1, and the boxcar unit 1 analyzes the return signal on Signal Input 1. Both the AWG and the Boxcar are referenced to oscillator 1. The following table summarizes the necessary settings.

Table 4.47: Settings: configure the Boxcar averager and AWG common reference

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Boxcar	Boxcar	Signal Input	1	Osc	1
Boxcar	Boxcar	Signal Input	1	Input Signal	Sig In 1
AWG	Control	Output 1		Mode	Plain
Lock-in	All	Demodulators	1	Osc	1
Lock-in	All	Oscillators	1	Frequency	200 kHz

Synchronization between the AWG and oscillator 1 is achieved with the `waitDemodOscPhase` used like in the exemplary Sequencer program below. Please refer to Controlling the AWG Repetition Rate for more information.

```
wave w_gauss = gauss(640, 320, 50);

while (true) {
  waitDemodOscPhase(1);
  playWave(1, w_gauss);
}
```

Please consider that the duration of the AWG pattern played after the `waitPhaseOfDemod` command should be shorter than one period of the reference oscillator.

The Periodic Waveform Analyzer (PWA) is a part of the Boxcar averager and usually serves to select a suitable Boxcar integration window. The working principle of the PWA and of the AWG impose some conditions on the repetition frequency. As explained in PWA Averager, the PWA requires a repetition frequency that is incommensurable with the base frequency of 450 MHz in order to faithfully measure a waveform. For the AWG on the contrary, it is preferable to have a repetition rate that is commensurable with the base frequency of 450 MHz, since otherwise the AWG signal jitters by one sequencer period of 4.44 ns.

In choosing between these contradicting requirements, it is usually better to select an repetition frequency that is optimized for a jitter-free AWG signal, i.e., a frequency that is commensurable with 450 MHz. Even though this means that the PWA signal will not look smooth, the Boxcar averager is unaffected by this and performs well. The PWA signal is usually still good enough to help adjusting the Boxcar gate start phase and width. Alternatively, you can use the LabOne Sweeper to vary the Boxcar gate parameters in order to maximize the Boxcar averager signal-to-noise ratio.

Pulse Counter

The Pulse Counter Tab supports two run modes that can make use of trigger signals generated by the AWG, gated free running and gated modes. In gated free running mode, the AWG trigger signal defines a pulse counting period by the rising and falling edge of its trigger signal. In gated mode, the AWG trigger signal resets the periodic Pulse Counter timer. Setting up AWG triggers for this purpose is in no way different than for external use and you can follow the explanations in Generating Triggers with the AWG. The following table summarizes the necessary settings in the Counter tab.

Table 4.48: Settings: configure the Counter gating by the AWG

Tab	Sub-tab	Section	#	Label	Setting / Value / State
Counter	1		1	Gate Input	AWG Trigger 1
Counter	1		1	Mode	Gated Free Running or Gated

Alternatively, AWG Trigger 2–4 can be used as Gate Input.

Lock-in Amplifier

The combination of AWG and lock-in amplifier enables a number of fast Sweeper and DAQ measurement modes described in Fast AWG Sweeper Modes and TV Mode Averaging with the Data Acquisition Tool .

4.9.9. Branching and Feed-Forward

Using its branching capabilities, the UHFAWG can select the next waveform based on external conditions such as the state of the 32-bit digital input, or internal conditions such as the value of a demodulated signal quadrature.

- Branching based on external conditions is typically used in automatic testing in order to allow for fast and flexible control of the AWG playback sequence.
- Branching based on internal conditions enables fast feed-forward protocols used for instance in quantum computing.

Inside a Sequencer program, branching is realized with the `if` statement or with the `switch...case` statement (cf. LabOne Sequence Programming to learn about the timing difference of the two). In the example below, we read the state of the AWG Analog Trigger Input 1 first, and depending on its state (1 or 0) we play a Gaussian waveform with amplitude 0.5 or 1.0.

```
wave w_gauss_low  = 0.5 * gauss(8000, 4000, 1000);
wave w_gauss_high = 1.0 * gauss(8000, 4000, 1000);

var trigger_state;
while (true) {
  trigger_state = getAnaTrigger(1);
  if (trigger_state == 0) {
```

```
    playWave(1, w_gauss_low);
  } else {
    playWave(1, w_gauss_high);
  }
}
```

The AWG output depends on how we configure the AWG Analog Trigger Input 1, and what physical signal we provide on that input. The Trigger source may be chosen with the Analog Trigger 1 Signal setting. For a sequence branching application, the Trigger would normally be used in a level-sensitive (as opposed to edge-sensitive) mode, which means that the Rise and Fall checkboxes would be disabled.

In order to set up branching based on external conditions, the Analog Trigger 1 Signal would be set to a physical trigger input, such the Ref / Trigger 2 input. This setting corresponds to a case in which the playback is controlled by an external TTL signal. Much more complex examples can be constructed by using the 32-bit DIO input. This input can be read using the `getDIO` command that works analogously to the `getAnaTrigger` command used here.

Branching based on internal conditions is available in the combination of UHF-AWG Arbitrary Waveform Generator with signal detection units such as the UHF-LI Lock-in Amplifier or the UHF-CNT Pulse Counter. In the following we will look into this unique configuration in more detail.

We shall consider the combination of UHFLI and UHF-AWG and realize the situation where a demodulator output signal figures as the AWG Analog Trigger 1 Signal in order to realize a fast feed-forward protocol. This could correspond to a controlled-reset protocol of a quantum bit (qubit), see Phys. Rev. Lett. 109, 240502 (2012). In this protocol, a qubit state is determined in a fast lock-in measurement, and if the measurement yields that the qubit is in an excited state, a reset pulse is applied immediately afterwards. We use the following Sequencer program in order to demonstrate this method.

```
wave w_gauss_low  = 0.5*gauss(8000, 4000, 1000);
wave w_gauss_high = 1.0*gauss(8000, 4000, 1000);

var trigger_state;
while (true) {
  waitDemodOscPhase(8);
  playWave(1, w_gauss_low);
  waitWave();
  trigger_state = getAnaTrigger(1);
  if (trigger_state == 0) {
    playWave(1, w_gauss_low);
  } else {
    playWave(1, w_gauss_high);
  }

  wait(3000);

  playWave(1, w_gauss_high);
  waitWave();
  trigger_state = getAnaTrigger(1);
  if (trigger_state == 0) {
    playWave(1, w_gauss_low);
  } else {
    playWave(1, w_gauss_high);
  }
}
```

The program consists of two almost identical blocks enclosed in an infinite `while` loop. In each block, we first play a Gaussian pulse - let's call this the measurement pulse. Then we obtain the Analog Trigger 1 state, and then we perform a conditional playback of another Gaussian pulse, let's call this one the reset pulse.

The two blocks only differ by the amplitude of the measurement pulse: it is either 0.5 or 1.0. This difference can be detected by a fast lock-in measurement. We let the AWG run in Modulation mode using a carrier frequency of 5 MHz, and we configure Demodulator 1 to measure at the same frequency. We set the Demodulator filter time constant to 3 µs, a value that is comparable to the width of the measurement pulse. This means that the demodulator filter roughly integrates the signal over the pulse width.

If we configure the AWG Analog Trigger Input 1 with the appropriate Signal (Demodulator 1 R) and Level (140 mV), the AWG will be able to discriminate the high- and low-amplitude measurement pulses. As a consequence, it will play a low-amplitude reset pulse after a low-amplitude measurement pulse, and a high-amplitude reset pulse after a high-amplitude measurement pulse. Note that the **`waitWave`** command ensures that the subsequent command (the **`getAnaTrigger`** command which evaluates the measurement value) is executed immediately after (and not during) the playback of the measurement pulse. In our case this is just the right timing to obtain a meaningful demodulator measurement taking into account the demodulator settling time.

The following table summarizes the settings to be made for the feed-forward experiment.

Table 4.49: Settings: configure the AWG and Demodulators for feed-forward

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG	Trigger	Analog Trigger	1	Rise	OFF
AWG	Trigger	Analog Trigger	1	Fall	OFF
AWG	Trigger	Analog Trigger	1	Signal	Demodulator 1 R
AWG	Trigger	Analog Trigger	1	Level	140 mV
AWG	Control	Output	1	Mode	Modulation
Lock-in	All	Low-Pass Filter	1	Order	1
Lock-in	All	Low-Pass Filter	1	TC	3 µs
Lock-in	All	Demodulators	1	Osc	1
Lock-in	All	Demodulators	8	Osc	2
Lock-in	All	Oscillators	1	Frequency	5 MHz
Lock-in	All	Oscillators	2	Frequency	1 kHz
Scope	Trigger	Trigger		Signal	Osc φ Demod 8
Scope	Trigger	Trigger		Enable	ON
Scope	Trigger	Trigger		Run / Stop	ON

Figure 4.51 shows the signal generated by the AWG in blue. With the UHF-DIG option, we can simultaneously display the R signal of Demodulator 1 in green. The Y2 cursor position shows the AWG Analog Trigger 1 Level of 140 mV. We can observe that the second and the fourth pulse are indeed played conditionally on the demodulator measurement which is evaluated immediately after the measurement pulse has ended. If you adjust the Trigger Level, you will see the live effect on the second and fourth pulse in the signal.

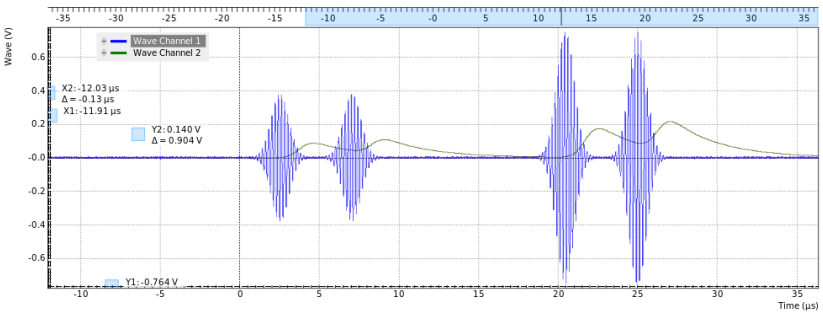


Figure 4.51: Signal generated by the AWG (blue) and demodulated signal (green; displaying this signal requires UHF-DIG option) captured by the LabOne Scope.

4.9.10. Fast AWG Sweeper Modes

The LabOne Sweeper offers special operation modes for fast measurement in combination with the UHF-AWG Arbitrary Waveform Generator. These modes take advantage of the powerful execution control of the LabOne AWG Sequencer and of the high measurement speed of the demodulators. Before using the Sweeper in the special modes, it's best to first become familiar with its basic operation. This is described in the Sweeper Tab of the Sweeper as well as in the Phase-locked Loop.

Note

For both operation modes described below, the demodulator measurement data rate can be pushed to very high values by gating the demodulator data stream with one of the AWG trigger output channels. Gating is activated using the Trigger setting of the Lock-in tab (collapsed by default). By this method, one can achieve that only the interesting data is transferred to the host PC, but at a much increased peak rate up to 14 MSa/s. This is attractive for applications relying on short and fast measurements interrupted by long dead times.

AWG Index Sweep

The Index Sweep mode allows for recording demodulator samples during a rapid pulse sequence played by the AWG. The speed of this mode can be much higher than that of a normal sweep because it has a much smaller overhead time for instrument communication. This is because the Sweeper tool on the host PC takes on a listener role almost all the time. Typically, a fast series of N pulse patterns is played by the AWG, and in each iteration one parameter is changed, e.g. a pulse length, a pulse amplitude, or a pulse delay. For each iteration, the Sweeper records one demodulator sample. The timing of the demodulator measurement relies on one of the AWG Trigger output channels controlled with the AWG `setTrigger` command. The attribution to a sweep point n=1,...,N relies on the `setID(n)` command in the AWG sequence program. This command tags the demodulator samples with an identifier number n. The Sweeper listens to the AWG Trigger channel selected in the Sweep Parameter setting. When it receives a trigger, it records a demodulator measurement and attributes it to the sweep point corresponding to the current identifier number. To fine-tune the measurement timing relative to the trigger time, it is advised to configure the settling time in the Settings section of the Sweeper (Advanced Mode).

In the example shown below, we play a Gaussian pulse of width \~1 µs in modulation mode and vary the delay of this pulse relative to a trigger in 1000 steps. We demodulate the signal with demodulator 1 with maximum bandwidth and obtain this data with the Sweeper. As a result, we are able to recover the envelope of the fast pulse in the Sweeper.

In order for the Index Sweep mode to work as desired, the AWG sequence program needs to be compatible with the settings in the Sweeper and in the Lock-in tab. This means the AWG program should contain a `for` or `while` loop with a loop count identical to the sweep Length parameter (1000 in this example). The `setID` needs to be applied inside the loop with the loop count variable as an argument (in the example the count variable is called `n`). The `setTrigger` command needs to be applied twice inside the loop: once to set the trigger to the high state, and once to set it back to the low state. The demodulator in use needs to be enabled in the Lock-in tab with a sufficiently high bandwidth compared to the AWG pulse pattern. For this example we set the Filter bandwidth to 5.6 MHz (1st order) and the demodulator sample rate to 400 kSa/s. Finally, it's important that the AWG pulse pattern and the demodulator sample rate are compatible: this is easily achieved by using the `waitDemodSample` command with the demodulator number as an argument (here number 1).

The following file shows an example AWG sequence program in which a pulse is generated with a varying delay relative to the AWG Trigger 1. The pulse is to be played in AWG Modulation mode and measured with demodulator 1.

```
var sweepPoints = 1000;

// define user utility function
void wait_us(const us) {
  wait(us / 4.444e-3);
}

wave w_gauss = gauss(8000, 4000, 1000);

// define user function
void user_func(var index) {
  // Set ID for assignment to sweep point
  setID(index);
  // Wait to synchronize AWG and demodulator sampling
  waitDemodSample(1);
  // Set Trigger for Sweeper measurement
  setTrigger(1);
  // Wait a variable time (sweep parameter)
  wait(index);
```

```
  // Play waveform
  playWave(1, w_gauss);
  waitWave();
  wait_us(3);
  setTrigger(0);
}

// Loop over sweeper variable (waiting time)
var n;
for (n = 0; n < sweepPoints; n = n + 1) {
  user_func(n);
  setID(0);
  wait_us(100);
}
setID(0);
```

Once a suitable sequence program is loaded, configure the Sweeper for the measurement. Select AWG 1 → Index Sweep Triggers → AWG Trigger 1 as the Sweep Parameter, corresponding to the trigger channel we used in the Sequence program. Set the sweep Length to 1000. In the Sweeper settings sub-tab, set the Filter to Advanced Mode in order to enable "AWG Control". The following table summarizes the settings to be made for this example.

Note

The AWG Index Sweep mode is implicitly enabled when the following two settings are made in the Sweeper: 1) AWG Control is enabled 2) one of AWG Index Trigger 1–4 is selected as the sweep parameter.

Table 4.50: Settings: configure the AWG and Sweeper for AWG Index Sweep mode

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG	Control	Output 1/2		Mode	Modulation
Lock-in	All	Demodulators	1	Enable	ON
Lock-in	All	Demodulators	1	Rate	400 kSa/s
Lock-in	All	Demodulators	1	Low-Pass Filter order	1
Lock-in	All	Demodulators	1	Low-Pass Filter bandwidth	5.6 MHz
Sweeper	Control	Horizontal		Sweep Param	AWG Index Sweep Triggers, Trigger 1
Sweeper	Settings			Filter	Advanced Mode
Sweeper	Settings			BW Mode	Manual
Sweeper	Settings	Statistics		AWG Control	ON
Sweeper	Control			Single	ON

If you start the Sweeper now, it will automatically start the AWG and capture the data very quickly. The result is shown in the following figure.

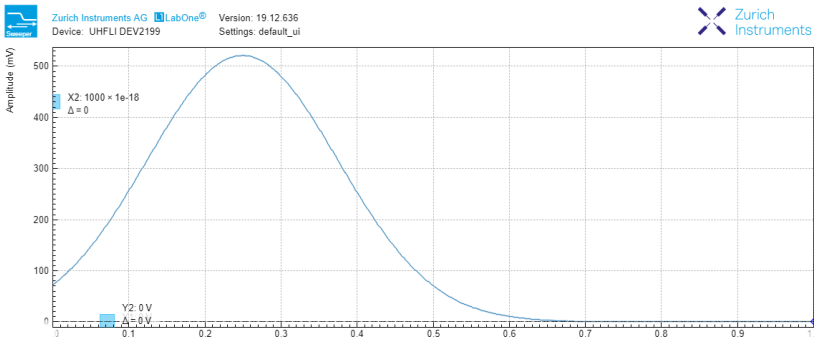


Figure 4.52: AWG Index Sweep measurement trace as displayed in the LabOne Sweeper.

AWG Parameter Sweep

The AWG Parameter Sweep mode allows for precisely timed demodulator measurements as a function of a large selection of device parameters. This mode combines elements of the basic Sweeper mode and of the AWG Index Sweep mode. Like in the basic Sweeper mode, the Sweeper sets a device parameter (such as an oscillator frequency, an AWG amplitude, or an AWG user register value) and starts measuring the data after that with the possibility to average data for some time. Like in the AWG Index Sweep mode, the precise timing is determined by the AWG Trigger output channel which is controlled with the AWG `setTrigger` command. When the Sweeper receives a trigger, it starts recording demodulator data for the defined averaging period. To fine-tune the measurement timing relative to the trigger time, it is advised to configure the settling time in the Settings section of the Sweeper (Advanced Mode).

In the example shown here, we play a long pulse in the AWG modulation mode and vary the amplitude of the AWG output with the Sweeper. We demodulate the signal with demodulator 1 with maximum bandwidth and obtain this data with the Sweeper.

For such a measurement, the AWG sequence program should contain two `setTrigger` commands that define the measurement time: once to set the trigger channel 1 to the high state, and once to set it back to the low state. The demodulator in use needs to be enabled in the Lock-in tab with a sufficiently high bandwidth compared to the AWG pulse pattern speed. Finally, it's important that the AWG pulse pattern and the demodulator sample rate are compatible: this is easily achieved by using the `waitDemodSample` command with the demodulator number as an argument (here number 1). The following sequence program is used in this example.

```
// Play the example at reduced rate of 28 MSa/s
const RATE = AWG_RATE_28MHZ;
const FS = 1.8e9/pow(2, RATE);

// Wait a number of microseconds
void wait_us(const us) {
  wait(1e-6*us*225e6);
}

void user_func() {
  // Total waveform length, 1 ms
  const N = 1e-3*FS;
  // Length of rising edge, 100 us
  const M = 100e-6*FS;
  // Create waveform
  wave edges = blackman(2*M, 1.0, 0.2);
  wave w_pulse = join(cut(edges, 0, M-1),
                      rect(N-2*M, 1.0),
                      cut(edges, M, 2*M-1));
  // Synchronize with demodulator data
  waitDemodSample(1);
  // Start waveform playback
  playWave(1, w_pulse, RATE);
}

// Function for enabling sweeper recording, from/to in microseconds
void sweeper_record(const from_us, const to_us) {
  // Wait a bit, then make sweeper record data
  wait_us(from_us);
  setTrigger(1);
  wait_us(to_us-from_us);
  setTrigger(0);
}

// Execute the subprograms
user_func();
sweeper_record(100, 900);
```

Note

The AWG Parameter Sweep mode is implicitly enabled when the following two settings are made in the Sweeper: 1) AWG Control is enabled 2) the sweep parameter is anything except AWG Index Trigger 1–4.

The following table summarizes the settings to be made for this example.

Table 4.51: Settings: configure the AWG and Sweeper for AWG Parameter Sweep mode

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG	Control	Output 1/2		Mode	Modulation
Lock-in	All	Demodulators	1	Enable	ON
Lock-in	All	Demodulators	1	Rate	400 kSa/s
Lock-in	All	Demodulators	1	Low-Pass Filter order	1
Lock-in	All	Demodulators	1	Low-Pass Filter bandwidth	5.6 MHz
Sweeper	Control	Horizontal		Sweep Param	AWG Output 1 Amplitude
Sweeper	Settings			Filter	Advanced Mode
Sweeper	Settings			BW Mode	Manual
Sweeper	Settings	Statistics		AWG Control	ON
Sweeper	Control			Single	ON

If you start the Sweeper now, it will automatically start the AWG and capture the data very quickly. The result is shown in the following figure.

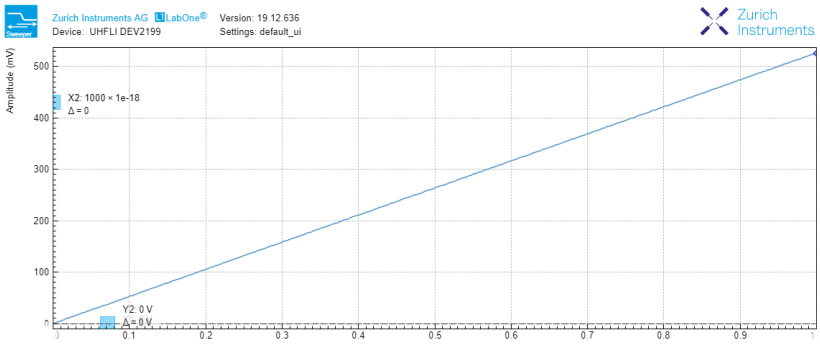


Figure 4.53: AWG parameter sweep measurement trace as displayed in the LabOne Sweeper.

4.9.11. TV Mode Averaging with the Data Acquisition Tool

The LabOne Data Acquisition (DAQ) tool in combination with the UHF-AWG offers a special cyclic data acquisition and averaging also known as TV mode averaging. This can be used to realize drift-resistant measurements as a function of pulse parameters such as pulse width, amplitude, or carrier frequency. One classic example is the measurement of Rabi oscillations of a quantum bit (qubit). In this section, we'll explain the Data Acquisition tool cyclic averaging using an example inspired by the Rabi oscillation measurement. Since it's not that easy to simulate the behavior of a qubit in a table-top experiment, the example doesn't correspond exactly to a real pulse pattern, but it is intended to demonstrate the typical signal generation and acquisition tasks that occur in an actual experiment, and to produce a data set that resembles that of an actual experiment.

In the example, we'll make use of the following features:

- Data Acquisition tool Grid Mode to capture 2D data sets
- UHF-AWG data tagging for 2D data row assignment
- UHF-AWG indexed waveform playback
- Demodulator gated data streaming for highest peak data rates

The following file shows the AWG sequence program for the Rabi example. Copy and upload this program to the AWG or select it from the examples drop-down menu.

```
const rows = 100;      // number of different pulse amplitudes
const f_s = 1.8e9;     // AWG sampling rate
const f_seq = 225e6;   // sequencer clock frequency
const pulse_width_sec = 100e-9; // width of the Gaussian pulse (s)
const gate_window_sec = 1.5e-6; // width of the demodulator data gating window
(s)
const period_sec = 30e-6; // time between successive pulses (s)

void gate_start() {
  setTrigger(0b01); // activate gate signal (AWG Trigger 1)
  wait(20);         // wait a certain time (at least the inverse of the
                    // demodulator sample rate) to make the rising edge
                    // of AWG Trigger 2 visible in the demodulator data stream
  setTrigger(0b11); // activate trigger signal for Data Acquisition tool (AWG
Trigger 2)
}

void gate_stop() {
  setTrigger(0b00); // reset gate and trigger signal
}

const gate_window = gate_window_sec*f_seq;

// Waveform generation
const pulse_width = pulse_width_sec*f_s;
const waveform_length = 7*pulse_width_sec*f_s;

cvar i;
cvar amplitude;
const decay_time = 100;
const rabi_period = 40;
wave w_array;
for (i=0; i<rows; i=i+1) {
  amplitude = 0.5 + exp(-i/decay_time)*0.5*cos(2*M_PI*i/rabi_period);
  wave w_segment = amplitude*gauss(waveform_length, waveform_length/2,
pulse_width);
  w_array = join(w_array, w_segment);
}

// Beginning of the core sequencer program executed on the UHF at run time
var t;
var id = 0; // Row ID to be used by the Data Acquisition tool for data triage
const loop_end = rows*waveform_length;
while (true) {
  id = 0;
  for (t = 0; t < loop_end; t = t+waveform_length) {
    setID(id);          // set the row ID
    wait(period_sec*f_seq);
    waitDemodSample(1); // synchronize the AWG with the demodulator sample rate
    playWaveIndexed(w_array, t, waveform_length); // play waveform segment
    wait(200);
    gate_start();       // start the demodulator gate signal
    wait(gate_window);
    gate_stop();        // stop the demodulator gate signal
    id = id+1;          // increment row ID
  }
}
```

In the first part of the program, we define two user functions `gate_start()` and `gate_stop()` which allow us to define a demodulator gating window using the AWG Trigger 1. Inside the gating window defined by AWG Trigger 1, we place a rising edge of AWG Trigger 2. The latter figures as a trigger signal for the Data Acquisition tool.

In the second part of the program, we use a `for` loop to define an array of pulse shapes that we intend to play back cyclically in the sequence program. The number of pulses is controlled using the constant `rows` which we set to 100. All 100 pulse waveforms have a Gaussian shape with different amplitudes. We combine all these 100 pulse shapes into one long waveform called `w_array` which we are going to access in a segmented manner during the playback. By using only `cvar` (compile-type variable) data types, we ensure that this loop is evaluated at compile time.

In the third part of the program, we define the waveform playback and other instructions to be executed at run time. The program features a `for` loop over `rows` iterations. At the beginning of each iteration, we use the `setID` command to tag the demodulator data with the row number `i`. This index will be read by the Data Acquisition tool to assign its data to the correct row. After the `setID` command and some waiting and synchronization time, we start the segmented waveform playback using the `playWaveIndexed` command. By using the run-time variable `t` as a sample index, we access one segment of the long waveform `w_array` that corresponds to one of the 100 pulses. After the start of the waveform playback, we use the `gate_start` and `gate_stop` functions to define a gate window of about 1.5 µs width for the measurement.

In the Lock-in tab, we configure demodulator 1 for fast measurements. We first set the demodulator 1 Trigger from Continuous to AWG Trigger 1 and the Trig Mode to High (the Trigger section to the right of the Demodulator section is collapsed by default). Because now the demodulator will only stream data during the short windows when the AWG Trigger is high, we can in turn increase its data rate to the maximum of 14 MSa/s without reaching a limitation by the interface. We also set the measurement bandwidth to the maximum of 5.6 MHz (1st-order filter).

In the DAQ tab tab (Settings sub-tab), we first set the Hold Off and Delay both to 0 s. We set Trigger Signal to AWG Trigger 2. In the Grid sub-tab, we enable the Grid Mode by setting Mode to Linear. In this mode, the Data Acquisition tool will resample the data on the horizontal axis using linear interpolation to a number of samples defined by the Columns settings. We can set the Columns to 200 and the Duration to 2 µs. In this way, we cover optimally the short pulses generated by the AWG. The Rows setting has to match the number of different pulse shapes defined by the constant `rows` in the AWG program. By enabling AWG Control, we tell the Data Acquisition tool to assign each shot of data to the row corresponding to the AWG tag `setID` on the data. By using the Average operation, and a certain number of repetitions, e.g. 10, we obtain a 2D data set averaged over several full periods of the experiment. To display this set using a color scale graph, set Plot Type to 2D.

All necessary settings are summarized in Table 4.52. Once these settings are made, we can start the Data Acquisition tool and subsequently the AWG using the "Single" buttons. The Data Acquisition tool should then capture and display a graph such as the one shown in Figure 4.54. In this graph, the horizontal axis corresponds to the time dependence given by the Gaussian pulse shape. The vertical axis shows the row number 1...100 which corresponds to the pulse parameter varied in the experiment. In this example, we artificially generated pulses with an exponentially decaying, oscillating amplitude similar to what would be seen in a Rabi oscillation experiment.

Table 4.52: Settings: configure the AWG and Data Acquisition tool

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG	Control	Output 1/2		Mode	Modulation
Lock-in	All	Demodulators	1	Enable	ON
Lock-in	All	Demodulators	1	Trigger	AWG Trigger 1
Lock-in	All	Demodulators	1	Trig Mode	High
Lock-in	All	Demodulators	1	Enable	ON
Lock-in	All	Demodulators	1	Rate	14 MSa/s
Lock-in	All	Demodulators	1	Low-Pass Filter order	1
Lock-in	All	Demodulators	1	Low-Pass Filter bandwidth	5.6 MHz
Lock-in	All	Signal Outputs		Signal Output 1	ON
DAQ	Settings	Trigger Settings		Trigger Signal	AWG Trigger 2
DAQ	Settings	Horizontal		Hold Off Time	0 s
DAQ	Settings	Horizontal		Delay	0 s
DAQ	Grid	Grid Settings		Mode	Linear
DAQ	Grid	Grid Settings		Operation	Average
DAQ	Grid	Grid Settings		Duration	2 µs
DAQ	Grid	Grid Settings		Columns	200

Tab	Sub-tab	Section	#	Label	Setting / Value / State
DAQ	Grid	Grid Settings		Rows	100
DAQ	Grid	Grid Settings		Repetitions	10
DAQ	Grid	Grid Settings		AWG Control	ON
DAQ	Grid	Display		Plot Type	2D

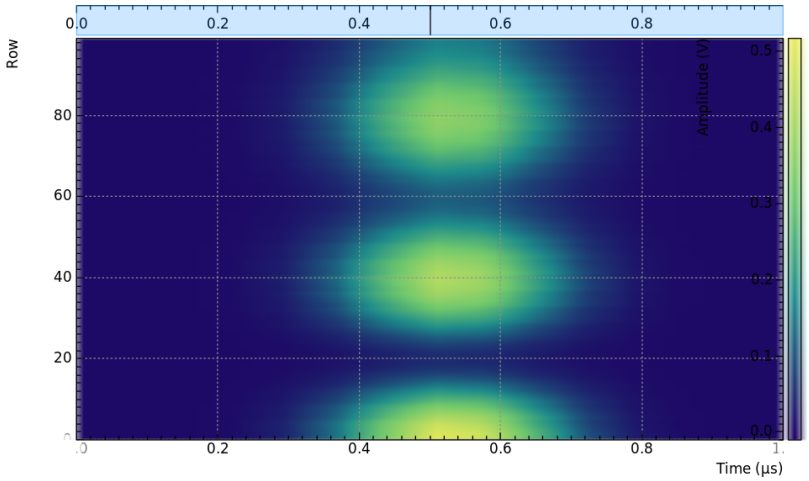


Figure 4.54: 2D data captured with the Data Acquisition tool in Grid Mode. The horizontal axis shows the time, the vertical axis is the row number, and the color scale shows the demodulator amplitude.

4.9.12. Four-channel Aux Output

For applications requiring more channels and/or higher voltages, the UHF-AWG can generate signals on the auxiliary outputs of the UHF instrument. To this end the AWG resources for one fast channel (1.8 GSa/s) can be reallocated so as to generate four independent signals at 14 MSa/s and 16 bit resolution in a ±10 V range.

In the sequence program, the functionality is available through the `playAuxWave` function. The function requires four waveforms of equal length as arguments.

We configure the multi-purpose Auxiliary Outputs for AWG signal generation by setting the Auxiliary Output Signal to AWG in the Auxiliary tab. Each Auxiliary Output corresponds to one of the four rows in the tab. The Channel setting allows you to route one of the four AWG Outputs (the four waveforms of the `playAuxWave` command) to the given Auxiliary Output. Typically for the first row the Channel is set to 1, for the second row to 2, and so forth.

We intend to monitor the individual Auxiliary signals with the Scope on Signal Input 1. Before making the corresponding BNC connections, it's good practice to adjust the Auxiliary Output Lower and Upper Limits in order to prevent damage to the Signal Input. You can use the Scale and the Offset setting in order to modify the signal.

The Auxiliary Outputs have a much lower analog bandwidth than the Signal Outputs. It is therefore necessary to work at a sampling rate of 14 MSa/s or less. In order to combine slow Auxiliary Output signals with fast signals on the Signal Outputs, it's useful to set the sampling rate for every individual waveform play command. In the following example, we first play four waveforms in parallel on the Auxiliary Outputs at reduced sampling rate, and then one waveform on the Signal Output 2 at full sampling rate.

```
// Sampling rate of the system, adjust accordingly if the rate is reduced
const FS = 1800e6;
// Frequency of the 'sine' in the SINC waveform
const F_SINC = 42e6;

// Generate the four-channel auxiliary output waveform
wave aux_ch1 =  1.0*gauss(8000, 4000, 1000);
wave aux_ch2 =  0.5*gauss(8000, 4000, 1000);
wave aux_ch3 = -0.5*gauss(8000, 4000, 1000);
wave aux_ch4 = -1.0*gauss(8000, 4000, 1000);
```

```
// Generate a waveform to be played on Signal Output 2
wave w_sinc = sinc(8000, 4000, FS/F_SINC);

while (true) {
  // play the four Aux Output channels at reduced rate
  playAuxWave(aux_ch1, aux_ch2, aux_ch3, aux_ch4, AWG_RATE_14MHZ);
  // play a wave on Signal Output 2
  playWave(w_sinc, AWG_RATE_1800MHZ);
}
```

The following table summarizes the settings to be made for this example.

Table 4.53: Settings: configure the AWG for generating signals on the Auxiliary Outputs

Tab	Sub-tab	Section	#	Label	Setting / Value / State
AWG	Control	Output 1/2		Mode	Plain
Auxiliary		Aux Output	1-4	Lower/Upper Limit	–1.5 V/+1.5 V
Auxiliary		Aux Output	1-4	Signal	AWG
Auxiliary		Aux Output	1	Channel	1
Auxiliary		Aux Output	2	Channel	2
Auxiliary		Aux Output	3	Channel	3
Auxiliary		Aux Output	4	Channel	4

Note

The four-channel AWG mode features a sample hold functionality: the output voltage of the last sample of a waveform remains fixed after the waveform playback is over. This can be used to control the output voltage between pulses.

4.9.13. Debugging Sequencer Programs

When generating fast signals and observing them with the LabOne Scope, in some configurations you may observe timing jitter or unexpected delays in the generated signal. There are two main reasons for that. The first reason is linked to the AWG's memory architecture, which is based on a main memory and a cache memory. Waveform data stored in the main memory (128 MSa per channel) must be copied to the cache memory (32 kSa per channel) prior to playback. The bandwidth available for this data transfer is less than that required by the AWG for dual-channel operation at 1.8 GSa/s. Therefore, if the AWG is configured to play waveforms longer than what fits in the cache memory in dual-channel mode at 1.8 GSa/s, interruptions in the generated signal may be observed. The second reason is connected to the AWG compiler concept explained in Description. When a program in the Sequence Editor is compiled into machine code that can be executed by the Sequencer hardware, single lines of code may be expanded into several machine instructions. Each instruction requires one clock cycle (4.44 ns) for execution. Therefore, the final timing of the generated waveform may not always be completely apart from looking solely at the high-level sequencer program. The compiled program, which defines the actual timing, is displayed in the Advanced sub-tab.

Please take the following tips into consideration when operating the UHF-AWG. They should help you prevent and solve timing problems.

- The Scope and the AWG share the same memory, which means that operating them together at high sampling rates affects the performance of both of them. Note that this is only a concern when the AWG is playing back waveforms that are too large to fit in the cache memory. If this is the case it may prove difficult to visualize the generated AWG signal using the LabOne Scope. One option for visualizing such long waveforms is to reduce the sampling rate of both the AWG and the Scope to 225 MHz, which allows both the AWG and the Scope to operate in dual-channel mode simultaneously. The overall shape of the generate AWG signal can then be visualized and evaluated. The sampling rate of the AWG can then be increased once you are satisfied with the shape of the generated signals.
- Minimize waveform memory (1): use the possibility to vary the sample rate during playback. The `playWave` command (and related commands) accept a sampling rate parameter, which means slow and fast signal components can be played at different rates.
-

Minimize waveform memory (2): take advantage of the amplitude modulation mode in order to generate signals at the full bandwidth, but with reduced envelope sampling rate.
- Minimize waveform memory (3): in four-channel (Auxiliary Output) mode, the signal amplitude of the last sample after a waveform playback is held. This eliminates the need for long waveforms with constant amplitude, e.g. on a pulse plateau.
- Check the occupied waveform cache memory in the Waveform sub-tab. If you stay below 100%, the performance is best and there is no interference with the LabOne Scope.
- Take advantage of the AWG state signals available on the Trigger outputs. In the DIO tab you can select from a number of options for outputting the AWG state as TTL signals, such as "fetching", or "playing". Monitoring these signals on a scope can help in understanding the AWG timing.
- When possible, use the `repeat` loop instead of the `for` and `while` loops. The `for` and `while` loops evaluate and compare run-time variables, which makes them slower to execute in comparison to the `repeat` loop.
- Fill up sequencer waiting time with useful commands. Placing commands and run-time variable operations just before a `wait` command (and related commands) in the sequence program means they will be executed when the sequencer has time.
- When you need sample-precise timing between analog and digital output signals, use the AWG Markers rather than the AWG Triggers or the DIO outputs.
- When using the four-channel Auxiliary Output mode, be aware that the timing between Signal Output and Aux outputs is not well-defined. Use a scope to adjust inter-channel delays.
- Be aware that the sequencer instruction memory is also segmented into a cache memory and main memory. Very long sequence programs therefore require fetching operations, which costs some time. You can read the memory usage in the Advanced sub-tab.

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 UHF Series. 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.

On top of standard functionality like acquiring and saving data points, this UI provides a wide variety of measurement tools for time and frequency domain analysis of measurement data as well as for convenient servo loop implementation.

5.1. User Interface Overview

5.2. 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 UHF Series 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.

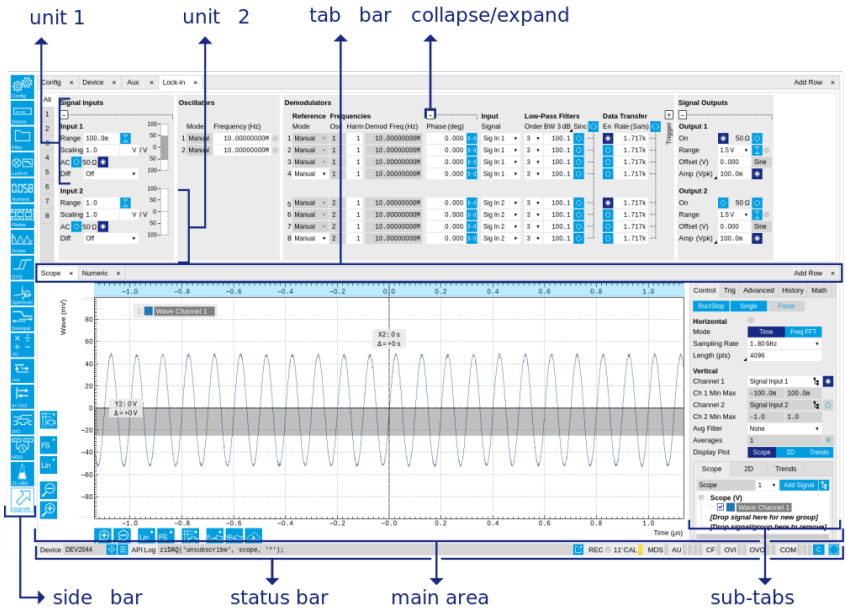


Figure 5.1: LabOne User Interface (default view)

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
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

Further items are highlighted in Figure 5.2.

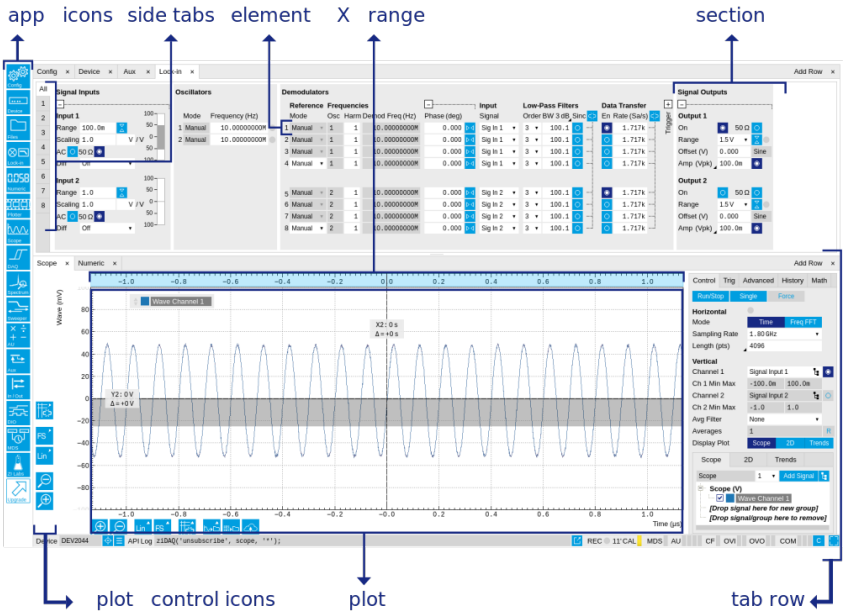


Figure 5.2: LabOne User Interface (more items)

5.2.1. Unique Set of Analysis Tools

All instruments feature a comprehensive tool set for time and frequency domain analysis for both raw and demodulated signals.

The app icons on the left side of the UI can be roughly divided into two categories: settings and tools.

Settings-related tabs are in direct connection to the instrument hardware, allowing the user to control all the settings and instrument states.

Tools-related tabs place a focus on the display and analysis of gathered measurement data.

There is no strict distinction between settings and tools, e.g. the Sweeper will change certain demodulator settings while performing a frequency sweep. Within the tools one can often further discriminate between time domain and frequency domain analysis. Moreover, a distinction can be made between the analysis of fast input signals - typical sampling rate of 1.8 GSa/s - and the measurement of orders of magnitude slower data - typical sampling rate of <28 MSa/s - derived for instance from demodulator outputs and auxiliary inputs. Table 5.2 provides a brief classification of the tools.

Table 5.2: Tools for time domain and frequency domain analysis

	Time Domain	Frequency Domain
Fast signals (1.8 GSa/s)	Oscilloscope (Scope tab)	FFT Analyzer (Scope tab)
Periodic Waveform Analyzer (Boxcar tab)	Multi-Harmonic Analyzer (Boxcar tab)	
Slow signals (<28 MSa/s)	Numeric	Spectrum Analyzer (Spectrum tab)
Plotter	Sweeper	
Data Acquisition	Multi-harmonic Analyzer (Out PWA tab)	
Periodic Waveform Analyzer (Out PWA tab)	-	

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 5.3: Overview of app icons and short description

Control/ Tool	Option/ Range	Description
Lock-in		Quick overview and access to all the settings and properties for signal generation and demodulation.
Lock-in MF		Quick overview and access to all the settings and properties for signal generation and demodulation.
Files		Access settings and measurement data files on the host computer.
Numeric		Access to all continuously streamed measurement data as numerical values.
Plotter		Displays various continuously streamed measurement data as traces over time (roll mode).
Scope		Displays shots of data samples in time and frequency domain (FFT) representation.
DAQ		Provides complex trigger functionality on all continuously streamed data samples and time domain display.
Spectrum		Provides FFT functionality to all continuously streamed measurement data.
Sweeper		Sweep frequencies, voltages, and other quantities over a defined range and display various response functions including statistical operations.
AU		Real-time arithmetic operations on demodulator outputs.
Aux		Controls all settings regarding the auxiliary inputs and auxiliary outputs.
In/Out		Gives access to all controls relevant for the Signal Inputs and Signal Outputs of each channel.
DIO		Gives access to all controls relevant for the digital inputs and outputs including DIO, Trigger Inputs, and Marker Outputs.
Config		Provides access to software configuration.
Device		Provides instrument specific settings.
PID		Features all control, analysis, and simulation capabilities of the PID controllers.
MOD		Control panel to enable (de)modulation at linear combinations of oscillator frequencies.
Boxcar		Boxcar settings and periodic waveform analyzer for fast input signals.
Out PWA		Multi-channel boxcar settings and measurement analysis for boxcar outputs.
AWG		Generate arbitrary signals using sequencing and sample-by-sample definition of waveforms.

Control/ Tool	Option/ Range	Description
Counter		Configure the Pulse Counters for analysis of pulse trains on the digital signal inputs.
MDS		Synchronize multiple instruments.
ZI Labs		Experimental settings and controls.

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

Control/ Tool	Option/ Range	Description
Next Calibration	Time or "M"	Remaining minutes until the first calibration is executed or a recalibration is requested. A time interval longer than 99 minutes is not displayed. Manual calibration mode is indicated by an "M".
CAL	grey/ yellow/red	State of device self calibration. Yellow: device is warming up and will automatically execute a self calibration after 16 minutes. Grey: device is warmed-up and self calibrated. Red: it is recommended to manually execute a self calibration to assure operation according to specifications.
AWG	grey/green	Arbitrary Waveform Generator - Green: indicates that the AWG core is enabled.
CNT	grey/green	Pulse Counter - Green: indicates which of the pulse counter modules is enabled.
DAC Error	grey/green	Red indicates that the digital to analog converter at the output encountered an error during operation. An error leads to additional jitter in the output wave, scrambled output or no output at all. If an error is encountered, please contact Zurich Instruments for support.
AU	grey/green/ red	Arithmetic Unit - Green: indicates which of the arithmetic units is enabled. Red: indicates overflow.
OVI	grey/ yellow/red	Signal Input Overload - Red: present overload condition on the signal input also shown by the red front panel LED. Yellow: indicates an overload occurred in the past.
OVO	grey/ yellow/red	Overload Signal Output - Red: present overload condition on the signal output. Yellow: indicates an overload occurred in the past.
COM	grey/red	Stall - Red: indicates that the sample transfer rates have been reset to default values to prevent severe communication failure. This is typically caused by high sample transfer rates on a slow host computer.
RUB	grey/ yellow/ green	Rubidium Clock - Grey: no rubidium clock is installed. Yellow: Rubidium clock is warming up (takes approximately 300 s). Green: Rubidium clock is warmed up and locked.
BOX	grey/green	Boxcar - Green: indicates which of the boxcar units is enabled.
MOD	grey/green	MOD - Green: indicates which of the modulation kits is enabled.
PID	grey/green	PID - Green: indicates which of the PID units is enabled. Red: indicates PID unit is in PLL or ExtRef mode but is not locked. Yellow: indicates PID unit was not locked in the past.
PLL	grey/green	PLL - Green: indicates which of the PLLs is enabled.
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

Control/ Tool	Option/ Range	Description
MDS	grey/green/ red/yellow	Multiple device synchronization indicator. Grey: Nothing to synchronize - single device on the UI. Green: All devices on the UI are correctly synchronized. Yellow: MDS sync in progress or only a subset of the connected devices is synchronized. Red: Devices not synchronized or error during MDS sync.
REC	grey/red	A blinking red indicator shows ongoing data recording (related to global recording settings in the Config tab).
CF	grey/ yellow/red	Clock Failure - Red: present malfunction of the external 10 MHz reference oscillator. Yellow: indicates a malfunction occurred in the past.
COM	grey/ yellow/red	Packet Loss - Red: present loss of data between the device and the host PC. Yellow: indicates a loss occurred in the past.
COM	grey/ yellow/red	Sample Loss - Red: present loss of sample data between the device and the host PC. Yellow: indicates a loss occurred in the past.
C	c	Reset status flags: Clear the current state of the status flags
Full Screen	⛶	Toggles the browser between full screen and normal mode.

Status bar description

5.2.2. Plot Functionality

Several tools provide a graphical display of measurement data in the form of plots. These are multi-functional tools with zooming, panning and cursor capability. This section introduces some of the highlights.

Plot Area Elements

Plots consist of the plot area, the X range and the range controls. The X range (above the plot area) indicates which section of the wave is displayed by means of the blue zoom region indicators. The two ranges show the full scale of the plot which does not change when the plot area displays a zoomed view. The two axes of the plot area instead do change when zoom is applied.

The mouse functionality inside of a plot greatly simplifies and speeds up data viewing and navigation.

Table 5.4: Mouse functionality inside plots

Name	Action	Description	Performed inside
Panning	left click on any location and move around	moves the waveforms	plot area
Zoom X axis	mouse wheel	zooms in and out the X axis	plot area
Zoom Y axis	shift + mouse wheel	zooms in and out the Y axis	plot area
Window zoom	shift and left mouse area select	selects the area of the waveform to be zoomed in	plot area
Absolute jump of zoom area	left mouse click	moves the blue zoom range indicators	X and Y range, but outside of the blue zoom range indicators
Absolute move of zoom area	left mouse drag-and-drop	moves the blue zoom range indicators	X and Y range, inside of the blue range indicators
Full Scale	double click	set X and Y axis to full scale	plot area

Each plot area contains a legend that lists all the shown signals in the respective color. The legend can be moved to any desired position by means of drag-and-drop.

The X range and Y range plot controls are described in Table 5.5.

Note

Plot data can be conveniently exported to other applications such as Excel or Matlab by using LabOne's Net Link functionality, see LabOne Net Link for more information.

Table 5.5: Plot control description

Control/ Tool	Option/ Range	Description
Axis scaling mode	Auto FS Man Auto	Selects between automatic, full scale and manual axis scaling.
Axis mapping mode	Lin Lin Log dB	Select between linear, logarithmic and decibel axis mapping.
Axis zoom in		Zooms the respective axis in by a factor of 2.
Axis zoom out		Zooms the respective axis out by a factor of 2.
Rescale axis to data		Rescale the foreground Y axis in the selected zoom area.
Save figure		Generates PNG, JPG or SVG of the plot area or areas for dual plots to the local download folder.
Save data		Generates a CSV file consisting of the displayed wave or histogram data (when histogram math operation is enabled). Select full scale to save the complete wave. The save data function only saves one shot at a time (the last displayed wave).
Cursor control	Off Off	Cursors can be switch On/Off and set to be moved both independently or one bound to the other one.
Net Link		Provides a LabOne Net Link to use displayed wave data in tools like Excel, MATLAB, etc.

Cursors and Math

The plot area provides two X and two Y cursors which appear as dashed lines inside of the plot area. The four cursors are selected and moved by means of the blue handles individually by means of drag-and-drop. For each axis, there is a primary cursor indicating its absolute position and a secondary cursor indicating both absolute and relative position to the primary cursor.

Cursors have an absolute position which does not change upon pan or zoom events. In case a cursor position moves out of the plot area, the corresponding handle is displayed at the edge of the plot area. Unless the handle is moved, the cursor keeps the current position. This functionality is very effective to measure large deltas with high precision (as the absolute position of the other cursors does not move).

The cursor data can also be used to define the input data for the mathematical operations performed on plotted data. This functionality is available in the Math sub-tab of each tool. The Table 5.6 gives an overview of all the elements and their functionality. The chosen Signals and Operations are applied to the currently active trace only.

Note

Cursor data can be conveniently exported to other applications such as Excel or MATLAB by using LabOne's Net Link functionality, see LabOne Net Link for more information.

Table 5.6: Plot math description

Control/Tool	Option/Range	Description
Source Select		Select from a list of input sources for math operations.
Cursor Loc	Cursor coordinates as input data.	
Cursor Area	Consider all data of the active trace inside the rectangle defined by the cursor positions as input for statistical functions (Min, Max, Avg, Std).	
Tracking	Display the value of the active trace at the position of the horizontal axis cursor X1 or X2.	
Plot Area	Consider all data of the active trace currently displayed in the plot as input for statistical functions (Min, Max, Avg, Std).	
Peak	Find positions and levels of up to 5 highest peaks in the data.	
Trough	Find positions and levels of up to 5 lowest troughs in the data.	
Histogram	Display a histogram of the active trace data within the x-axis range. The histogram is used as input to statistical functions (Avg, Std). Because of binning, the statistical functions typically yield different results than those under the selection Plot Area.	
Resonance	Display a curve fitted to a resonance.	
Linear Fit	Display a linear regression curve.	
Operation Select		Select from a list of mathematical operations to be performed on the selected source. Choice offered depends on the selected source.
Cursor Loc: X1, X2, X2-X1, Y1, Y2, Y2-Y1, Y2 / Y1	Cursors positions, their difference and ratio.	
Cursor Area: Min, Max, Avg, Std	Minimum, maximum value, average, and bias-corrected sample standard deviation for all samples between cursor X1 and X2. All values are shown in the plot as well.	
Tracking: Y(X1), Y(X2), ratioY, deltaY	Trace value at cursor positions X1 and X2, the ratio between these two Y values and their difference.	
Plot Area: Min, Max, Pk Pk, Avg, Std	Minimum, maximum value, difference between min and max, average, and bias-corrected sample standard deviation for all samples in the x axis range.	
Peak: Pos, Level	Position and level of the peak, starting with the highest one. The values are also shown in the plot to identify the peak.	
Histogram: Avg, Std, Bin Size, (Plotter tab only: SNR, Norm Fit, Rice Fit)	A histogram is generated from all samples within the x-axis range. The bin size is given by the resolution of the screen: 1 pixel = 1 bin. From this histogram, the average and bias-corrected sample standard deviation is calculated, essentially assuming all data points in a bin lie in the center of their respective bin. When used in the plotter tab with demodulator or boxcar signals, there additionally are the options of SNR estimation and fitting statistical distributions to the histogram (normal and rice distribution).	

Control/Tool	Option/Range	Description
Resonance: Q, BW, Center, Amp, Phase, Fit Error	A curve is fitted to a resonator. The fit boundaries are determined by the two cursors X1 and X2. Depending on the type of trace (Demod R or Demod Phase) either a Lorentzian or an inverse tangent function is fitted to the trace. The Q is the quality factor of the fitted curve. BW is the 3dB bandwidth (FWHM) of the fitted curve. Center is the center frequency. Amp gives the amplitude (Demod R only), whereas Phase returns the phase at the center frequency of the resonance (demod Phase only). The fit error is given by the normalized root-mean-square deviation. It is normalized by the range of the measured data.	
Linear Fit: Intercept, Slope, R^2	A simple linear least squares regression is performed using a QR decomposition routine. The fit boundaries are determined by the two cursors X1 and X2. The parameter outputs are the Y-axis intercept, slope and the R^2-value, which is the coefficient of determination to determine the goodness-of-fit.	
Add	Add	Add the selected math function to the result table below.
Add All	Add All	Add all operations for the selected signal to the result table below.
Clear Selected	Clear	Clear selected lines from the result table above.
Clear All	Clear All	Clear all lines from the result table above.
Copy	Copy	Copy selected row(s) to Clipboard as CSV
Unit Prefix		Adds a suitable prefix to the SI units to allow for better readability and increase of significant digits displayed.
CSV	CSV	Values of the current result table are saved as a text file into the download folder.
Net Link	Link	Provides a LabOne Net Link to use the data in tools like Excel, MATLAB, etc.
Help	Help	Opens the LabOne User Interface help.

Note

The standard deviation is calculated using the formula $\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$ for the unbiased estimator of the sample standard deviation with a total of N samples x_i and an arithmetic average $\bar{x}$. The formula above is used as-is to calculate the standard deviation for the Histogram Plot Math tool. For large number of points (Cursor Area and Plot Area tools), the more accurate pairwise algorithm is used (Chan et al., "Algorithms for Computing the Sample Variance: Analysis and Recommendations", The American Statistician 37 (1983), 242-247).

Note

The fitting functions used in the Resonance Plot Math tool depend on the selected signal source. The demodulator R signal is fitted with the following function:

$$\begin{equation}\tag{1} R(f)=C+A\frac{f}{\sqrt{f^2+\left(\frac{Q}{f_0}\right)^2\left(f^2-f^2_0\right)^2}}\end{equation}$$

where **C** accounts for a possible offset in the output, **A** is the amplitude, **Q** is the quality factor and f_0 is the center frequency. The demodulator **ϕ** signal s fitted with the following function:

$$\begin{equation}\tag{2} \phi(f)=tan^{-1}\left(Q\frac{1-\left(\frac{f}{f_0}\right)^2}{\frac{f}{f_0}}\right) \end{equation}$$

using the same parameters as above.

Tree Selector

The Tree selector allows one to access streamed measurement data in a hierarchical structure by checking the boxes of the signals that should be displayed. The tree selector also supports data selection from multiple instruments, where available. Depending on the tool, the Tree selector is either displayed in a separate Tree sub-tab, or it is accessible by a click on the button.

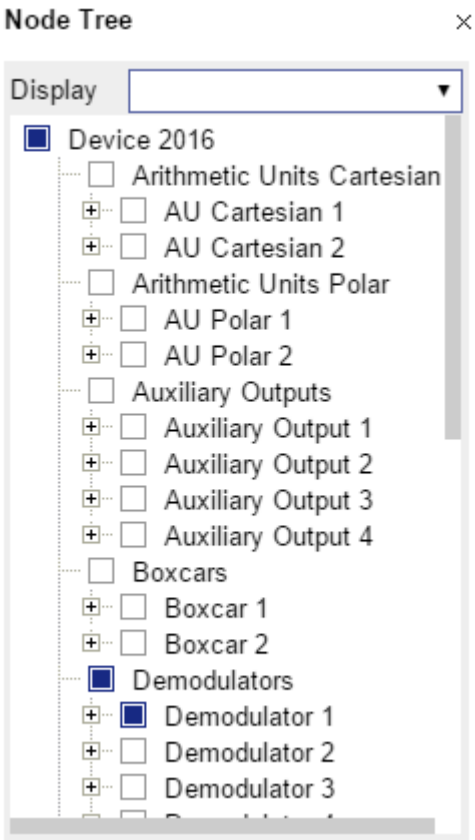


Figure 5.3: Tree selector with Display drop-down menu

Vertical Axis Groups

Vertical Axis groups are available as part of the plot functionality in many of the LabOne tools. Their purpose is to handle signals with different axis properties within the same plot. Signals with different units naturally have independent vertical scales even if they are displayed in the same plot. However, signals with the same unit should preferably share one scaling to enable quantitative comparison. To this end, the signals are assigned to specific axis group. Each axis group has its own axis system. This default behavior can be changed by moving one or more signals into a new group.

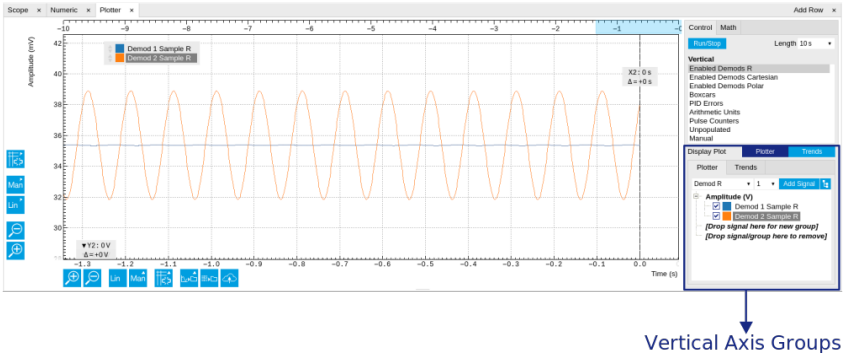


Figure 5.4: Vertical Axis Group in Plotter tool

The tick labels of only one axis group can be shown at once. This is the foreground axis group. To define the foreground group click on one of the group names in the Vertical Axis Groups box. The current foreground group gets a high contrast color.

Select foreground group

Click on a signal name or group name inside the Vertical Axis Groups. If a group is empty the selection is not performed.

Split the default vertical axis group

Use drag-and-drop to move one signal on the field **[Drop signal here to add a new group]**. This signal will now have its own axis system.

Change vertical axis group of a signal

Use drag-and-drop to move a signal from one group into another group that has the same unit.

Group separation

In case a group hosts multiple signals and the unit of some of these signals changes, the group will be split in several groups according to the different new units.

Remove a signal from the group

In order to remove a signal from a group drag-and-drop the signal to a place outside of the Vertical Axis Groups box.

Remove a vertical axis group

A group is removed as soon as the last signal of a custom group is removed. Default groups will remain active until they are explicitly removed by drag-and-drop. If a new signal is added that match the group properties it will be added again to this default group. This ensures that settings of default groups are not lost, unless explicitly removed.

Rename a vertical axis group

New groups get a default name "Group of ...". This name can be changed by double-clicking on the group name.

Hide/show a signal

Uncheck/check the check box of the signal. This is faster than fetching a signal from a tree again.

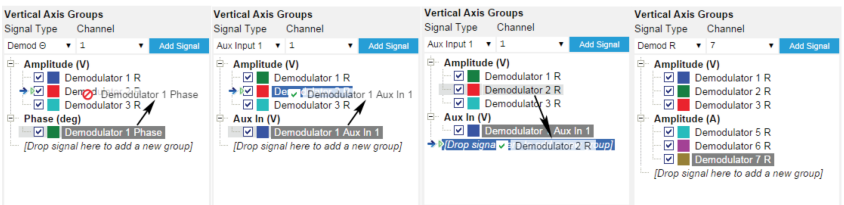


Figure 5.5: Vertical Axis Group typical drag and drop moves.

Table 5.7: Vertical Axis Groups description

Control/ Tool	Option/ Range	Description
Vertical Axis Group		Manages signal groups sharing a common vertical axis. Show or hide signals by changing the check box state. Split a group by dropping signals to the field [Drop signal here to add new group]. Remove signals by dragging them on a free area. Rename group names by editing the group label. Axis tick labels of the selected group are shown in the plot. Cursor elements of the active wave (selected) are added in the cursor math tab.
Signal Type	Demod X, Y, R, Theta	Select signal types for the Vertical Axis Group.
	Frequency	
	Aux Input 1, 2	
	HW Trigger	
	PID Error	
	PID Shift	
	PID Value	
	Boxcar	
	AU Cartesian	
	AU Polar	
Channel	integer value	Selects a channel to be added.
Signal	integer value	Selects signal to be added.
Add Signal	Add Signal	Adds a signal to the plot. The signal will be added to its default group. It may be moved by drag and drop to its own group. All signals within a group share a common y-axis. Select a group to bring its axis to the foreground and display its labels.
Window Length	2 s to 12 h	Window memory depth. Values larger than 10 s may cause excessive memory consumption for signals with high sampling rates. Auto scale or pan causes a refresh of the display for which only data within the defined window length are considered.

Trends

The Trends tool lets the user monitor the temporal evolution of signal features such as minimum and maximum values, or mean and standard deviation. This feature is available for the Scope , Spectrum, Plotter, and DAQ tab. Using the Trends feature, one can monitor all the parameters obtained in the Math sub-tab of the corresponding tab.

The Trends tool allows the user to analyze recorded data on a different and adjustable time scale much longer than the fast acquisition of measured signals. It saves time by avoiding post-processing of recorded signals and it facilitates fine-tuning of experimental parameters as it extracts and shows the measurement outcome in real time.

To activate the Trends plot, enable the Trends button in the Control sub-tab of the corresponding main tab. Various signal features can be added to the plot from the Trends sub-tab in the Vertical Axis Groups . The vertical axis group of Trends has its own Run/Stop button and Length setting independent from the main plot of the tab. Since the Math quantities are derived from the raw signals in the main plot, the Trends plot is only shown together with the main plot. The Trends feature is only available in the LabOne user interface and not at the API level.

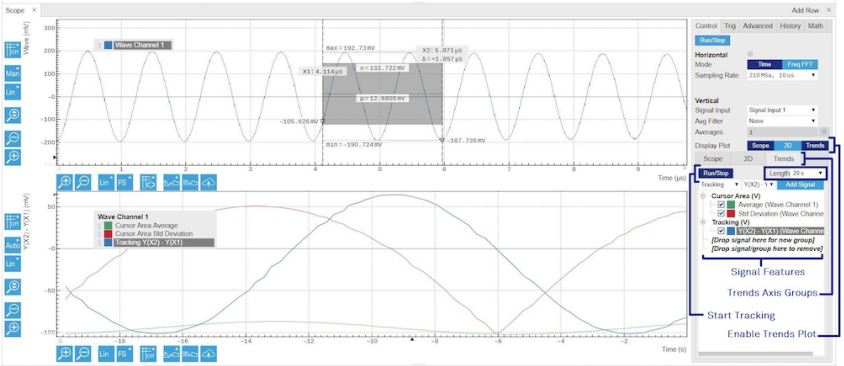


Figure 5.6: Top: main plot of the Scope tab showing the signal trace. Bottom: corresponding Trends plot tracking an average, standard deviation, and difference signal derived from the cursor positions in the main plot. The example shown is part of the HF2LI user interface. The controls of the Trends feature and their layout are very similar in all tabs and product platforms where this feature is available.

5.3. Saving and Loading Data

5.4. Overview

In this section we discuss how to save and record measurement data with the UHF Series Instrument using the LabOne user interface. In the LabOne user interface, there are 3 ways to save data:

- Saving the data that is currently displayed in a plot
- Continuously recording data in the background
- Saving trace data in the History sub-tab

Furthermore, the History sub-tab supports loading data. In the following, we will explain these methods.

5.4.1. Saving Data from Plots

A quick way to save data from any plot is to click on the Save CSV icon at the bottom of the plot to store the currently displayed curves as a comma-separated value (CSV) file to the download folder of your web browser. Clicking on will save a graphics file instead.

5.4.2. Recording Data

The recording functionality allows you to store measurement data continuously, as well as to track instrument settings over time. The Config Tab gives you access to the main settings for this function. The Format selector defines which format is used: HDF5, CSV, or MATLAB. The CSV delimiter character can be changed in the User Preferences section. The default option is Semicolon.

The node tree display of the Record Data section allows you to browse through the different measurement data and instrument settings, and to select the ones you would like to record. For instance, the demodulator 1 measurement data is accessible under the path of the form `Device 0000/Demodulators/Demod 1/Sample`. An example for an instrument setting would be the filter time constant, accessible under the path `Device 0000/Demodulators/Demod 1/Filter Time Constant`.

The default storage location is the LabOne Data folder which can, for instance, be accessed by the Open Folder button . The exact path is displayed in the Folder field whenever a file has been written.

Clicking on the Record checkbox will initiate the recording to the hard drive. In case of demodulator and boxcar data, ensure that the corresponding data stream is enabled, as otherwise no data will be saved.

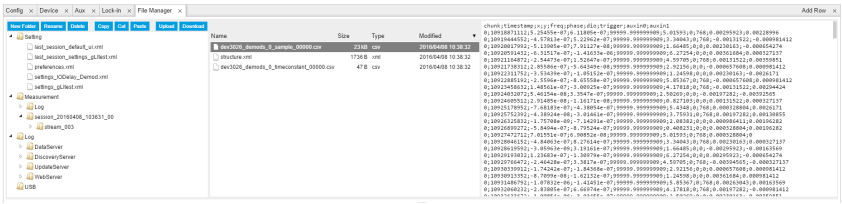


Figure 5.7: Browsing and inspecting files in the LabOne File Manager tab

In case HDF5 or MATLAB is selected as the file format, LabOne creates a single file containing the data for all selected nodes. For the CSV format, at least one file for each of the selected nodes is created from the start. At a configurable time interval, new data files are created, but the maximum size is capped at about 1 GB for easier data handling. The storage location is indicated in the Folder field of the Record Data section.

The File Manager Tab is a good place to inspect CSV data files. The file browser on the left of the tab allows you to navigate to the location of the data files and offers functionalities for managing files in the LabOne Data folder structure. In addition, you can conveniently transfer files between the folder structure and your preferred location using the Upload/Download buttons. The file viewer on the right side of the tab displays the contents of text files up to a certain size limit. Figure 5.7 shows the Files tab after recording Demodulator Sample and Filter Time Constant for a few seconds. The file viewer shows the contents of the demodulator data file.

Note

The structure of files containing instrument settings and of those containing streamed data is the same. Streaming data files contain one line per sampling period, whereas in the case of instrument settings, the file usually only contains a few lines, one for each change in the settings. More information on the file structure can be found in the LabOne Programming Manual.

5.4.3. History List

Tabs with a history list such as Sweeper Tab, Data Acquisition Tab , Scope Tab, Spectrum Analyzer Tab support feature saving, autosaving, and loading functionality. By default, the plot area in those tools displays the last 100 measurements (depending on the tool, these can be sweep traces, scope shots, DAQ data sets, or spectra), and each measurement is represented as an entry in the History sub-tab. The button to the left of each list entry controls the visibility of the corresponding trace in the plot; the button to the right controls the color of the trace. [1]Double-clicking on a list entry allows you to rename it. All measurements in the history list can be saved with Save All. Clicking on the Save Sel button (note the dropdown button ▼) saves only those traces that were selected by a mouse click. Use the Control or Shift button together with a mouse click to select multiple traces. The file location can be accessed by the Open Folder button ↗. Figure 5.10.8 illustrates some of these features. Figure 5.8 illustrates the data loading feature.

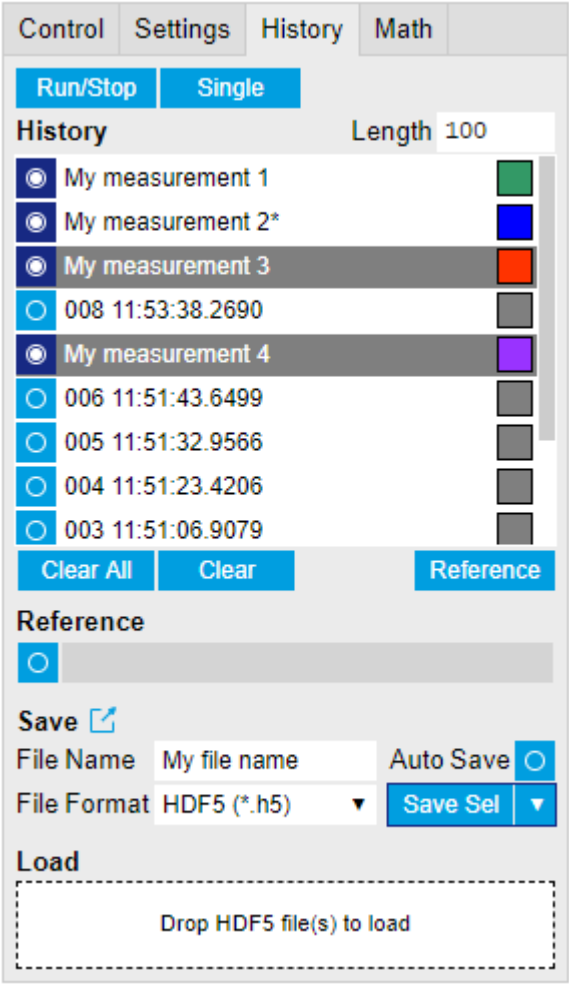


Figure 5.8: History sub-tab features. The entries "My measurement 1" etc. were renamed by the user. Measurement 1, 2, 3, 4 are currently displayed in the plot because their left-hand-side button is enabled. Clicking on Save Sel would save "My measurement 3" and "My measurement 4" to a file, because these entries were selected (gray overlay) by a Control key + mouse click action.

Which quantities are saved depends on which signals have been added to the Vertical Axis Groups section in the **Control** sub-tab. Only data from demodulators with enabled Data Transfer in the Lock-in tab can be included in the files.

The history sub-tab supports an **autosave** functionality to store measurement results continuously while the tool is running. Autosave directories are differentiated from normal saved directories by the text "autosave" in the name, e.g. sweep_autosave_000. When running a tool continuously (Run/Stop button) with Autosave activated, after the current measurement (history entry) is complete, all measurements in the history are saved. The same file is overwritten each time, which means that old measurements will be lost once the limit defined by the history Length setting has been reached. When performing single measurements (Single button) with Autosave activated, after each measurement, the elements in the history list are saved in a new directory with an incrementing count, e.g. `sweep_autosave_001`, `sweep_autosave_002`.

Data which was saved in HDF5 file format can be loaded back into the history list. Loaded traces are marked by a prefix "loaded " that is added to the history entry name in the user interface. The `createdtimestamp` information in the header data marks the time at which the data were measured.

- Only files created by the Save button in the History sub-tab can be loaded.
- Loading a file will add all history items saved in the file to the history list. Previous entries are kept in the list.
- Data from the file is only displayed in the plot if it matches the current settings in the Vertical Axis Group section the tool. Loading e.g. PID data in the Sweeper will not be shown, unless it is selected in the Control sub-tab.
- Files can only be loaded if the devices saving and loading data are of the same product family. The data path will be set according to the device ID loading the data.

Figure 5.9 illustrates the data loading feature.

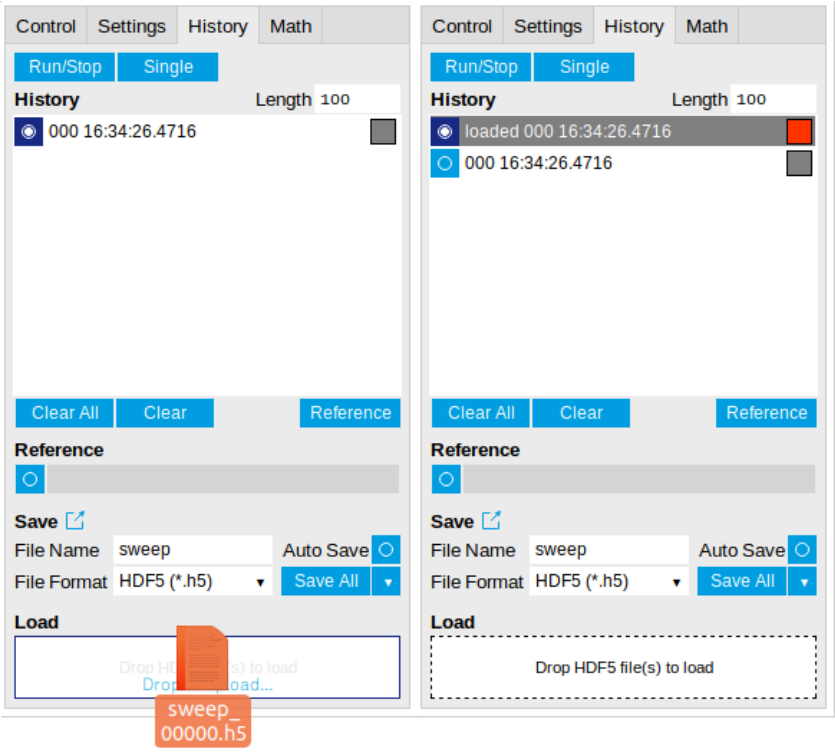


Figure 5.9: History data loading feature. Here, the file sweep_00000.h5 is loaded by drag-and-drop. The loaded data are added to the measurements in the history list.

5.4.4. Supported File Formats

HDF5

Hierarchical Data File 5 (HDF5) is a widespread memory-efficient, structured, binary, open file format. Data in this format can be inspected using the dedicated viewer HDFview. HDF5 libraries or import tools are available for Python, MATLAB, LabVIEW, C, R, Octave, Origin, Igor Pro, and others. The following example illustrates how to access demodulator data from a sweep using the h5py library in Python:

```
import h5py
filename = 'sweep_00000.h5'
f = h5py.File(filename, 'r')
x = f['000/dev3025/demods/0/sample/frequency']
```

The data loading feature of LabOne supports HDF5 files, while it is unavailable for other formats.

MATLAB

The MATLAB File Format (.mat) is a proprietary file format from MathWorks based on the open HDF5 file format. It has thus similar properties as the HDF5 format, but the support for importing .mat files into third-party software other than MATLAB is usually less good than that for importing HDF5 files.

SXM

SXM is a proprietary file format by Nanonis used for SPM measurements.

5.4.5. LabOne Net Link

Measurement and cursor data can be downloaded from the browser as CSV data. This allows for further processing in any application that supports CSV file formats. As the data is stored internally on the web server it can be read by direct server access from other applications. Most up-to-date

software supports data import from web pages or CSV files over the internet. This allows for automatic import and refresh of data sets in many applications. To perform the import the application needs to know the address from where to load the data. This link is supplied by the LabOne User Interface. The following chapter lists examples of how to import data into some commonly used applications.

The CSV data sent to the application is a snap-shot of the data set on the web server at the time of the request. Many applications support either manual or periodic refresh functionality.

Since tabs can be instantiated several times within the same user interface, the link is specific to the tab that it is taken from. Changing the session on the LabOne User Interface or removing tabs may invalidate the link.

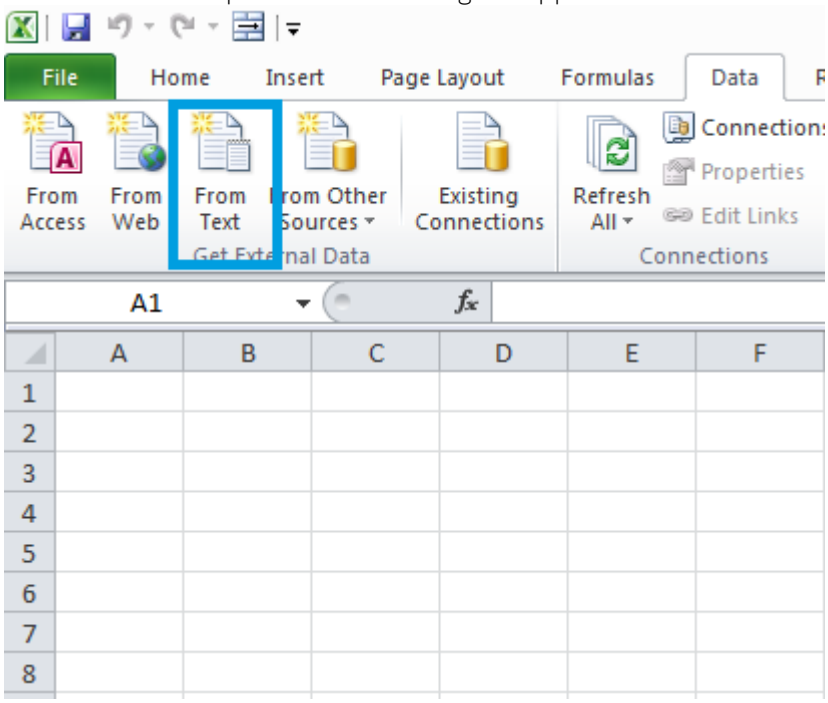
Supported applications:

- Excel
- MATLAB
- Python
- C#.NET
- Igor Pro
- Origin

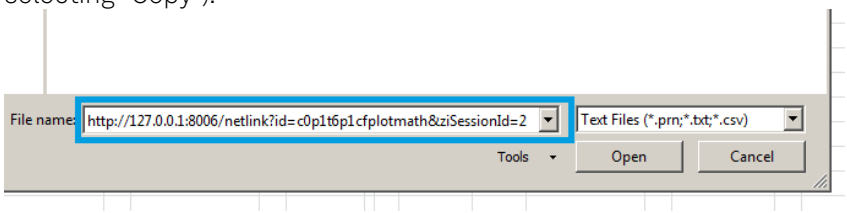
Excel

These instructions are for Excel 2010 (English). The procedure for other versions may differ.

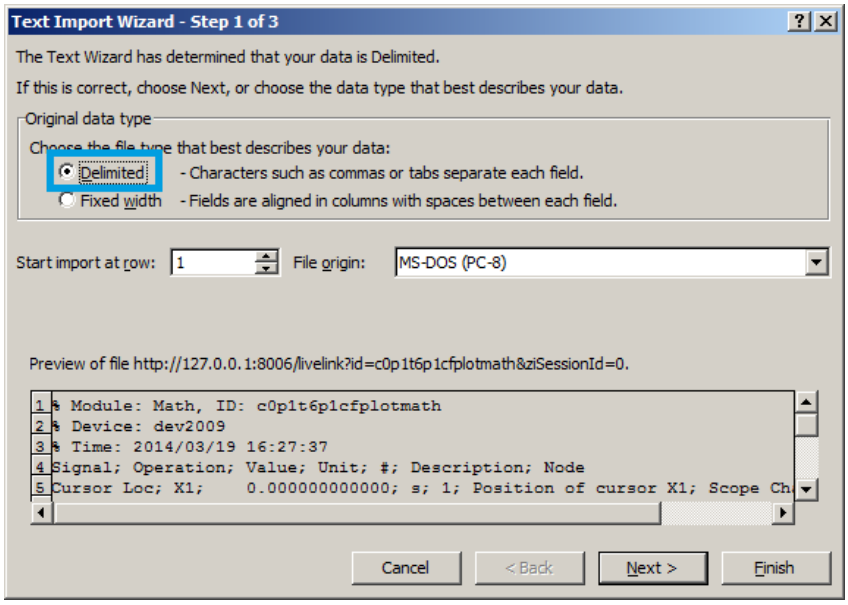
1. In Excel, click on the cell where the data is to be placed. From the Data ribbon, click the "From Text" icon. The "Import Text File" dialog will appear.



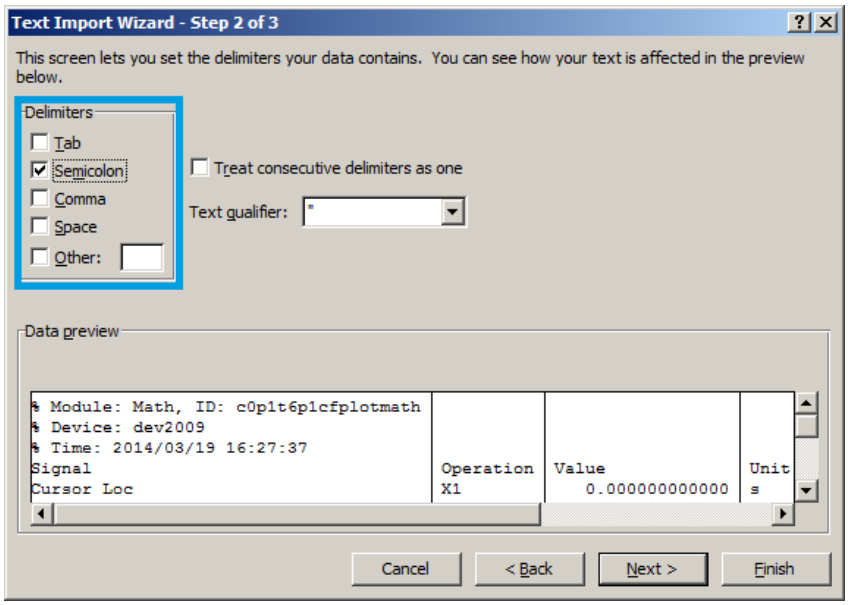
2. In LabOne, click the "Link" button of the appropriate Math tab. Copy the selected text from the "LabOne Net Link" dialog to the clipboard (either with Ctrl-C or by right clicking and selecting "Copy").



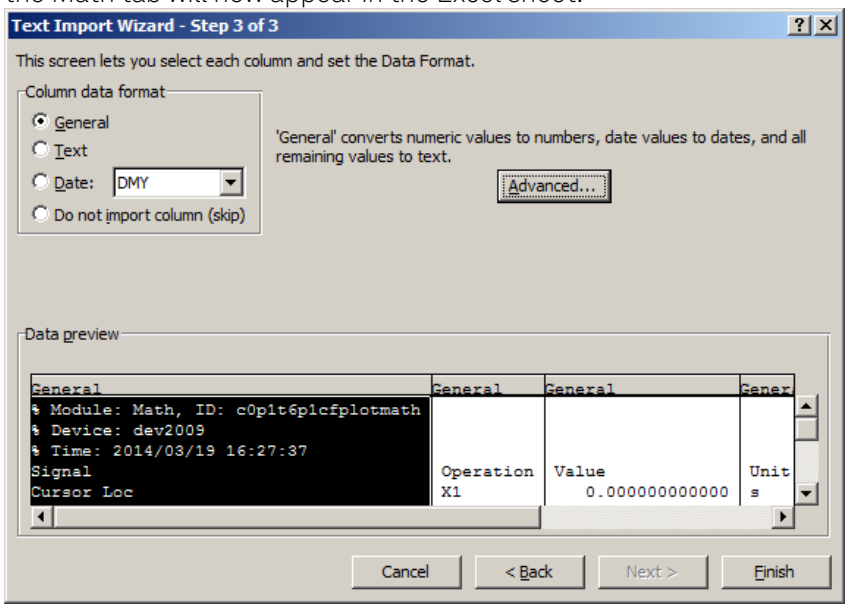
3. In Excel, paste the link into the "File name" entry field of the "Import Text File" dialog and click the "Open" button. This will start the text import wizard. Ensure that the "Delimited" button is checked before clicking the "Next" button.



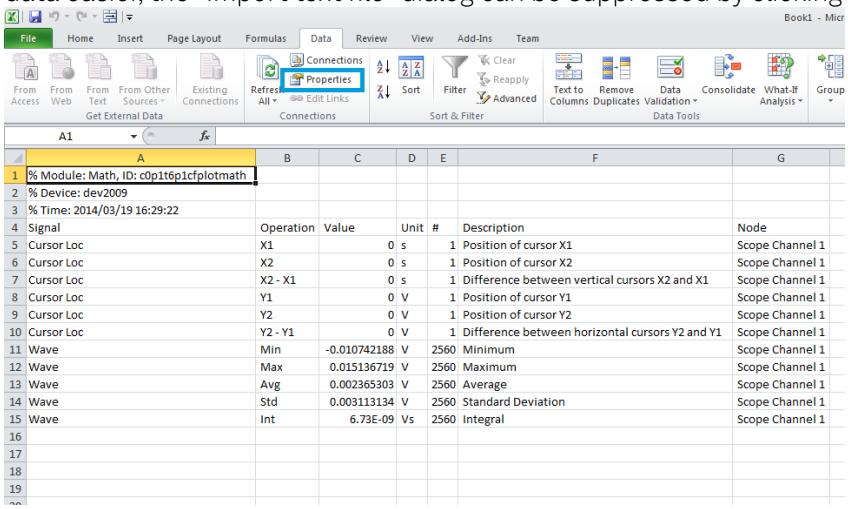
4. In the next dialog, select the delimiter character corresponding to that selected in LabOne (this can be found in the "Sessions" section of the Config tab). The default is semicolon. Click the "Next" button.



5. In the next dialog, click on "Finish" and then "OK" in the "Import Data" dialog. The data from the Math tab will now appear in the Excel sheet.

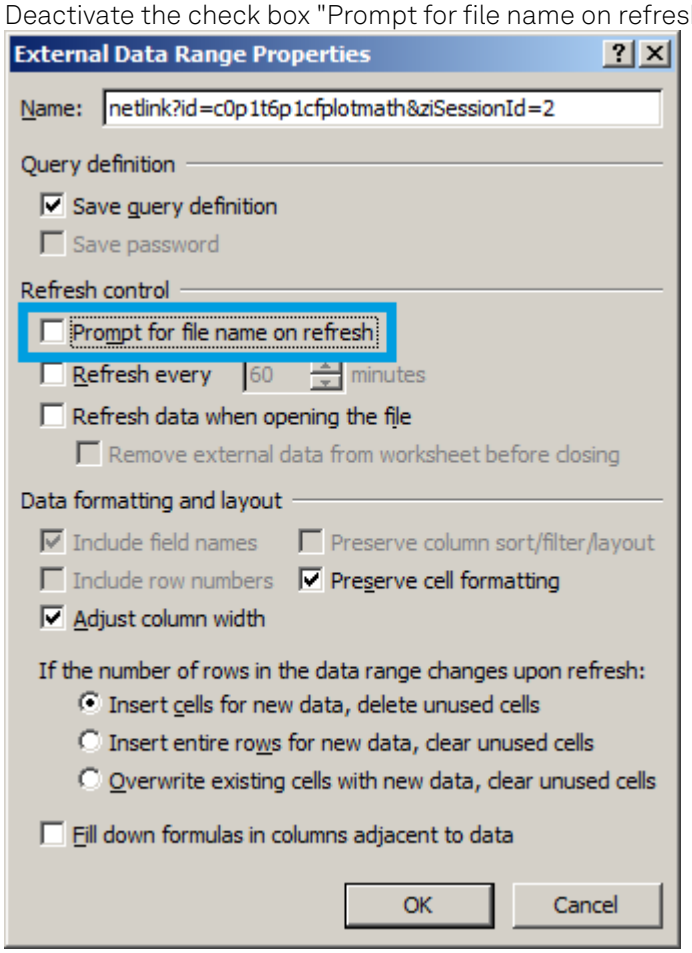


6. The data in the sheet can be updated by clicking the "Refresh All" icon. To make updating the data easier, the "Import text file" dialog can be suppressed by clicking on "Properties".



	A	B	C	D	E	F	G
1	% Module: Math, ID: c0p1t6p1cfplotmath						
2	% Device: dev2009						
3	% Time: 2014/03/19 16:29:22						
4	Signal	Operation	Value	Unit	#	Description	Node
5	Cursor Loc	X1	0	s	1	Position of cursor X1	Scope Channel 1
6	Cursor Loc	X2	0	s	1	Position of cursor X2	Scope Channel 1
7	Cursor Loc	X2 - X1	0	s	1	Difference between vertical cursors X2 and X1	Scope Channel 1
8	Cursor Loc	Y1	0	V	1	Position of cursor Y1	Scope Channel 1
9	Cursor Loc	Y2	0	V	1	Position of cursor Y2	Scope Channel 1
10	Cursor Loc	Y2 - Y1	0	V	1	Difference between horizontal cursors Y2 and Y1	Scope Channel 1
11	Wave	Min	-0.010742188	V	2560	Minimum	Scope Channel 1
12	Wave	Max	0.015136719	V	2560	Maximum	Scope Channel 1
13	Wave	Avg	0.002365303	V	2560	Average	Scope Channel 1
14	Wave	Std	0.003113134	V	2560	Standard Deviation	Scope Channel 1
15	Wave	Int	6.73E-09	Vs	2560	Integral	Scope Channel 1

7. Deactivate the check box "Prompt for file name on refresh".



MATLAB

By copying the link text from the "LabOne Net Link" dialog to the clipbord, the following code snippet can be used in MATLAB to read the data.

```
textscan(urlread(clipboard('paste')),'%s%s%f%s%d%s%s','Headerlines',
4,'Delimiter', ';')
```

Python

The following code snippet can be used in Python 2 to read the LabOne Net Link data, where "url" is assigned to the text copied from the "LabOne Net Link" dialog.

```
import csv
import urllib2
url = "http://127.0.0.1:8006/netlink?id=c0p5t6p1cfplotmath&ziSessionId=0"
webpage = urllib2.urlopen(url)
datareader = csv.reader(webpage)
data = []
for row in datareader:
    data.append(row)
```

C#.NET

The .NET Framework offers a WebClient object which can be used to send web requests to the LabOne WebServer and download LabOne Net Link data. The string with comma separated content can be parsed by splitting the data at comma borders.

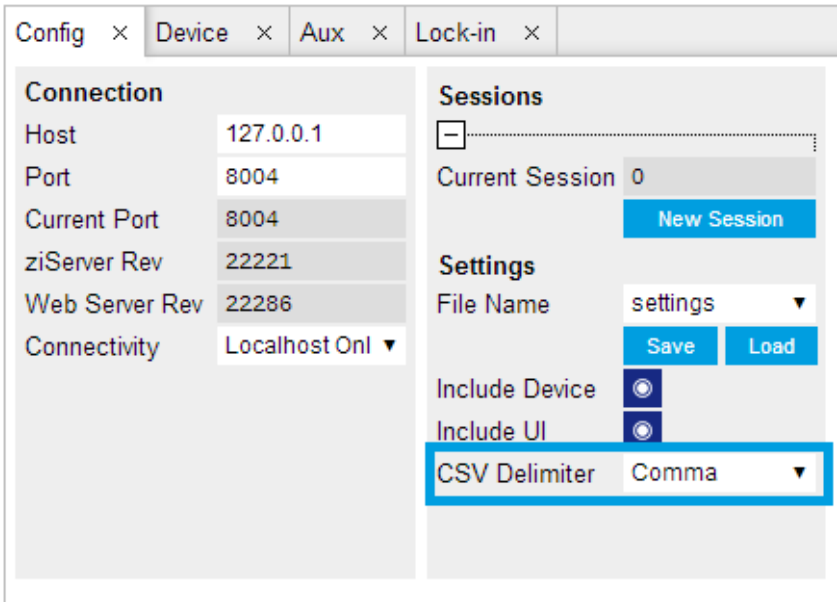
```
using System;
using System.Text;
using System.Net;

namespace ExampleCSV
{
  class Program
  {
    static void Main(string[] args)
    {
      try
      {
        WebClient wc = new WebClient();
        byte[] buffer = wc.DownloadData("http://127.0.0.1:8006/netlink?
id=c0p1t6p1cfplotmath&ziSessionId=0");
        String doc = Encoding.ASCII.GetString(buffer);
        // Parse here CSV lines and extract data
        // ...
        Console.WriteLine(doc);
      } catch (Exception e) {
        Console.WriteLine("Caught exception: " + e.Message);
      }
    }
  }
 }
```

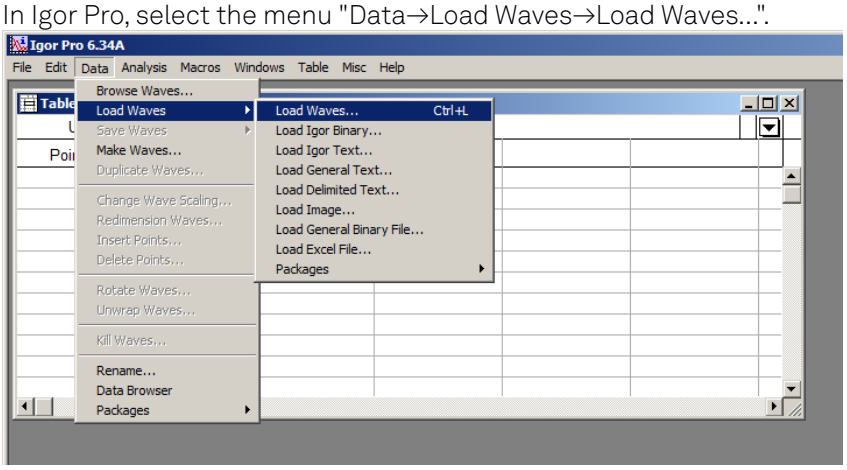
Igor Pro

These instructions are for Igor Pro 6.34A English. The procedure for other versions may differ.

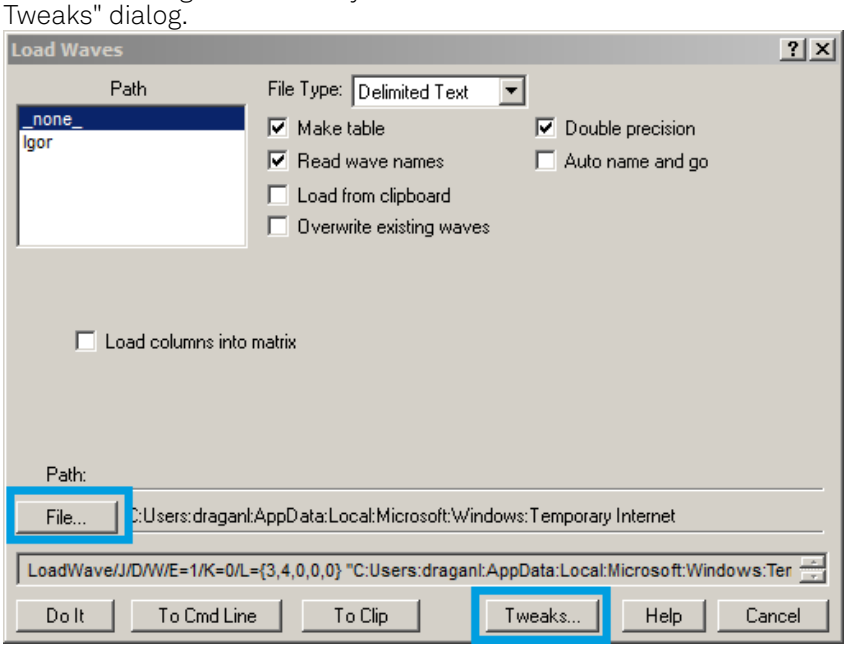
1. For Igor Pro, the CSV separator has to be the comma. Set this in the LabOne Config tab as follows:



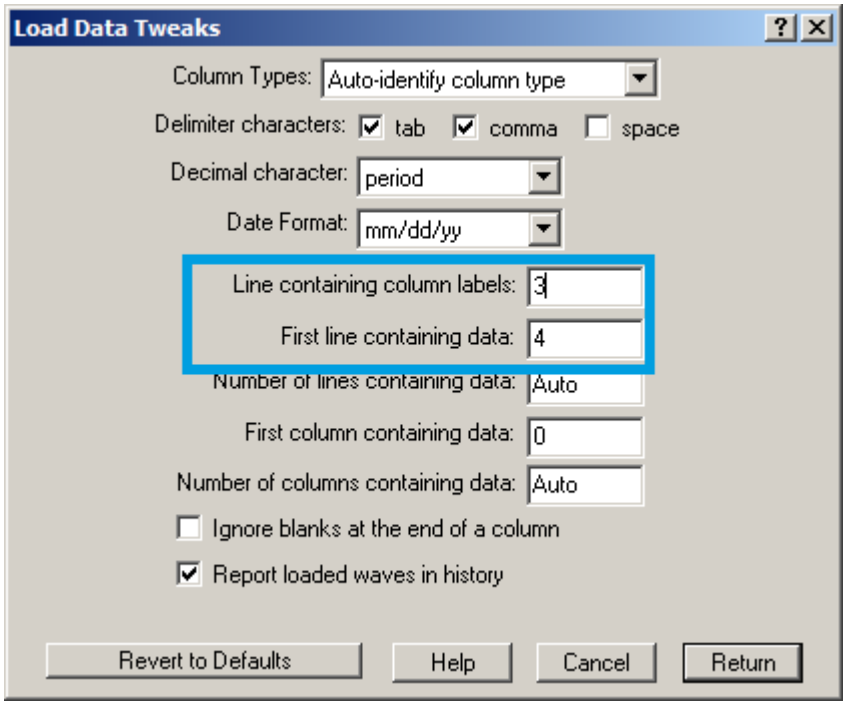
2. In Igor Pro, select the menu "Data→Load Waves→Load Waves...".



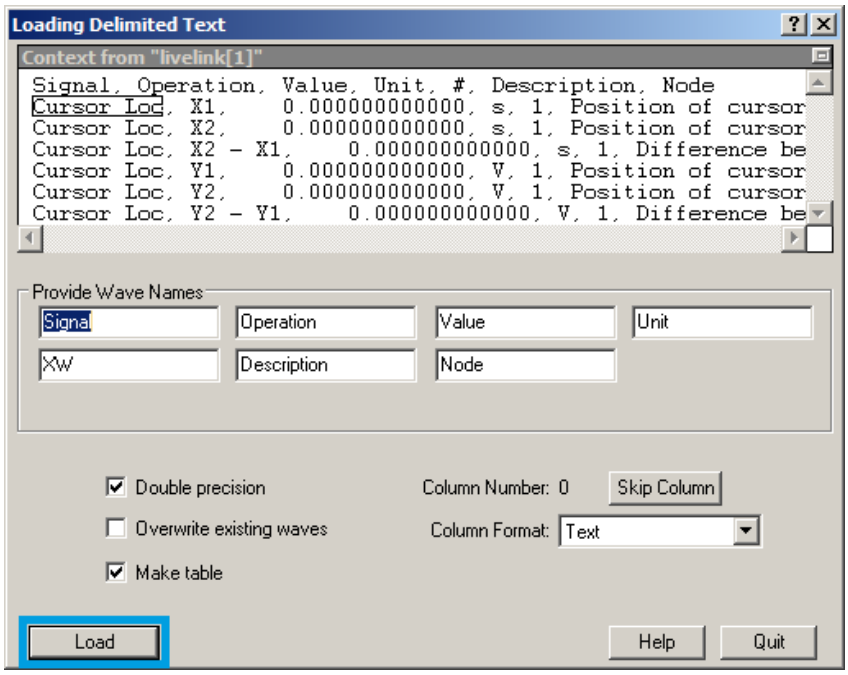
3. In the "Load Waves" dialog, click the "File..." button and paste the link text from the "LabOne Net Link" dialog into the entry field. Then click the "Tweaks..." button to open the "Load Data Tweaks" dialog.



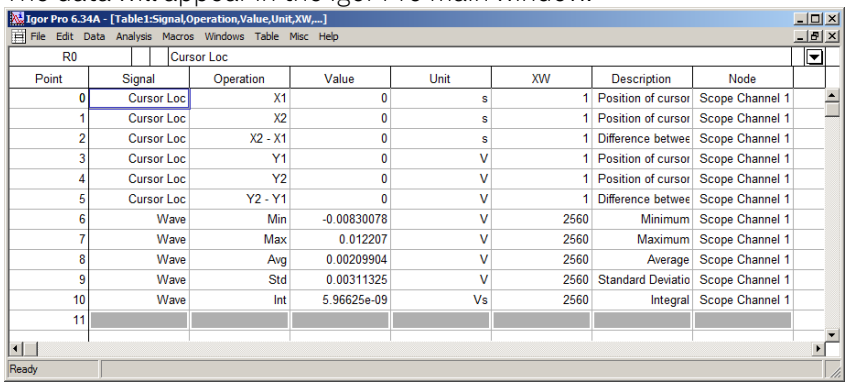
4. Adjust values as highlighted below and click "Return". The "Loading Delimited Data" dialog will appear.



5. Click the "Load" button to read the data.



6. The data will appear in the Igor Pro main window.

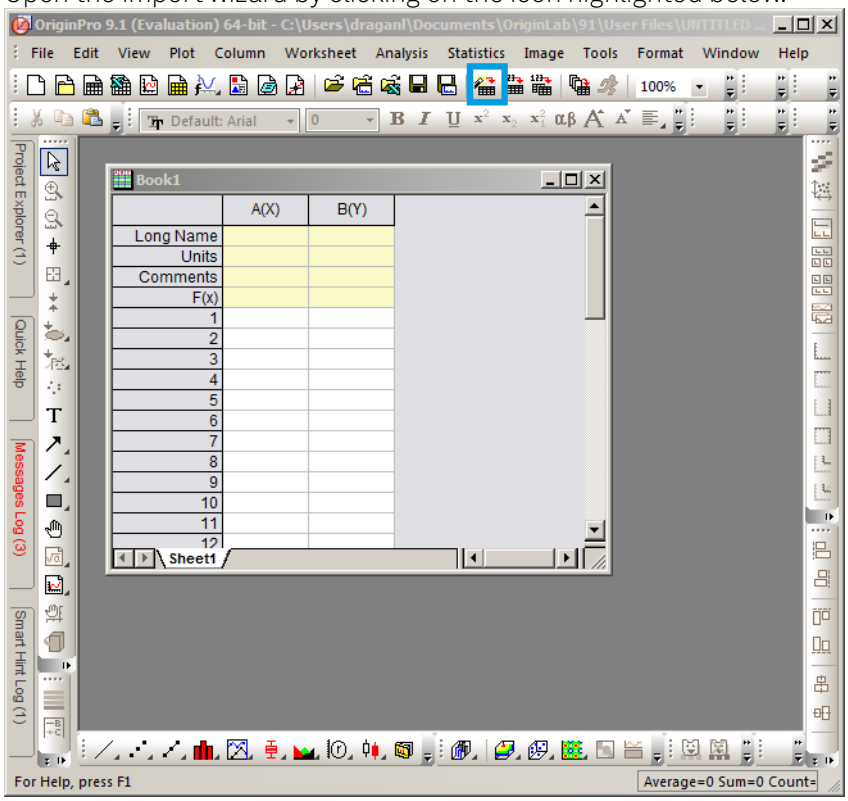


Point	Signal	Operation	Value	Unit	XW	Description	Node
0	Cursor Loc	X1	0	s	1	Position of cursor	Scope Channel 1
1	Cursor Loc	X2	0	s	1	Position of cursor	Scope Channel 1
2	Cursor Loc	X2 - X1	0	s	1	Difference betwee	Scope Channel 1
3	Cursor Loc	Y1	0	V	1	Position of cursor	Scope Channel 1
4	Cursor Loc	Y2	0	V	1	Position of cursor	Scope Channel 1
5	Cursor Loc	Y2 - Y1	0	V	1	Difference betwee	Scope Channel 1
6	Wave	Min	-0.00830078	V	2560	Minimum	Scope Channel 1
7	Wave	Max	0.012207	V	2560	Maximum	Scope Channel 1
8	Wave	Avg	0.00209904	V	2560	Average	Scope Channel 1
9	Wave	Std	0.00311325	V	2560	Standard Deviatio	Scope Channel 1
10	Wave	Int	5.96625e-09	Vs	2560	Integral	Scope Channel 1
11							

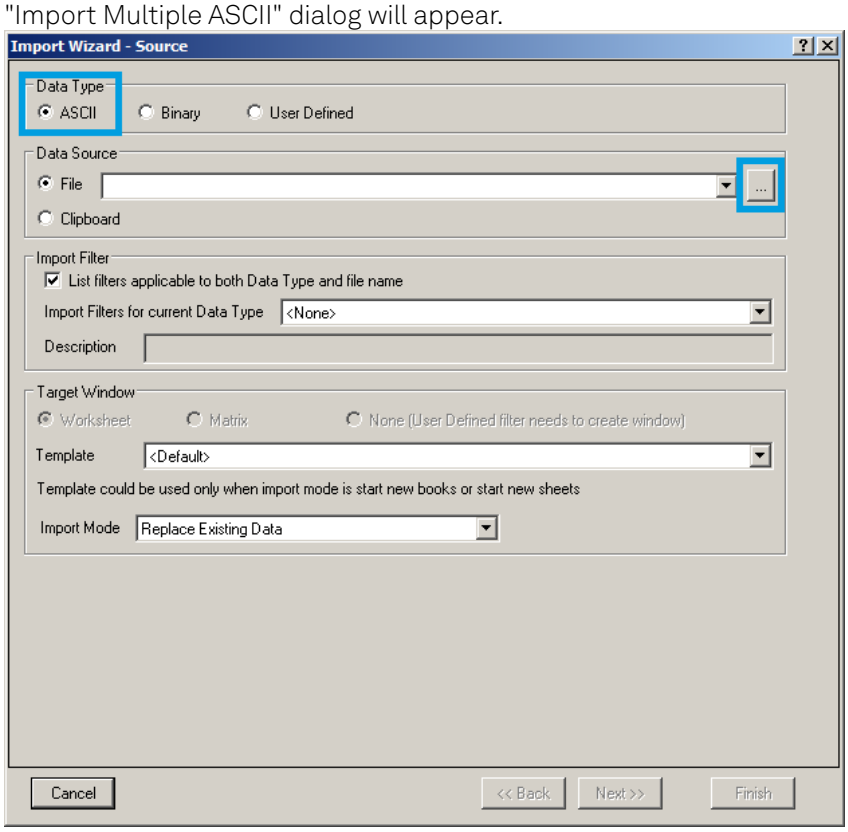
Origin

These instructions are for Origin 9.1 English. The procedure for other versions may differ.

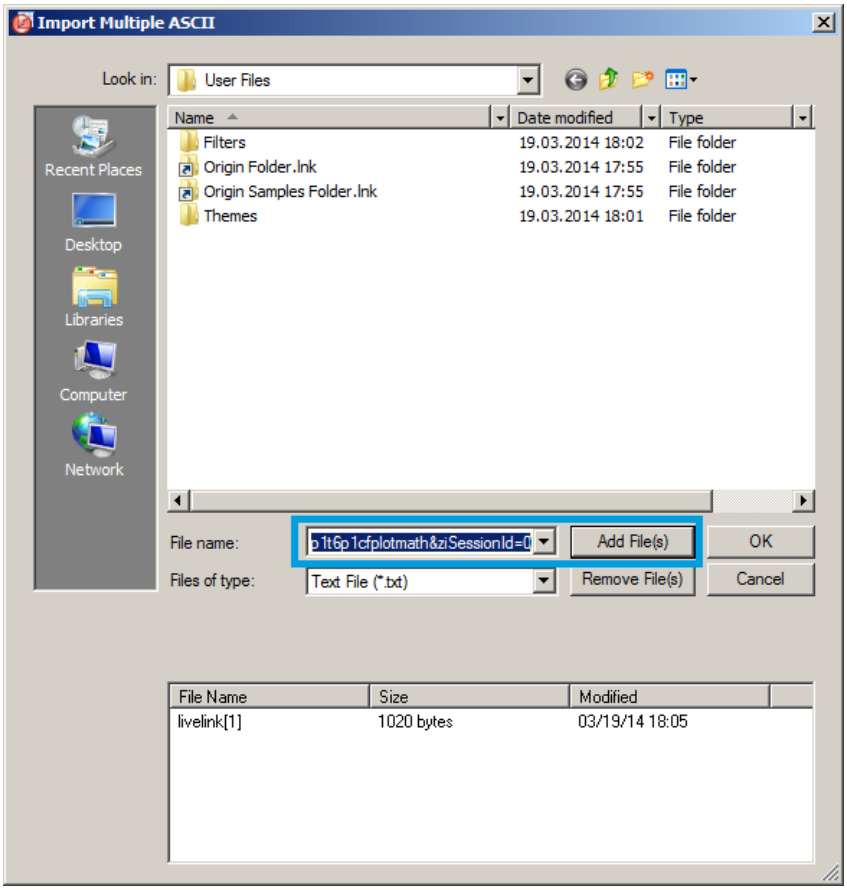
1. Open the import wizard by clicking on the icon highlighted below.



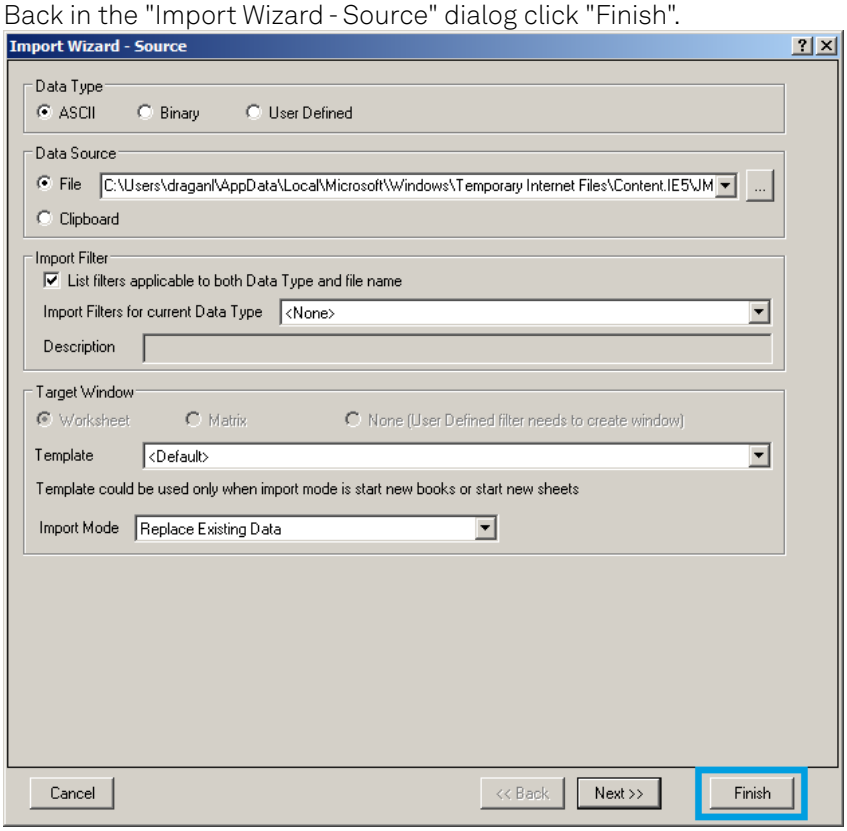
2. Ensure that the ASCII button is selected. Click the "..." button. See screenshot below. The "Import Multiple ASCII" dialog will appear.



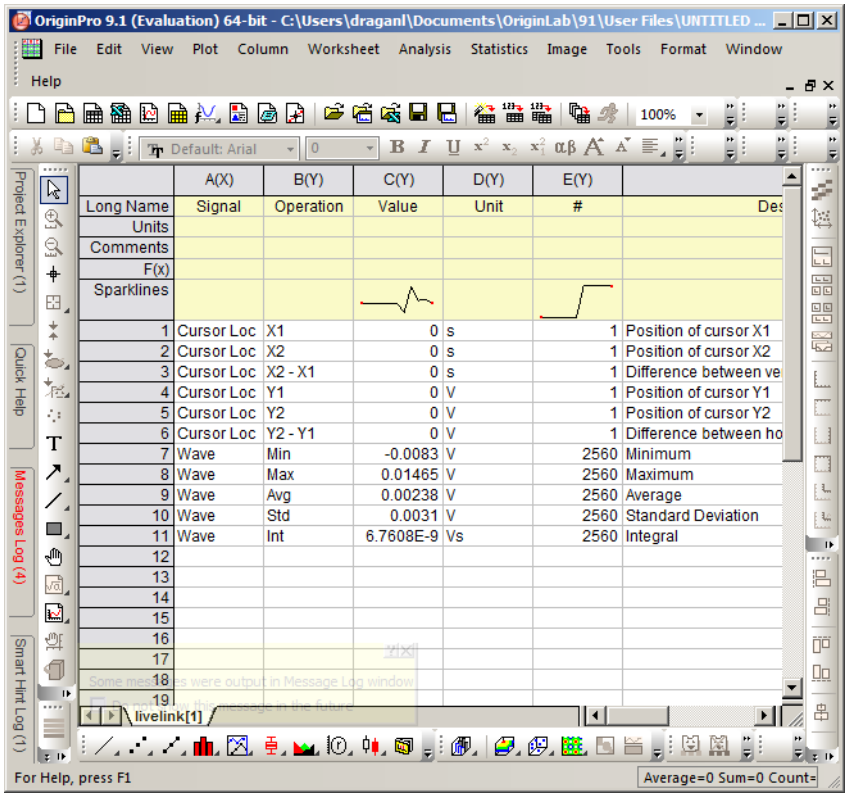
3. Paste the link text from the "LabOne Net Link" dialog into the entry field highlighted below. Then click "Add File(s)" followed by "OK".



4. Back in the "Import Wizard - Source" dialog click "Finish".



5. The data will appear in the Origin main window.



1. Among the mentioned tools, the Scope is exceptional: it displays the most recent acquisition, and its display color is fixed. However, the Persistence feature represents a more specialized functionality for multi-trace display. ↩

5.5. Lock-in Tab

This tab is the main lock-in amplifier control panel. Users with instruments with UHF-MF Multi-frequency option installed are kindly referred to Lock-in Tab (UHF-MF option)

5.5.1. Features

- Functional block diagram with access to main input, output and demodulator controls
- Parameter table with main input, output and demodulator controls
- Control elements for 8 configurable demodulators
- Auto ranging, scaling, arbitrary input units for both input channels
- Control for 2 oscillators
- Settings for main signal inputs and signal outputs
- Flexible choice of reference source, trigger options and data transfer rates

5.5.2. Description

The Lock-in tab is the main control center of the instrument and open after start up by default. 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
Lock-in		Quick overview and access to all the settings and properties for signal generation and demodulation.

The default view of the Lock-in tab is the parameter table view. It is accessible under the side tab labeled All and provides controls for all demodulators in the instrument. Moreover, for each

individual demodulator there is a functional block diagram available. It is accessible under the side tab labeled with the corresponding demodulator number.

Parameter Table

The parameter table (see Figure 5.10) consists of 4 vertical sections: Signal Inputs, Oscillators, Demodulators and Signal Outputs. The Demodulator section is horizontally divided into two identical groups. The upper group is tied to oscillator 1 and the lower group is tied to oscillator 2. That means demodulators 1 to 4 (5 to 8) can demodulate input signals at the frequency of oscillator 1 (2) and higher multiples. Demodulators 4 and 8 can be used for external referencing. Every demodulator can be connected to any of the possible inputs and outputs. Signal Input 1 and 2 are identical in all aspects, the same holds for the Signal Outputs 1 and 2.

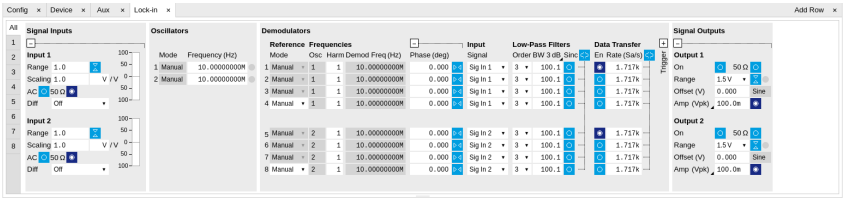


Figure 5.10: LabOne User Interface Lock-in tab - Parameter table (All)

The **Signal Inputs section** allows the user to define all relevant settings specific to the signal entered as for example input coupling, range, etc. Some of the available options like phase adjustment and the trigger functionality are collapsed by default. It takes one mouse click on the "+" icon in order to expand those controls. On the right-hand side of the Lock-in tab the Signal Outputs section allows defining signal amplitudes, offsets and range values.

The Scaling field below the Range field can be used to multiply the Signal Input data for instance to account for the gain of an external amplifier. In case there is a transimpedance gain of 10 V/A applied to the input signal externally, then the Scaling field can be set to 0.1 and the Units field can be set to A in order to show the actual current readings through the entire user interface. Below the Scaling field there is the AC/DC button and the 50 Ω/ 1 MΩ . The AC/DC button sets the coupling type: AC coupling has a high-pass cutoff frequency that can be used to block large DC signal components to prevent input signal saturation during amplification. The 50 Ω/ 1 MΩ button toggles the input impedance

between low (50 Ω) and high (approx. 1 MΩ) input impedance. 50 Ω input impedance should be selected for signal frequencies above 10 MHz to avoid artifacts generated by multiple signal reflections within the cable. With 50 Ω input impedance, one will expect a reduction of a factor of 2 in the measured signal if the signal source also has an output impedance of 50 Ω.

The **Oscillator section** indicates the frequencies of both internal oscillators . Where the Mode indicator shows Manual, the user can define the oscillator frequency manually defined by typing a frequency value in the field. In case the oscillator is referenced to an external source, the Mode indicator will show ExtRef and the frequency field is set to read-only. External reference requires a PLL to do the frequency mapping onto an internal oscillator. Successful locking is indicated by a green light right next to the frequency field. When the Modulation unit or the PID controller determine the frequency value of an oscillator, MOD or PID are indicated in the Mode field and the user cannot change the frequency manually.

In the following, we discuss the **Demodulators settings** in more detail. The block diagram displayed in Figure 5.11 indicates the main demodulator components and their interconnection. The understanding of the wiring is essential for successfully operating the instrument.

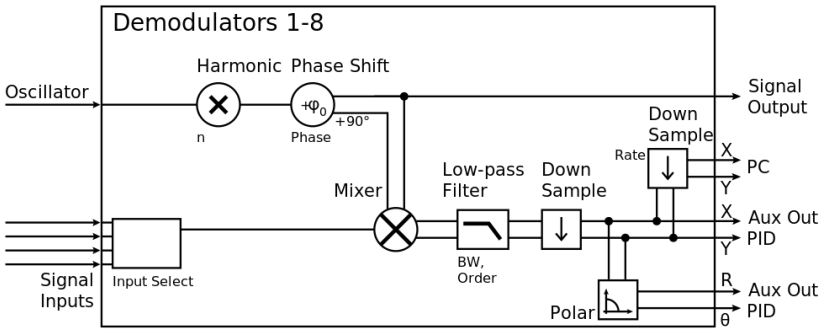


Figure 5.11: Demodulator block diagram without UHF-MF Multi-frequency option.

Every line in the Demodulators section represents one demodulator. The Mode column is read-only for all demodulators except 4 and 8, which can be to set to either internal reference (Demod) or external reference mode (ExtRef). When internal reference mode is selected, it is possible to demodulate the input signal with 4 demodulators simultaneously, using different filter settings or at different harmonic frequencies of the reference frequency. For external reference mode, one demodulator is used for the reference recovery and a few settings are greyed-out, and therefore 3 demodulators remain for simultaneous measurements. In the Input Signal column one defines the signal that is taken as input for a given demodulator. A wide choice of signals can be selected: Signal Inputs, the Trigger Inputs, the Auxiliary Inputs and Auxiliary Outputs. This allows using the instrument for many different measurement topologies. For each demodulator an additional phase shift can be introduced to the associated oscillator by entering the phase offset in the Phase column. This phase is added both to the reference channel and to the output of the demodulator. Hence, when the frequency is generated and detected using the same demodulator, signal phase and reference phase change by the same amount and no change will be visible in the demodulation result. Demodulation of frequencies that are integer multiples of any of the oscillator frequencies is achieved by entering the desired factor in the Harm column. The result of the demodulation, i.e. the amplitude and phase can be read e.g. using the Numeric tab which is described in Numeric Tab.

In the middle of the Lock-in tab is the Low-Pass Filters section where the filter order can be selected in the drop-down list for each demodulator and the filter bandwidth (BW 3dB) can be chosen by typing a numerical value. Alternatively, the time constant of the filter (TC) or the noise equivalent power filter bandwidth (BW NEP) can be chosen by clicking on the column's header. For example, setting the filter order to 4 corresponds to a roll off of 24 dB/oct or 80 dB/dec i.e. an attenuation of 10^4 for a tenfold frequency increase. If the Low-Pass Filter bandwidth is comparable to or larger than the demodulation frequency, the demodulator output may contain frequency components at the frequency of demodulation and its higher harmonics. In this case, the additional Sinc Filter should be enabled. It attenuates those unwanted harmonic components in the demodulator output. The Sinc Filter is useful when measuring at low frequencies, since it allows one to apply a Low-Pass Filter bandwidth closer to the demodulation frequency, thus speeding up the measurement time.

The data transfer of demodulator outputs is activated by the En button in the Data Transfer section where also the sampling rate (Rate) for each demodulator can be defined.

The Trigger section next to the Data Transfer allows for setting trigger conditions in order to control and initiate data transfer from the Instrument to the host PC by the application of logic signals (e.g. TTL) to either Trigger Input 3 or 4 on the instrument back panel.

In the **Signal Outputs section** the On buttons are used to activate the Signal Outputs. This is also the place where the output amplitudes for each of the Signal Outputs can be set in adjustable units (Vpk, Vrms, or dBm). The Range drop-down list is used to select the proper output range setting. On each Signal Output a digital offset voltage (Offset) can be defined. The maximum output signal permitted is ± 1.5 V.

Block Diagram

The block diagram view of the main instrument functions is also sometimes called the "Graphical Lock-in Tab". A set of indexed side tabs in the Lock-in Tab give access to a block diagram for each demodulator. The block diagrams are fully functional and provide the user with a visual feedback of what is going on inside the instrument. All control elements that are available in the Parameter Table detailed in the previous section are also present in the graphical representation.

The block diagram in Figure 5.12 shows the signal path through the instrument for the case when the internal oscillator is used as reference. The Signal Inputs and Reference/Internal Frequency are shown on the left-hand side. The actual demodulation, i.e. the mixing and low-pass filtering is represented in the center of the tab. On the bottom right the user can set Signal Output parameters. On the top right there are the settings related to the output of the measurement data, either by digital means (PC Data Transfer) or by analog means (Auxiliary Outputs 1 to 4).

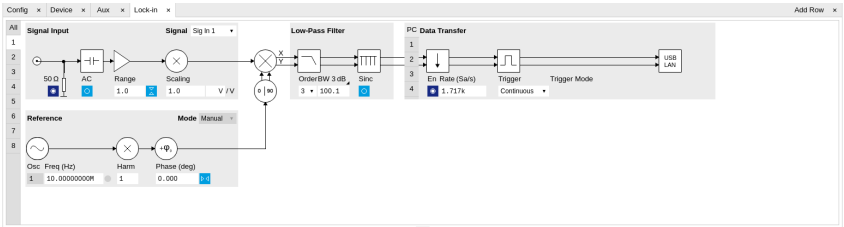


Figure 5.12: LabOne User Interface Lock-in tab - Graphical Lock-in tab in Internal Reference mode

The block diagram in Figure 5.13 shows the signal path through the instrument for the case when an external reference is used. This setting is only available for demodulators 4 and 8. In order to map an external frequency to oscillator 1/2 go to the Reference section of demodulator 4/8 and change the mode to ExtRef. This demodulator will then be used as a phase detector within the phase locked loop. The software will choose the appropriate filter settings according to the frequency and properties of the reference signal. Once demodulator 4/8 is used to map an external frequency on to one of the internal oscillators, it is no longer available for other measurements.

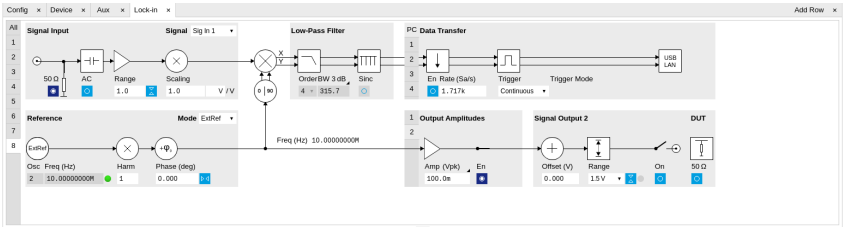


Figure 5.13: LabOne User Interface Lock-in tab - Graphical Lock-in tab in External Reference mode

5.5.3. Functional Elements

Table 5.9: Lock-in tab

Control/Tool	Option/ Range	Description
Range	10 mV to 1.5 V	Defines the gain of the analog input amplifier. The range should exceed the incoming signal by roughly a factor two including a potential DC offset. The instrument selects the next higher available range relative to a value inserted by the user. A suitable choice of this setting optimizes the accuracy and signal-to-noise ratio by ensuring that the full dynamic range of the input ADC is used.
On	ON / OFF	Enable Signal Input.
Auto		Automatic adjustment of the Range to about two times the maximum signal input amplitude measured over about 100 ms.
Scaling	numeric value	Applies an arbitrary scale factor to the input signal.
Measurement Unit	unit acronym	Defines the physical unit of the input signal. Use *, / and ^ operators, e.g., m or m/s^2. The value in this field modifies the readout of all measurement tools in the user interface. Typical uses of this field is to make measurements in the unit before the sensor/transducer, e.g. to take an transimpedance amplifier into account and to directly read results in Ampere instead of Volts.
Coupling	OFF: DC coupling	Defines the input coupling for the Signal Inputs. AC coupling inserts a high-pass filter.
	ON: AC coupling	
50 Ω	OFF: 1 MΩ	Switches between 50 Ω (ON) and 1 MΩ (OFF).
	ON: 50 Ω	
Diff	Input 2 - Input 1	Switch input mode between normal (OFF), inverted, and differential. The differential modes are implemented digitally and are not suited for analog common-mode rejection.
	Off	
	Inverted	
	Input 1 - Input 2	
Mode		Indicates how the frequency of the corresponding oscillator is controlled (manual, external reference, PLL, PID). Read only flag.
	Manual	The user setting defines the oscillator frequency.

Control/Tool	Option/ Range	Description
	ExtRef	An external reference is mapped onto the oscillator frequency.
	PLL	The UHF-PID option controls the oscillator frequency.
	PID	The UHF-PID option controls the oscillator frequency.
Frequency (Hz)	0 to 600 MHz	Frequency control for each oscillator.
Locked	ON / OFF	Oscillator locked to external reference when turned on.
Mode		Select the reference mode (manual or external reference) or indicate the unit that uses the demodulator (e.g. PLL).
	Manual	Default lock-in operating mode with manually set reference frequency.
	Mod	The demodulator is used by the UHF-MOD option, e.g. for the direct demodulation of carrier and sideband signals.
	PLL	The demodulator is used in PLL mode for frequency tracking of the signal. This function requires the UHF-PID option.
	ExtRef	The demodulator is used for external reference mode and tracks the frequency of the selected reference input. The demodulator bandwidth is set automatically to adapt to the signal properties.
	ExtRef Low BW	The demodulator is used for external reference mode and tracks the frequency of the selected reference input. The demodulator bandwidth is fixed at low value. Use when automatic bandwidth adjustment interferes with a stable lock at a fixed frequency.
	ExtRef High BW	The demodulator is used for external reference mode and tracks the frequency of the selected reference input. The demodulator bandwidth is fixed at a high value. Use when automatic bandwidth adjustment interferes with tracking a strongly fluctuating frequency.
Osc	oscillator index	Connects the selected oscillator with the demodulator corresponding to this line. Number of available oscillators depends on the installed options.
Harm	1 to 1023	Multiplies the demodulator's reference frequency with the integer factor defined by this field. If the demodulator is used as a phase detector in external reference mode (PLL), the effect is that the internal oscillator locks to the external frequency divided by the integer factor.
Demod Freq (Hz)	0 to 600 MHz	Indicates the frequency used for demodulation and for output generation. The demodulation frequency is calculated with oscillator frequency times the harmonic factor. When the UHF-MOD option is used linear combinations of oscillator frequencies including the harmonic factors define the demodulation frequencies.
Phase (deg)	-180° to 180°	Phase shift applied to the reference input of the demodulator.
Zero	⋈	Adjust the phase of the demodulator reference automatically in order to read zero degrees at the demodulator output. This action maximizes the X output, zeros the Y output, zeros the Θ output, and leaves the R output unchanged.
Signal		Selects the signal source to be associated to the demodulator.
	Sig In 1	Signal Input 1 is connected to the corresponding demodulator.
	Sig In 2	Signal Input 2 is connected to the corresponding demodulator.
	Trigger 1	Trigger 1 is connected to the corresponding demodulator.
	Trigger 2	Trigger 2 is connected to the corresponding demodulator.
	Aux Out 1	Internal value of Auxiliary Output 1 is applied to the input of the corresponding demodulator.

Control/Tool	Option/ Range	Description
	Aux Out 2	Internal value of Auxiliary Output 2 is applied to the input of the corresponding demodulator.
	Aux Out 3	Internal value of Auxiliary Output 3 is applied to the input of the corresponding demodulator.
	Aux Out 4	Internal value of Auxiliary Output 4 is applied to the input of the corresponding demodulator.
	Aux In 1	Auxiliary Input 1 is connected to the corresponding demodulator.
	Aux In 2	Auxiliary Input 2 is connected to the corresponding demodulator.
	Oscillator Phase Demod 4	Oscillator Phase of Demod 4 is connected to the corresponding demodulator. This selection combined with the demodulator's ExtRef Mode can be used to phase-lock two internal oscillators.
	Oscillator Phase Demod 8	Oscillator Phase of Demod 8 is connected to the corresponding demodulator. This selection combined with the demodulator's ExtRef Mode can be used to phase-lock two internal oscillators.
Order		Selects the filter roll off between 6 dB/oct and 48 dB/oct.
	1	1st order filter 6 dB/oct
	2	2nd order filter 12 dB/oct
	3	3rd order filter 18 dB/oct
	4	4th order filter 24 dB/oct
	5	5th order filter 30 dB/oct
	6	6th order filter 36 dB/oct
	7	7th order filter 42 dB/oct
	8	8th order filter 48 dB/oct
TC/BW Select		Defines the display unit of the low-pass filters: time constant (TC) in seconds, noise equivalent power bandwidth (BW NEP) in Hz, 3 dB bandwidth (BW 3 dB) in Hz.
	TC	Defines the low-pass filter characteristic using time constant (s) of the filter.
	BW NEP	Defines the low-pass filter characteristic using the noise equivalent power bandwidth (Hz) of the filter.
	BW 3 dB	Defines the low-pass filter characteristic using the 3 dB cut-off frequency (Hz) of the filter.
TC/BW Value	numeric value	Defines the low-pass filter characteristic in the unit defined above.
Sinc	ON / OFF	Enables the sinc filter. When the filter bandwidth is comparable to or larger than the demodulation frequency, the demodulator output may contain frequency components at the frequency of demodulation and its higher harmonics. The sinc is an additional filter that attenuates these unwanted components in the demodulator output.
Filter Lock		Makes all demodulator filter settings equal (order, time constant, bandwidth). Enabling the lock copies the settings from demodulator 1 to all other demodulators. With locked filters, any modification to a filter setting is applied to all other filters, too. Releasing the lock does not change any setting.

Control/Tool	Option/ Range	Description
Enable Streaming	ON / OFF	Enables the data acquisition and streaming of demodulated samples to the host computer for the corresponding demodulator. The streaming rate is defined in the field on the right hand side. Enabling a stream activates a corresponding element in the numeric tab and allows for demodulated samples to be visualized and analyzed in any of the LabOne measurement tools. Note: increasing number of active demodulators increases load on physical connection to the host computer.
Rate (Sa/s)	0.42 Sa/s to 14 MSa/s	Defines the demodulator sampling rate, the number of samples that are sent to the host computer per second. A rate of about 7-10 higher as compared to the filter bandwidth usually provides sufficient aliasing suppression. This is also the rate of data received by LabOne Data Server and saved to the computer hard disk. This setting has no impact on the sample rate on the auxiliary outputs connectors. Note: the value inserted by the user may be approximated to the nearest value supported by the instrument.
Demodulator Sampling Rate Lock		Makes all demodulator sampling rates equal. Enabling the lock copies the settings from demodulator 1 to all other demodulators. With locked sampling rates, any modification to a sampling rate is applied to all other sampling rate fields, too. Releasing the lock does not change any setting.
Trigger		Selects the acquisition mode of demodulated samples. Continuous trigger means data are streamed to the host computer at the Rate indicated.
	Continuous	Selects continuous data acquisition mode. The demodulated samples are streamed to the host computer at the Rate indicated on the left hand side. In continuous mode the numerical and plotter tools are continuously receiving and display new values.
	Trigger In 3	Selects external triggering by means of the Trigger 3 connector. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning. Note: some UHF Instruments feature Trigger 1/2 on the back panel instead of Trigger 3/4.
	Trigger In 4	Selects external triggering by means of the Trigger 4 connector. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning. Note: some UHF Instruments feature Trigger 1/2 on the back panel instead of Trigger 3/4.
	Trigger In 3|4	Same functionality as above, but triggering is based on a logical OR function of Trigger 3 and Trigger 4. Note: some UHF Instruments feature Trigger 1/2 on the back panel instead of Trigger 3/4.
	AWG Trigger 1	Selects internal triggering by means of the AWG Trigger Output 1. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning.
	AWG Trigger 2	Selects internal triggering by means of the AWG Trigger Output 2. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning.
	AWG Trigger 3	Selects internal triggering by means of the AWG Trigger Output 3. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning.

Control/Tool	Option/ Range	Description
	AWG Trigger 4	Selects internal triggering by means of the AWG Trigger Output 4. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning.
Trigger Mode		Defines the edge or level trigger mode for the selected Trigger input. Note: this field only appears when a non-continuous trigger is selected in the Trigger field.
	Rising	Selects triggered sample acquisition mode on rising edge of the selected Trigger input.
	Falling	Selects triggered sample acquisition mode on falling edge of the selected Trigger input.
	Both	Selects triggered sample acquisition mode on both edges of the selected Trigger input.
	High	Selects continuous sample acquisition mode on high level of the selected Trigger input. In this selection, the sample rate field determines the frequency in which demodulated samples are sent to the host computer.
	Low	Selects continuous sample acquisition mode on low level of the selected Trigger input. In this selection, the sample rate field determines the frequency in which demodulated samples are sent to the host computer.
Amplitude Unit	Vpk, Vrms, dBm	Select the unit of the displayed amplitude value. The dBm value is only valid for a system with 50 Ω termination.
Amplitude Mode		Indicates the type or source of the waveform being generated. 'Sine' indicates a sinusoidal waveform from the internal oscillator. When the AWG option is used and the AWG is enabled, the text 'AWG' will be displayed.
Amplitude Enable	ON / OFF	Enables individual output signal amplitude. When the UHF-MF option is used, it is possible to generate signals being the linear combination of the available demodulator frequencies.
On	ON / OFF	Main switch for the Signal Output corresponding to the blue LED indicator on the instrument front panel.
50Ω	ON / OFF	Select the load impedance between 50Ω and HiZ. The impedance of the output is always 50Ω. For a load impedance of 50Ω the displayed voltage is half the output voltage to reflect the voltage seen at the load.
Range		Defines the maximum output voltage that is generated by the corresponding Signal Output. This includes the potential multiple Signal Amplitudes and Offsets summed up. Select the smallest range possible to optimize signal quality. This setting ensures that no levels or peaks above the setting are generated, and therefore it limits the values that can be entered as output amplitudes. Therefore selected output amplitudes are clipped to the defined range and the clipping indicator turns on. If 50 Ω target source or differential output is enabled the possible maximal output range will be half.
	150 mV	Selects output range ±150 mV.
	1.5 V	Selects output range ±1.5 V.
Auto Range		Selects the most suited output range automatically.
Output Clipping	grey/red	Indicates that the specified output amplitude(s) exceeds the range setting. Signal clipping occurs and the output signal quality is degraded. Adjustment of the range or the output amplitudes is required.

Control/Tool	Option/ Range	Description
Offset	-range to range	Defines the DC voltage that is added to the dynamic part of the output signal.
Output	-range to range	Defines the output amplitude for each demodulator frequency as rms or peak-to-peak value. A negative amplitude value is equivalent to a phase change of 180 degree. Demodulator 4 is the signal source for Signal Output 1, demodulator 8 is the source for Signal Output 2.

5.6. Lock-in Tab (UHF-MF option)

This tab is the main lock-in amplifier control panel for UHFLI Instruments with the UHF-MF Multi-frequency option installed. Users with instruments without this option installed are kindly referred to Lock-in Tab.

5.6.1. Features

- Functional block diagram with access to main input, output and demodulator controls
- Parameter table with main input, output and demodulator controls
- Controls for 8 individually configurable demodulators
- Auto ranging, scaling, arbitrary input units for both input channels
- Control for 8 oscillators
- Settings for main signal inputs and signal outputs
- Choice of reference source, trigger options and data transfer rates

5.6.2. Description

The Lock-in tab is the main control center of the instrument and open after start up by default. 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
Lock-in MF		Quick overview and access to all the settings and properties for signal generation and demodulation.

The default view of the Lock-in tab is the parameter table view. It is accessible under the side tab labeled All and provides controls for all demodulators in the instrument. Moreover, for each individual demodulator there is a functional block diagram available. It is accessible under the side tab labeled with the corresponding demodulator number.

Parameter Table

The parameter table (see Figure 5.14) consists of 5 vertical sections: Signal Inputs, Oscillators, Demodulators, Output Amplitudes and Signal Outputs. The Demodulator section contains 8 rows each of them providing access to the settings of one dual phase demodulator. Demodulators 4 and 8 can be used for external referencing. Every demodulator can be connected to any of the possible inputs, outputs and oscillators. Signal Input 1 and 2 are identical in all aspects, the same holds for the Signal Outputs 1 and 2.

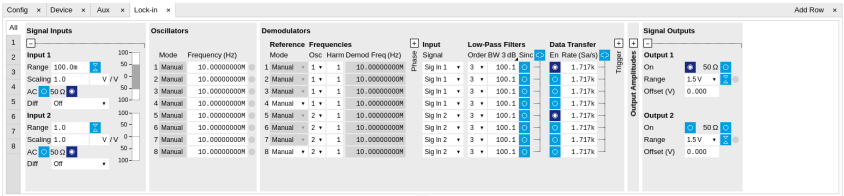


Figure 5.14: LabOne User Interface Lock-in tab with UHF-MF Multi-frequency option.

The **Signal Inputs section** allows the user to define all relevant settings specific to the signal entered as for example input coupling, range, etc. Some of the available options like phase adjustment and the trigger functionality are collapsed by default. It takes one mouse click on the "+" icon in order to expand those controls. On the right-hand side of the Lock-in tab the Signal Outputs section allows to define signal amplitudes, offsets and range values.

The Scaling field below the Range field can be used to multiply the Signal Input data for instance to account for the gain of an external amplifier. In case there is a transimpedance gain of 10 V/A applied to the input signal externally, then the Scaling field can be set to 0.1 and the Units field can be set to A in order to show the actual current readings through the entire user interface.

There are two buttons below the Scaling field that can be toggled: the AC/DC button and the 50 Ω/ 1 MΩ . The AC/DC button sets the coupling type: AC coupling has a high-pass cutoff frequency that can be used to block large DC signal components to prevent input signal saturation during amplification. The 50 Ω/ 1 MΩ button toggles the input impedance between low (50 Ω) and high (approx. 1 MΩ) input impedance. 50 Ω input impedance should be selected for signal frequencies above 10 MHz to avoid artifacts generated by multiple signal reflections within the cable. With 50 Ω input impedance, one will expect a reduction of a factor of 2 in the measured signal if the signal source also has an output impedance of 50 Ω.

The **Oscillator section** indicates the frequencies of all 8 internal oscillators . Where the Mode indicator shows Manual the user can define the oscillator frequency manually defined by typing a frequency value in the field. In case the oscillator is referenced to an external source the Mode indicator will show ExtRef and the frequency field is set to read-only. External reference requires a PLL to do the frequency mapping onto an internal oscillator. Successful locking is indicated by a green light right next to the frequency field. When the MOD option or the PID determine the frequency value of an oscillator, MOD and PID are indicated in the Mode field and the user cannot change the frequency manually.

The next section contains the **Demodulators settings**. The block diagram displayed in Figure 5.15 indicates the main demodulator components and their interconnection. The understanding of the wiring is essential for successfully operating the instrument.

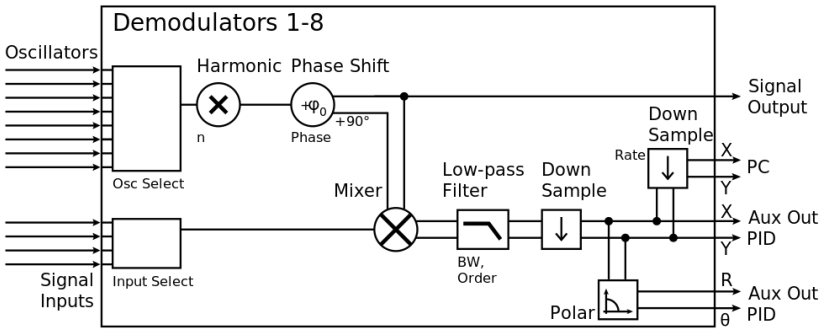


Figure 5.15: Demodulator block diagram with UHF-MF Multi-frequency option.

Every line in the Demodulators section represents one demodulator. The Mode column is read-only for all demodulators except 4 and 8, which can be to set to either internal reference (Demod) or external reference mode (ExtRef). When internal reference mode is selected, it is possible to demodulate the input signal with 8 demodulators simultaneously at 8 independent frequencies and using different filter settings. For external reference mode, one demodulator is used for the reference recovery and a few settings are greyed-out, and therefore 7 demodulators remain for simultaneous measurements. In the Input Signal column one defines the signal that is taken as input for the demodulator. A wide choice of signals can be selected: Signal Inputs, the Trigger Inputs, the Auxiliary Inputs and Auxiliary Outputs. This allows to use the instrument for many different measurement topologies.

For each demodulator an additional phase shift can be introduced to the associated oscillator by entering the phase offset in the Phase column. This phase is added both, to the reference channel and the output of the demodulator. Hence, when the frequency is generated and detected using the same demodulator, signal phase and reference phase change by the same amount and no change will be visible in the demodulation result. Demodulation of frequencies that are integer multiples of any of the oscillator frequencies is achieved by entering the desired factor in the Harm column. The demodulator readout can be obtained using the Numeric tab which is described in Numeric Tab.

In the middle of the Lock-in tab is the Low-Pass Filters section where the filter order can be selected in the drop down list for each demodulator and the filter bandwidth (BW 3dB) can chosen by typing a numerical value. Alternatively the time constant of the filter (TC) or the noise equivalent power filter bandwidth (BW NEP) can be chosen by clicking on the column's header. For example, setting the

filter order to 4 corresponds to a roll off of 24 dB/oct or 80 dB/dec i.e. an attenuation of 10^4 for a tenfold frequency increase. If the Low-Pass Filter bandwidth is comparable to or larger than the demodulation frequency, the demodulator output may contain frequency components at the frequency of demodulation and its higher harmonics. In this case, the additional Sinc Filter can be enabled. It attenuates those unwanted harmonic components in the demodulator output. The Sinc Filter is also useful when measuring at low frequencies, since it allows to apply a Low-Pass Filter bandwidth closer to the demodulation frequency, thus speeding up the measurement time.

The data transfer of demodulator outputs is activated by the En button in the Data Transfer section where also the sampling rate (Rate) for each demodulator can be defined.

The Trigger section next to the Data Transfer allows for setting trigger conditions in order to control and initiate data transfer from the Instrument to the host PC by the application of logic signals (e.g. TTL) to either Trigger Input 3 or 4 on the back panel.

The **Output Amplitudes section** is only available for Instruments with the UHF-MF option installed and allows for the flexible adjustment of output amplitudes of different demodulators and their summation on either Signal Output 1 or Signal Output 2. In order to avoid signal clipping the sum of amplitudes of each signal output needs to be smaller than the range defined in the Signal Outputs section on the right. By clicking the headline of each column one can switch between amplitude definitions in terms of root mean square values, peak-to-peak values or even units of dBm, when the 50 Ω option in the Signal Output section is activated. In the **Signal Outputs section** the On buttons allow to activate each of the Signal Outputs of the front panel. The Range drop down list is used to select the proper output range setting. On each Signal Output a digital offset voltage (Offset) can be defined. The maximum output signal permitted is ± 1.5 V.

Block Diagram

The block diagram view of the main instrument functions is also sometimes called the "Graphical Lock-in Tab". Depending on how many demodulators are available in the instrument a set of numbered side tabs are available giving access to a Graphical Lock-in Tab for each demodulator. The block diagrams are fully functional and provide the user with a visual feedback of what is going on inside the instrument. All control elements that are available in the Parameter Table detailed in the previous section are also present in the graphical representation.

The block diagram in Figure 5.16 describes the signal path throughout the instrument for the case when the internal oscillator is used as reference. The Signal Inputs and Reference/Internal Frequency are described on the left side, the core of demodulation with the mixer and low-pass filter is located in the center of the tab and the Signal Outputs, the Auxiliary Outputs as well as the data transfer to the PC is sketched on the right.

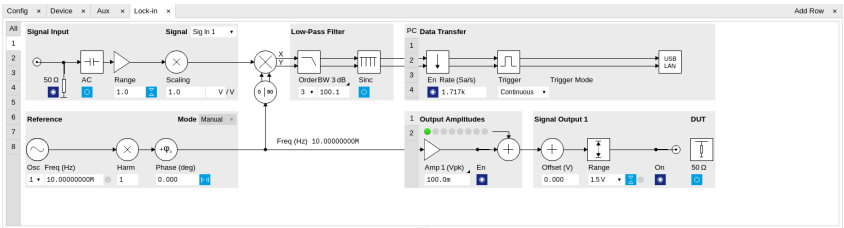


Figure 5.16: LabOne User Interface Lock-in tab - Graphical Lock-in tab in Internal Reference mode

The block diagram in Figure 5.17 describes the signal path throughout the instrument for the case when an external reference is used. This setting is only available for demodulator 4/8. In order to map an external frequency to any of the oscillators, go to the Reference section of demodulator 4/8 and change the mode to ExtRef. This demodulator will then be used as a phase detector within the phase-locked loop. The software will choose the appropriate filter settings according to the frequency and properties of the reference signal. Once a demodulator is used to map an external frequency on to one of the internal oscillators, it is no longer available for other measurements.

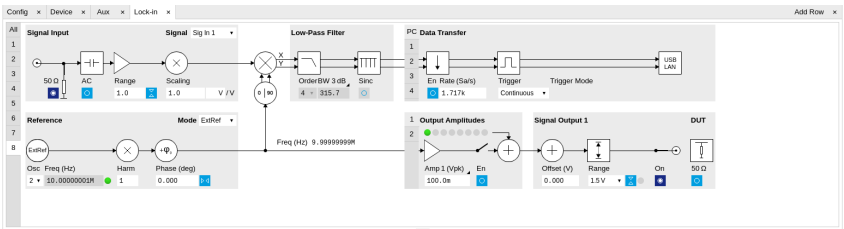


Figure 5.17: LabOne User Interface Lock-in tab - Graphical Lock-in tab in External Reference mode

5.6.3. Functional Elements

Table 5.11: Lock-in MF tab

Control/Tool	Option/ Range	Description
Range	10 mV to 1.5 V	Defines the gain of the analog input amplifier. The range should exceed the incoming signal by roughly a factor two including a potential DC offset. The instrument selects the next higher available range relative to a value inserted by the user. A suitable choice of this setting optimizes the accuracy and signal-to-noise ratio by ensuring that the full dynamic range of the input ADC is used.
On	ON / OFF	Enable Signal Input.
Auto		Automatic adjustment of the Range to about two times the maximum signal input amplitude measured over about 100 ms.
Scaling	numeric value	Applies an arbitrary scale factor to the input signal.
Measurement Unit	unit acronym	Defines the physical unit of the input signal. Use *, / and ^ operators, e.g., m or m/s^2. The value in this field modifies the readout of all measurement tools in the user interface. Typical uses of this field is to make measurements in the unit before the sensor/transducer, e.g. to take an transimpedance amplifier into account and to directly read results in Ampere instead of Volts.
Coupling	OFF: DC coupling	Defines the input coupling for the Signal Inputs. AC coupling inserts a high-pass filter.
	ON: AC coupling	
50 Ω	OFF: 1 MΩ	Switches between 50 Ω (ON) and 1 MΩ (OFF).
	ON: 50 Ω	
Diff	Input 2 - Input 1	Switch input mode between normal (OFF), inverted, and differential. The differential modes are implemented digitally and are not suited for analog common-mode rejection.
	Off	
	Inverted	
	Input 1 - Input 2	
Mode		Indicates how the frequency of the corresponding oscillator is controlled (manual, external reference, PLL, PID). Read only flag.
	Manual	The user setting defines the oscillator frequency.
	ExtRef	An external reference is mapped onto the oscillator frequency.
	PLL	The UHF-PID option controls the oscillator frequency.
	PID	The UHF-PID option controls the oscillator frequency.
Frequency (Hz)	0 to 600 MHz	Frequency control for each oscillator.
Locked	ON / OFF	Oscillator locked to external reference when turned on.
Mode		Select the reference mode (manual or external reference) or indicate the unit that uses the demodulator (e.g. PLL).
	Manual	Default lock-in operating mode with manually set reference frequency.

Control/Tool	Option/ Range	Description
	Mod	The demodulator is used by the UHF-MOD option, e.g. for the direct demodulation of carrier and sideband signals.
	PLL	The demodulator is used in PLL mode for frequency tracking of the signal. This function requires the UHF-PID option.
	ExtRef	The demodulator is used for external reference mode and tracks the frequency of the selected reference input. The demodulator bandwidth is set automatically to adapt to the signal properties.
	ExtRef Low BW	The demodulator is used for external reference mode and tracks the frequency of the selected reference input. The demodulator bandwidth is fixed at low value. Use when automatic bandwidth adjustment interferes with a stable lock at a fixed frequency.
	ExtRef High BW	The demodulator is used for external reference mode and tracks the frequency of the selected reference input. The demodulator bandwidth is fixed at a high value. Use when automatic bandwidth adjustment interferes with tracking a strongly fluctuating frequency.
Osc	oscillator index	Connects the selected oscillator with the demodulator corresponding to this line. Number of available oscillators depends on the installed options.
Harm	1 to 1023	Multiplies the demodulator's reference frequency with the integer factor defined by this field. If the demodulator is used as a phase detector in external reference mode (PLL), the effect is that the internal oscillator locks to the external frequency divided by the integer factor.
Demod Freq (Hz)	0 to 600 MHz	Indicates the frequency used for demodulation and for output generation. The demodulation frequency is calculated with oscillator frequency times the harmonic factor. When the UHF-MOD option is used linear combinations of oscillator frequencies including the harmonic factors define the demodulation frequencies.
Phase (deg)	-180° to 180°	Phase shift applied to the reference input of the demodulator.
Zero	⋈	Adjust the phase of the demodulator reference automatically in order to read zero degrees at the demodulator output. This action maximizes the X output, zeros the Y output, zeros the Θ output, and leaves the R output unchanged.
Signal		Selects the signal source to be associated to the demodulator.
	Sig In 1	Signal Input 1 is connected to the corresponding demodulator.
	Sig In 2	Signal Input 2 is connected to the corresponding demodulator.
	Trigger 1	Trigger 1 is connected to the corresponding demodulator.
	Trigger 2	Trigger 2 is connected to the corresponding demodulator.
	Aux Out 1	Internal value of Auxiliary Output 1 is applied to the input of the corresponding demodulator.
	Aux Out 2	Internal value of Auxiliary Output 2 is applied to the input of the corresponding demodulator.
	Aux Out 3	Internal value of Auxiliary Output 3 is applied to the input of the corresponding demodulator.
	Aux Out 4	Internal value of Auxiliary Output 4 is applied to the input of the corresponding demodulator.
	Aux In 1	Auxiliary Input 1 is connected to the corresponding demodulator.
	Aux In 2	Auxiliary Input 2 is connected to the corresponding demodulator.
	Oscillator Phase Demod 4	Oscillator Phase of Demod 4 is connected to the corresponding demodulator. This selection combined with the demodulator's ExtRef Mode can be used to phase-lock two internal oscillators.

Control/Tool	Option/ Range	Description
	Oscillator Phase Demod 8	Oscillator Phase of Demod 8 is connected to the corresponding demodulator. This selection combined with the demodulator's ExtRef Mode can be used to phase-lock two internal oscillators.
Order		Selects the filter roll off between 6 dB/oct and 48 dB/oct.
	1	1st order filter 6 dB/oct
	2	2nd order filter 12 dB/oct
	3	3rd order filter 18 dB/oct
	4	4th order filter 24 dB/oct
	5	5th order filter 30 dB/oct
	6	6th order filter 36 dB/oct
	7	7th order filter 42 dB/oct
	8	8th order filter 48 dB/oct
TC/BW Select		Defines the display unit of the low-pass filters: time constant (TC) in seconds, noise equivalent power bandwidth (BW NEP) in Hz, 3 dB bandwidth (BW 3 dB) in Hz.
	TC	Defines the low-pass filter characteristic using time constant (s) of the filter.
	BW NEP	Defines the low-pass filter characteristic using the noise equivalent power bandwidth (Hz) of the filter.
	BW 3 dB	Defines the low-pass filter characteristic using the 3 dB cut-off frequency (Hz) of the filter.
TC/BW Value	numeric value	Defines the low-pass filter characteristic in the unit defined above.
Sinc	ON / OFF	Enables the sinc filter. When the filter bandwidth is comparable to or larger than the demodulation frequency, the demodulator output may contain frequency components at the frequency of demodulation and its higher harmonics. The sinc is an additional filter that attenuates these unwanted components in the demodulator output.
Filter Lock		Makes all demodulator filter settings equal (order, time constant, bandwidth). Enabling the lock copies the settings from demodulator 1 to all other demodulators. With locked filters, any modification to a filter setting is applied to all other filters, too. Releasing the lock does not change any setting.
Enable Streaming	ON / OFF	Enables the data acquisition and streaming of demodulated samples to the host computer for the corresponding demodulator. The streaming rate is defined in the field on the right hand side. Enabling a stream activates a corresponding element in the numeric tab and allows for demodulated samples to be visualized and analyzed in any of the LabOne measurement tools. Note: increasing number of active demodulators increases load on physical connection to the host computer.
Rate (Sa/s)	0.42 Sa/s to 14 MSa/s	Defines the demodulator sampling rate, the number of samples that are sent to the host computer per second. A rate of about 7-10 higher as compared to the filter bandwidth usually provides sufficient aliasing suppression. This is also the rate of data received by LabOne Data Server and saved to the computer hard disk. This setting has no impact on the sample rate on the auxiliary outputs connectors. Note: the value inserted by the user may be approximated to the nearest value supported by the instrument.

Control/Tool	Option/ Range	Description
Demodulator Sampling Rate Lock		Makes all demodulator sampling rates equal. Enabling the lock copies the settings from demodulator 1 to all other demodulators. With locked sampling rates, any modification to a sampling rate is applied to all other sampling rate fields, too. Releasing the lock does not change any setting.
Trigger		Selects the acquisition mode of demodulated samples. Continuous trigger means data are streamed to the host computer at the Rate indicated.
	Continuous	Selects continuous data acquisition mode. The demodulated samples are streamed to the host computer at the Rate indicated on the left hand side. In continuous mode the numerical and plotter tools are continuously receiving and display new values.
	Trigger In 3	Selects external triggering by means of the Trigger 3 connector. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning. Note: some UHF Instruments feature Trigger 1/2 on the back panel instead of Trigger 3/4.
	Trigger In 4	Selects external triggering by means of the Trigger 4 connector. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning. Note: some UHF Instruments feature Trigger 1/2 on the back panel instead of Trigger 3/4.
	Trigger In 3|4	Same functionality as above, but triggering is based on a logical OR function of Trigger 3 and Trigger 4. Note: some UHF Instruments feature Trigger 1/2 on the back panel instead of Trigger 3/4.
	AWG Trigger 1	Selects internal triggering by means of the AWG Trigger Output 1. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning.
	AWG Trigger 2	Selects internal triggering by means of the AWG Trigger Output 2. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning.
	AWG Trigger 3	Selects internal triggering by means of the AWG Trigger Output 3. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning.
	AWG Trigger 4	Selects internal triggering by means of the AWG Trigger Output 4. Demodulated samples are sent to the host computer for each event defined in the Trig Mode field. When edge trigger is selected the rate field is greyed out and has no meaning.
Trigger Mode		Defines the edge or level trigger mode for the selected Trigger input. Note: this field only appears when a non-continuous trigger is selected in the Trigger field.
	Rising	Selects triggered sample acquisition mode on rising edge of the selected Trigger input.
	Falling	Selects triggered sample acquisition mode on falling edge of the selected Trigger input.
	Both	Selects triggered sample acquisition mode on both edges of the selected Trigger input.
	High	Selects continuous sample acquisition mode on high level of the selected Trigger input. In this selection, the sample rate field determines the frequency in which demodulated samples are sent to the host computer.

Control/Tool	Option/ Range	Description
	Low	Selects continuous sample acquisition mode on low level of the selected Trigger input. In this selection, the sample rate field determines the frequency in which demodulated samples are sent to the host computer.
Amplitude Unit	Vpk, Vrms, dBm	Select the unit of the displayed amplitude value. The dBm value is only valid for a system with 50 Ω termination.
Amplitude Mode		Indicates the type or source of the waveform being generated. 'Sine' indicates a sinusoidal waveform from the internal oscillator. When the AWG option is used and the AWG is enabled, the text 'AWG' will be displayed.
Amplitude Enable	ON / OFF	Enables individual output signal amplitude. When the UHF-MF option is used, it is possible to generate signals being the linear combination of the available demodulator frequencies.
Amplitude (V)	-range to range	Defines the output amplitude for each demodulator frequency as rms or peak-to-peak value. A negative amplitude value is equivalent to a phase change of 180 degree. Linear combination of multiple amplitude settings on the same output are clipped to the range setting. Note: the value inserted by the user may be approximated to the nearest value supported by the Instrument.
AWG	AWG is ON	Indicates that the output amplitude is generated by the AWG.
On	ON / OFF	Main switch for the Signal Output corresponding to the blue LED indicator on the instrument front panel.
50Ω	ON / OFF	Select the load impedance between 50Ω and HiZ. The impedance of the output is always 50Ω. For a load impedance of 50Ω the displayed voltage is half the output voltage to reflect the voltage seen at the load.
Range		Defines the maximum output voltage that is generated by the corresponding Signal Output. This includes the potential multiple Signal Amplitudes and Offsets summed up. Select the smallest range possible to optimize signal quality. This setting ensures that no levels or peaks above the setting are generated, and therefore it limits the values that can be entered as output amplitudes. Therefore selected output amplitudes are clipped to the defined range and the clipping indicator turns on. If 50 Ω target source or differential output is enabled the possible maximal output range will be half.
	150 mV	Selects output range ±150 mV.
	1.5 V	Selects output range ±1.5 V.
Auto Range		Selects the most suited output range automatically.
Output Clipping	grey/red	Indicates that the specified output amplitude(s) exceeds the range setting. Signal clipping occurs and the output signal quality is degraded. Adjustment of the range or the output amplitudes is required.
Offset	-range to range	Defines the DC voltage that is added to the dynamic part of the output signal.

5.7. Numeric Tab

The Numeric tab provides a powerful time domain based measurement display as introduced in Unique Set of Analysis Tools. It is available on all UHFLI instruments. On UHFAWG instruments, the tab is available if at least one of the options UHF-BOX or UHF-LIA is installed.

5.7.1. Features

- Display of demodulator output data and other streamed data, e.g. auxiliary inputs, PID errors, Boxcar data, demodulator frequencies, AU data, etc.
- Graphical and numerical range indicators
- Polar and Cartesian formats
- Support for Input Scaling and Input Units

5.7.2. Description

The Numeric tab serves as the main numeric overview display of multiple measurement data. The display can be configured by both choosing the values displayed and also rearrange the display tiles by drag-and-drop. 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 5.12: App icon and short description

Control/ Tool	Option/ Range	Description
Numeric	0.058	Access to all continuously streamed measurement data as numerical values.

The numeric tab (see Figure 5.18) is divided into a display section on the left and a configuration section on the right. The configuration section is further divided into a number of sub-tabs.

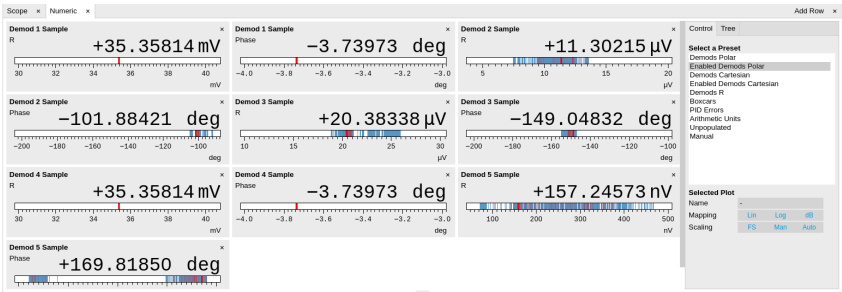


Figure 5.18: LabOne UI: Numeric tab

The numeric tab can be deployed to display the demodulated signal, phase, frequency as well as the signal levels at the auxiliary inputs. By default, the user can display the demodulated data either in polar coordinates (R, Θ) or in Cartesian coordinates (X, Y) which can be toggled using the presets. To display other measurement quantities as available from any of the presets simply click on the tree tab next to the preset tab. The desired display fields can be selected under each demodulator's directory tree structure.

5.7.3. Functional Elements

Table 5.13: Numeric tab: Presets sub-tab

Control/ Tool	Option/Range	Description
Select a Preset		Select numerical view based on a preset. Alternatively, the displayed value may also selected based on tree elements.
	Demods Polar	Shows R and Phase of all demodulators.
	Enabled Demods Polar	Shows R and Phase of enabled demodulators.
	Demods Cartesian	Shows X and Y of all demodulators.
	Enabled Demods Cartesian	Shows X and Y of enabled demodulators.
	Demods R	Shows R of all demodulators.
	Boxcars	Shows amplitude of all boxcars.
	PID Errors	Shows error of all PID.

Control/ Tool	Option/Range	Description
	Arithmetic Units	Shows output of all Cartesian and polar arithmetic units.
	Unpopulated	Shows no signals.
	Manual	If additional signals are added or removed the active preset gets manual.

For the Tree sub-tab please see the section called "Tree Selector".

Table 5.14: Numeric tab: Settings sub-tab

Control/ Tool	Option/ Range	Description
Name	text label	Name of the selected plot(s). The default name can be changed to reflect the measured signal.
Mapping		Mapping of the selected plot(s)
	Lin	Enable linear mapping.
	Log	Enable logarithmic mapping.
	dB	Enable logarithmic mapping in dB.
Scaling	Manual/Full Scale	Scaling of the selected plot(s)
Zoom To Limits		Adjust the zoom to the current limits of the displayed histogram data.
Start Value	numeric value	Start value of the selected plot(s). Only visible for manual scaling.
Stop Value	numeric value	Stop value of the selected plot(s). Only visible for manual scaling.

5.8. Plotter Tab

The Plotter is one of the powerful time-domain measurement tools as introduced in Unique Set of Analysis Tools and is available on all UHFLI instruments. On UHFAWG instruments, the tab is available if at least one of the options UHF-DIG, UHF-CNT, UHF-BOX, or UHF-LIA is installed.

5.8.1. Features

- Plotting of all streamed data, e.g. demodulator data, auxiliary inputs, auxiliary outputs, Boxcar data, PID, etc.
- Plotting of all streamed data, e.g. demodulator data, auxiliary inputs, auxiliary outputs, etc.
- Plotting of all streamed data, e.g. impedance data, demodulator data, auxiliary inputs, auxiliary outputs, etc.
- Plotting of streamed data from the pulse counters (requires HDAWG-CNT option) or streamed trigger input states.
- Plotting of Scope data, e.g. Signal Inputs (requires UHF-DIG option)
- Vertical axis grouping for flexible axis scaling
- Polar and Cartesian data format
- Histogram and Math functionality for data analysis
- 4 cursors for data analysis
- Support for Input Scaling and Input Units

5.8.2. Description

The Plotter serves as graphical display for time domain data in a roll mode, i.e. continuously without triggering. 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 5.15: App icon and short description

Control/ Tool	Option/ Range	Description
Plotter		Displays various continuously streamed measurement data as traces over time (roll mode).

The Plotter tab (see Figure 5.19) is divided into a display section on the left and a configuration section on the right.

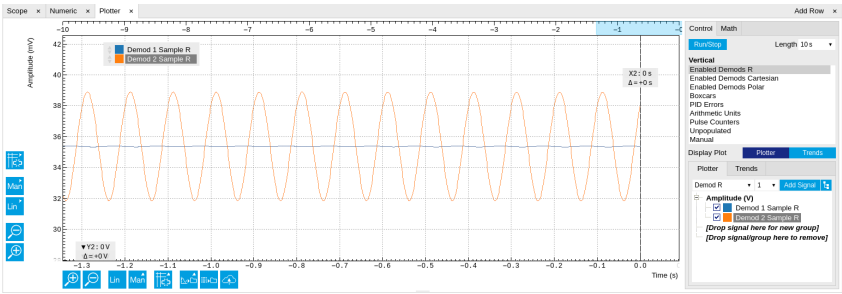


Figure 5.19: LabOne UI: Plotter tab

The Plotter can be used to monitor the evolution of demodulated data and other streamed data continuously over time. Just as in the numeric tab any continuously streamed quantity can be displayed, for instance R, Θ, X, Y, frequency, and others. New signals can be added by either using the presets in the Control sub-tab or by going through the tree and selecting the signals of interest in the tree structure. The vertical and horizontal axis can be displayed in Lin, Log or dB scale. The Plotter display can be zoomed in and out with the magnifier symbols, or through Man (Manual), Auto (Automatic) and FS (Full Scale) button settings (see also Plot Functionality). The maximum duration data is kept in the memory can be defined through the window length parameter in the Settings sub-tab. The window length also determines the file size for the Record Data functionality.

Note

Setting the window length to large values when operating at high sampling rates can lead to memory problems at the computer hosting the data server.

The sampling rate of the demodulator data is determined by the Rate value in Sa/s set in the Lock-in tab ; similarly the rates for PID and Boxcar related data are set in the associated tabs . The Plotter data can be continuously saved to disk by clicking the record button in the Config tab which will be indicated by a green Recording (REC) LED in the status bar. See Saving and Loading Data for more information on data saving.

5.8.3. Functional Elements

Table 5.16: Plotter tab: Control sub-tab

Control/ Tool	Option/Range	Description
Run/Stop	Run/Stop	Start and stop continuous data plotting (roll mode)
Select a Preset		Select a pre-defined group signals. A signal group is defined by a common unit and signal type. They should have the same scaling behavior as they share a y-axis. Split a group if the signals have different scaling properties.
	Enabled Demods R	Selects the amplitude of all enabled demodulators.
	Enabled Demods Cartesian	Selects X and Y of all enabled demodulators.
	Enabled Demods Polar	Selects amplitude and phase of all enabled demodulators.
	Boxcars	Selects the amplitude of boxcar 1 and 2.
	PID Errors	Selects the error of all PID.

Control/ Tool	Option/Range	Description
	Arithmetic Units	Selects the output of all Cartesian and polar arithmetic units.
	Pulse Counters	Selects the output of all pulse counter units.
	Unpopulated	Shows no signals.
	Manual	Selects the signals as defined in the tree sub-tab.

For the Vertical Axis Groups, please see the table "Vertical Axis Groups description" in the section called "Vertical Axis Groups".

For the Math sub-tab please see the table "Plot math description" in the section called "Cursors and Math".

5.9. Scope Tab

The Scope is a powerful time domain and frequency domain measurement tool as introduced in Unique Set of Analysis Tools and is available on all UHF Series instruments.

5.9.1. Features

- One input channel with 64 kSa of memory; upgradable to two channels with 128 MSa memory per channel (UHF-DIG option)
- 12 bit nominal resolution
- Simultaneous display of both input channels with up to 1.8 GSa/s (requires UHF-DIG option)
- Segmented recording (requires UHF-DIG option)
- Colorscale display for imaging (requires UHF-DIG option)
- Fast Fourier Transform (FFT): up to 900 MHz span, spectral density and power conversion, choice of window functions
- Sampling rates from 27 kSa/s to 1.8 GSa/s; up to 36 µs acquisition time at 1.8 GSa/s or 2.3 s at 27 kSa/s
- 8 signal sources including Signal Inputs and Trigger Inputs; up to 8 trigger sources and 2 trigger methods
- Independent hold-off, hysteresis, pre-trigger and trigger level settings `
- Support for Input Scaling and Input Units
- Continuous recording of both input channels at up to 7 MSa/s over USB and 14 MSa/s over 1GbE

5.9.2. Description

The Scope tab serves as the graphical display for time domain data. 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 5.17: App icon and short description

Control/ Tool	Option/ Range	Description
Scope		Displays shots of data samples in time and frequency domain (FFT) representation.

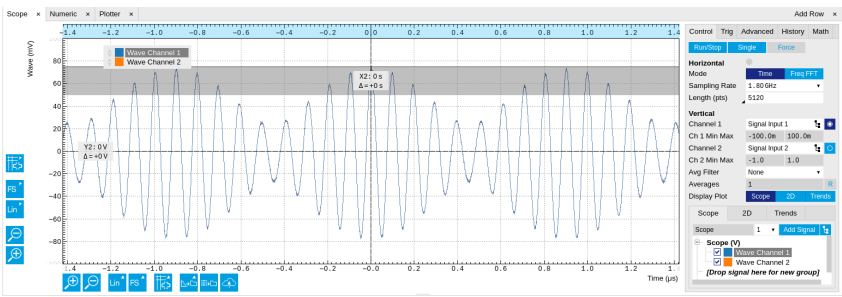


Figure 5.20: LabOne UI: Scope tab - Time domain

The Scope tab consists of a plot section on the left and a configuration section on the right. The configuration section is further divided into a number of sub-tabs. It gives access to a single-channel oscilloscope that can be used to monitor a choice of signals in the time or frequency domain. Hence the X axis of the plot area is time (for time domain display, Figure 5.20) or frequency (for frequency domain display, Figure 5.22). It is possible to display the time trace and the associated FFT simultaneously by opening a second instance of the Scope tab. The Y axis displays the selected signal that can be modified and scaled using the arbitrary input unit feature found in the Lock-in tab.

The Scope records data from a single channel at up to 1.8 GSa/s. The channel can be selected among the two Signal Inputs, Auxiliary Inputs, Trigger Inputs and Demodulator Oscillator Phase. The Scope records data sets of up to 64 kSa samples in the standard configuration, which corresponds to an acquisition time of 36 µs at the highest sampling rate. The performance of the Scope is comparable to that of entry-level GHz sampling rate oscilloscopes. The Scope may be upgraded with the UHF-DIG Digitizer option, which enables two channels to be recorded in parallel, increases the available memory to 128 MSa/channel, and allows recording and color scale display of data in a segmented fashion. The UHF-DIG Digitizer option also enables a continuous recording mode with a sampling rate of up to 28 MSa/s.

The product of the inverse sampling rate and the number of acquired points (Length) determines the total recording time for each shot. Hence, longer time intervals can be captured by reducing the sampling rate. The Scope can perform sampling rate reduction either using decimation or BW Limitation as illustrated in Figure 5.21. BW Limitation is activated by default, but it can be deactivated in the Advanced sub-tab. The figure shows an example of an input signal at the top, followed by the Scope output when the highest sample rate of 1.8 GSa/s is used. The next signal shows the Scope output when a rate reduction by a factor of 4 (i.e. 450 MSa/s) is configured and the rate reduction method of decimation is used. For decimation, a rate reduction by a factor of N is performed by only keeping every N^{th} sample and discarding the rest. The advantage of this method is its simplicity, but the disadvantage is that the signal is undersampled because the input filter bandwidth of the UHF Series instrument is fixed at 600 MHz. As a consequence, the Nyquist sampling criterion is no longer satisfied and aliasing effects may be observed. The default rate reduction mechanism of BW Limitation is illustrated by the lowermost signal in the figure. BW Limitation means that for a rate reduction by a factor of N, each sample produced by the Scope is computed as the average of N samples acquired at the maximum sampling rate. The effective signal bandwidth is thereby reduced and aliasing effects are largely suppressed. As can be seen from the figure, with a rate reduction by a factor of 4, every output sample is simply computed as the average of 4 consecutive samples acquired at 1.8 GSa/s.

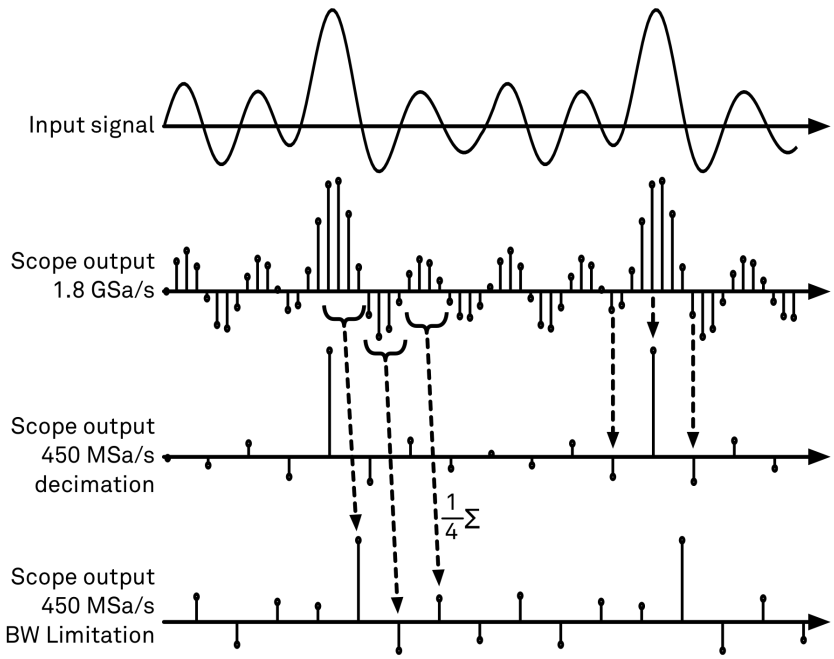


Figure 5.21: Illustration of how the Scope output is generated in BW Limitation and decimation mode when the sampling rate is reduced from the default of 1.8 GSa/s to 450 MSa/s

The frequency domain representation is activated in the Control sub-tab by selecting Freq Domain FFT as the Horizontal Mode. It allows the user to observe the spectrum of the acquired shots of samples. All controls and settings are shared between the time domain and frequency domain representations.

The Scope supports averaging over multiple shots. The functionality is implemented by means of an exponential moving average filter with configurable filter depth. Averaging helps to suppress noise

components that are uncorrelated with the main signal. It is particularly useful in combination with the Frequency Domain FFT mode where it can help to reveal harmonic signals and disturbances that might otherwise be hidden below the noise floor.

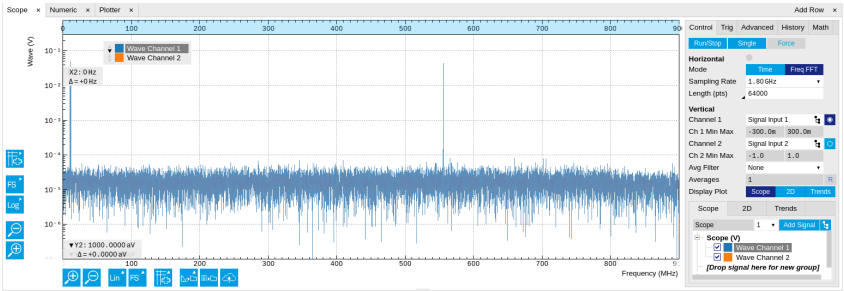


Figure 5.22: LabOne UI: Scope tab - Frequency domain

The Trigger sub-tab offers all the controls necessary for triggering on different signal sources. When the trigger is enabled, then oscilloscope shots are acquired whenever the trigger conditions are met. Trigger and Hysteresis levels can be indicated graphically in the plot. A disabled trigger is equivalent to continuous oscilloscope shot acquisition.

Digitizer upgrade option

The UHF-DIG Digitizer option greatly enhances the performance of the Scope with the addition of the following features

- Simultaneous recording of two Scope channels
- Memory depth of 128 MSa for both Scope channels
- Additional input signal sources (Boxcar, Demodulator, Arithmetic Unit and PID data)
- Segmented recording
- Imaging support with colorscale display
- XY display of two channels
- Trigger gating
- Additional trigger input sources that allow for cross-domain triggering
- Additional trigger/marker output sources based on the state of the Scope
- Continuous scope data streaming

This additional functionality can be enabled on any UHF Series instrument with an option key. Please contact Zurich Instruments to get more information. The following sections explain the Digitizer features in more detail.

Two channels and extended memory depth

The UHF-DIG option enables simultaneous dual-channel recording. This allows for very exact relative timing measurements. With the XY display, it's possible to plot two signals against each other, e.g. for quick visualization of phase offsets or characteristic curves. Each channel can be assigned a different signal source. Trigger settings, sampling rate, and recording length settings are shared between both channels. An increased shot length of up to 128 MSa compared to the standard 64 kSa allows for longer recording times and FFTs with finer frequency resolution for the same frequency span.

Additional input sources

Besides the Signal Input, Trigger Input, Auxiliary Input, and Oscillator Phase the UHF-DIG option also allows for recording of Demodulator, PID, Boxcar and Arithmetic Unit signals. This functionality is very powerful in that it allows short bursts to be recorded with very high sampling rates. In order to optimally use the vertical resolution, the upper and lower limit of these input signals should be chosen appropriately. Before sampling, a scaling and an offset are applied to the input signal in order to get 12 bit resolution between the lower and upper limit. The applied scaling and offset values are transferred together with the scope data, which allows for recovery of the original physical signal strength in absolute values. For directly sampled input signals like the Signal Inputs or Trigger Inputs, the limits are read-only values and reflect the selected input range.

Segmented data recording and imaging

The segmented data recording mode allows for a significant reduction of the hold-off time between scope shots to less than 100 µs. This is achieved by intermediate storage of a burst of up to 32768 scope shots, called segments, in the instrument memory. In this way, the Scope does not have to wait after each shot until the data transfer to the host computer is completed. In segmented data recording mode, the Scope provides a two-dimensional color scale display of the data particularly useful for imaging applications. When used over the API, the data of each shot will contain information on the segment number.

Trigger gating

With the UHF-DIG option installed the user can make full use of the Trigger Engine. If trigger gating is enabled, a trigger event will only be accepted if the gating input is active.

Additional trigger input sources

By using a Demodulator , PID, Boxcar, or Arithmetic Unit signal as trigger source, the Scope can be used in a cross-domain triggering mode. This allows, for example, for time domain signals to be recorded in a synchronous fashion triggered by the result from analyzing a signal in the frequency domain by means of a demodulator.

Note

Choose a negative delay (pre-trigger) to compensate for the delay of the Demodulator, PID, Boxcar or Arithmetic Unit.

Continuous Scope data streaming

Normal scope operation records scope shots into the instrument memory. This allows for recording of up to 1.8 GSa/s until the memory is full. After each scope shot there will be a dead time, also known as hold-off time, to re-arm the trigger, address the next memory block and transfer the data to the PC. Due to this dead time scope shots cannot be recorded back to back. In order to record very long scope shots (digitizer mode) the Scope data can be streamed directly to the host computer bypassing the instrument memory. This allows for continuous recording of very long Scope traces that exceed the available memory depth of the instrument. The streamed Scope data is available for display in the Plotter tab together with all other streaming data. Due to the limited transfer bandwidth over the TCPIP or USB interface, the maximum sampling rate is smaller than for shot operation. The sampling rate for the Scope streaming channels and the enabling of each channel is controlled in the Advanced sub-tab of the Scope. As the sampling rate of the Scope streaming can be adjusted independently from the Scope shot sampling rate it is possible to record continuous data together with triggered high sampling rate Scope shots.

Scope state output on Trigger Output

The UHF-DIG option extends the list of available Trigger Outputs by the six elements: Scope Trigger, Scope Armed, Scope Active and their logically inverse signals. The Trigger Output signals are controlled on the DIO tab (DIO Tab). Figure 5.23 shows an illustration of the signal that will be generated on the Trigger Output when one of the six new Scope-related sources is selected. An example input signal is shown at the top of the figure. It is assumed that the Scope is configured to trigger on this input signal on a rising edge crossing the level indicated by the stippled line.

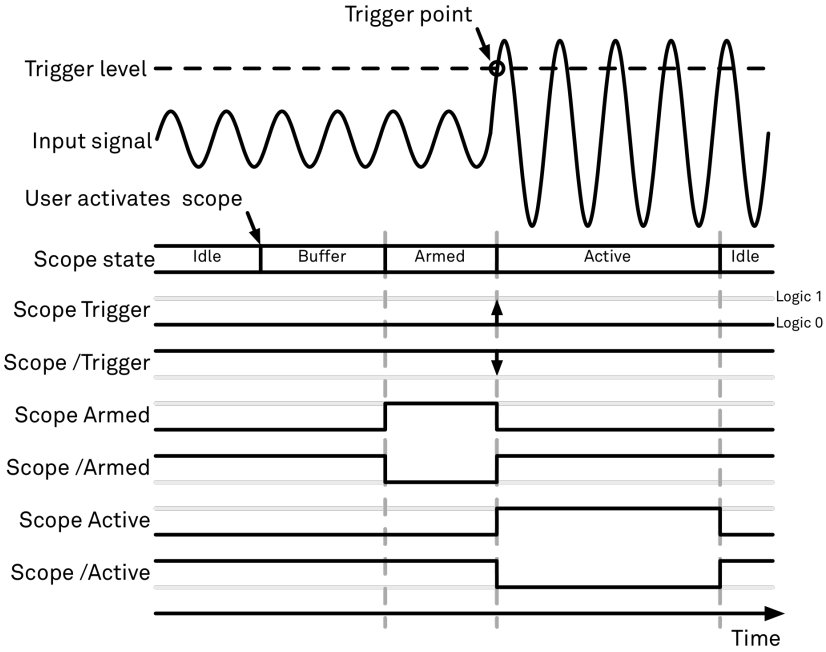


Figure 5.23: Illustration of the signal that will appear on the Trigger Output when one of the six Scope related sources is selected.

The Scope can be thought of as having a state, which changes over time. The state is shown below the input signal in the figure. When the Scope is completely inactive, it is said to be in the Idle state. When the user then activates the Scope, it will transition into a Buffer state. In this state the Scope will start to record the input signal. It will remain in this state until sufficient data has been recorded to fulfill the user requirement for recording data prior to the trigger point as controlled by the trigger Reference and Delay fields in the user interface. Once sufficient data has been recorded, the Scope will transition to the Armed state. In this state the Scope is ready to accept the trigger signal. Note that the Scope will continue to record data for as long as it is in the Armed state, and that if no trigger is defined, the Scope will simply pass straight through the Armed state. Once the input signal passes the Trigger level the Scope will trigger, and at the same time its state will change from Armed to Active. The Scope will remain in the Active state, where it also records data, until sufficient data has been recorded to fulfill the Length requirement configured in the user interface. Once enough data has been acquired, the Scope will transition back into the Idle state where it will wait for the time configured with the Hold-off time before it either starts the next measurement automatically (in case Run is active) or waits for the user to reactivate it.

The trigger source selector allows information about the Scope state to be reproduced on the Trigger Output in a number of ways. The signal that will appear on the output is shown with the six bottommost traces in the figure. Note that these traces are shown as digital signals with symbolic values of logic 0 and 1. These values will of course be actual voltages when measured on the device itself.

First, if Scope Trigger is selected then the trigger output will have a signal that is asserted, which means that it goes high, when the scope triggers, i.e. changes from the Armed to the Active state. The signal will normally have a very short duration and, therefore, it is shown with an arrow in the figure. The duration can be increased by means of the Width input field, which can be found next to the Output Signal selector on the DIO tab. If Scope/Trigger is selected, then the same signal will appear on the output, but it will simply be inverted logically.

Next, if the Scope Armed source is selected, the trigger output will be asserted as long as the Scope is in the Armed state. Again, this means that the Scope has recorded enough data to proceed with the acquisition and is waiting for the trigger condition to become satisfied. In this example, since a rising edge trigger is defined, the trigger condition becomes satisfied when the input signal goes from below the trigger level to above the trigger level.

Similarly, if Scope /Armed is selected, the trigger output will be asserted (i.e. at logic 1) whenever the Scope is in a state different from the Armed state. The same explanation holds for the remaining two configuration options, except here the trigger output is asserted when the Scope is in the Active state or when it is not in the Active state.

5.9.3. Functional Elements

Table 5.18: Scope tab: Control sub-tab

Control/ Tool	Option/ Range	Description
Run/Stop	Run/Stop	Runs the scope/FFT continuously.
Single	Single	Acquires a single shot of samples.
Force	Force	Force a trigger event.
Mode	Freq Domain (FFT)	Switches between time and frequency domain display.
	Time Domain	
Sampling Rate	27.5 kSa/s to 1.8 GSa/s	Defines the sampling rate of the scope. The numeric values are rounded for display purposes. The exact values are equal to the base sampling rate divided by 2^n, where n is an integer.
Sampling Rate		Defines the sampling rate of the scope. The numeric values are rounded for display purposes. The exact values are equal to the base sampling rate divided by 2^n, where n is an integer. Warning: Due to the lack of sample averaging feature, reduced sampling rates can cause aliasing and thus artifacts in the signal spectrum. Currently, the Scope tool only supports sample decimation, but in the future it will also offer sample averaging.
Length Mode		Switches between length and duration display.
	Length (pts)	The scope shot length is defined in number of samples. The duration is given by the number of samples divided by the sampling rate. The UHF-DIG option greatly increases the available length.
	Duration (s)	The scope shot length is defined as a duration. The number of samples is given by the duration times the sampling rate.
Length (pts) or Duration (s)	numeric value	Defines the length of the recorded scope shot. Use the Length Mode to switch between length and duration display.
Channel 1/2		Selects the signal source for the corresponding scope channel. Navigate through the tree view that appears and click on the required signal. Note: Channel 2 requires the DIG option.
Min	numeric value	Lower limit of the scope full scale range. For demodulator, PID, Boxcar, and AU signals the limit should be adjusted so that the signal covers the specified range to achieve optimal resolution.
Max	numeric value	Upper limit of the scope full scale range. For demodulator, PID, Boxcar, and AU signals the limit should be adjusted so that the signal covers the specified range to achieve optimal resolution.
Enable	ON / OFF	Activates the display of the corresponding scope channel. Note: Channel 2 requires the DIG option.
Average Filter		Enable Exponential Moving Average (EMA) filter that is applied when the average of several scope shots is computed and displayed. Depending on the mode, the source data for averaging is either the Time or the Freq FFT trace.
	Off	Averaging is turned off.
	On	Consecutive scope shots are averaged with an exponential weight.
Averages	integer value	The number of shots required to reach 63% settling. Twice the number of shots yields 86% settling.
Averages	integer value	The number of shots to average on the device before returning the data.
Reset	R	Resets the averaging filter.

For the Vertical Axis Groups, please see the table "Vertical Axis Groups description" in the section called "Vertical Axis Groups".

Table 5.19: Scope tab: Trigger sub-tab

Control/ Tool	Option/Range	Description
Trigger	grey/green/ yellow	When flashing, indicates that new scope shots are being captured and displayed in the plot area. The Trigger must not necessarily be enabled for this indicator to flash. A disabled trigger is equivalent to continuous acquisition. Scope shots with data loss are indicated by yellow. Such an invalid scope shot is not processed.
Enable	ON / OFF	When triggering is enabled scope data are acquired every time the defined trigger condition is met. If disabled, scope shots are acquired continuously.
Signal		Selects the trigger source signal. Navigate through the tree view that appears and click on the required signal.
Slope	Rising or falling edge trigger	Select the signal edge that should activate the trigger.
	None	
	Rising edge trigger	
	Falling edge trigger	
Level (V)	trigger signal range (negative values permitted)	Defines the trigger level.
Hysteresis Mode		Selects the mode to define the hysteresis strength. The relative mode will work best over the full input range as long as the analog input signal does not suffer from excessive noise.
	Hysteresis (V)	Selects absolute hysteresis.
	Hysteresis (%)	Selects a hysteresis relative to the adjusted full scale signal input range.
Hysteresis (V)	trigger signal range (positive values only)	Defines the voltage the source signal must deviate from the trigger level before the trigger is rearmed again. Set to 0 to turn it off. The sign is defined by the Edge setting.
Hysteresis (%)	numeric percentage value (positive values only)	Hysteresis relative to the adjusted full scale signal input range. A hysteresis value larger than 100% is allowed.
Show Level	ON / OFF	If enabled shows the trigger level as grey line in the plot. The hysteresis is indicated by a grey box. The trigger level can be adjusted by drag and drop of the grey line.
Trigger Gating		Select the signal source used for trigger gating if gating is enabled. This feature requires the UHF-DIG option.
	Trigger In 3 High	Only trigger if the Trigger Input 3 is at high level.
	Trigger In 3 Low	Only trigger if the Trigger Input 3 is at low level.
	Trigger In 4 High	Only trigger if the Trigger Input 4 is at high level.
	Trigger In 4 Low	Only trigger if the Trigger Input 4 is at low level.
Trigger Gating Enable	ON / OFF	If enabled the trigger will be gated by the trigger gating input signal. This feature requires the UHF-DIG option.
Holdoff Mode		Selects the holdoff mode.
	Holdoff (s)	Holdoff is defined as time.
	Holdoff (events)	Holdoff is defined as number of events.
Holdoff (s)	numeric value	Defines the time before the trigger is rearmed after a recording event.

Control/ Tool	Option/Range	Description
Holdoff (events)	1 to 1048575	Defines the trigger event number that will trigger the next recording after a recording event. A value one will start a recording for each trigger event.
Reference (%)	percent value	Trigger reference position relative to the plot window. Default is 50% which results in a reference point in the middle of the acquired data.
Delay (s)	numeric value	Trigger position relative to reference. A positive delay results in less data being acquired before the trigger point, a negative delay results in more data being acquired before the trigger point.
Enable	ON / OFF	Enable segmented scope recording. This allows for full bandwidth recording of scope shots with a minimum dead time between individual shots. This functionality requires the DIG option.
Segments	1 to 32768	Specifies the number of segments to be recorded in device memory. The maximum scope shot size is given by the available memory divided by the number of segments. This functionality requires the DIG option.
Shown Trigger	integer value	Displays the number of triggered events since last start.
Plot Type		Select the plot type.
	None	No plot displayed.
	2D	Display defined number of grid rows as one 2D plot.
	Row	Display only the trace of index defined in the Active Row field.
	2D + Row	Display 2D and row plots.
Active Row	integer value	Set the row index to be displayed in the Row plot.
Track Active Row	ON / OFF	If enabled, the active row marker will track with the last recorded row. The active row control field is read-only if enabled.
Palette	Solar	Select the colormap for the current plot.
	Viridis	
	Inferno	
	Balance	
	Turbo	
	Grey	
Colorscale	ON / OFF	Enable/disable the colorscale bar display in the 2D plot.
Mapping		Mapping of colorscale.
	Lin	Enable linear mapping.
	Log	Enable logarithmic mapping.
	dB	Enable logarithmic mapping in dB.
Scaling	Full Scale/ Manual/Auto	Scaling of colorscale.
Clamp To Color	ON / OFF	When enabled, grid values that are outside of defined Min or Max region are painted with Min or Max color equivalents. When disabled, Grid values that are outside of defined Min or Max values are left transparent.
Start	numeric value	Lower limit of colorscale. Only visible for manual scaling.
Stop	numeric value	Upper limit of colorscale. Only visible for manual scaling.

Table 5.20: Scope tab: Advanced sub-tab

Control/ Tool	Option/ Range	Description
FFT Window	Cosine squared (ring-down)	Several different FFT windows to choose from. Each window function results in a different trade-off between amplitude accuracy and spectral leakage. Please check the literature to find the window function that best suits your needs.
	Rectangular	
	Hann	
	Hamming	
	Blackman Harris	
	Flat Top	
	Exponential (ring-down)	
	Cosine (ring-down)	
Resolution (Hz)	mHz to Hz	Spectral resolution defined by the reciprocal acquisition time (sample rate, number of samples recorded).
Absolute Frequency	ON / OFF	Shifts x-axis labeling to show the absolute frequency in the center as opposed to 0 Hz, when turned off.
Spectral Density	ON / OFF	Calculate and show the spectral density. If power is enabled the power spectral density value is calculated. The spectral density is used to analyze noise.
Power	ON / OFF	Calculate and show the power value. To extract power spectral density (PSD) this button should be enabled together with Spectral Density.
Persistence	ON / OFF	Keeps previous scope shots in the display. The color scheme visualizes the number of occurrences at certain positions in time and amplitude by a multi-color scheme.
BW Limit		Selects between sample decimation and sample averaging. Averaging avoids aliasing, but may conceal signal peaks. Channel 2 requires the DIG option.
	OFF	Selects sample decimation for sample rates lower than the maximal available sampling rate.
	ON	Selects sample averaging for sample rates lower than the maximal available sampling rate.
Rate	27.5 kHz to 28.1 MHz	Streaming rate of the scope channels. The streaming rate can be adjusted independent from the scope sampling rate. The maximum rate depends on the interface used for transfer. Note: scope streaming requires the DIG option.
Enable	ON / OFF	Enable scope streaming for the specified channel. This allows for continuous recording of scope data on the plotter and streaming to disk. Note: scope streaming requires the DIG option.
X Axis		Select the x-axis for xy-plot display mode.
	Time/Freq	The xy-plot mode is off. The x-axis is either time or frequency.
	Channel 1	The xy-plot is enabled with the first channel used for the x-axis.
	Channel 2	The xy-plot is enabled with the second channel used for the x-axis.

Table 5.21: Scope tab: History sub-tab

Control/ Tool	Option/ Range	Description
History	History	Each entry in the list corresponds to a single trace in the history. The number of traces displayed in the plot is limited to 20. Use the toggle buttons to hide or show individual traces. Use the color picker to change the color of a trace in the plot. Double click on a list entry to edit its name.

Control/ Tool	Option/ Range	Description
Length	integer value	Maximum number of records in the history. The number of entries displayed in the list is limited to the 100 most recent ones.
Clear All	Clear All	Remove all records from the history list.
Clear	Clear	Remove selected records from the history list.
Load file		Load data from a file into the history. Loading does not change the data type and range displayed in the plot, this has to be adapted manually if data is not shown.
Name		Enter a name which is used as a folder name to save the history into. An additional three digit counter is added to the folder name to identify consecutive saves into the same folder name.
Auto Save		Activate autosaving. When activated, any measurements already in the history are saved. Each subsequent measurement is then also saved. The autosave directory is identified by the text "autosave" in the name, e.g. "sweep_autosave_001". If autosave is active during continuous running of the module, each successive measurement is saved to the same directory. For single shot operation, a new directory is created containing all measurements in the history. Depending on the file format, the measurements are either appended to the same file, or saved in individual files. For HDF5 and ZView formats, measurements are appended to the same file. For MATLAB and SXM formats, each measurement is saved in a separate file.
File Format		Select the file format in which to save the data.
Save	Save	Save the traces in the history to a file accessible in the File Manager tab. The file contains the signals in the Vertical Axis Groups of the Control sub-tab. The data that is saved depends on the selection from the pull-down list. Save All: All traces are saved. Save Sel: The selected traces are saved.

For the Math sub-tab please see the table "Plot math description" in the section called "Cursors and Math".

5.10. Data Acquisition Tab

The Data Acquisition tool is one of the powerful time domain measurement tools as introduced in Unique Set of Analysis Tools and is available on all UHFLI instruments . On UHFAWG instruments, the tab is available if at least one of the options UHF-BOX, UHF-CNT, or UHF-LIA is installed. This tab used to be named Software Trigger tab in previous versions of the LabOne software.

5.10.1. Features

- Time-domain and frequency domain display for all continuously streamed data
- Capture and color scale display of imaging data
- Frame averaging and pixel interpolation
- Automatic trigger level determination
- Display of multiple traces
- Adjustable record history
- Mathematical toolkit for signal analysis

5.10.2. Description

The Data Acquisition tab features display and recording of shot-wise and imaging data sets upon a trigger event. 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 5.22: App icon and short description

Control/ Tool	Option/ Range	Description
DAQ		Provides complex trigger functionality on all continuously streamed data samples and time domain display.

The Data Acquisition tab (see LabOne UI: Data Acquisition tab) is divided into a display section on the left and a configuration section on the right. The configuration section is further divided into a number of sub-tabs.

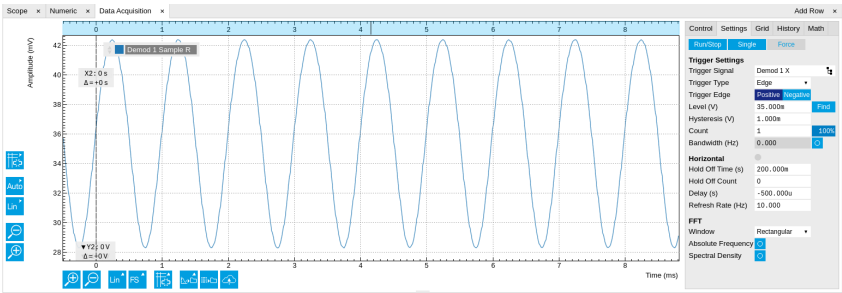


Figure 5.24: LabOne UI: Data Acquisition tab

The Data Acquisition tool brings the trigger functionality of a scope with FFT capability to the demodulator signals and other streamed data. The user can choose between a variety of different trigger and display options in the time and frequency domain.

Use the **Control** sub-tab to configure which signals are measured, both in time and in frequency domain. Measurement signals can be added to the Vertical Axis Groups section as described in Vertical Axis Groups. There is one vertical axis group for each the time domain and the frequency domain data.

The trigger condition is configured in the **Settings** sub-tab. Among the selection of Trigger Types provided here, Edge and Pulse are applicable to analog trigger sources such as demodulator data, auxiliary voltages, or oscillator frequencies. The trigger time resolution is enhanced above the sampling rate of the analog data by using interpolation. Instead of manually setting a Trigger Level, you can click on **Find** to have LabOne find a value by analyzing the data stream. In case of noisy trigger sources, both the Bandwidth and the Hysteresis setting can help preventing false trigger events. The Bandwidth setting provides a configurable low-pass filter applied to the trigger source. When enabling this function, be sure to choose a sufficiently high bandwidth to resolve the signal feature that should be triggered upon, i.e., the signal edge or pulse. The Bandwidth setting does not affect the recorded data.

For trigger sources with a slowly varying offset, the Tracking Edge and Tracking Pulse Trigger Types provide continuous adjustment of the Level and Hysteresis. In Tracking mode, the Bandwidth setting plays a different role than for the Edge and Pulse trigger types. Here, the Bandwidth should be chosen sufficiently low to filter out all fast features and only let pass the slow offset. The Trigger Types HW Trigger and Digital are used for TTL signals on the Trigger Inputs or on the DIO lines. Using the Bits and Bit Mask setting, complex multi-bit trigger conditions on the DIO lines can be defined. The timing resolution for digital triggers is given by the demodulator sample rate because the state of the DIO line is transferred together with demodulator data.

The Horizontal section of the Settings sub-tab contains the settings for shot Duration and Delay (negative delays correspond to pre-trigger time). Also minimum and maximum pulse width for the Pulse and Tracking Pulse trigger types are defined here.

The **Grid** sub-tab provides imaging functionality to capture and display two-dimensional data sets organized in frames consisting of rows and columns. By default, the number of rows is 1, which means the Data Acquisition tool operates similar to a scope. With a Rows setting larger than 1, every newly captured shot of data is assigned to a row until the number of rows is reached and the frame is complete. After completion of a full frame, the new data either replace the old or averaging is performed, according to the selected Operation and Repetitions setting. On the horizontal axis, the Duration of a shot is divided into a number of samples specified with the Columns setting. The Mode settings provides the functionality for post-processing of the streamed data for interpolation, resampling, and alignment with the trigger event. This is particularly helpful when capturing data from several sources, e.g. demodulators and PID controllers. As illustrated in Figure 5.25, in such situation the streamed data don't lie on the same temporal grid by default. This can be changed by setting Mode to Linear or Nearest. In these modes, the streams from several sources will be up-sampled to match the sampling rate and temporal grid of the fastest data stream. This means data processing after saving becomes more convenient, however note that the actual streamed data rate is not increased, and the data don't gain in time resolution. A two-dimensional color scale image of

the data can be enabled and controlled in the Display section. The display features configurable scaling, range, and color scale.

With enabled grid mode, the data of a completed frame after averaging appear as a list entry in the **History** sub-tab. See History List for more details on how data in the history list can be managed and stored.

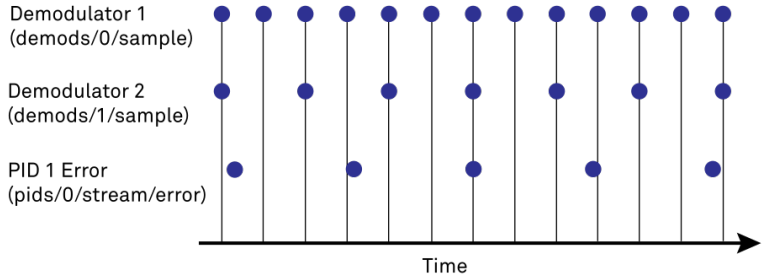


Figure 5.25: Samples from different sources configured with different rates: demodulator 1 at 2N kSa/s, demodulator 2 at N kSa/s and PID Error 1 at M kSa/s (N not divisible by M). Although each stream consists of equidistantly spaced samples in time, the sample timestamps from different streams are not necessarily aligned due to the different sampling rates

5.10.3. Functional Elements

Table 5.23: DAQ tab: Control sub-tab

Control/ Tool	Option/ Range	Description
Run/Stop	Run/Stop	Start and stop the Data Acquisition tool
Single	Single	Run the Data Acquisition tool once (record Count trigger events)
Force	Force	Forces a single trigger event.
Triggered	grey/green	When green, indicates that new trigger shots are being captured and displayed in the plot area.

For the Vertical Axis Groups, please see the table "Vertical Axis Groups description" in the section called "Vertical Axis Groups".

Table 5.24: DAQ tab: Settings sub-tab

Control/ Tool	Option/ Range	Description
Trigger Signal		Source signal for trigger condition. Navigate through the tree view that appears and click on the required signal.
Trigger Type		Select the type of trigger to use. Selectable options depend on the selected trigger signal.
	Continuous	Continuous triggering.
	Edge	Analog edge triggering based on high and low level. Hysteresis on the levels and low-pass filtering can be used to reduce the risk of wrong trigger for noisy trigger signals.
	Digital	Digital triggering on the 32-bit DIO lines. The bit value defines the trigger condition. The bit mask controls the bits that are used for trigger evaluation. When using a Positive Edge trigger setting, a trigger event occurs as soon as the equality (DIO Value)AND(Bit Mask) = (Bits)AND(Bit Mask) is fulfilled (and was not previously fulfilled). In order to trigger on DIO0 set bit value to 1 and bit mask to 1; to trigger on DIO1 set bit value to 2 and bit mask to 2.
	Pulse	Triggers if a pulse on an analog signal is within the min and max pulse width. Pulses can be defined as either low to high then high to low (positive), the reverse (negative) or both.

Control/ Tool	Option/ Range	Description
	Tracking Edge	Edge triggering with automatic adjustment of trigger levels to compensate for drifts. The tracking speed is controlled by the bandwidth of the low-pass filter. For this filter noise rejection can only be achieved by level hysteresis.
	HW Trigger	Trigger on one of the four trigger inputs. Ensure that the trigger level and the trigger coupling is correctly adjusted. The trigger input state can be monitored on the plotter.
	Tracking Pulse	Pulse triggering with automatic adjustment of trigger levels to compensate for drifts. The tracking speed is controlled by the bandwidth of the low-pass filter. For this filter noise rejection can only be achieved by level hysteresis.
	Pulse Count	Trigger on trigger events supplied by the pulse counter. This allows for high resolution triggering. The pulse counter must be enabled and configured on the pulse counter tab. This functionality requires the UHF-CNT option.
Pulse Type	Positive/ Negative/ Both	Select between negative, positive or both pulse forms in the signal to trigger on.
Trigger Edge	Positive/ Negative/ Both	Triggers when the trigger input signal is crossing the trigger level from either high to low, low to high or both. This field is only displayed for trigger type Edge, Tracking Edge and Event Count.
Bits	0 to 2^32-1	Specify the value of the DIO to trigger on. All specified bits have to be set in order to trigger. This field is only displayed for trigger type Digital.
Bit Mask	0 to 2^32-1	Specify a bit mask for the DIO trigger value. The trigger value is bits AND bit mask (bitwise). This field is only displayed for trigger type Digital.
Level	full signal range	Specify the trigger level value.
Find	Find	Automatically find the trigger level based on the current signal.
Hysteresis	full signal range	The hysteresis is important to trigger on the correct edge in the presence of noise. The hysteresis is applied below the trigger level for positive trigger edge selection. It is applied above for negative trigger edge selection, and on both sides for triggering on both edges.
Count Type		Type of pulse to trigger on for pulse counter trigger signals.
	Any	Trigger on every sample from the pulse counter, regardless of the counter value.
	Increment	Trigger on incrementing counter values.
Count	integer number	Number of trigger events to record (in Single mode)
Trigger progress	0% to 100%	The percentage of triggers already acquired (in Single mode)
Bandwidth (Hz)	0 to 0.5 * Sampling Rate	Bandwidth of the low-pass filter applied to the trigger signal. For edge and pulse trigger use a bandwidth larger than the signal sampling rate divided by 20 to keep the phase delay. For tracking filter use a bandwidth smaller than signal sampling frequency divided by 100 to just track slow signal components like drifts.
Enable	ON / OFF	Enable low-pass filtering of the trigger signal.
Hold Off Time (s)	positive numeric value	Hold off time before the trigger is rearmed. A hold off time smaller than the duration will lead to overlapping trigger frames.
Hold Off Count	integer value	Number of skipped triggers until the next trigger is recorded again.
Delay (s)	-Duration to Duration	Time delay of trigger frame position (left side) relative to the trigger edge. For delays smaller than 0, trigger edge inside trigger frame (pre trigger). For delays greater than 0, trigger edge before trigger frame (post trigger)

Control/ Tool	Option/ Range	Description
Refresh Rate	100 mHz to 10 Hz	Set the maximum refresh rate for plot updates. The actual refresh rate depends on other factors such as the hold-off time and duration.
Pulse Min (s)	0 to Duration	Minimum pulse width to trigger on.
Pulse Max (s)	0 to Duration	Maximum pulse width to trigger on.
Window	Cosine squared (ring-down)	Several different FFT windows to choose from. Depending on the application it makes a huge difference which of the provided window function is used. Please check the literature to find out the best trade off for your needs.
	Rectangular	
	Hann	
	Hamming	
	Blackman Harris	
	Flat Top	
	Exponential (ring-down)	
	Cosine (ring-down)	
Absolute Frequency	ON / OFF	Shifts x-axis labeling to show the demodulation frequency in the center as opposed to 0 Hz, when turned off.
Spectral Density	ON / OFF	Calculate and show the spectral density. If power is enabled the power spectral density value is calculated. The spectral density is used to analyze noise.

Table 5.25: DAQ tab: Grid sub-tab

Control/ Tool	Option/ Range	Description
Mode		Select resampling method for two-dimensional data recording.
	Off	Two-dimensional data recording is disabled.
	Nearest	Resampling is performed using substitution by closest data point.
	Linear	Resampling is performed using linear interpolation.
	Exact (on-grid)	Adjust the duration so that the grid distance matches the maximal sampling rate of the selected signals. This allows for on-grid sampling of measurement data. If a signal uses lower sampling rate it will be up-sampled by linear interpolation.
On Grid Sampling	Green or yellow	When green, indicates that all the captured data is aligned to the grid. When yellow, indicates that some data is not aligned to the grid and is interpolated. This can happen when one or more data sources have different sampling rates, or when a sampling rate changes.
Operation		Select row update method.
	Replace	New row replaces old row.
	Average	The data for each row is averaged over a number of repetitions.
	Std	The data for each row is the standard deviation over a number of repetitions.
Columns	numeric value	Number of columns. The data along the horizontal axis are resampled to a number of samples defined by this setting.
Duration	up to 1000 s	Recording length for each triggered data set. In exact sampling mode the duration is a read-only field. The duration is then defined by the maximal sampling rate and column size.

Control/ Tool	Option/ Range	Description
Rows	numeric value	Number of rows
Scan Direction		Select the scan direction and mode
	Forward	Scan direction from left to right
	Reverse	Scan direction from right to left
	Bidirectional	Alternate scanning in both directions
Repetitions	numeric value	Number of repetitions used for averaging
Row-wise repetition	ON / OFF	Enable row-wise repetition. With row-wise repetition, each row is calculated from successive repetitions before starting the next row. With grid-wise repetition, the entire grid is calculated with each repetition.
Waterfall	ON / OFF	Enable to show the 2D plot in waterfall mode. It will always update the last line.
Overwrite	ON / OFF	Enable to overwrite the grid in continuous mode. History will not be collected. A history element will only be created when the analysis is stopped.
AWG Control	ON / OFF	If enabled, the row number is identified based on the digital row ID number set by the AWG. If disabled, every new trigger event is attributed to a new row sequentially.
Plot Type		Select the plot type.
	None	No plot displayed.
	2D	Display defined number of grid rows as one 2D plot.
	Row	Display only the trace of index defined in the Active Row field.
	2D + Row	Display 2D and row plots.
Active Row	integer value	Set the row index to be displayed in the Row plot.
Track Active Row	ON / OFF	If enabled, the active row marker will track with the last recorded row. The active row control field is read-only if enabled.
Palette	Solar	Select the colormap for the current plot.
	Viridis	
	Inferno	
	Balance	
	Turbo	
	Grey	
Colorscale	ON / OFF	Enable/disable the colorscale bar display in the 2D plot.
Mapping		Mapping of colorscale.
	Lin	Enable linear mapping.
	Log	Enable logarithmic mapping.
	dB	Enable logarithmic mapping in dB.
Scaling	Full Scale/ Manual/Auto	Scaling of colorscale.
Clamp To Color	ON / OFF	When enabled, grid values that are outside of defined Min or Max region are painted with Min or Max color equivalents. When disabled, Grid values that are outside of defined Min or Max values are left transparent.
Start	numeric value	Lower limit of colorscale. Only visible for manual scaling.
Stop	numeric value	Upper limit of colorscale. Only visible for manual scaling.

Table 5.26: DAQ tab: History sub-tab

Control/ Tool	Option/ Range	Description
History	History	Each entry in the list corresponds to a single trace in the history. The number of traces displayed in the plot is limited to 20. Use the toggle buttons to hide or show individual traces. Use the color picker to change the color of a trace in the plot. Double click on a list entry to edit its name.
Length	integer value	Maximum number of records in the history. The number of entries displayed in the list is limited to the 100 most recent ones.
Clear All	Clear All	Remove all records from the history list.
Clear	Clear	Remove selected records from the history list.
Load file		Load data from a file into the history. Loading does not change the data type and range displayed in the plot, this has to be adapted manually if data is not shown.
Name		Enter a name which is used as a folder name to save the history into. An additional three digit counter is added to the folder name to identify consecutive saves into the same folder name.
Auto Save		Activate autosaving. When activated, any measurements already in the history are saved. Each subsequent measurement is then also saved. The autosave directory is identified by the text "autosave" in the name, e.g. "sweep_autosave_001". If autosave is active during continuous running of the module, each successive measurement is saved to the same directory. For single shot operation, a new directory is created containing all measurements in the history. Depending on the file format, the measurements are either appended to the same file, or saved in individual files. For HDF5 and ZView formats, measurements are appended to the same file. For MATLAB and SXM formats, each measurement is saved in a separate file.
File Format		Select the file format in which to save the data.
Save	Save	Save the traces in the history to a file accessible in the File Manager tab. The file contains the signals in the Vertical Axis Groups of the Control sub-tab. The data that is saved depends on the selection from the pull-down list. Save All: All traces are saved. Save Sel: The selected traces are saved.

For the Math sub-tab please see the table "Plot math description" in the section called "Cursors and Math".

5.11. Spectrum Analyzer Tab

The Spectrum Analyzer is one of the powerful frequency domain measurement tools as introduced in Unique Set of Analysis Tools and is available on all UHF Series instruments.

5.11.1. Features

- Fast, high-resolution FFT spectrum analyzer
- Signals: demodulated data (X+iY, R, Θ, f and dΘ/dt/(2π)), PID, Boxcar, Auxiliary Inputs, and more
- Variable center frequency, frequency resolution and frequency span
- Auto bandwidth
- Waterfall display
- Choice of 4 different FFT window functions
- Continuous and block-wise acquisition with different types of averaging
- Detailed noise power analysis
- Support for Input Scaling and Input Units
- Mathematical toolbox for signal analysis

5.11.2. Description

The Spectrum Analyzer provides frequency domain analysis of demodulator data. 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 5.27: App icon and short description

Control/ Tool	Option/ Range	Description
Spectrum		Provides FFT functionality to all continuously streamed measurement data.

The Spectrum tab (see LabOne UI: Spectrum analyzer tab) is divided into a display section on the left and a configuration section on the right. The configuration section is further divided into a number of sub-tabs.

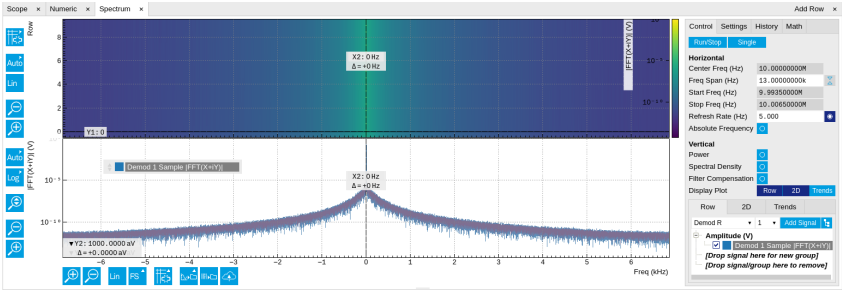


Figure 5.26: LabOne UI: Spectrum analyzer tab

The Spectrum Analyzer allows for spectral analysis of all the demodulator data by performing the fast Fourier transform (FFT) on the complex demodulator data samples X+iY (with i as the imaginary unit). The result of this FFT is a spectrum centered around the demodulation frequency, whereas applying a FFT directly on the raw input data would produce a spectrum centered around zero frequency. The latter procedure corresponds to the Frequency Domain operation in the Scope Tab. The main difference between the two is that the Spectrum Analyzer tool can acquire data for a much longer periods of time and therefore can achieve very high frequency resolution around the demodulation frequency. By default, the spectrum is displayed centered around zero. Sometimes however it is convenient to shift the frequency axis by the demodulation frequency which allows one to identify the frequencies on the horizontal axis with the physical frequencies at the signal inputs. This can be done by activating Absolute Frequency on the Settings sub-tab.

By default, the display section contains a line plot of the spectrum together with a color waterfall plot of the last few acquired spectra. The waterfall plot makes it easier to see the evolution of the spectrum over time. The display layout as well as the number of rows in the color plot can be configured in the Settings sub-tab.

Data shown in the Spectrum tab have passed a low-pass filter with a well-defined order and bandwidth. This is most clearly noted by the shape of the noise floor. One has to take care that the selected frequency span, which equals the demodulator sampling rate, is 5 to 10 times higher than the filter bandwidth in order to prevent measurement errors due to aliasing. The Auto Bandwidth button A adjusts the sampling rate so that it suits the filter settings. The Spectrum tab features FFT display of a selection of data available in the Signal Type drop-down list in addition to the complex demodulator samples X+iY. Looking at the FFT of polar demodulator values R and Theta allows one to discriminate between phase noise components and amplitude noise components in the signal. The FFT of the phase derivative dΘ/dt provides a quantitative view of the spectrum of demodulator frequencies. That is particularly useful in conjunction with the PLL or the ExtRef functionalities. The FFT of the frequency samples then provide a quantitative view of what frequency noise components are present in the reference signal and also helps to find the optimal PLL bandwidth to track the signal. Note that many of the signals in the Signal Type list are real-valued, rather than complex-valued. Their spectra are single-sided with minimum frequency of 0 Hz.

5.11.3. Functional Elements

Table 5.28: Spectrum tab: Settings sub-tab

Control/Tool	Option/ Range	Description
Run/Stop	Run/Stop	Run the FFT spectrum analysis continuously
Single	Single	Run the FFT spectrum analysis once
Center Freq (Hz)	numeric value	Demodulation frequency of the selected demodulator used as input for the spectrum. For complex FFT(X+iY) the demodulation frequency defines the center frequency of the displayed FFT.

Control/Tool	Option/ Range	Description
Frequency Span (Hz)	numeric value	Set the frequency span of interest for the complex FFT. A FFT based on real input data will display half of the frequency span up to the Nyquist frequency.
Auto Bandwidth	A	Automatic adjustment of the demodulator bandwidths to obtain optimal alias rejection for the selected frequency span which is equivalent to the sampling rate. The functionality is only available if the spectrum is enabled.
Start Frequency (Hz)	numeric value	Indicates the start frequency of the FFT.
Stop Frequency (Hz)	numeric value	Indicates the end frequency of the FFT.
Refresh Rate (Hz)	numeric value	Set the maximum plot refresh rate. The actual refresh rate also depends on other parameters such as FFT length. In overlapped mode the refresh rate defines the amount of overlapping.
Overlapped FFT	ON / OFF	Enable overlapped FFTs. If disabled, FFTs are performed on distinct abutting data sets. If enabled, the data sets of successive FFTs overlap based on the defined refresh rate.
Power	ON / OFF	Calculate and show the power value. To extract power spectral density (PSD) this button should be enabled together with spectral density.
Spectral Density	ON / OFF	Calculate and show the spectral density. If power is enabled the power spectral density value is calculated. The spectral density is used to analyze noise.
Filter Compensation	ON / OFF	Spectrum is corrected by demodulator filter transfer function. Allows for quantitative comparison of amplitudes of different parts of the spectrum.
Absolute Frequency	ON / OFF	Shifts x-axis labeling to show the demodulation frequency in the center as opposed to 0 Hz, when turned off.
FFT length	numeric value	The number of samples used for the FFT. Values entered that are not a binary power are truncated to the nearest power of 2.
Sampling Progress	0% to 100%	The percentage of the FFT buffer already acquired. The progress includes the number of rows and averages.
FFT Duration (s)	numeric value	Indicates the length in time of the samples used for a single FFT.
Window	Cosine squared (ring-down)	Several different FFT windows to choose from. Depending on the application it makes a huge difference which of the provided window function is used. Please check the literature to find out the best trade off for your needs.
	Rectangular	
	Hann	
	Hamming	
	Blackman Harris	
	Flat Top	
	Exponential (ring-down)	
	Cosine (ring-down)	
Resolution (Hz)	mHz to Hz	Spectral resolution defined by the reciprocal acquisition time (sample rate, number of samples recorded).
Rows	numeric value	Number of rows
Averages	numeric value	Number of FFT averaged for each row. Setting the value to 1 will disable any averaging.
Waterfall	ON / OFF	Enable to show the 2D plot in waterfall mode. It will always update the lowest line.

Control/Tool	Option/ Range	Description
Overwrite	ON / OFF	Enable to overwrite the grid in continuous mode. History will not be collected. A history element will only be created when the analysis is stopped.
AWG Control	ON / OFF	If enabled, the row number is identified based on the digital row ID number set by the AWG. If disabled, every new trigger event is attributed to a new row sequentially.
Plot Type		Select the plot type.
	None	No plot displayed.
	2D	Display defined number of grid rows as one 2D plot.
	Row	Display only the trace of index defined in the Active Row field.
	2D + Row	Display 2D and row plots.
Active Row	integer value	Set the row index to be displayed in the Row plot.
Track Active Row	ON / OFF	If enabled, the active row marker will track with the last recorded row. The active row control field is read-only if enabled.
Palette	Solar	Select the colormap for the current plot.
	Viridis	
	Inferno	
	Balance	
	Turbo	
	Grey	
Colorscale	ON / OFF	Enable/disable the colorscale bar display in the 2D plot.
Mapping		Mapping of colorscale.
	Lin	Enable linear mapping.
	Log	Enable logarithmic mapping.
	dB	Enable logarithmic mapping in dB.
Scaling	Full Scale/ Manual/Auto	Scaling of colorscale.
Clamp To Color	ON / OFF	When enabled, grid values that are outside of defined Min or Max region are painted with Min or Max color equivalents. When disabled, Grid values that are outside of defined Min or Max values are left transparent.
Start	numeric value	Lower limit of colorscale. Only visible for manual scaling.
Stop	numeric value	Upper limit of colorscale. Only visible for manual scaling.

For the Math sub-tab please see the table "Plot math description" in the section called "Cursors and Math".

5.12. Sweeper Tab

The Sweeper is a highly versatile measurement tool available on all UHFLI instruments. The Sweeper enables scans of an instrument parameter over a defined range and simultaneous measurement of a selection of continuously streamed data. In the special case where the sweep parameter is an oscillator frequency, the Sweeper offers the functionality of a frequency response analyzer (FRA), a well-known class of instruments. On UHFAWG instruments, the tab is available if at least one of the options UHF-BOX, UHF-CNT, or UHF-LIA is installed.

5.12.1. Features

- Full-featured parametric sweep tool for frequency, phase shift, output amplitude, DC output voltages, etc.
- Simultaneous display of data from different sources (Demodulators, PIDs, auxiliary inputs, and others)
- Different application modes, e.g. Frequency response analyzer (Bode plots), noise amplitude sweeps, etc.
- Different sweep types: single, continuous (run / stop), bidirectional, binary
- Persistent display of previous sweep results
- XY Mode for Nyquist plots or I-V curves
- Normalization of sweeps
- Auto bandwidth, averaging and display normalization
- Support for Input Scaling and Input Units
- Phase unwrap
- Full support of sinc filter

5.12.2. Description

The Sweeper supports a variety of experiments where a parameter is changed stepwise and numerous measurement data can be graphically displayed. Open the tool by clicking the corresponding icon in the UI side bar. The Sweeper tab (see Figure 5.27) is divided into a plot section on the left and a configuration section on the right. The configuration section is further divided into a number of sub-tabs.

Table 5.29: App icon and short description

Control/ Tool	Option/ Range	Description
Sweeper		Sweep frequencies, voltages, and other quantities over a defined range and display various response functions including statistical operations.

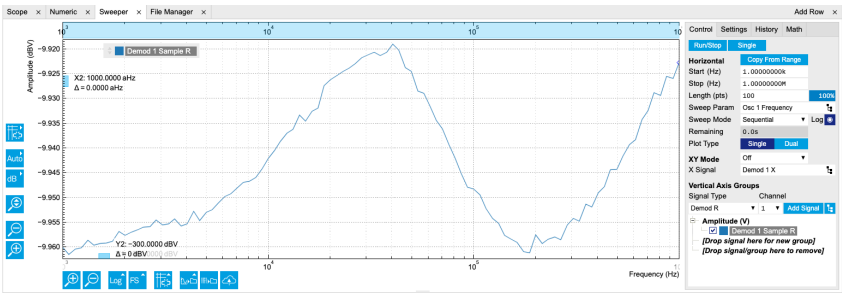


Figure 5.27: LabOne UI: Sweeper tab

The **Control** sub-tab holds the basic measurement settings such as Sweep Parameter, Start/Stop values, and number of points (Length) in the Horizontal section. Measurement signals can be added in the Vertical Axis Groups section. A typical use of the Sweeper is to perform a frequency sweep and measure the response of the device under test in the form of a Bode plot. As an example, AFM and MEMS users require to determine the resonance frequency and the phase delay of their oscillators. The Sweeper can also be used to sweep parameters other than frequency, for instance signal amplitudes and DC offset voltages. A sweep of the Auxiliary Output offset can for instance be used to measure current-voltage (I-V) characteristics. The XY Mode allows one to use a measured signal, rather than the sweep parameter, on the horizontal axis. This is useful to obtain Nyquist plots in impedance measurements, or to display an I-V curve in a four-probe measurement of a nonlinear device.

For frequency sweeps, the sweep points are distributed logarithmically, rather than linearly, between the start and stop values by default. This feature is particularly useful for sweeps over several decades and can be disabled with the Log checkbox. The Sweep Mode setting is useful for identifying measurement problems caused by hysteretic sample behavior or too fast sweeping speed. Such problems would cause non-overlapping curves in a bidirectional sweep.

Note

The Sweeper actively modifies the main settings of the demodulators and oscillators. So in particular for situations where multiple experiments are served maybe even from different control computers great care needs to be taken so that the parameters altered by the sweeper module do not have unwanted effects elsewhere.

The Sweeper offers two operation modes differing in the level of detail of the accessible settings: the Application Mode and the Advanced Mode. Both of them are accessible in the **Settings** sub-tab. The Application Mode provides the choice between six measurement approaches that should help to quickly obtain correct measurement results for a large range of applications. Users who like to be in control of all the settings can access them by switching to the Advanced Mode.

In the Statistics section of the Advanced Mode one can control how data is averaged at each sweep point: either by specifying the Sample count, or by specifying the number of filter time constants (TC). The actual measurement time is determined by the larger of the two settings, taking into account the demodulator sample rate and filter settings. The Algorithm settings determines the statistics calculated from the measured data: the average for general purposes, the deviation for noise measurements, or the mean square for power measurements. The Phase Unwrap features ensures continuity of a phase measurement curve across the PM180 degree boundary. Enabling the Sinc Filter setting means that the demodulator Sinc Filter gets activated for sweep points below 50 Hz in Auto and Fixed mode. This speeds up measurements at small frequencies as explained in the Sinc Filtering .

In the Settling section one can control the waiting time between a parameter setting and the first measurement. Similarly to the Statistics setting, one has the choice between two different representations of this waiting time. The actual settling time is the maximum of the values set in units of absolute time and a time derived from the demodulator filter and a desired inaccuracy (e.g. 1 m for 0.1%). Let's consider an example. For a 4th order filter and a 3 dB bandwidth of 100 Hz we obtain a step response the attains 90% after about 4.5 ms. This can be easily measured by using the Data Acquisition tool as indicated in Figure 5.28. It is also explained in Discrete-Time Filters. In case the full range is set to 1 V this means a measurement has a maximum error caused by imperfect settling of about 0.1 V. However, for most measurements the neighboring values are close compared to the full range and hence the real error caused is usually much smaller.

In the Filter section of the Advanced mode, the Bandwidth Mode setting determines how the filters of the activated demodulators are configured. In Manual mode, the current setting (in the Lock-in tab) remains unchanged by the Sweeper. In Fixed mode, the filter settings can be controlled from within the Sweeper tab. In Auto mode, the Sweeper determines the filter bandwidth for each sweep point based on a desired ω suppression. The ω suppression depends on the measurement frequency and the filter steepness. For frequency sweeps, the bandwidth will be adjusted for every sweep point within the bound set by the Max Bandwidth setting. The Auto mode is particularly useful for frequency sweeps over several decades, because the continuous adjustment of the bandwidth considerably reduces the overall measurement time.

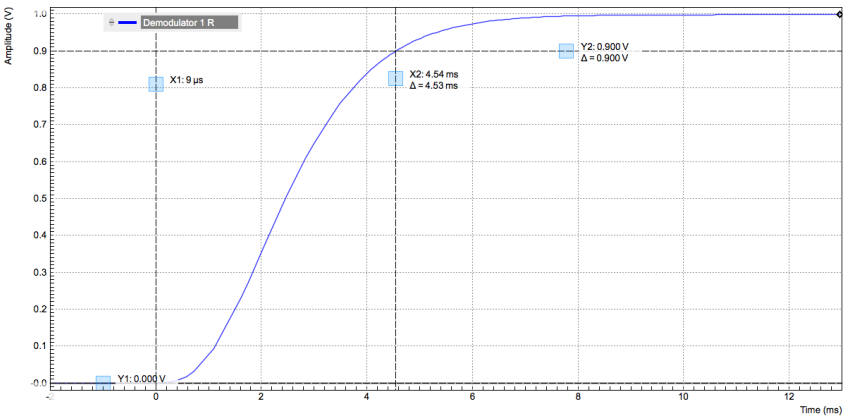


Figure 5.28: Demodulator settling time and inaccuracy: measurement carried out with the Data Acquisition tool to illustrate the settling time for a 4

By default the plot area keeps the memory and display of the last 100 sweeps represented in a list in the **History** sub-tab. See History List for more details on how data in the history list can be managed and stored. With the Reference feature, it is possible to divide all measurements in the history by a reference measurement. This is useful for instance to eliminate spurious effects in a frequency response sweep. To define a certain measurement as the reference, mark it in the list and click on Reference . Then enable the Reference mode with the checkbox below the list to update the plot

display. Note that the Reference setting does not affect data saving: saved files always contain raw data.

Note

The Sweeper can get stuck whenever it does not receive any data. A common mistake is to select to display demodulator data without enabling the data transfer of the associated demodulator in the Lock-in tab.

Note

Once a sweep is performed the sweeper stores all data from the enabled demodulators and auxiliary inputs even when they are not displayed immediately in the plot area. These data can be accessed at a later point in time simply by choosing the corresponding signal display settings (Input Channel).

5.12.3. Functional Elements

Table 5.30: Sweeper tab: Control sub-tab

Control/ Tool	Option/ Range	Description
Run/Stop	Run/Stop	Runs the sweeper continuously.
Single	Single	Runs the sweeper once.
Copy From X-Axis		Takes over start and stop value from the X-axis.
Copy From X-Cursors		Takes over start and stop value from X-cursors. Button is disabled when one or both X cursors are not visible.
Start (unit)	numeric value	Start value of the sweep parameter. The unit adapts according to the selected sweep parameter.
Stop (unit)	numeric value	Stop value of the sweep parameter. The unit adapts according to the selected sweep parameter.
Length	integer value	Sets the number of measurement points.
Progress	0 to 100%	Reports the sweep progress as ratio of points recorded.
Sweep Param		Selects the parameter to be swept. Navigate through the tree view that appears and click on the required parameter. The available selection depends on the configuration of the device.
Sweep Mode		Select the scanning type, default is sequential (incremental scanning from start to stop value)
	Sequential	Sequential sweep from Start to Stop value
	Binary	Non-sequential sweep continues increase of resolution over entire range
	Bidirectional	Sequential sweep from Start to Stop value and back to Start again
	Reverse	Reverse sweep from Stop to Start value
X Distribution	Linear / Logarithmic	Selects between linear and logarithmic distribution of the sweep parameter.
Remaining	numeric value	Reporting of the remaining time of the current sweep. A valid number is only displayed once the sweeper has been started. An undefined sweep time is indicated as NaN.
Invert Y Axis	ON / OFF	The xy-plot is displayed with inverted y-axis. This mode is used for Nyquist plots that allow for displaying -imag(z) on the y-axis and real(z) on the x-axis.

Control/ Tool	Option/ Range	Description
X Signal		Selects the signal that defines the x-axis for xy-plots. The available selection depends on the configuration of the device. Displaying the selected signal source will result in a diagonal straight line.

For the Vertical Axis Groups, please see the table "Vertical Axis Groups description" in the section called "Vertical Axis Groups".

Table 5.31: Sweeper tab: Settings sub-tab

Control/ Tool	Option/ Range	Description
Filter		Application Mode: preset configuration. Advanced Mode: manual configuration.
	Application Mode	The sweeper sets the filters and other parameters automatically.
	Advanced Mode	The sweeper uses manually configured parameters.
Application		Select the sweep application mode
	Parameter Sweep	Only one data sample is acquired per sweeper point.
	Parameter Sweep Averaged	Multiple data samples are acquired per sweeper point of which the average value is displayed.
	Noise Amplitude Sweep	Multiple data samples are acquired per sweeper point of which the standard deviation is displayed (e.g. to determine input noise). For accurate noise measurement, the signal amplitude R is replaced by its quadrature components X and Y.
	Freq Response Analyzer	Narrow band frequency response analysis. Averaging is enabled.
	3-Omega Sweep	Optimized parameters for 3-omega application. Averaging is enabled.
	FRA (Sinc Filter)	The sinc filter helps to speed up measurements for frequencies below 50 Hz in FRA mode. For higher frequencies it is automatically disabled. Averaging is off.
	Impedance	This application mode uses narrow bandwidth filter settings to achieve high calibration accuracy.
Precision		Choose between a high speed scan speed or high precision and accuracy.
	Low -> fast sweep	Medium accuracy/precision is optimized for sweep speed.
	High -> standard speed	Medium accuracy/precision takes more measurement time.
	Very high -> slow sweep	High accuracy/precision takes more measurement time.
Bandwidth Mode		Automatically is recommended in particular for logarithmic sweeps and assures the whole spectrum is covered.
	Auto	All bandwidth settings of the chosen demodulators are automatically adjusted. For logarithmic sweeps the measurement bandwidth is adjusted throughout the measurement.
	Fixed	Define a certain bandwidth which is taken for all chosen demodulators for the course of the measurement.
	Manual	The sweeper module leaves the demodulator bandwidth settings entirely untouched.

Control/ Tool	Option/ Range	Description
Time Constant/ Bandwidth Select		Defines the display unit of the low-pass filter to use for the sweep in fixed bandwidth mode: time constant (TC), noise equivalent power bandwidth (NEP), 3 dB bandwidth (3 dB).
	TC	Defines the low-pass filter characteristic using time constant of the filter.
	Bandwidth NEP	Defines the low-pass filter characteristic using the noise equivalent power bandwidth of the filter.
	Bandwidth 3 dB	Defines the low-pass filter characteristic using the cut-off frequency of the filter.
Time Constant/ Bandwidth	numeric value	Defines the measurement bandwidth for Fixed bandwidth sweep mode, and corresponds to either noise equivalent power bandwidth (NEP), time constant (TC) or 3 dB bandwidth (3 dB) depending on selection.
Order	numeric value	Selects the filter roll off to set on the device in Fixed and Auto bandwidth modes. It ranges from 1 (6 dB/octave) to 8 (48 dB/ octave).
Max Bandwidth (Hz)	numeric value	Maximal bandwidth used in auto bandwidth mode. The effective bandwidth will be calculated based on this max value, the frequency step size, and the omega suppression. The noise-equivalent power (NEP) is correctly taken into account for demodulation bandwidths of up to 1.25 MHz.
BW Overlap	ON / OFF	If enabled the bandwidth of a sweep point may overlap with the frequency of neighboring sweep points. The effective bandwidth is only limited by the maximal bandwidth setting and omega suppression. As a result, the bandwidth is independent of the number of sweep points. For frequency response analysis bandwidth overlap should be enabled to achieve maximal sweep speed.
Omega Suppression (dB)	numeric value	Suppression of the omega and 2-omega components. Large suppression will have a significant impact on sweep time especially for low filter orders.
Min Settling Time (s)	numeric value	Minimum wait time in seconds between a sweep parameter change and the recording of the next sweep point. This parameter can be used to define the required settling time of the experimental setup. The effective wait time is the maximum of this value and the demodulator filter settling time determined from the Inaccuracy value specified.
Inaccuracy	numeric value	Demodulator filter settling inaccuracy defining the wait time between a sweep parameter change and recording of the next sweep point. The settling time is calculated as the time required to attain the specified remaining proportion [1e-13, 0.1] of an incoming step function. Typical inaccuracy values: 10 m for highest sweep speed for large signals, 100 u for precise amplitude measurements, 100 n for precise noise measurements. Depending on the order the settling accuracy will define the number of filter time constants the sweeper has to wait. The maximum between this value and the settling time is taken as wait time until the next sweep point is recorded.
Settling Time (TC)	numeric value	Calculated wait time expressed in time constants defined by the specified filter settling inaccuracy.
Algorithm		Selects the measurement method.
	Averaging	Calculates the average on each data set.
	Standard Deviation	Calculates the standard deviation on each data set.
	Average Power	Calculates the electric power based on a 50 Ω input impedance.

Control/ Tool	Option/ Range	Description
Count (Sa)	integer number	Sets the number of data samples per sweeper parameter point that is considered in the measurement. The maximum between samples, time and number of time constants is taken as effective calculation time.
Count (s)	numeric value	Sets the time during which data samples are processed. The maximum between samples, time and number of time constants is taken as effective calculation time.
Count (TC)	0/5/15/50/100 TC	Sets the effective measurement time per sweeper parameter point that is considered in the measurement. The maximum between samples, time and number of time constants is taken as effective calculation time.
Phase Unwrap	ON / OFF	Allows for unwrapping of slowly changing phase evolutions around the +/-180 degree boundary.
Spectral Density	ON / OFF	Selects whether the result of the measurement is normalized versus the demodulation bandwidth.
Sinc Filter	ON / OFF	Enables sinc filter if sweep frequency is below 50 Hz. Will improve the sweep speed at low frequencies as omega components do not need to be suppressed by the normal low-pass filter.
AWG Control	ON / OFF	If enabled the sweeper starts automatically the AWG when a sweep is started. If sweeps are performed on nodes Index Sweep Triggers the AWG control is enabled automatically. Enable AWG control if some parameters should be recorded based on AWG generated signals.

Table 5.32: Sweeper tab: History sub-tab

Control/ Tool	Option/ Range	Description
History	History	Each entry in the list corresponds to a single trace in the history. The number of traces displayed in the plot is limited to 20. Use the toggle buttons to hide or show individual traces. Use the color picker to change the color of a trace in the plot. Double click on a list entry to edit its name.
Length	integer value	Maximum number of records in the history. The number of entries displayed in the list is limited to the 100 most recent ones.
Clear All	Clear All	Remove all records from the history list.
Clear	Clear	Remove selected records from the history list.
Load file		Load data from a file into the history. Loading does not change the data type and range displayed in the plot, this has to be adapted manually if data is not shown.
Name		Enter a name which is used as a folder name to save the history into. An additional three digit counter is added to the folder name to identify consecutive saves into the same folder name.
Auto Save		Activate autosaving. When activated, any measurements already in the history are saved. Each subsequent measurement is then also saved. The autosave directory is identified by the text "autosave" in the name, e.g. "sweep_autosave_001". If autosave is active during continuous running of the module, each successive measurement is saved to the same directory. For single shot operation, a new directory is created containing all measurements in the history. Depending on the file format, the measurements are either appended to the same file, or saved in individual files. For HDF5 and ZView formats, measurements are appended to the same file. For MATLAB and SXM formats, each measurement is saved in a separate file.
File Format		Select the file format in which to save the data.
Save	Save	Save the traces in the history to a file accessible in the File Manager tab. The file contains the signals in the Vertical Axis Groups of the Control sub-tab. The data that is saved depends on the selection from the pull-down list. Save All: All traces are saved. Save Sel: The selected traces are saved.

Control/ Tool	Option/ Range	Description
Reference	Reference	Use the selected trace as reference for all active traces.
Reference On	ON / OFF	Enable/disable the reference mode.
Reference name	name	Name of the reference trace used.

For the Math sub-tab please see the table "Plot math description" in the section called "Cursors and Math".

5.13. Arithmetic Unit Tab

The Arithmetic Unit (AU) tab allows the user to define arithmetic operations that are performed on demodulator data in real time. The results of the AUs can be provided to Auxiliary output connectors or to other functional units within the instrument. This functionality and tab is available on all UHF instruments.

5.13.1. Features

- Four arithmetic units, more than 50 input parameters
- Add and subtract demodulator samples (X, Y, R, θ) and Boxcar output samples
- Multiply and divide demodulator samples (X, Y, R, θ) and Boxcar output samples
- Calculate polar coordinates from arbitrary Cartesian demodulator outputs
- Fixed coefficients and auxiliary inputs as scaling factors
- Results available on auxiliary outputs and with that they can also be used as demodulator inputs
- Results available as PID input (requires UHF-PID option)
- Streaming to host computer

5.13.2. Description

The AU tab is the tool used to define and monitor mathematical operations on measurement data in real time. 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 5.33: App icon and short description

Control/Tool	Option/Range	Description
AU	× ÷ + −	Real-time arithmetic operations on demodulator outputs.

There are four expandable sections (see Figure 5.29), each corresponding to one arithmetic unit. Each unit operates independently and can be considered always ON, hence the defined operation is calculated all the time and the result is available to be used elsewhere in the system. Moreover, when streaming is enabled, the results can be transferred to the host computer, observed in the user interface, and stored to disk. A wide selection of input parameters including demodulator outputs and auxiliary inputs can be taken as operands.

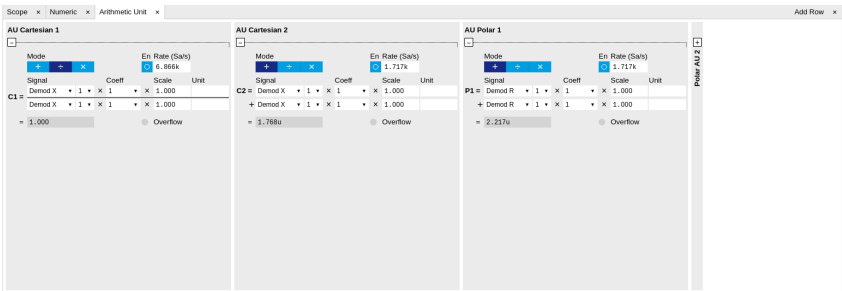


Figure 5.29: LabOne UI: Arithmetic unit tab

In total there are four units, two for Cartesian operations and two for polar operations. Each unit produces a scalar output along with a unit, both indicated in the last line. The Cartesian units can either add, multiply or divide two distinct X and Y values of all demodulators or alternatively the

output samples of either Boxcar unit. In addition scaling factors can be applied based on adjustable variables, derived from the auxiliary inputs or even the other Cartesian unit. The polar units can perform similar computations on demodulator magnitude (Demod R) and angle (Demod θ). In addition, the polar units can also operate on the magnitude and angle of a complex value computed from the two Cartesian units as C1 + iC2 (R(C1+iC2) or θ(C1+iC2), respectively). Each polar unit must operate entirely on either magnitude or angle values. Similarly to the Cartesian units, the magnitude and angle values can be multiplied with an adjustable variable, a value from one of the auxiliary inputs or even the result of the other Polar arithmetic unit.

5.13.3. Functional Elements

Table 5.34: Arithmetic unit tab

Control/ Tool	Option/ Range	Description
Mode		Selects the operation mode of the arithmetic unit
	Add	The arithmetic unit is in add mode: two independent demodulator outputs can be added together.
	Divide	The arithmetic unit is in divide mode: two independent demodulator outputs can be divided by each other.
	Multiply	The arithmetic unit is in multiply mode: two independent demodulator outputs can be multiplied with each other.
En	ON / OFF	Enables the streaming of arithmetic unit results to the host computer. The arithmetic unit is always operative, but streaming allows to use the results in other LabOne measurement and analysis tools.
Rate	0.2 to 1.75 MSa/s	Defines the number of arithmetic unit result samples that are sent to the host computer per second.
Signal		Select the arithmetic unit input signal
	Demod X	Use demodulator X (for Cartesian AU only).
	Demod Y	Use demodulator Y (for Cartesian AU only).
	Boxcar	Use Boxcar (for Cartesian AU only).
Channel	index	Select demodulator and/or Boxcar channel number.
Coeff		Select a coefficient to be applied to the selected Signal. Default: 1.
	1	A coefficient of 1 is used (default).
	Aux In 1	The signal on Aux In 1 is used as coefficient.
	Aux In 2	The signal on Aux In 2 is used as coefficient.
	C1	Output of Cartesian AU 1 (C1) is used as coefficient (for Cartesian AU only).
	C2	Output of Cartesian AU 2 (C2) is used as coefficient (for Cartesian AU only).
Signal		Select the arithmetic unit input signal
	Demod R	Use demodulator R (for polar AU only).
	Demod Θ	Use demodulator Θ (for polar AU only).
	R(C1 + iC2)	Use the magnitude of C1 + iC2 (for polar AU only).
	Θ(C1 + iC2)	Use the angle of C1 + iC2 (for polar AU only).
Channel	index	Select demodulator channel number.
Coeff		Select a coefficient to be applied to the selected Signal. Default: 1.
	1	A coefficient of 1 is used (default).
	Aux In 1	The signal on Aux In 1 is used as coefficient.
	Aux In 2	The signal on Aux In 2 is used as coefficient.
	P1	Output of Polar AU 1 (P1) is used as coefficient (for Polar AU only).
	P2	Output of Polar AU 2 (P2) is used as coefficient (for Polar AU only).

Control/ Tool	Option/ Range	Description
Scale	Real number	Custom scaling factor.
Unit	Text	Unit of "Scale", for example "m/V".
Result value	Real number	Shows the result of the arithmetic unit.
Result unit	Text	Shows the unit of the result of the arithmetic unit. If the unit formula is not valid, it will be indicated as #Invalid! and invalid formula can be corrected by adjusting scaling units.
Overflow	Text	When red, indicates that an overflow has occurred in the arithmetic unit.

5.14. Auxiliary Tab

The Auxiliary tab provides access to the settings of the Auxiliary Inputs and Auxiliary Outputs; it is available on all UHF Series instruments.

5.14.1. Features

- Monitor signal levels of auxiliary input connectors
- Monitor signal levels of auxiliary output connectors
- Auxiliary output signal sources: Demodulators, PIDs, Boxcars, AUs, AWG and manual setting
- Define Offsets and Scaling for auxiliary output values
- Control auxiliary output range limitations

5.14.2. Description

The Auxiliary tab serves mainly to monitor and control the auxiliary inputs and outputs. 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 5.35: App icon and short description

Control/ Tool	Option/ Range	Description
Aux		Controls all settings regarding the auxiliary inputs and auxiliary outputs.

The Auxiliary tab (see Figure 5.30) is divided into three sections. The Aux Input section gives two graphical and two numerical monitors for the signal amplitude applied to the auxiliary inputs on the back panel. In the middle of the tab the Aux Output section allows to associate any of the measured signals to one of the 4 auxiliary outputs on the instrument front panel. With the action buttons next to the Preoffset and Offset values the effective voltage on the auxiliary outputs can be automatically set to zero. The analog output voltages can be limited to a certain range in order to avoid damaging the parts connected to the outputs.

Note

Please note the change of units of the scaling factor depending on what measurement signal is chosen.

Two Aux Output Levels on the right provides 4 graphical and 4 numerical indicators to monitor the voltages currently set on the auxiliary outputs.

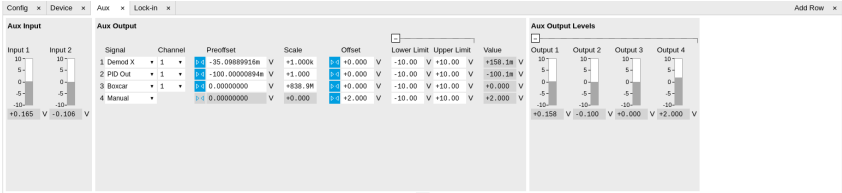


Figure 5.30: LabOne UI: Auxiliary tab

5.14.3. Functional Elements

Table 5.36: Auxiliary tab

Control/ Tool	Option/ Range	Description
Auxiliary Input Voltage	-10 V to 10 V	Voltage measured at the Auxiliary Input.
Signal		Select the signal source to be represented on the Auxiliary Output.
	X	Select the demodulator X component for auxiliary output.
	Y	Select the demodulator Y component for auxiliary output.
	R	Select the demodulator magnitude component for auxiliary output.
	Θ	Select the demodulator phase component for auxiliary output.
	PID Out	Select one of the PID controllers output. UHF-PID option needs to be installed.
	PID Shift	Select one of the PID controllers' shift signal. UHF-PID option needs to be installed.
	PID Error	Select one of the PID controllers' error signal. UHF-PID option needs to be installed.
	Boxcar	Select one of the two Boxcar units for auxiliary output. UHF-BOX option needs to be installed.
	AU Cartesian	Select one of the two Arithmetic Cartesian units for auxiliary output.
	AU Polar	Select one of the two Arithmetic Polar units for auxiliary output.
	AWG	Select one of the AWG Outputs for auxiliary output when running the AWG in four-channel mode. UHF-AWG option needs to be installed.
	CNT Out	Select one of the Pulse Counter signals for auxiliary output. UHF-CNT option needs to be installed.
	Manual	Manually define an auxiliary output voltage using the offset field.
Channel	index	Select the channel according to the selected signal source.
Preoffset	numerical value in signal units	Add a pre-offset to the signal before scaling is applied. Auxiliary Output Value = (Signal+Preoffset)*Scale + Offset
Auto-zero	▷◁	Automatically adjusts the Pre-offset to set the Auxiliary Output Value to zero.
Scale	numerical value	Multiplication factor to scale the signal. Auxiliary Output Value = (Signal+Preoffset)*Scale + Offset
Auto-zero	▷◁	Automatically adjusts the Offset to set the Auxiliary Output Value to zero.
Offset	numerical value in Volts	Add the specified offset voltage to the signal after scaling. Auxiliary Output Value = (Signal+Preoffset)*Scale + Offset
Lower Limit	-10 V to 10 V	Lower limit for the signal at the Auxiliary Output. A smaller value will be clipped.
Upper Limit	-10 V to 10 V	Upper limit for the signal at the Auxiliary Output. A larger value will be clipped.

Control/ Tool	Option/ Range	Description
Value	-10 V to 10 V	Voltage present on the Auxiliary Output port. The value is obtained by clamping the calculated signal (Signal+Preoffset)*Scale + Offset based on Lower and Upper Limits.

5.15. Inputs/Outputs Tab

The In / Out tab provides access to the settings of the Instrument's main Signal Inputs and Signal Outputs. It is available on all UHF Series instruments.

5.15.1. Features

- Signal input configuration
- Signal output configuration

5.15.2. Description

The In / Out tab gives access to the same settings as do the left-most and the right-most sections of the Lock-in tab. It is mainly intended to be used on small screens that can not show the entire the Lock-in tab at once. 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 5.37: App icon and short description

Control/ Tool	Option/ Range	Description
In/Out		Gives access to all controls relevant for the Signal Inputs and Signal Outputs of each channel.

The In / Out tab contains one section for the signal inputs and one for the signal outputs. All of the corresponding connectors are placed on the instrument front panel. The In / Out tab looks differently depending on whether the UHF-MF Multi-frequency option is installed or not.

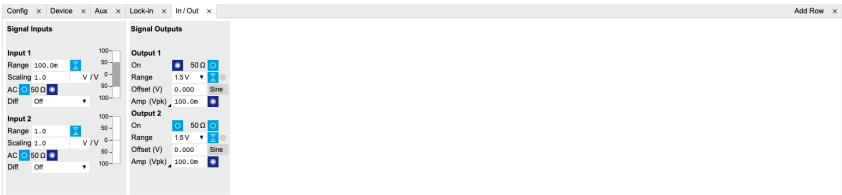


Figure 5.31: LabOne UI: Inputs/Outputs tab (base configuration)

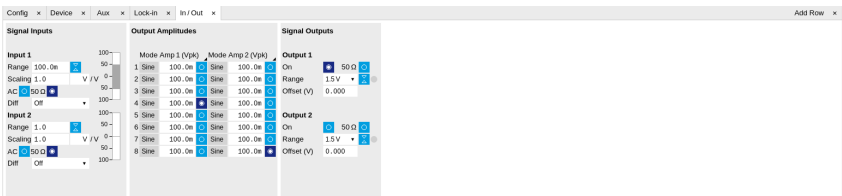


Figure 5.32: LabOne UI: Inputs/Outputs tab (with UHF-MF Multi-frequency option)

5.15.3. Functional Elements

All functional elements are equivalent to the ones on the Lock-in tab. See the Lock-in Tab or Lock-in MF Tab for a detailed description of the functional elements.

5.16. DIO Tab

The DIO tab provides access to the settings and controls of the digital I/O as well as the Trigger channels and is available on all UHF Series instruments.

5.16.1. Features

- Monitor and control of digital I/O connectors
- Control settings for external reference and triggering

5.16.2. Description

The DIO tab is the main panel to control the digital inputs and outputs as well as the trigger levels and external reference channels . 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 5.38: App icon and short description

Control/ Tool	Option/ Range	Description
DIO		Gives access to all controls relevant for the digital inputs and outputs including DIO, Trigger Inputs, Trigger Outputs, and Marker Outputs.

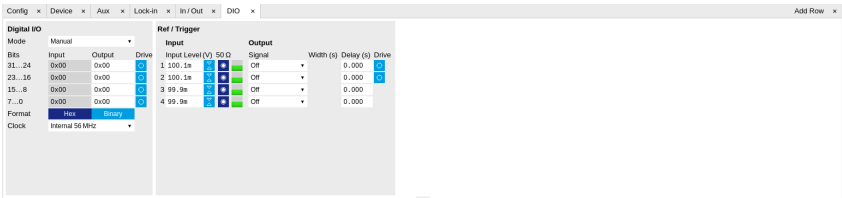


Figure 5.33: LabOne UI: DIO tab

The Digital I/O section provides numerical monitors to observe the states of the digital inputs and outputs. Moreover, with the values set in the Output column and the Drive button activated the states can also be actively set in different numerical formats.

The Ref/Trigger section shows the settings for the 6 reference and trigger inputs and outputs. The two BNC connectors on the front panel are numbered 1 and 2 and can act as inputs as well as outputs. The first two lines in this section are associated to these front panel connectors. On the back panel of the Instrument are 2 more trigger inputs (line 3 and 4, left columns) and 2 more trigger outputs (line 3 and 4, right columns). All four are SMA connectors.

Note

The Input Level determines the trigger threshold for trigger state discrimination. Also a 100 mV hysteresis is applied that cannot be adjusted such that a minimum amplitude of more than 100 mV is needed for the Trigger inputs to work reliably.

5.16.3. Functional Elements

Table 5.39: Digital input and output channels, reference and trigger

Control/ Tool	Option/Range	Description
DIO mode		Select DIO mode
	Manual	Enables manual control of the DIO output bits.
	AWG Sequencer	Enables setting of DIO output values by AWG sequencer commands.
	QA Results	Enables setting of DIO output values by QA results.

Control/ Tool	Option/Range	Description
	QA Results QCCS	Enables setting of DIO output values by QA results compatible with the QCCS.
QA Result Overflow	grey/yellow/red	Red: present overflow condition on the DIO interface during readout. Yellow: indicates an overflow occurred in the past. An overflow can happen if readouts are triggered faster than the maximum possible data-rate of the DIO interface.
DIO bits	label	Partitioning of the 32 bits of the DIO into 4 buses of 8 bits each. Each bus can be used as an input or output.
DIO input	numeric value in either Hex or Binary format	Current digital values at the DIO input port.
DIO output	numeric value in either hexadecimal or binary format	Digital output values. Enable drive to apply the signals to the output.
DIO drive	ON / OFF	When on, the corresponding 8-bit bus is in output mode. When off, it is in input mode.
Format		Select DIO view format.
	Hexadecimal	DIO view format is hexadecimal.
	Binary	DIO view format is binary.
Clock		Select DIO internal or external clocking.
	Internal 56 MHz	The DIO is internally clocked with a frequency of 56.25 MHz.
	Clk Pin 68	The DIO is externally clocked with a clock signal connected to DIO Pin 68. Available frequency range 1 Hz to 56.25 MHz.
	Internal 50 MHz	The DIO is internally clocked with a frequency of 50 MHz.
Trigger level	-5 V to 5 V	Trigger voltage level at which the trigger input toggles between low and high. Use 50% amplitude for digital input and consider the trigger hysteresis.
Auto Threshold	Press once	Automatically adjust the trigger threshold. The level is adjusted to fall in the center of the applied transitions.
50 Ω	50 Ω/1 kΩ	Trigger input impedance: When on, the trigger input impedance is 50 Ω, when off 1 kΩ.
Trigger Input Low status		Indicates the current low level trigger state.
	Off	A low state is not being triggered.
	On	A low state is being triggered.
Trigger Input High status		Indicates the current high level trigger state.
	Off	A high state is not being triggered.
	On	A high state is being triggered.
Trigger output signal		Select the signal assigned to the trigger output.
	Off	The output trigger is disabled.
	Osc Phase Demod 4/8	Oscillator phase of demod 4 (trigger output channel 1) or demod 8 (trigger output channel 2).Trigger event is output for each zero crossing of the oscillator phase.
	Scope Trigger	Trigger output is asserted when the scope trigger condition is satisfied.
	Scope /Trigger	Trigger output is deasserted when the scope trigger condition is satisfied.
	Scope Armed	Trigger output is asserted when the scope is waiting for the trigger condition to become satisfied.

Control/ Tool	Option/Range	Description
	Scope /Armed	Trigger output is deasserted when the scope is waiting for the trigger condition to become satisfied.
	Scope Active	Trigger output is asserted when the scope has triggered and is recording data.
	Scope /Active	Trigger output is deasserted when the scope has triggered and is recording data.
	AWG Marker 1	Trigger output is assigned to one of the AWG Marker channels attached to AWG waveform data.
	AWG Marker 2	Trigger output is assigned to one of the AWG Marker channels attached to AWG waveform data.
	AWG Marker 3	Trigger output is assigned to one of the AWG Marker channels attached to AWG waveform data.
	AWG Marker 4	Trigger output is assigned to one of the AWG Marker channels attached to AWG waveform data.
	AWG Active	Trigger output is asserted when the AWG is enabled.
	AWG Waiting	Trigger output is asserted when the AWG is waiting for external triggers, for a clock timer, or for other events.
	AWG Fetching	Trigger output is asserted when the AWG is fetching data from the main waveform and instruction memory.
	AWG Playing	Trigger output is asserted when the AWG is playing waveforms.
	AWG Trigger 1	Trigger output is assigned to one of the AWG Trigger channels controlled by AWG sequencer commands.
	AWG Trigger 2	Trigger output is assigned to one of the AWG Trigger channels controlled by AWG sequencer commands.
	AWG Trigger 3	Trigger output is assigned to one of the AWG Trigger channels controlled by AWG sequencer commands.
	AWG Trigger 4	Trigger output is assigned to one of the AWG Trigger channels controlled by AWG sequencer commands.
	MDS Clock Out	Trigger output is driven by the multi-device synchronisation clock.
	MDS Sync Out	Trigger output is driven by the multi-device synchronisation signal.
Delay (s)		This delay adds an offset that acts only on the trigger/marker output. The total delay to the trigger/marker output is the sum of this value and the value of the output delay node.
Width	0 s to 0.149 s	Defines the minimal pulse width for the case of Scope and AWG Trigger/Active events written to the trigger outputs of the device.
Delay	0 ns to 2.4 ns	Controls the delay of the Ref / Trigger output. The resolution is 78 ps.
Trigger drive	ON / OFF	When on, the bidirectional trigger on the front panel is in output mode. When off, the trigger is in input mode.

5.17. Config Tab

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

5.17.1. 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

5.17.2. 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 5.40: App icon and short description

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

The Config tab (see Figure 5.34) is divided into four sections to control connections, sessions, settings, user interface appearance and data recording.

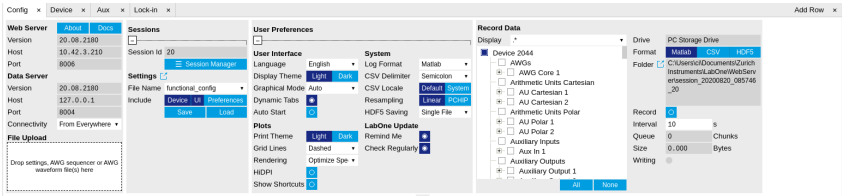


Figure 5.34: LabOne UI: Config tab

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 UHF Series instrument is used from the same computer account.

For low ambient light conditions the use of the dark display theme is recommended (see Figure 5.35).

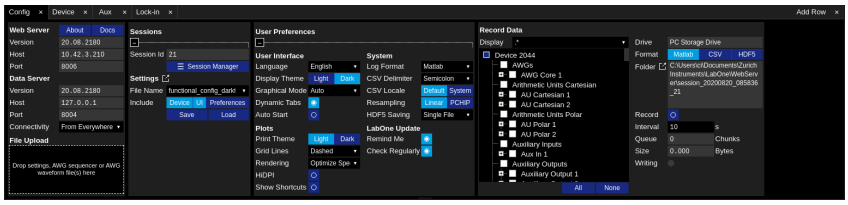


Figure 5.35: LabOne UI: Config tab - dark theme

The **Record Data** section contains all settings necessary to obtain hard copies of measurement data. The tree structure (see Tree Selector section) provides access to a large number of signals and instrument settings. Use the View Filter in order to reduce the tree structure to the most commonly used nodes such as the demodulator sample nodes. Whenever the Record button is enabled, all selected nodes get saved continuously in MATLAB, comma-separated value (CSV), or other supported file formats. For each selected node at least one file gets generated, but the data may be distributed over several files during long recordings. See Saving and Loading Data for more information on data saving. The quickest way to inspect the files after recording is to use the File Manager tab described in File Manager Tab. Apart from the numerical data and settings, the files contain timestamps. These integer numbers encode the measurement time in units of the instrument clock period 1/(1.8 GHz). The timestamps are universal within one instrument and can be used to align the data from different files.

5.17.3. Functional Elements

Table 5.41: 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		Enable Save/Load of particular settings.
	No Include Settings	Please enable settings type to be included during Save/ Load.
	Include Device	Enable Save/Load of Device settings.
	Include UI	Enable Save/Load of User Interface settings.
	Include UI and Device	Enable Save/Load of User Interface and Device settings.
	Include Preferences	Enable loading of User Preferences from settings file.
	Include UI, Device and Preferences	Enable Save/Load of User Interface, Device and User Preferences.
Save	Save	Save the user interface and device setting to a file.
Load	Load	Load the user interface and device setting from a file.
Language		Choose the language for the tooltips.
Display Theme	Dark	Choose theme of the user interface.

Control/ Tool	Option/Range	Description
	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	
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.

Control/ Tool	Option/Range	Description
	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.
	PC Storage Drive	Storage of the PC on which the LabOne Web Server is running.
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.

For more information on the tree functionality in the Record Data section, please see Tree Selector.

5.18. Device Tab

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

5.18.1. Features

- Option and upgrade management
- External clock referencing (10 MHz)
- Auto calibration settings
- Instrument connectivity parameters
- Device monitor

5.18.2. 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 5.42: App icon and short description

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

The Device tab (see Figure 5.36) is divided into five sections: general instrument information, configuration, communication parameters, device presets, and a device monitor.

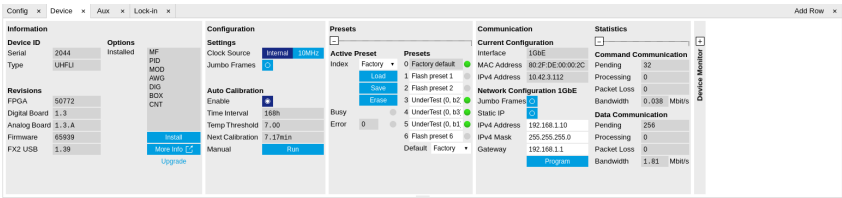


Figure 5.36: LabOne UI: Device tab

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 MHz reference. The reference is to be connected to the Clock Input on the instrument back panel.

The **Presets** section allows you to define a custom instrument start-up configuration different from the factory default. This configuration is stored in the instrument itself and are applied independently of the control PC. This saves time in cases where the control PC is not routinely needed, for instance when using only analog interfaces the instrument configuration is fixed.

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

Note

Activating Jumbo Frames is essential to achieve maximum data rates and also reduces load on the host PC.

The **Statistics** section gives an overview on communication statistics. In particular the current data rate (Bandwidth) that is consumed.

Note

Packet loss on data streaming over UDP or USB: data packets may be lost if total bandwidth exceeds the available physical interface bandwidth. Data may also be lost if the host computer is not able to handle high-bandwidth data. Network card setting optimization and Jumbo frame enabling may increase the maximal effective bandwidth.

Note

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

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

Self Calibration

Note

The calibration routine takes about 200 ms for that time the transfer of measurement data is stopped. That will lead to the following visible effects on the UI:

- missing data on the plotter
- the UI will shortly freeze
- the data loss flag will not report data loss (as the server intentionally trashed data)
- Sweeper, Data Acquisition tool and Scope will behave as usual and wait until they get data again
- The Spectrum tool will restart as it can only analyze continuously sampled data

Please see also additional remarks regarding calibration in specifications.

5.18.3. Functional Elements

Table 5.43: Device tab

Control/Tool	Option/Range	Description
Serial	1-4 digit number	Device serial number
Type	string	Device type
FPGA	integer number	HDL firmware revision.
Digital Board	version number	Hardware revision of the FPGA base board.
Analog Board	version indicator	Hardware revision of the analog board.
Firmware	integer number	Revision of the device internal controller software.
FX2 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.
Clock Source		10MHz reference clock source.
	Internal	The internal 10MHz clock is used as the frequency and time base reference.
	Clk 10MHz	An external 10MHz clock is intended to be used as the frequency and time base reference. Provide a clean and stable 10MHz reference to the appropriate back panel connector.

Control/Tool	Option/Range	Description
Jumbo Frames	ON / OFF	Enables jumbo frames (4k) on the TCP/IP interface. This will reduce the load on the PC and is required to achieve maximal throughput. Make sure that jumbo frames (4k) are enabled on the network card as well. If one of the devices on the network is not able to work with jumbo frames, the connection will fail.
Enabled	ON / OFF	Enables an automatic instrument self calibration about 16 min after start up. In order to guarantee the full specification, it is recommended to perform a self calibration after warm-up of the device.
Time interval	time	Time interval for which the self calibration is valid. After this time it is recommended to rerun the auto calibration. A LED indicator in the status bar indicates when another self calibration is recommended.
Calibration Temperature Threshold	temperature in °C	When the temperature changes by the specified amount, it is recommended to rerun the self calibration. A LED indicator in the status bar indicates when another self calibration is recommended.
Next calibration	time	Remaining time until the first calibration is executed or a recalibration is requested.
Manual self calibration	Run	Initiate self calibration to improve input digitizer linearity.
Index		Select between factory preset or presets stored in internal flash memory.
	Factory	Select factory preset.
	Flash 1-6	Select one of the presets stored in internal flash memory 1-6.
Load	Load	Load the selected preset.
Save	Save	Save the actual setting as preset.
Erase	Erase	Erase the selected preset.
Busy	grey/green	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.
Valid LED	grey/green	Turns green if a valid preset is stored at the respective location.
Presets		Shows a list of available presets including factory preset.
	0	Factory default preset. The name of the factory default preset is given and can not be edited.
	1	Flash preset 1. The name of this preset can be edited.
	2	Flash preset 2. The name of this preset can be edited.
	3	Flash preset 3. The name of this preset can be edited.
	4	Flash preset 4. The name of this preset can be edited.
	5	Flash preset 5. The name of this preset can be edited.
	6	Flash preset 6. The name of this preset can be edited.
Default		Indicates the preset which is used as default preset at start-up of the device.

Control/Tool	Option/Range	Description
	Factory	Select factory preset as default preset.
	Flash 1-6	Select one of the presets stored in internal flash memory 1-6 as default preset.
Interface	1. USB, 2. 1GbE	Active interface between device and data server. In case multiple options are available, the priority as indicated on the left applies.
MAC Address	80:2F:DE:xx:xx:xx	MAC address of the device. The MAC address is defined statically, cannot be changed and is unique for each device.
IPv4 Address	default 192.168.1.10	Current IP address of the device. This IP address is assigned dynamically by a DHCP server, defined statically, or is a fall-back IP address if the DHCP server could not be found (for point to point connections).
Jumbo Frames	ON / OFF	Enable jumbo frames for this device and interface as default.
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
Program	Program	Click to program the specified IPv4 address, IPv4 Mask and Gateway to the device.
Pending	integer value	Number of buffers ready for receiving command packets from the device.
Processing	integer value	Number of buffers being processed for command packets. Small values indicate proper performance. For a TCP/IP interface, command packets are sent using the TCP protocol.
Packet Loss	integer value	Number of command packets lost since device start. Command packets contain device settings that are sent to and received from the device.
Bandwidth	numeric value	Command streaming bandwidth usage on the physical network connection between device and data server.
Pending	integer value	Number of buffers ready for receiving data packets from the device.
Processing	integer value	Number of buffers being processed for data packets. Small values indicate proper performance. For a TCP/IP interface, data packets are sent using the UDP protocol.
Packet Loss	integer value	Number of data packets lost since device start. Data packets contain measurement data.
Bandwidth	numeric value	Data streaming bandwidth usage on the physical network connection between device and data server.
FW Load	numeric value	Indicates the CPU load on the processor where the firmware is running.

5.19. File Manager Tab

The File Manager tab provides a quick access to measurement files, log files and setting files in the local file system.

5.19.1. Features

- Quick access to measurement files, log files and settings files

File preview for settings files and log files

5.19.2. Description

The File Manager tab provides standard tools to see and organize the files relevant for the use of the instrument. Files can be conveniently copied, renamed and deleted. 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 5.44: App icon and short description

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

The Files tab (see Figure 5.37) provides three windows for exploring. The left window allows one to browse through the directory structure, the center window shows the files of the folder selected in the left window, and the right window displays the content of the file selected in the center window, e.g. a settings file or log file.

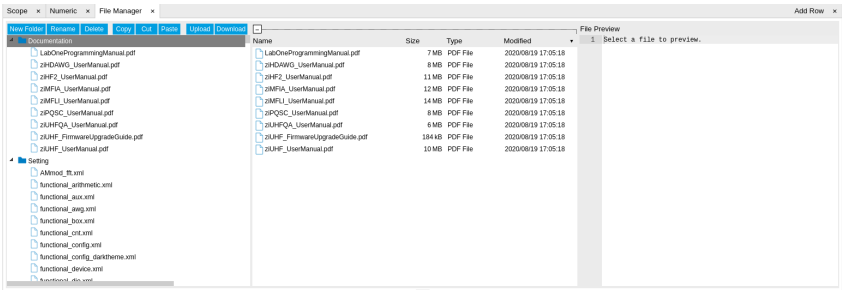


Figure 5.37: LabOne UI: File Manager tab

5.19.3. Functional Elements

Table 5.45: File tab

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

5.20. PID / PLL Tab

The PID / PLL tab is only available if the UHF-PID Quad PID/PLL Controller option is installed on the UHF Series Instrument (the installed options are displayed in the Device tab).

Note

The feedback controllers provide general-purpose PID functionality, phase-locked loop (PLL) functionality, and External Reference functionality. When the user sets one of the demodulators to ExtRef mode (see Lock-in tab, Demodulators section, Mode column), one of the PID controllers will be reserved for that purpose.

Note

Some settings in the PID / PLL tab are interdependent with settings that are accessible from other tabs. If the PID output controls a certain variable, e.g. Signal Output Offset, this variable will be shown as read-only where it appears in other tabs (i.e. in the Lock-in tab for this case).

5.20.1. Features

- Four fully programmable proportional, integral, derivative (PID) controllers
- Two fully programmable 600 MHz phased-locked loops
- PID / PLL Advisor with multiple DUT models, transfer function, and step function modeling
- Auto Tune: Automatic minimization of the amplitude of the PID error signal
- High speed operation with up to 300 kHz loop filter bandwidth
- Input parameters: demodulator data, auxiliary inputs, auxiliary outputs and arithmetic unit
- Output parameters: output amplitudes, oscillator frequencies, demodulator phase, auxiliary outputs, signal output offsets, and AWG output gain
- Phase unwrap for demodulator Θ data (± 1024 π), e.g. for optical phase-locked loops
- Bandwidth limit for the derivative (D) feedback component
- Programmable PLL center frequency and phase setpoint
- Programmable PLL phase detector filter settings
- Auto-zero functions for PLL center frequency and setpoint
- Generation of sub-multiple frequencies by use of harmonic multiplication factor

5.20.2. Description

The PID / PLL tab is the main control center for the feedback loop controllers in the instrument. 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 5.46: App icon and short description

Control/ Tool	Option/ Range	Description
PID		Features all control, analysis, and simulation capabilities of the PID controllers.

The PID / PLL tab (see LabOne UI: PID / PLL tab) consists of four identical side-tabs, each of them providing access to the functionality of one of the four PID / PLL controllers and the associated PID Advisor.

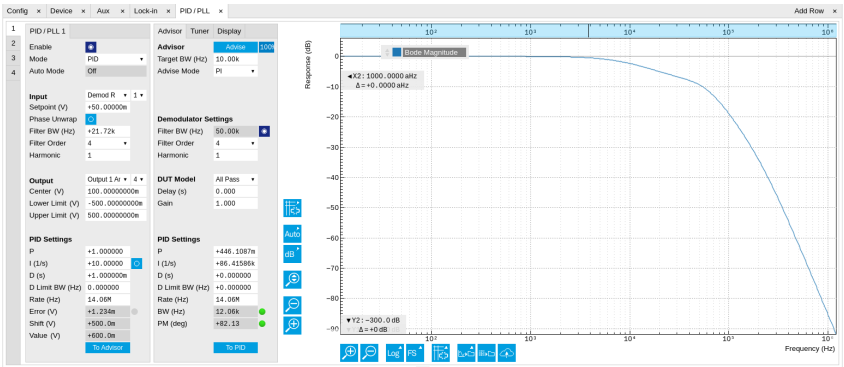


Figure 5.38: LabOne UI: PID / PLL tab

With their variety of different input and output connections, the LabOne PID controllers are extremely versatile and can be used in a wide range of different applications including laser locking

or high-speed SPM. Figure 5.39 shows a block diagram of all PID controller components, their interconnections and the variables to be specified by the user.

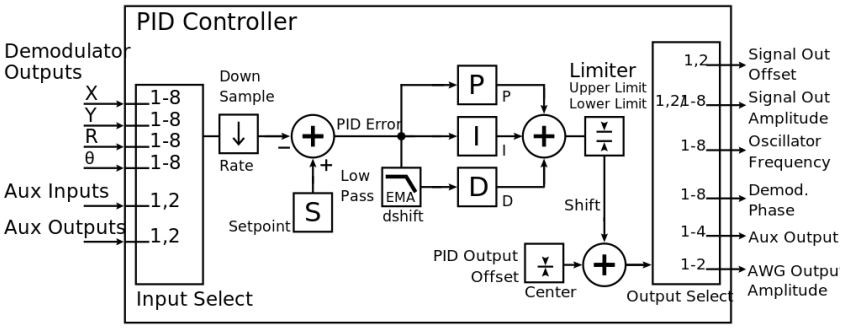


Figure 5.39: PID controller block diagram

Setting up a Control Loop

Depending on the application there are a number of ways to set up a control loop. Let's consider a few different approaches and see how the Advisor can help to reduce the effort and improve on the result and understanding of the setup.

Manual Setup

In cases where the transfer function of the device under test (DUT) is unknown and only little noise couples into the system from the environment, a manual approach is often the quickest way to get going. For manual configuration of a new control loop it is recommended to start with a small value for P and set the other parameters (I, D, D Limit) to zero. By enabling the controller one will then immediately see if the sign of P is correct and if the feedback is acting on the correct output parameter for instance by checking the numbers (Error, Shift, Out) displayed in the PID / PLL tab. A stepwise increase of the integral gain I will then help to zero the PID error signal completely. Enabling the derivative gain D can increase the speed of the feedback loop, but it can also cause an instable feedback loop behavior which sometimes can be mitigated by activating the associated low-pass filter . Monitoring the PID error in the Plotter Tab in parallel can be a great help at this stage. The math tools offered by the Plotter allow us to display the standard deviation and the average value of the error. These values should be minimized by tweaking the PID parameters and the associated histogram should have a symmetric (ideally Gaussian) envelope.

In order to characterize the feedback loop quantitatively, you can measure the step response in the Data Acquisition Tab. To do that measurement, change the PID setpoint manually after you have configured the DAQ Trigger level half way in between the old and new setpoint. DAQ Delay and Duration are chosen to roughly match the expected bandwidth. For a step response curve with fine time resolution, the PID data rate should be high enough.

PID Advisor

For many experimental situations the external device or DUT can be well approximated by a simple model. The LabOne PID Advisor allows you to simulate the behavior of a number of different DUT types in a feedback loop and choose feedback gain parameters based on the simulation. The DUTs are characterized by a model function with a number of parameters found on the Advisor sub-tab. All models include a setting for the delay that occurs outside the instrument. Depending on the targeted servo bandwidth, the external delay can often be the limiting factor and should be sensibly chosen.

Note

The delay specified for each model is the earliest possible response to a stepwise change of the instrument output to be seen on the instrument input. It describes the causality of the system and does not affect the shape of the DUT transfer function. Standard coaxial cables cause a signal delay of about 5 ns/m.

The most simple approach to modeling is to assume a DUT with a unity transfer function by using All Pass. The low-pass filters allow for limiting the bandwidth, to set an overall gain and a damping for the second order filter. With a Gain set to 1 and a Delay set to 0, All Pass can be used to model the PID controller independent of the external device. Resonator Frequency is a model that applies well in situations with a passive external component, e.g. a AFM cantilever or a quartz resonator, whose frequency should be tracked by a PLL over time. In cases where the amplitude of the resonator signal needs to be stabilized with a second control loop (automatic gain control), the Resonator Amplitude model is the right choice. Setting the resonance frequency and the Q factor, both can be obtained before by a frequency scan over the resonance using the Sweeper Tab, allows the Advisor to estimate the gain and low-pass behavior of the resonator. Internal PLL is used whenever an external oscillating signal is provided that shall be followed by one of the internal oscillators. The VCO setting describes a situation where the input variable of the DUT is a voltage and the output is a frequency. The gain parameter specifies how much voltage change on the input causes how much frequency shift on the VCO output. In case the frequency of the VCO can be tracked by using the external reference mode, one can easily measure this gain with the Sweeper Tab by scanning the Auxiliary Output voltage and displaying the resulting oscillator frequency. The gain is given by the slope of the resulting line at the frequency of interest.

With a model and parameters set to best describe the actual measurement situation, one can now continue by defining a target bandwidth for the entire control loop and the Advise Mode, i.e. the feedback gain parameters that shall be used for the control operation. Whenever the input signal is derived from one of the demodulators it is convenient to activate the box next to target bandwidth. With that in place the Advise algorithm will automatically adjust the demodulator bandwidth to a value about 5 times higher than the target bandwidth in order to avoid to be limited by demodulation speed. The Advisor algorithm will now calculate a target step response function that it will try to achieve by adjusting the feedback gain parameters in the next step. Before doing so in case of a newly set up DUT model, the algorithm will first try to estimate the PID parameters by using the Ziegler-Nichols method. When there has been a previous run, the user can also change the parameters in the model manually which will the be used as new start parameters of the next Advise run. Starting from the initial parameters, the Advisor will then perform a numerical optimization in order to achieve a least-squares fit of the calculated step response to a target step response determined from the Target Bandwidth. The result is numerically characterized by an achieved bandwidth (BW) and a phase margin (PM). Moreover, the large plot area on the right can be used to characterize the result by displaying transfer functions, magnitude and phase, and step responses between different signal nodes inside the loop. Once the modeling is completed one can copy the resulting parameters to the physical PID by clicking on To PID.

Table 5.47: DUT transfer functions

Name	Function	Parameters
All pass	$H(s) = g$	1. Gain g
Low-pass 1st	$H(s) = g\frac{1}{t_c s+1} = g\frac{\omega_n}{s+\omega_n}$	1. Gain g 2. Filter bandwidth (BW) $f_{-3dB} = \omega_n/2\pi$
Low-pass 2nd	$H(s) = g\frac{\omega_n^2}{s^2+2\omega_n\zeta s+\omega_n^2}$	1. Gain g 2. Resonance frequency $f_{res} = \omega_n/2\pi$ 3. Damping ratio ζ with $f_{-3dB} = 2\zeta f_{res}$
Resonator frequency	$H(s) = -360^\circ\frac{t_c}{t_c s+1}$ with $t_c = \frac{1}{2\pi BW} = \frac{2Q}{2\pi f_{res}}$	1. Resonance frequency f_{res} 2. Quality factor Q

Name	Function	Parameters
Resonator amplitude	$H(s) = g\frac{\omega/(2Q)}{s+\omega/(2Q)}$ with $\omega = 2\pi f_{res}$	1. Gain g 2. Resonance frequency f_{res} 3. Quality factor Q
Internal PLL	$H(s) = -\frac{360^\circ}{s}$	
VCO	$H(s) = g\frac{360^\circ}{s(t_c s+1)}$ with $t_c = \frac{1}{2\pi f_{res}}$	1. Gain g (Hz/V) 2. Bandwidth (BW) f_{-3dB}

Note

It is recommended to use the Advisor in a stepwise approach where one increases the free parameters from P to PI, to PID , and then to PIDF . This can save time because it prevents optimizing into local minima. Also it can be quite illustrative to see which of the feedback parameters leads to which effect in the feedback behavior.

Note

The low-pass filter in the differential part is implemented as an exponential moving average filter described by $y_t = (1-\alpha)\cdot y_{t-1} + \alpha x_t$ with $\alpha = 2^{-dshift}$, x_t the filter input, and y_t the filter output. The default value for dshift is 0 which corresponds to a disabled filter. On the UI the filter properties can be changed in units of bandwidth or a time constant.

In case the feedback output is a voltage applied to sensitive external equipment it is recommended to make use of the center value and the upper and lower limit values. This will guarantee that the output stays in the defined range even when the lock fails and the integrator goes into saturation.

Auto Tune

The Auto Tune feature found on the Tuner sub-tab can now help to minimize the residual error signal. Auto Tune will vary the feedback gain parameters, as selected in the Advise Mode field in the Advisor sub-tab, in order to minimize the root mean square of the PID error signal. Being based on measurement, Auto Tune can often improve on the results of the model-based PID Advisor because it can take into account the real experimental noise and device transfer function. For Auto Tune to deliver good results, it should be applied in the actual operating conditions of the PID loop, as otherwise the PID bandwidth may end up too low. E.g. it makes no sense to apply Auto Tune to a PLL on a lifted AFM cantilever when the PLL later is used to track the cantilever during scanning.

The transfer function of the chosen PID settings can always be checked by copying the values to the Advisor by clicking the To Advisor button and selecting the Advanced Mode in the Display sub-tab. With the Response In set to Setpoint, the Response Out set to PID Output and with Closed-Loop deactivated one can visualize the Bode Magnitude of the PID controller's transfer function. This graph is what is usually shown in textbooks and is independent of the model function chosen in the DUT section. However, in order to simulate step responses or to calculate a bandwidth, a suitable model for the entire loop is required.

Setting up a phase-locked loop (PLL)

The PID controllers 1 and 2 can be set to PLL mode which the Mode selector. Changing to PLL mode sets the PID controller input to a demodulator phase, the PID controller output to the frequency of an internal oscillator, and changes some of the parameters to appropriate default values. Figure 5.40 shows a block diagram of a PLL with its components, their interconnections and the variables to be specified by the user. The demodulator and the PID controller are slightly simplified for this sketch. Their full detailed block diagrams are given in Demodulator block diagram without UHF-MF Multi-

frequency option., Demodulator block diagram with UHF-MF Multi-frequency option., and Figure 5.39 respectively.

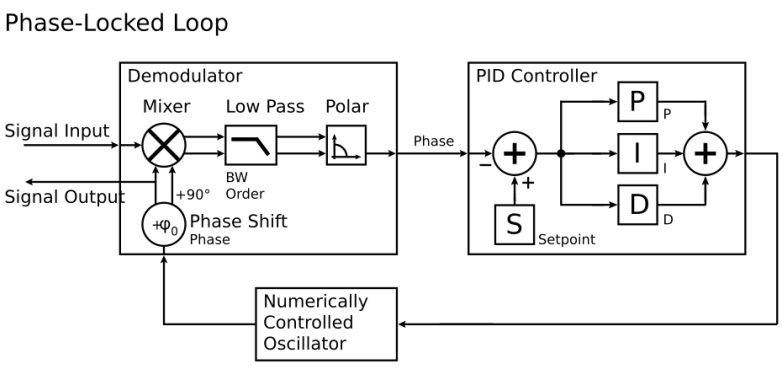


Figure 5.40: Phase-Locked Loop block diagram (components simplified)

In a typical procedure to set up a PLL one would first define the center frequency, frequency limits, and the phase setpoint in the left section. If the frequency is not known beforehand, it can often be determined using the Sweeper or Spectrum tool. Then one would set a target bandwidth in the Advisor section and click on the Advise button. The feedback parameters calculated by the Advisor will be shown in the fields just below. A graphical representation of the calculated transfer function is shown in the plot on the right-hand side. Once satisfied with the result, one can transfer the values to the instrument by clicking the To PID button, and then enable the PLL. If the Error/ PLL Lock field now displays very small values, the phase lock was successful. One can now iterate the process and e.g. play with the target bandwidth in the PLL Advisor to calculate a new set of feedback parameters. Displaying the oscillator frequency in the Plotter along with a Histogram and Math function (e.g. standard deviation) can help to quantify the residual phase error and further improve the lock performance by manual tweaking.

Note

The span set by the PLL frequency limits should exceed the target bandwidth by a factor of 5 to 10 or more.

Note

In the PID/PLL tab you select which of the demodulators you use as a phase detector. Open the Lock-in tab to check if the right Signal Input is associated with the demodulator in use.

5.20.3. Functional Elements

Table 5.48: PID tab: PID section

Control/ Tool	Option/Range	Description
Enable	ON / OFF	Enable the PID controller
Mode		Operation mode of the PID module.
	PID	The PID is used for a general application.
	PLL	The PID is used to control an internal oscillator.
	ExtRef	The PID is used by the external reference to control an internal oscillator.
Auto Mode		This defines the type of automatic adaptation of parameters in the PID.
	Off	No automatic adaptation.
	PID Coeff	The coefficients of the PID controller are automatically set.
	Coeff + BW (low)	The PID coefficients, the filter bandwidth and the output limits are automatically set using a low bandwidth.

Control/ Tool	Option/Range	Description
	Coeff + BW (high)	The PID coefficients, the filter bandwidth and the output limits are automatically set using a high bandwidth.
	Adaptive	All parameters of the PID including the center frequency are adapted.
Input		Select input source of PID controller
	Demodulator X	Demodulator cartesian X component
	Demodulator Y	Demodulator cartesian Y component
	Demodulator R	Demodulator magnitude component
	Demodulator Theta	Demodulator phase
	Aux Input	Auxiliary Input
	Aux Output	Internal value of Auxiliary Output
	Arithmetic Unit Cartesian	Cartesian output of arithmetic unit
	Arithmetic Unit Polar	Polar output of arithmetic unit
Input Channel	index	Select input channel of PID controller.
Setpoint	numeric value	PID controller setpoint
Phase Unwrap	ON / OFF	Enables the phase unwrapping to track phase errors past the +/-180 degree boundary and increase PLL bandwidth.
Filter BW	numeric value	Bandwidth of the demodulator filter used as an input.
Filter Order		Selects the filter roll off between 6 dB/oct and 48 dB/oct of the current demodulator.
	1	1st order filter 6 dB/oct
	2	2nd order filter 12 dB/oct
	3	3rd order filter 18 dB/oct
	4	4th order filter 24 dB/oct
	5	5th order filter 30 dB/oct
	6	6th order filter 36 dB/oct
	7	7th order filter 42 dB/oct
	8	8th order filter 48 dB/oct
Harmonic	1 to 1023	Multiplier of the for the reference frequency of the current demodulator.
Output		Select output of the PID controller
	Sig Out 1 Amplitude	Feedback to the main signal output amplitude 1
	Sig Out 2 Amplitude	Feedback to the main signal output amplitude 2
	Oscillator Frequency	Feedback to any of the internal oscillator frequencies
	Demodulator Phase	Feedback to any of the 8 demodulator phase set points
	Aux Output Offset	Feedback to any of the 4 Auxiliary Output's Offset
	Signal Output Offset	Feedback to the main Signal Output offset adjustment

Control/ Tool	Option/Range	Description
	AWG Output Gain	Feedback to the AWG Output Gain
Output Channel	index	Select output channel of PID controller.
Center	numeric value	After adding the Center value to the PID output, the signal is clamped to Center + Lower Limit and Center + Upper Limit.
Lower Limit	numeric value	After adding the Center value to the PID output, the signal is clamped between Center - Lower Limit, and Center + Upper Limit.
Upper Limit	numeric value	After adding the Center value to the PID output, the signal is clamped between Center - Lower Limit, and Center + Upper Limit.
P (Hz/deg)	numeric value	PID proportional gain P
I (Hz/deg/s)	numeric value	PID integral gain I
D (Hz/deg*s)	numeric value	PID derivative gain D
D Limit TC/ BW 3 dB	102 ns to 2.33 ms/68.3 Hz to 1.56 MHz	The cutoff of the low-pass filter for the D limitation, shown as either the filter time constant or the 3 dB cutoff frequency, depending on the selected TC mode. When set to 0, the low-pass filter is disabled.
Rate	109.9 kHz to 14 MHz	PID sampling rate and update rate of PID outputs. Needs to be set substantially higher than the targeted loop filter bandwidth. The numerical precision of the controller is influenced by the loop filter sampling rate. If the target bandwidth is below 1 kHz is starts to make sense to adjust this rate to a value of about 100 to 500 times the target bandwidth. If the rate is set too high for low bandwidth applications, integration inaccuracies can lead to non linear behavior.
Error	numeric value	Error = Set point - PID Input
Lock LED	grey/green	Indicates when the PID (configured as PLL) is locked. The PLL error is sampled at 5 Sa/s and its absolute value is calculated. If the result is smaller than 5 degrees the loop is considered locked. Only works if mode is PLL or ExtRef.
Shift	numeric value	Difference between the current output value Out and the Center. Shift = P*Error + I*Int(Error, dt) + D*dError/dt
Value	numeric value	Current output value
To Advisor	To Advisor	Copy the current PID settings to the PID Advisor.

Table 5.49: PID tab: Advisor sub-tab

Control/ Tool	Option/ Range	Description
Advise	Advise	Calculate the PID coefficients based on the used DUT model and the given target bandwidth. If optimized values can be found the coefficients are updated and the response curve is updated on the plot. Only PID coefficients specified with the advise mode are optimized. The Advise mode can be used incremental, means current coefficients are used as starting point for the optimization unless other model parameters are changed in-between.
Progress		The percentage of design algorithm already done when the Advisor is in progress.
Target BW (Hz)	numeric value	Target bandwidth for the closed loop feedback system which is used for the advising of the PID parameters. This bandwidth defines the trade-off between PID speed and noise.

Control/ Tool	Option/ Range	Description
Advise Mode		Select the PID coefficients that are optimized. The other PID coefficients remain unchanged but are used during optimization. This enables keeping selected coefficients at a fixed value while optimizing the rest. The advise time will increase significantly with the number of parameters to be optimized.
	P	Only optimize the proportional gain.
	I	Only optimize the integral gain.
	PI	Only optimize the proportional and the integral gain.
	PID	Optimize the proportional, integral, and derivative gains.
	PIDF	Optimize the proportional, integral, and derivative gains. Also the derivative gain bandwidth will be optimized.
Filter BW	numeric Value	Defines the low-pass filter characteristic of the selected demodulator input.
Auto Bandwidth	ON / OFF	Adjusts the demodulator bandwidth to fit best to the specified target bandwidth of the full system. If disabled, a demodulator bandwidth too close to the target bandwidth may cause overshoot and instability. In special cases the demodulator bandwidth can also be selected smaller than the target bandwidth.
Filter Order		Selects the filter roll off between 6 dB/oct and 48 dB/oct of the modelled demodulator.
	1	1st order filter 6 dB/oct
	2	2nd order filter 12 dB/oct
	3	3rd order filter 18 dB/oct
	4	4th order filter 24 dB/oct
	5	5th order filter 30 dB/oct
	6	6th order filter 36 dB/oct
	7	7th order filter 42 dB/oct
	8	8th order filter 48 dB/oct
Harmonic	1 to 1023	Multiplier of the for the reference frequency of the modelled demodulator.
DUT Model		Type of model used for the external device to be controlled by the PID. A detailed description of the transfer function for each model is found in the previous section.
	All Pass	The external device is modelled by an all pass filter. Parameters to be configured are delay and gain.
	LP 1st	The external device is modelled by a first-order low-pass filter. Parameters to be configured are delay, gain and filter bandwidth.
	LP 2nd	The external device is modelled by a second-order low-pass filter. Parameters to be configured are delay, gain, resonance frequency and damping ratio.
	Resonator Frequency	The external device is modelled by a resonator. Parameters to be configured are delay, center frequency and quality factor.
	Internal PLL	The DUT is the internal oscillator locked to an external signal through a phase-locked loop. The parameter to be configured is the delay.
	VCO	The external device is modelled by a voltage controlled oscillator. Parameters to be configured are delay, gain and bandwidth.

Control/ Tool	Option/ Range	Description
	Resonator Amplitude	The external device is modelled by a resonator. Parameters to be configured are delay, gain, center frequency and quality factor.
Delay	numeric value	Parameter that determines the earliest response for a step change. This parameter does not affect the shape of the DUT transfer function.
Gain	numeric value	Parameter that determines the gain of the DUT transfer function.
BW (Hz)	numeric value	Parameter that determines the bandwidth of the first-order low-pass filter respectively the bandwidth of the VCO.
Damping Ratio	numeric value	Parameter that determines the damping ratio of the second-order low-pass filter.
Res Freq	numeric value	Parameter that determines the resonance frequency of the of the modelled resonator.
Q	numeric value	Parameter that determines the quality factor of the modelled resonator.
P (Hz/deg)	numeric value	Proportional gain P coefficient used for calculation of the response of the PID model. The parameter can be optimized with PID advise or changed manually. The parameter only gets active on the PID after pressing the button To PLL.
I (Hz/deg/s)	numeric value	Integral gain I coefficient used for calculation of the response of the PID model. The parameter can be optimized with PID advise or changed manually. The parameter only gets active on the PID after pressing the button To PLL.
D (Hz/deg*s)	numeric value	Derivative gain D coefficient used for calculation of the response of the PID model. The parameter can be optimized with PID advise or changed manually. The parameter only gets active on the PID after pressing the button To PLL.
D Limit TC/ BW 3 dB	numeric value	The cutoff of the low-pass filter for the D limitation, shown as either the filter time constant or the 3 dB cutoff frequency, depending on the selected TC mode. When set to 0, the low-pass filter is disabled.
Rate	109.9 kHz to 14 MHz	PID sampling rate used for simulation. The advisor will update the rate to match with the specified target bandwidth. A sampling rate close to the target bandwidth and excessive higher bandwidth will results in a simulation mismatch.
BW (Hz)	numeric value	Simulated bandwidth of the full close loop model with the current PID settings. This value should be larger than the target bandwidth.
Target BW LED	green/red	Green indicates that the target bandwidth can be achieved. For very high PID bandwidth the target bandwidth might be only achieved using marginal stable PID settings. In this case, try to lower the bandwidth or optimize the loop delays of the PID system.
PM (deg)	numeric value	Simulated phase margin of the PID with the current settings. The phase margin should be greater than 45 deg for internal PLL and 60 deg for all other DUT for stable conditions. An Infinite value is shown if no unity gain crossing is available to determine a phase margin.
Stable LED	green/red	Green indicates that the phase margin is fulfilled and the PID system should be stable.
To PID	To PID	Copy the PID Advisor settings to the PID.

Table 5.50: PID tab: Tuner sub-tab

Control/ Tool	Option/ Range	Description
Auto Tune	Run/Stop	Optimize the PID parameters so that the noise of the closed-loop system gets minimized. The tuning method needs a proper starting point for optimization (away from the limits). The tuning process can be interrupted and restarted. The tuning will try to match the PID bandwidth with the loop bandwidth of the DUT, signal input (demodulator), and signal output.
Mode		Select the PID coefficients that are optimized. The other PID coefficients remain unchanged but are used during optimization. This allows to force coefficients to a value while optimizing the rest.
	P	Only optimize the proportional gain.
	I	Only optimize the integral gain.
	PI	Only optimize the proportional and the integral gain.
	PID	Optimize the proportional, integral, and derivative gains.
	PIDF	Optimize the proportional, integral, and derivative gains. Also the derivative gain bandwidth will be optimized.
Advanced Mode	ON / OFF	Enables manual selection of tuner averaging and setpoint toggling.
Iteration Time	numeric Value	Set the length for a tuner optimization iteration.

Table 5.51: PID tab: Display sub-tab

Control/ Tool	Option/ Range	Description
Advanced Mode	ON / OFF	Enables manual selection of display and advice properties. If disabled the display and advise settings are automatically with optimized default values.
Display		Select the display mode used for rendering the system frequency or time response.
	Bode Magnitude	Display the Bode magnitude plot.
	Bode Phase	Display the Bode phase plot.
	Step Resp	Display the step response plot.
Start (Hz)	numeric value	Start frequency for Bode plot display. For disabled advanced mode the start value is automatically derived from the system properties and the input field is read-only.
Stop (Hz)	numeric value	Stop frequency for Bode plot display. For disabled advanced mode the stop value is automatically derived from the system properties and the input field is read-only.
Start (s)	numeric value	Start time for step response display. For disabled advanced mode the start value is zero and the field is read-only.
Stop (s)	numeric value	Stop time for step response display. For disabled advanced mode the stop value is automatically derived from the system properties and the input field is read-only.
Transfer Function Selector		Selection of the displayed transfer function of the loop. 2 presets and a manual selection are possible. In closed loop configuration all elements from output to input will be included as feedback elements.
	System	From Setpoint to System Output.
	PID	From Setpoint to PID Output.
	Manual	Any transfer function in the open or closed loop can be visualized.

Control/ Tool	Option/ Range	Description
Response In		Start point for the plant response simulation for open or closed loops. In closed loop configuration all elements from output to input will be included as feedback elements.
	Demod Input	Start point is at the demodulator input.
	Setpoint	Start point is at the setpoint in front of the PID.
	PID Output	Start point is at PID output.
	Instrument Output	Start point is at the instrument output.
	DUT Output	Start point is at the DUT output and instrument input.
Response Out		End point for the plant response simulation for open or closed loops. In closed loop configuration all elements from output to input will be included as feedback elements.
	PID Output	End point is at PID output.
	Instrument Output	End point is at the instrument output.
	DUT Output	End point is at the DUT output and instrument input.
	Demod Input	End point is at the demodulator input.
	System Output	End point is at the output of the controlled system.
Closed-Loop	ON / OFF	Switch the display of the system response between closed or open loop.
TC Mode	ON / OFF	Enables time constant representation of PID parameters.
Set Limits	ON / OFF	Switch the writing of PID limits when 'To PID' is pressed. Only applies in case of internal PLL.
Advisor Link		Automatically copy cursor values displayed below to the PID advisor. To enable cursor helpers, switch Advanced Mode on and set Display to Bode Magnitude with PID Transfer Function. Cursors will be displayed in Log and dB axis scale combinations.
P		Cursor value representing PID proportional gain P. Drag the plot cursor with the mouse pointer or directly insert numerical value here.
I		Cursor value representing PID integral gain I. Drag the plot cursor with the mouse pointer or directly insert numerical value here.
D		Cursor value representing PID derivative gain D. Drag the plot cursor with the mouse pointer or directly insert numerical value here.
Max Rate (Sa/s)	1 to 14 MSa/s	Target Rate for PID output data sent to PC. This value defines the applied decimation for sending data to the PC. It does not affect any other place where PID data are used.
Data Rate (Sa/s)		Actual data transfer rate of the PID stream data sent to PC. It is defined based on Max Rate which can be adjusted in Display tab of PID Advisor.
Data Rate (Sa/s)		Actual data transfer rate of the PID stream data sent to PC. It is defined based on Max Rate.
Stream Overflow Indicator		Indicates an overflow in the stream.

5.21. MOD Tab

The MOD tab provides access to the settings of the amplitude and frequency modulation units. This tab is only available when the UHF-MOD AM/FM Modulation option is installed on the Instrument (see Information section in the Device tab).

Note

The UHF-MOD AM/FM Modulation option requires the UHF-MF Multi-frequency option.

5.21.1. Features

- Phase coherently add and subtract oscillator frequencies and their multiples
- Control for AM and FM demodulation
- Control for AM and narrow-band FM generation
- Direct analysis of higher order carrier frequencies and sidebands

5.21.2. Description

The MOD tab offers control in order to phase coherently add and subtract the frequencies of multiple numerical oscillators. 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 5.52: App icon and short description

Control/ Tool	Option/ Range	Description
MOD		Control panel to enable (de)modulation at linear combinations of oscillator frequencies.

The MOD tab (see Figure 5.41) is divided into two horizontal sections, one for each modulation unit.

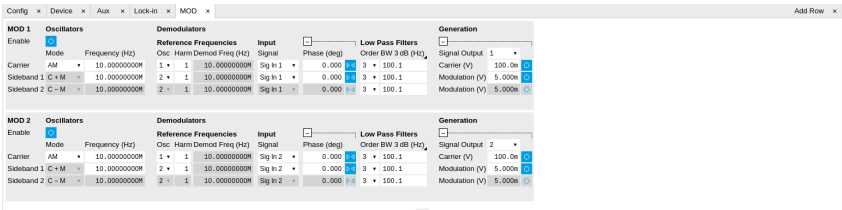


Figure 5.41: LabOne UI: MOD tab

The modulation units are designed for experiments involving multiple frequencies. For many of such experiments the associated spectrum reveals a dominant center frequency, often called the carrier, and one or multiple sidebands symmetrically placed around the carrier. Typical examples are amplitude modulated (AM) signals with one carrier and two sidebands separated from the carrier by the AM modulation frequency. Another example is frequency modulation (FM) where multiple sidebands to the left and right of the carrier can appear. The relative amplitude of the sideband for both AM and FM depends on the modulation depth, which is often expressed by the modulation index.

The classical approach of analyzing such signals (in particular when only analog instruments are available) is to use a configuration called tandem demodulation. This is essentially the serial cascading of lock-in amplifiers. The first device is referenced to the carrier frequency and outputs the in-phase component. This is then fed into the subsequent lock-in amplifiers in order to extract the different sideband components. There are several downsides to this scheme:

- The quadrature component of the first lock-in tuned to the carrier has to be continuously zeroed out by adjusting the reference phase. Otherwise a serious part of the signal power is lost for the analysis which usually leads to a drop in SNR.
- The scheme scales badly in terms of the hardware resources needed, in particular if multiple sideband frequencies need to be extracted.
- Every time a signal enters or exits an instrument the SNR gets smaller (e.g. due to the instrument inputs noise). Multiple such steps can deteriorate signal quality significantly.

All these shortcomings are nicely overcome by providing the ability to generate linear combinations of oscillator frequencies and use these combinations as demodulation references.

The MOD tab contains two sections MOD 1 and MOD 2. Both are identical in all aspects except that MOD 1 is linked to demodulators 1, 2 and 3, whereas MOD 2 is linked to demodulators 5, 6, and 7. Each of the MOD units can make use of up to 3 oscillators, which can be even referenced to an external source by using ExtRef or a PLL. Figure 5.42 gives an overview of the different components involved and their interconnections.

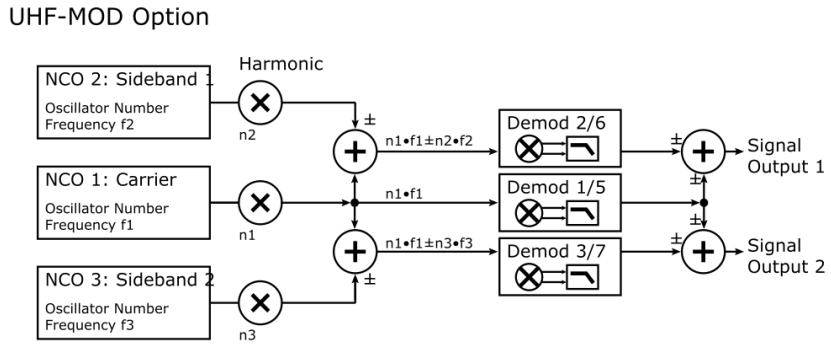


Figure 5.42: Modulation Option block diagram

For convenience the UI provides access to presets for AM and FM in the Mode column. In Manual Mode all settings can be chosen freely. When there are more than three frequencies present on a single signal one can even associate both sections MOD 1 and 2 to the same Signal Input.

Note

Whenever a MOD unit is enabled, all the settings in the Lock-in tab that are controlled by this unit will be set to read-only.

On top of signal analysis the UHF-MOD AM/FM Modulation option can also be utilized for signal generation. The Generation section provides all the necessary controls to adjust the carrier and sideband amplitudes.

Note

FM signals are generated by coherent superposition of the carrier signal with two sideband frequencies on either side that have the same amplitudes but opposite phases. The phase shift is achieved by using negative amplitudes as displayed in the Lock-in tab. This FM generation method approximates true FM as long as the modulation index is well below 1, i.e. higher-order sidebands can be neglected. For a modulation index of 1 true FM provides more than 13% of signal power in the second and higher order sidebands.

More details regarding AM and FM signal analysis and generation can be found on the Zurich Instruments web page.

5.21.3. Functional Elements

Table 5.53: MOD tab

Control/ Tool	Option/Range	Description
Enable	ON / OFF	Enable the modulation
Mode	AM/FM/manual	Select the modulation mode.
Mode		Enabling of the first sideband and selection of the position of the sideband relative to the carrier frequency for manual mode.
	Off	First sideband is disabled. The sideband demodulator behaves like a normal demodulator.
	C + M	First sideband to the right of the carrier
	C - M	First sideband to the left of the carrier
Mode		Enabling of the second sideband and selection of the position of the sideband relative to the carrier frequency for manual mode.

Control/ Tool	Option/Range	Description
	Off	Second sideband is disabled. The sideband demodulator behaves like a normal demodulator.
	C + M	Second sideband to the right of the carrier
	C - M	Second sideband to the left of the carrier
Frequency (Hz)	0 to 600 MHz	Sets the frequency of the carrier.
Frequency (Hz)	0 to 600 MHz	Frequency offset to the carrier from the first sideband.
Frequency (Hz)	0 to 600 MHz	Frequency offset to the carrier from the second sideband.
Carrier	oscillator index	Select the oscillator for the carrier signal.
Sideband 1	oscillator index	Select the oscillator for the first sideband.
Sideband 2	oscillator index	Select the oscillator for the second sideband.
Harm	1 to 1023	Set harmonic of the carrier frequency. 1=Fundamental
Harm	1 to 1023	Set harmonic of the first sideband frequency. 1 = fundamental
Harm	1 to 1023	Set harmonic of the second sideband frequency. 1 = fundamental
Demod Freq (Hz)	0 to 600 MHz	Carrier frequency used for the demodulation and signal generation on the carrier demodulator.
Demod Freq (Hz)	0 to 600 MHz	Absolute frequency used for demodulation and signal generation on the first sideband demodulator.
Demod Freq (Hz)	0 to 600 MHz	Absolute frequency used for demodulation and signal generation on the second sideband demodulator.
Channel	Signal Inputs, Trigger Inputs, Auxiliary Inputs, Auxiliary Outputs, Phase Demod 4, Phase Demod 8	Select Signal Input for the carrier demodulation
Channel	Signal Inputs, Trigger Inputs, Auxiliary Inputs, Auxiliary Outputs, Phase Demod 4, Phase Demod 8	Select Signal Input for the sideband demodulation
Phase	-180° to 180°	Phase shift applied to the reference input of the carrier demodulator and also to the carrier signal on the Signal Outputs
Phase	-180° to 180°	Phase shift applied to the reference input of the sideband demodulator and also to the sideband signal on the Signal Outputs
Zero	⋈	Adjust the carrier demodulator's reference phase automatically in order to read zero degrees at the demodulator output. This action maximizes the X output, zeros the Y output, zeros the Θ output, and leaves the R output unchanged.
Zero	⋈	Adjust the sideband demodulator's reference phase automatically in order to read zero degrees at the demodulator output. This action maximizes the X output, zeros the Y output, zeros the Θ output, and leaves the R output unchanged.
Order	1 to 8	Filter order used for carrier demodulation

Control/ Tool	Option/Range	Description
Order	1 to 8	Filter order used for sideband demodulation
TC/BW Value	numeric value	Defines the low-pass filter characteristic in the unit defined above for the carrier demodulation
TC/BW Value	numeric value	Defines the low-pass filter characteristic in the unit defined above for the sideband demodulation
Signal Output	1, 2 or both	Select Signal Output 1, 2 or none
Carrier (V)	-range to range	Set the carrier amplitude
Modulation (V)	-range to range	Set the amplitude of the first sideband component.
Modulation (V)	-range to range	Set the amplitude of the second sideband component.
Index	-range to range	In FM mode, set modulation index value. The modulation index equals peak deviation divided by modulation frequency.
Peak Dev (Hz)	-range to range	In FM mode, set peak deviation value.
Enable FM Peak Mode	ON / OFF	In FM mode, choose to work with either modulation index or peak deviation. The modulation index equals peak deviation divided by modulation frequency.
Enable	ON / OFF	Enable the signal generation for the first sideband
Enable	ON / OFF	Enable the signal generation for the second sideband
Enable	ON / OFF	Enable the carrier signal

5.22. Boxcar Tab

The Boxcar tab relates to the UHF-BOX Boxcar Averager option and is only available if this option is installed on the UHF Instrument (see Information section in the Device tab).

5.22.1. Features

- 2 equivalent boxcar units with up to 450 MHz repetition rate
- Baseline suppression for each Boxcar unit
- up to 450 MHz repetition rate
- Dead time free operation for frequencies below 450 MHz
- Period waveform analyzer (PWA) allows display of waveform and convenient graphical setting of Boxcar averaging windows
- PWA frequency domain view allows for simultaneous analysis of up to 512 harmonics of the reference frequency

5.22.2. Description

The Boxcar tab provides access to the gated averaging functionality of the UHF Instrument. 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 5.54: App icon and short description

Control/ Tool	Option/ Range	Description
Boxcar		Boxcar settings and periodic waveform analyzer for fast input signals.

Each Boxcar unit is shown in a separate sub-tab (see Figure 5.43) that consists of a plot area and three control tabs on the right-hand side.

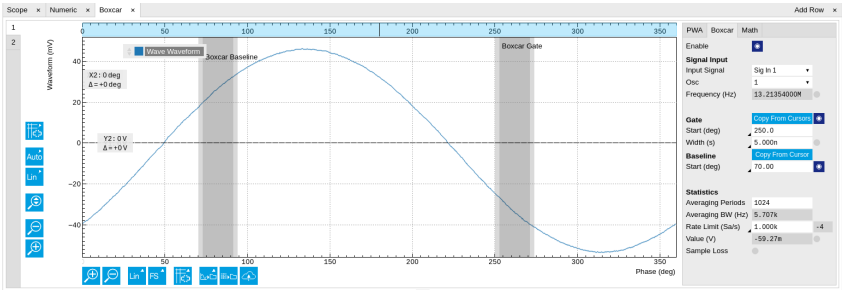


Figure 5.43: LabOne UI: Boxcar tab

Similar to the lock-in amplifier functionality the Boxcar offers a large reduction of the incoming signal bandwidth sampled with 1.8 GSa/s to a regime where much lower sampling rates suffice that can easily be transferred to a PC over USB or Ethernet cable for further analysis and post processing. For both methods ideally no piece of signal information is lost during the data reduction but huge parts if the initial signal are discarded that contain no or a negligible piece of relevant information. The operation of the lock-in amplifier can most easily be understood considering the inputs signal in the frequency domain where the lock-in acts as a sophisticated bandpass filter with adjustable center frequency and bandwidth (if we generously ignore phase sensitivity here for the sake of simplicity). In contrast, the Boxcar does a very similar thing in the time domain where it allows to cut out only the signal components that contain information. A very common use case are experiments with pulsed lasers. In particular when duty cycles are low, the fraction of the time domain signal where there is actual information can be quite small and so the idea is to record only the parts when for instance the laser is on.

In classical analog instruments this is typically realized by a switch, that can by triggered externally, and a subsequent integrator. Most often the trigger functionality also allows to configure a time delay and a certain window for as long as the switch shall open up for each trigger and the signal will be integrated. The signal output from the integrator is then passed through an adjustable low-pass filter for further noise reduction.

One of the biggest limitations of analog boxcar instruments is their trigger re-arm time (caused by the finite time required to erase the integrator) which is usually several 10 ms long. During that time no signals can be acquired. For periodic signals this means a limitation to frequencies of a few 10 kHz when signal loss cannot be afforded, measurement time needs to be minimized while high SNR is crucial.

Note

The Zurich Instruments Boxcar uses a synchronous detection approach instead of the traditional triggering method described above. A reference frequency has to be provided - either from external or an internal oscillator can be used - instead of a trigger signal and the Boxcar window is defined in terms of the phases of that reference frequency.

Note

Using a synchronous detection scheme in combination with a fixed input sampling rate of 1.8 GSa/s excludes all commensurate signal frequencies from proper analysis. The UI provides warnings whenever the reference frequency is anywhere close to any of these. Potential issues can be easily quantified by displaying the bin counts in the PWA sub-tab.

Figure 5.44 shows a detailed block diagram how signal processing is performed.

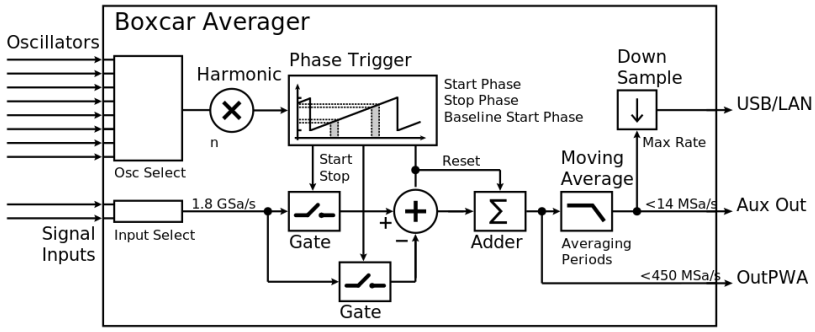


Figure 5.44: Boxcar averager block diagram

The input signal is sampled at a rate of 1.8 GSa/s. Depending on the phase of the reference oscillator and the set Start Phase and Window Width each of these samples is added up and output from the Adder after each period. From there one branch is directly connected to the outPWA (see Out PWA Tab) for a further step of synchronous detection. The other signal path way is subject to a Moving Average filter that allows to average over an adjustable number of reference oscillator periods.

Note

The moving average filter provides up to 512 intermittent results. That means if Averaging Periods is set to 1024 the Output is updated with a new value every second oscillator period whereas for smaller numbers of averaging Periods this update is performed on every cycle.

Another big advantage of the Zurich Instruments Boxcar is the graphical display of the input signal termed Periodic Waveform Analyzer. Each Boxcar unit is equipped with a PWA unit that can be either bound to the Boxcar settings or used on any other signal input and oscillator independently. Figure 5.45 shows a block diagram of the PWA.

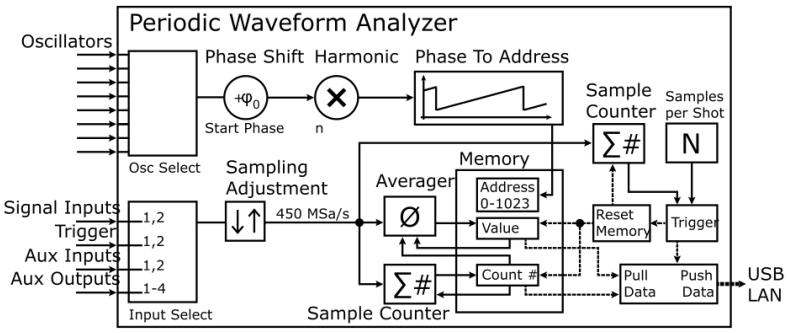


Figure 5.45: Periodic Waveform Analyzer block diagram

The user can select from a variety of different input signals, all of which will be re-sampled - either up or down, where no averaging is provided - at the input to a sampling rate of 450 MSa/s. Depending on the phase of the reference oscillator each data sample is associated to one of 1024 memory units which records the average values and the number of samples. These 1024 can be spread over the entire 360 degree of the reference oscillator period or a smaller span by using the Zoom mode. After an adjustable number of total input samples the entire memory is transferred to the PC and the memory is reset.

Each shot of data contains 1024 average values and sample counts each associated to a certain phase window. In case the reference frequency is sufficiently stable over the course of one shot it makes perfect sense to switch from the phase domain view to the time domain, which for some experiments might be the more natural way of consideration.

5.22.3. Functional Elements

Table 5.55: Boxcar tab: PWA sub-tab

Control/Tool	Option/Range	Description
Run/Stop	Run/Stop	Continuously run and stop PWA acquisition.
Single	Single	Single acquisition of a PWA data set.
Input Signal	Signal Inputs, Trigger Inputs, Auxiliary Inputs, Auxiliary Outputs, Phase Demod 4, Phase Demod 8	Select PWA input signal.
Input Interlock	ON / OFF	Interlock PWA and Boxcar Input settings
Osc	oscillator index	Select reference oscillator for PWA signal acquisition.
PWA Frequency	numeric value	Actual frequency at which the PWA operates based on set oscillator frequency and harmonic scaling factor.
Commensurability	grey/red	Traffic light showing whether the number of samples acquired is evenly distributed over all bins.
Mode	Harmonics (FFT)	Measurement data can be interpreted in four different modes and displayed over either phase (native), time, frequency (FFT) or harmonics of the base frequency (FFT).
	Phase	
	Time	
	Freq Domain (FFT)	
Copy from range	Copy From Range	Change PWA start and span according to plot range.
Reset	R	Reset the start and width value to show the full 360 deg.
Start	numeric value	Defines the start of PWA range in time or phase.
Width	numeric value	Defines width of PWA range in time or phase.
Samples	1 to 2^47	Defines the number of samples acquired of each PWA data set (450 MSa/s).
Acq Time (s)	numeric value	Estimated time needed for recording of the specified number of samples.
Overflow	grey/red	Indicates whether the number of samples collected per bin or the amplitude exceeds the numerical limit. Reduce number of samples and/or change frequency.
Infinite Acq Time	string	The signal source of this unit (Boxcar) is not producing any data. Once it is configured and enabled, this field will indicate the duration of a single measurement.
Progress (%)	0 to 100%	Show state of the PWA acquisition in percent.
3dB Bandwidth	numeric value	3dB bandwidth in Hz.
Max Harmonics	numeric value	Maximum number of displayed harmonics.
Signal	Count	Select signal to be displayed.
	Waveform	

Table 5.56: Boxcar tab: Boxcar sub-tab

Control/Tool	Option/ Range	Description
Enable	ON / OFF	Enable the BOXCAR unit
Input Signal	1/2	Select Signal Input used for the boxcar analysis.
Osc	oscillator index	Selection of the oscillator used for the boxcar analysis.

Control/Tool	Option/ Range	Description
Frequency (Hz)	frequency value	Oscillator frequency used for the boxcar analysis.
Too high frequency	grey/red	Frequency for the boxcar is above or equal 450 MHz. Sticky flag cleared by restarting the boxcar. The boxcar output may not be reliable any more.
Copy from cursors	Copy From Cursors	Take cursor values to define Window Start and Window span values.
Show Gate Opening	ON / OFF	Show gate opening on the PWA plot.
Start Mode		Selects the mode to specify the start of the boxcar averaging gate opening. The phase (deg) is the native mode for the device.
	Start (deg)	Native definition of the boxcar averaging gate start as phase.
	Start (s)	Definition of the boxcar averaging gate start as time. Due to the conversion to phase on the device a small uncertainty window exists.
Start (deg)	0 to 360	Boxcar averaging gate opening start in degrees. It can be converted to time assuming 360 equals to a full period of the driving oscillator.
Start Time (s)	0 to period	Boxcar averaging gate opening start in seconds based on one oscillator frequency period equals 360 degrees. Boxcar must be disabled to edit input field.
Width Mode		Selects the mode to specify the width of the boxcar averaging gate opening. The time (s) is the native mode for the device.
	Width (deg)	Definition of the averaging gate width as phase.
	Width (s)	Native definition of the averaging gate width as time.
	Width (pts)	Definition of the averaging gate width in samples.
Width (deg)	0 to 360	Boxcar averaging gate opening width in degrees based on one oscillator frequency period equals 360 degrees. Boxcar must be disabled to edit input field.
Width (s)	555 ps to period	Boxcar averaging gate opening width in seconds. It can be converted to phase assuming 360 equals to a full period of the driving oscillator.
Width (pts)	Integer value	Boxcar averaging gate opening width in samples at 1.8 GHz rate.
Too large gate width	grey/red	Boxcar averaging gate opening width is more than one cycle of the signal and should be reduced.
Copy from cursor	Copy from cursor	Take cursor value to define Baseline Start value.
Start Mode		Selects the mode to specify the start of the boxcar baseline suppression gate opening. The phase (deg) is the native mode for the device.
	Start (deg)	Native definition of the boxcar baseline suppression gate start as phase.
	Start (s)	Definition of the boxcar baseline suppression gate start as time.
	Offset (deg)	Definition of the boxcar baseline suppression gate start relative to the gate opening start as phase.
	Offset (s)	Definition of the boxcar baseline suppression gate start relative to the gate opening start as time.
Start (deg)	0 to 360	Boxcar baseline suppression gate opening start in degrees based on one oscillator frequency period equals 360 degrees.
Start (s)	0 to period	Boxcar baseline suppression gate opening start in seconds based on one oscillator frequency period equals 360 degrees.
Start (deg)	0 to 360	Boxcar baseline suppression gate opening start in degrees relative to Gate Start.

Control/Tool	Option/ Range	Description
Start (s)	0 to period	Boxcar baseline suppression gate opening start in seconds relative to Gate Start.
Enable	ON / OFF	Enable Baseline Suppression
Averaging Periods	1 to 2^20	Number of periods to average. The output will be refreshed up to 512 times during the specified number of periods. This setting has no effect on Output PWAs.
Averaging BW	10 µHz to 7 MHz	The 3 dB signal bandwidth of the Boxcar Averager is determined by the oscillation frequency and the Number of Averaging Periods set. Note: internally the boxcar signal is sampled at a rate of 14 MSa/s and the signal bandwidth of the auxiliary output is 7 MHz.
Rate Limit (Sa/s)	1 to 14.06 MSa/s	Rate Limit for Boxcar output data sent to PC. This value does not affect the Aux Output for which the effective rate is given by min(14 MSa/s , Frequency / max(1, Averaging Periods/512)).
Rate (Sa/s)	1 to 14.06 MSa/s	Display of the currently effective rate used for data transfer to the PC given by min(14 MSa/s , Frequency / max(1, Averaging Periods/512)). This value is read-only.
Rate Limit (Sa/s) or Rate (Sa/s)		Switches between display of Rate Limit or Rate
	Rate Limit (Sa/s)	Display of the Rate Limit which defines the maximal transfer rate.
	Rate (Sa/s)	Display of the currently active transfer rate.
Oversampling	Integer value, ideally 0	Indicates, in powers of 2, the number of averager outputs sent to the PC while Averaging Periods Boxcar integrations are obtained. Positive integer values indicate oversampling. Negative integer values indicate undersampling. Examples for oversampling values: 0 : 2^0 = 1 averager output is sent to the PC during Averaging Periods Boxcar integrations. 2 : 2^2 = 4 averager outputs are sent to the PC during Averaging Periods Boxcar integrations. -1 : 2^-1 = 0.5, only every other Averaging Periods Boxcar integrations an averager output is sent to the PC.
Value	numeric value	The current boxcar output.
Value Overflow flag	grey/red	Overflow detected. Sticky flag cleared by restarting the boxcar. The boxcar output may not be reliable any more.
Sample Loss	grey/red	Data lost during streaming to PC. Sticky flag cleared by restarting the boxcar.

For the Math sub-tab please see Plot math description in Cursors and Math.

5.23. Out PWA Tab

The Out PWA tab relates to the UHF-BOX Boxcar Averager option and is only available if this option is installed on the UHF Instrument (see Information section in the Device tab).

5.23.1. Features

- Period waveform analyzer for boxcar output samples (multi-channel boxcar, deconvolution boxcar)
- Support signals derived from asynchronous optical sampling

5.23.2. Description

The Out PWA tab provides access to the period waveform analyzer that acts on boxcar output samples. This feature is also called multi-channel boxcar or deconvolution boxcar. 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 5.57: App icon and short description

Control/ Tool	Option/ Range	Description
Out PWA		Multi-channel boxcar settings and measurement analysis for boxcar outputs.

The Out PWA tab (see Figure 5.46) consists of a display section on the left and a configuration section on the right. The configuration section is further divided into a number of sub-tabs.

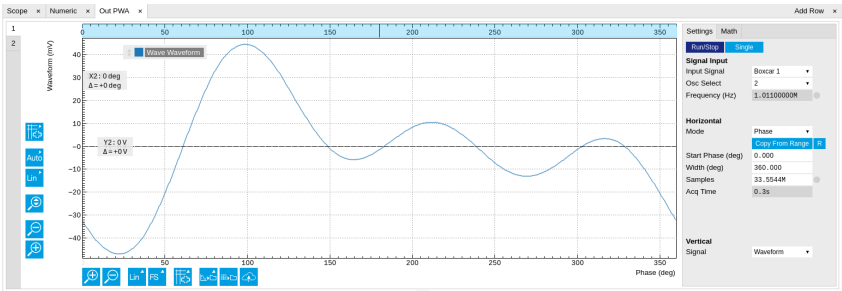


Figure 5.46: LabOne UI: Out PWA tab

The operation of the Output PWA is described in Multi-channel Boxcar Averager in the Tutorials chapter. The Output PWA works analogously to the PWA supplied in the Boxcar tabs (see boxcar tab) except that its inputs are given by the outputs of the two Boxcar units, rather than the Signal Inputs. The Boxcar output is routed to the input of the Output PWA without averaging, which means that the Averaging Periods setting made in the Boxcar tab is irrelevant for the measurement with the Output PWA.

5.23.3. Functional Elements

Table 5.58: Out PWA tab: Settings sub-tab

Control/Tool	Option/ Range	Description
Run/Stop	Run/Stop	Continuously run and stop PWA acquisition.
Single	Single	Single acquisition of a PWA data set.
Input Signal	Boxcar 2	Select PWA input signal.
	Boxcar 1	
Osc Select	oscillator index	Select reference oscillator for PWA signal acquisition.
Frequency	numeric value	Actual frequency at which the PWA operates based on set oscillator frequency and harmonic scaling factor.
Commensurability	grey/red	Traffic light showing whether the number of samples acquired is evenly distributed over all bins.
Mode	Harmonics (FFT)	Measurement data can be interpreted in four different modes and displayed over either phase (native), time, frequency (FFT) or harmonics of the base frequency (FFT).
	Phase	
	Time	
	Freq Domain (FFT)	
Copy from range	Copy From Range	Change PWA start and span according to plot range.
Reset	R	Reset the start and width value to show the full 360 deg.
Start	numeric value	Defines the start of PWA range in time or phase.
Width	numeric value	Defines width of PWA range in time or phase.
Samples	1 to 2^47	Defines the number of samples acquired of each PWA data set (450 MSa/s).

Control/Tool	Option/ Range	Description
Acq Time (s)	numeric value	Estimated time needed for recording of the specified number of samples.
Overflow	grey/red	Indicates whether the number of samples collected per bin or the amplitude exceeds the numerical limit. Reduce number of samples and/or change frequency.
Infinite Acq Time	string	The signal source of this unit (Boxcar) is not producing any data. Once it is configured and enabled, this field will indicate the duration of a single measurement.
Progress (%)	0 to 100%	Show state of the PWA acquisition in percent.
3dB Bandwidth	numeric value	3dB bandwidth in Hz.
Max Harmonics	numeric value	Maximum number of displayed harmonics.
Signal	Count	Select signal to be displayed.
	Waveform	

For the Math sub-tab please see Plot math description in Cursors and Math.

5.24. AWG Tab

The AWG tab is available on UHFAWG Arbitrary Waveform Generator instruments and on UHFLI Lock-in Amplifier instruments with installed UHF-AWG Arbitrary Waveform Generator option (see Information section in the Device tab).

5.24.1. Features

- Dual-channel arbitrary waveform generator
- 128 MSa waveform memory per channel
- Sequence branching
- Digital modulation
- Multi-instrument synchronization
- Sequence program distribution over multiple instruments
- Cross-domain trigger engine
- Sequence Editor with code highlighting and auto completion
- High-level programming language with waveform generation and editing toolset
- Waveform viewer

5.24.2. Description

The AWG tab gives access to the arbitrary waveform generator functionality. 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 5.59: App icon and short description

Control/ Tool	Option/ Range	Description
AWG		Generate arbitrary signals using sequencing and sample-by-sample definition of waveforms.

The AWG tab (see Figure 5.47) consists of a settings section on the right side and the Sequence and Waveform Viewer sub-tabs on the left side. The settings section is further divided into Control, Waveform, Trigger, and Advanced sub-tabs. The **Sequence** sub-tab is used for displaying, editing and compiling a LabOne sequence program. The sequence program defines which waveforms are played and in which order. The Sequence Editor is the main tool for operating the AWG.

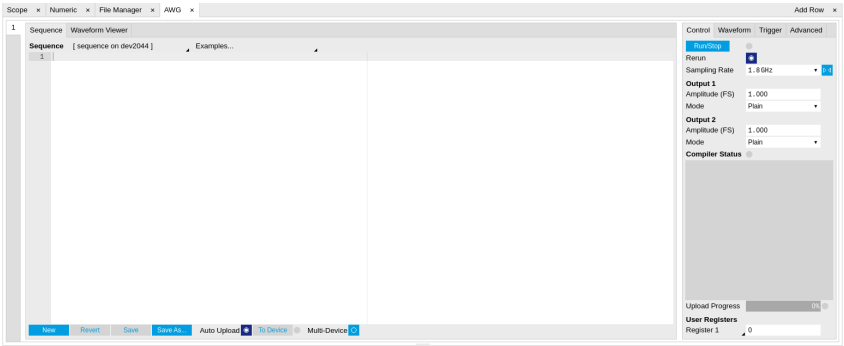


Figure 5.47: LabOne UI: AWG tab

A number of sequence programming examples are available through a drop-down menu at the top of the Sequence Editor, and additional ones can be found in Arbitrary Waveform Generator . The LabOne sequence programming language is specified in detail in LabOne Sequence Programming. The language comes with a number of predefined waveforms, such as Gaussian, Blackman, sine, or square functions. By combining those predefined waveforms using the waveform editing tools (add, multiply, cut, concatenate, etc), signals with a high level of complexity can be generated directly from the Sequence Editor window. Sample-by-sample definition of the output signal is possible by using comma-separated value (CSV) files specified by the user , see Waveform Generation and Playbackfor an example.

The AWG features a compiler which translates the high-level sequence program into machine instructions and waveform data to be stored in the instrument memory as shown in Figure 5.48. The sequence program is written using high-level control structures and syntax that are inspired by human language, whereas machine instructions reflect exactly what happens on the hardware level. Writing the sequence program using a high-level language represents a more natural and efficient way of working in comparison to writing lists of machine instructions, which is the traditional way of programming AWGs. Concretely, the improvements rely on features such as:

- combination of waveform generation, editing, and playback sequence in a single script
- easily readable syntax and naming for run-time variables and constants
- optimized waveform memory management, reduced transfers upon waveform changes
- definition of user functions and procedures for advanced structuring
- syntax validation

By design, there is no one-to-one link between the list of statements in the high-level language and the list of instructions executed by the Sequencer. There are cases in which a more detailed understanding of the Sequencer instruction list, and in particular its execution timing, is needed. Typically this is the case when observing delays or other signal timing properties that are unexpected from looking at the high-level script. Often such problems can be solved with a few adjustments to the program. Please see Debugging Sequencer Programs for practical advice.

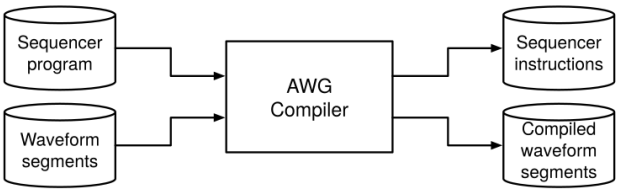


Figure 5.48: AWG sequence program compilation process

The **Sequence Editor** provides the editing, compilation, and transfer functionality for sequence programs. A program typed into the Editor is compiled upon clicking Save. If the compilation is successful and Automatic Upload is enabled, the program including all necessary waveform data is transferred to the device. If the compilation fails, the Status field will display debug messages. Clicking on Save as... allows you to choose a new name for the program. The name of the program that is currently edited is displayed at the top of the editor. External program files as well as waveform data files can be transferred to the right location easily using the file drag-and-drop zone in the Config Tab so they become accessible from the user interface. The files can be managed in the File Manager Tab and their location in the directory structure is shown in Table 5.60. The program name is displayed in a drop-down box. The box allows quick access to all programs in the standard sequence program location. It is possible to quickly switch between programs using the box. Changes made in one program will be preserved when switching to a different program. The file name of a program will be postfixed by an asterisk in case there are unsaved changes in the source file. Note that switching programs in the editor is not sufficient to also update the program in the instrument. In order to send a newly selected program to the instrument, the To Device button must be clicked.

Table 5.60: Sequence program and waveform file location

File type	Location
Waveform files (Windows)	`C:\Users\<user name>\Documents\Zurich Instruments\LabOne\WebServer\awg\waves`
Sequence programs (Windows)	`C:\Users\<user name>\Documents\Zurich Instruments\LabOne\WebServer\awg\src`
Waveform files (Linux)	`~/Zurich Instruments/LabOne/WebServer/awg/waves`
Sequence programs (Linux)	`~/Zurich Instruments/LabOne/WebServer/awg/src`

In the **Control sub-tab** the user configures signal parameters and controls the execution of the AWG. The AWG can be started in by clicking on Start. When enabling the Rerun button, the Sequencer will be restarted automatically when its program completes. The continuous mode is a simple way to create an infinite loop, but it results in a considerable timing jitter. To avoid this jitter, it is recommended to specify infinite loops directly in the sequence program.

The Sampling Rate field is used to control the default playback sampling rate of the AWG. The sampling rate is dynamic, i.e., can be specified for each waveform by using an optional argument in the waveform playback instructions in the sequence program. This allows for considerably reducing waveform upload time for signals that contain both fast and slow components. The two Output sections are used to configure the AWG output mode and signal amplitude. The AWG output channels are not the same as the physical Signal Outputs of the instrument. The AWG output channels are routed to the Signal Outputs of the device. The Amplitude value is a gain parameter, 1.0 by default, that is applied to waveforms on the way from the AWG output channel to the Signal Output. The Amplitude value gives a means to rescale the signal independently of the programmed waveforms. The Mode control is used to enable or disable the modulation mode , or to enable advanced modulation mode . With enabled modulation, the signal of an AWG Output is multiplied with an oscillator signal prior to being sent to the Signal Output. This is useful for the case where the desired signal can be described as a sinusoidal carrier with a shaped envelope. The advanced modulation mode allows you to modulate multiple carriers (up to 4) with individual envelope waveforms. Please read more about use cases, advantages, and practical examples in Modulation Mode.

The generation of the modulated signal depends on the settings made in the Lock-in tab. Figure 5.49 shows how the signals are routed internally on their way from the oscillators and the AWG to the Signal Outputs. There are two switches in the diagram. The upper switch is related to the AWG Mode selection. In Modulation mode, the signal coming from the AWG unit 1 (2) is multiplied with the oscillator signal of demodulator 4 (8). The phase and harmonic of the oscillator signal can be adjusted in the Lock-in tab. In Direct mode, the AWG signal is multiplied with a constant 1.0, in other words, it remains unchanged. The lower switch is related to the running state of the AWG, i.e, the Single buttons. When the AWG is idle, the Output Amplitude setting from the Lock-in tab takes the place of the AWG signal. This is the standard configuration for lock-in measurements. It is furthermore useful for defining the voltage appearing on the Signal Output when the AWG is off. The UHF-MF Multi-frequency option provides additional oscillators as well as an Oscillator Select switch matrix at the input of the demodulators, enabling the use of up to 8 independent frequencies for modulation. The Register values may be used as integer variables inside a sequence program, for instance to vary a delay between pulses manually or with the Sweeper Tab.

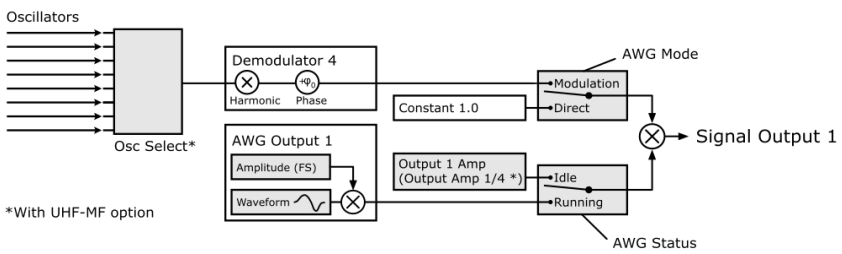


Figure 5.49: Amplitude modulation block diagram for AWG Output 1

The **Waveform sub-tab** displays information about the waveforms that are used by the current sequence program, such as their length and channel number. Together with the **Waveform viewer sub-tab**, it is a useful tool to visualize the waveforms used in the sequence program.

On the **Trigger sub-tab** you can configure the trigger inputs of the AWG and control the Cross-Domain Trigger functionality of the instrument. The AWG has four trigger input channels which can be configured to probe a variety of signals coming both from internal (e.g. demodulator output data) or external (e.g. Ref/Trigger input) sources . This means that the AWG trigger input channels are not the same as physical device inputs. Two of the trigger input channels are called analog (meaning

they can accept signals of continuous, analog-like character), and two are called digital (meaning they can accept binary signals). Trigger Level and Hysteresis may be configured for the Analog Triggers, and the user can select between rising and falling edge trigger functionality. The primary use of the triggers is to control the timing of the AWG signal relative to an external device. Another use of triggers is to implement sequence branching. See Triggering and Synchronization and Branching and Feed-Forward for practical examples on how to use the AWG trigger in- and outputs.

The **Advanced sub-tab** displays the compiled list of sequencer instructions and the current state of the sequencer on the instrument. This can help an advanced user in debugging a sequence program and understanding its execution.

Sequence Editor Keyboard Shortcuts

The tables below list a number of helpful keyboard shortcuts that are applicable in the LabOne Sequence Editor.

Table 5.61: Line Operations

Shortcut	Action
`Ctrl+D`	Remove line
`Alt+Shift+Down`	Copy lines down
`Alt+Shift+Up`	Copy lines up
`Alt+Down`	Move lines down
`Alt+Up`	Move lines up
`Alt+Del`	Remove to line end
`Alt+Backspace`	Remove to line start
`Ctrl+Backspace`	Remove word left
`Ctrl+Del`	Remove word right

Table 5.62: Selection

Shortcut	Action
`Ctrl+A`	Select all
`Shift+Left`	Select left
`Shift+Right`	Select right
`Ctrl+Shift+Left`	Select word left
`Ctrl+Shift+Right`	Select word right
`Shift+Home`	Select line start
`Shift+End`	Select line end
`Alt+Shift+Right`	Select to line end
`Alt+Shift+Left`	Select to line start
`Shift+Up`	Select up
`Shift+Down`	Select down
`Shift+Page Up`	Select page up
`Shift+Page Down`	Select page down
`Ctrl+Shift+Home`	Select to start
`Ctrl+Shift+End`	Select to end
`Ctrl+Shift+D`	Duplicate selection
`Ctrl+Shift+P`	Select to matching bracket

Table 5.63: Go to

Shortcut	Action
`Left`	Go to left
`Right`	Go to right
`Ctrl+Left`	Go to word left
`Ctrl+Right`	Go to word right
`Up`	Go line up
`Down`	Go line down
`Alt+Left`, `Home`	Go to line start
`Alt+Right`, `End`	Go to line end
`Page Up`	Go to page up
`Page Down`	Go to page down
`Ctrl+Home`	Go to start
`Ctrl+End`	Go to end
`Ctrl+L`	Go to line
`Ctrl+Down`	Scroll line down
`Ctrl+Up`	Scroll line up
`Ctrl+P`	Go to matching bracket

Table 5.64: Find/Replace

Shortcut	Action
`Ctrl+F`	Find
`Ctrl+H`	Replace
`Ctrl+K`	Find next
`Ctrl+Shift+K`	Find previous

Table 5.65: Folding

Shortcut	Action
`Alt+L`	Fold selection
`Alt+Shift+L`	Unfold

Table 5.66: Other

Shortcut	Action
`Tab`	Indent
`Shift+Tab`	Outdent
`Ctrl+Z`	Undo
`Ctrl+Shift+Z`, `Ctrl+Y`	Redo
`Ctrl+/`	Toggle comment
`Ctrl+Shift+U`	Change to lower case
`Ctrl+U`	Change to upper case
`Ins`	Overwrite
`Ctrl+Shift+E`	Macros replay

Shortcut	Action
`Ctrl+Alt+E`	Macros recording
`Del`	Delete

5.24.3. LabOne Sequence Programming

A Simple Example

The syntax of the LabOne AWG Sequencer programming language is based on C, but with a few simplifications. Each statement is concluded with a semicolon, several statements can be grouped with curly brackets, and comment lines are identified with a double slash. The following example shows some of the fundamental functionalities: waveform generation, repeated playback, triggering, and single/dual-channel waveform playback. See Arbitrary Waveform Generator for a step-by-step introduction with more examples.

```
// Define an integer constant
const N = 4096;
// Create two Gaussian pulses with length N points,
// amplitude +1.0 (-1.0), center at N/2, and a width of N/8
wave gauss_pos = 1.0*gauss(N, N/2, N/8);
wave gauss_neg = -1.0*gauss(N, N/2, N/8);
// execute playback sequence 100 times
repeat (100) {
  // Play pulse on AWG channel 1
  playWave(gauss_pos);
  // Play pulses simultaneously on both AWG channels
  playWave(gauss_pos, gauss_neg);
  // Wait 8000 samples
  playZero(8000);
}
```

Multi-Instrument Support

The UHFLI supports multi-instrument operation by two important features

1. Automatic synchronization
2. Multi-instrument sequence program compilation

The first feature ensures that signals of multiple AWGs are precisely aligned in time and the user does not have to worry about cable delays, or about varying trigger delays after power cycles. The second feature greatly simplifies writing sequence program, as it allows to treat a setup with multiple AWGs conceptually like a single instrument.

Automatic synchronization can be set up using the Multi-Device Sync tab and is explained in detail in Multi Device Sync Tab. We assume that two UHFLI have been successfully synchronized according to the instructions in this section. Here we show an example of a sequence program to generate synchronized signals on two instruments

As part of the synchronization procedure using MDS, the LabOne Data Server running on the host PC is connected to both instruments. In the AWG tab, enable the Multi-Device button. The LabOne AWG Compiler is then able to distribute the high-level, multi-channel program the user enters in the AWG tab across all instruments. The Signal Output on which a given wave `w` is played is controlled by the integer argument `sig_out` in the instruction `playWave(sig_out, w)`. The numbering of the Signal Outputs is as follows:

Channel number in sequence program	Instrument number (according to order in MDS tab)	Signal Output number
1	Leader	1
2	Leader	2

Channel number in sequence program	Instrument number (according to order in MDS tab)	Signal Output number
3	Follower 1	1
4	Follower 1	2
5	Follower 2	1
...	...	...

The sequence program below contains three `playWave` instructions: the first instruction generates a pulse on instrument no. 1, the second one on instrument no. 2, and the third `playWave` instruction generates pulses simultaneously on both instruments.

```
wave w_gauss = gauss(8000, 4000, 1000);

while (true) {
  setTrigger(1);
  // Pulse on AWG 1 (Signal Output 1):
  playWave(1, w_gauss);
  // Pulse on AWG 2 (Signal Output 1):
  playWave(3, w_gauss);
  // Pulse on AWG 1 (Signal Outputs 1 & 2)
  //   and on AWG 2 (Signal Outputs 1 & 2):
  playWave(1, w_gauss, 2, w_gauss,
           3, w_gauss, 4, w_gauss);
  setTrigger(0);
  wait(100);
}
```

Keywords and Comments

The following table lists the keywords used in the LabOne AWG Sequencer language.

Table 5.67: Programming keywords

Keyword	Description
`const`	Constant declaration
`var`	Integer variable declaration
`cvar`	Compile-time variable declaration
`string`	Constant string declaration
`true`	Boolean true constant
`false`	Boolean false constant
`for`	For-loop declaration
`while`	While-loop declaration
`repeat`	Repeat-loop declaration
`if`	If-statement
`else`	Else-part of an if-statement
`switch`	Switch-statement
`case`	Case-statement within a switch
`default`	Default-statement within a switch
`return`	Return from function or procedure, optionally with a return value

The following code example shows how to use comments.

```
const a = 10; // This is a line comment. Everything between the double
              // slash and the end of the line will be ignored.

/* This is a block comment. Everything between the start-of-block-comment
and end-of-block-comment markers is ignored.

For example, the following statement will be ignored by the compiler.
const b = 100;
*/
```

Constants and Variables

Constants may be used to make the program more readable. They may be of integer or floating-point type. It must be possible for the compiler to compute the value of a constant at compile time, i.e., on the host computer. Constants are declared using the `const` keyword.

Compile-time variables may be used in computations and loop iterations during compile time, e.g. to create large numbers of waveforms in a loop. They may be of integer or floating-point type. They are used in a similar way as constants, except that they can change their value during compile time operations. Compile-time variables are declared using the `cvar` keyword.

Variables may be used for making simple computations during run time, i.e., on the instrument. The Sequencer supports integer variables, addition, and subtraction. Not supported are floating-point variables, multiplication, and division. Typical uses of variables are to step waiting times , to output DIO values, or to tag digital measurement data with a numerical identifier. Variables are declared using the `var` keyword.

The following code example shows how to use variables.

```
var b = 100; // Create and initialize a variable

// Repeat the following block of statements 100 times
repeat (100) {
  b = b + 1; // Increment b
  wait(b);   // Wait 'b' cycles
}
```

The following table shows the predefined constants. These constants are intended to be used as arguments in certain run-time evaluated functions that encode device parameters with integer numbers. For example, the AWG Sampling Rate is specified as an integer exponent n in the expression (1.8 GSa/s)/2^n.

Table 5.68: Predefined Constants

Name	Value	Description
commandTableEntries	{4096}	
AWG_RATE_1800MHZ	0	Constant to set Sampling Rate to 1.8 GHz.
AWG_RATE_900MHZ	1	Constant to set Sampling Rate to 900 MHz.
AWG_RATE_450MHZ	2	Constant to set Sampling Rate to 450 MHz.
AWG_RATE_225MHZ	3	Constant to set Sampling Rate to 225 MHz.
AWG_RATE_112MHZ	4	Constant to set Sampling Rate to 112 MHz.
AWG_RATE_56MHZ	5	Constant to set Sampling Rate to 56 MHz.
AWG_RATE_28MHZ	6	Constant to set Sampling Rate to 28 MHz.
AWG_RATE_14MHZ	7	Constant to set Sampling Rate to 14 MHz.

Name	Value	Description
AWG_RATE_7MHZ	8	Constant to set Sampling Rate to 7 MHz.
AWG_RATE_3P5MHZ	9	Constant to set Sampling Rate to 3.5 MHz.
AWG_RATE_1P8MHZ	10	Constant to set Sampling Rate to 1.8 MHz.
AWG_RATE_880KHZ	11	Constant to set Sampling Rate to 880 kHz.
AWG_RATE_440KHZ	12	Constant to set Sampling Rate to 440 kHz.
AWG_RATE_220KHZ	13	Constant to set Sampling Rate to 220 kHz.
AWG_MONITOR_TRIGGER	0x0000020	Constant to activate the trigger of the Monitor unit.
AWG_INTEGRATION_TRIGGER	0x0000010	Constant to activate the trigger of the Integration units.
AWG_INTEGRATION_ARM	0x3ff0000	Constant to arm the Integration units.
DEVICE_SAMPLE_RATE	<actual device sample rate>	
AWG_CHAN1	1	Constant to select channel 1.
AWG_CHAN2	2	Constant to select channel 2.
AWG_MARKER1	1	Constant to select marker 1.
AWG_MARKER2	2	Constant to select marker 2.
AWG_OSC_PHASE_START	1	Constant to trigger the oscillator phase on the positive edge.
AWG_OSC_PHASE_MIDDLE	0	Constant to trigger the oscillator phase on the negative edge.
AWG_USERREG_SWEEP_COUNT0	35	Constant for the sweep count register 0.
AWG_USERREG_SWEEP_COUNT1	36	Constant for the sweep count register 1.

Numbers can be expressed using either of the following formatting.

```
const a = 10;            // Integer notation
const b = -10;           // Negative number
const h = 0xdeadbeef;    // Hexadecimal integer
const bin = 0b10101;     // Binary integer
const f = 0.1e-3;        // Floating point number.
const not_float = 10e3;  // Not a floating point number
```

Booleans are specified with the keywords `true` and `false`. Furthermore, all numbers that evaluate to a nonzero value are considered true. All numbers that evaluate to zero are considered false.

Strings are delimited using "" and are interpreted as constants. Strings may be concatenated using the + operator.

```
string AWG_PATH = "awgs/0/";
string AWG_GAIN_PATH = AWG_PATH + "gains/0";
```

Waveform Generation and Editing

The following table contains the definition of functions for waveform generation.

wave zeros(const samples)

Constant amplitude of 0 over the defined number of samples.

$$h(x) = 0$$

Args:

- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave ones(const samples)

Constant amplitude of 1 over the defined number of samples.

$$h(x) = 1$$

Args:

- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave sine(const samples, const amplitude=1.0, const phaseOffset, const nrOfPeriods)

Sine function with arbitrary amplitude (a), phase offset in radians (p), number of periods (f) and number of samples (N).

$$h(x) = a \cdot \sin(2\pi f \frac{x}{N} + p)$$

Args:

- `amplitude:` Amplitude of the signal (optional)
- `nrOfPeriods:` Number of Periods within the defined number of samples
- `phaseOffset:` Phase offset of the signal in radians
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave cosine(const samples, const amplitude=1.0, const phaseOffset, const nrOfPeriods)

Cosine function with arbitrary amplitude (a), phase offset in radians (p), number of periods (f) and number of samples (N).

$$h(x) = a \cdot \cos(2\pi f \frac{x}{N} + p)$$

Args:

- `amplitude:` Amplitude of the signal (optional)
- `nrOfPeriods:` Number of Periods within the defined number of samples
- `phaseOffset:` Phase offset of the signal in radians
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave sinc(const samples, const amplitude=1.0, const position, const beta)

Normalized sinc function with control of peak position (p), amplitude (a), width (\beta) and number of samples (N).

$$h(x) = \{a \quad \text{if } x = p \; a \cdot \frac{\sin(2\pi \cdot \text{beta} \cdot \frac{x-p}{N})}{2\pi \cdot \text{beta} \cdot \frac{x-p}{N}} \quad \text{else}$$

Args:

- `amplitude:` Amplitude of the signal (optional)
- `beta:` Width of the function
- `position:` Peak position of the function
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave ramp(const samples, const startLevel, const endLevel)

Linear ramp from the start (s) to the end level (e) over the number of samples (N).

$$h(x) = s + \frac{x(e-s)}{N-1}$$

Args:

- `endLevel:` level at the last sample of the waveform
- `samples:` Number of samples in the waveform
- `startLevel:` level at the first sample of the waveform

Returns:

resulting waveform

wave sawtooth(const samples, const amplitude=1.0, const phaseOffset, const nrOfPeriods)

Sawtooth function with arbitrary amplitude, phase in radians and number of periods.

Args:

- `amplitude:` Amplitude of the signal
- `nrOfPeriods:` Number of Periods within the defined number of samples
- `phaseOffset:` Phase offset of the signal in radians
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave triangle(const samples, const amplitude=1.0, const phaseOffset, const nrOfPeriods)

Triangle function with arbitrary amplitude, phase in radians and number of periods.

Args:

- `amplitude:` Amplitude of the signal
- `nrOfPeriods:` Number of Periods within the defined number of samples
- `phaseOffset:` Phase offset of the signal in radians
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave gauss(const samples, const amplitude=1.0, const position, const width)

Gaussian pulse with arbitrary amplitude (a), position (p), width (w) and number of samples (N).

$$h(x) = a \cdot e^{-\frac{(x-p)^2}{2 \cdot w^2}}$$

Args:

- `amplitude:` Amplitude of the signal (optional)
- `position:` Peak position of the pulse
- `samples:` Number of samples in the waveform
- `width:` Width of the pulse

Returns:

resulting waveform

wave drag(const samples, const amplitude=1.0, const position, const width)

Derivative of Gaussian pulse with arbitrary amplitude (a), position (p), width (w) and number of samples (N) normalized to a maximum amplitude of 1.

$$h(x) = a \cdot \frac{\sqrt{e}(p - x)}{w} \cdot e^{-\frac{(x-p)^2}{2 \cdot w^2}}$$

Args:

- `amplitude:` Amplitude of the signal (optional)
- `position:` Center point position of the pulse
- `samples:` Number of samples in the waveform
- `width:` Width of the pulse

Returns:

resulting waveform

wave blackman(const samples, const amplitude=1.0, const alpha)

Blackman window function with arbitrary amplitude (a), alpha parameter and number of samples (N).

$$h(x) = a \cdot (\alpha_0 - \alpha_1 \cos(\frac{2\pi x}{N-1}) \quad + \alpha_2 \cos(\frac{4\pi x}{N-1})) \; \alpha_0 = \frac{1-\alpha}{2}; \quad \alpha_1 = \frac{1}{2}; \quad \alpha_2 = \frac{\alpha}{2};$$

Args:

- `alpha:` Width of the function
- `amplitude:` Amplitude of the signal (optional)
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave hamming(const samples, const amplitude=1.0)

Hamming window function with arbitrary amplitude (a) and number of samples (N).

$$h(x) = a \cdot (\alpha - \beta \cos(\frac{2\pi x}{N-1})) \text{ with } \alpha = 0.54 \text{ and } \beta = 0.46$$

Args:

- `amplitude:` Amplitude of the signal (optional)
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave hann(const samples, const amplitude=1.0)

Hann window function with arbitrary amplitude (a) and number of samples (N).

$$h(x) = a \cdot 0.5 \cdot (1 - \cos(\frac{2\pi x}{N-1}))$$

Args:

- `amplitude:` Amplitude of the signal
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave rect(const samples, const amplitude)

Rectangle function, constants amplitude (a) over the defined number of samples.

$$h(x) = a$$

Args:

- `amplitude:` Amplitude of the signal
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave marker(const samples, const markerValue)

Generate a waveform with marker bits set to the specified value. The analog part of the waveform is zero.

Args:

- `markerValue:` Value of the marker bits
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave rand(const samples, const amplitude=1.0, const mean, const stdDev)

White noise with arbitrary amplitude, power and standard deviation.

Args:

- `amplitude:` Amplitude of the signal
- `mean:` Average signal level
- `samples:` Number of samples in the waveform
- `stdDev:` Standard deviation of the noise signal

Returns:

resulting waveform

wave randomGauss(const samples, const amplitude=1.0, const mean, const stdDev)

White noise with arbitrary amplitude, power and standard deviation.

Args:

- `amplitude:` Amplitude of the signal
- `mean:` Average signal level
- `samples:` Number of samples in the waveform
- `stdDev:` Standard deviation of the noise signal

Returns:

resulting waveform

wave randomUniform(const samples, const amplitude=1.0)

Random waveform with uniform distribution.

Args:

- `amplitude:` Amplitude of the signal
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave lfsrGaloisMarker(const samples, const markerBit, const polynomial, const initial)

Generate a waveform with specified marker bit set to the Galois LFSR (linear-feedback shift register) generated sequence. The analog part of the waveform is zero. The LFSR characteristic polynomial is a member of the Galois Field of two elements and represented in binary form. See wikipedia entries for "Finite field arithmetic" and "Linear-feedback shift register (Galois LFSR)".

Args:

- `initial:` LFSR initial state, any nonzero value will work, usually 0x1
- `markerBit:` Marker bit to set (1 or 2)
- `polynomial:` LFSR characteristic polynomial in binary representation (max shift length 32), use 0x90000 for QRSS / PRBS-20
- `samples:` Number of samples in the waveform

Returns:

resulting waveform

wave chirp(const samples, const amplitude=1.0, const startFreq, const stopFreq, const phase=0)

Frequency chirp function with arbitrary amplitude, start and stop frequency, initial phase in radians and number of samples. Start and stop frequency are expressed in units of the AWG Sampling Rate. The amplitude can only be defined if the initial phase is defined as well.

Args:

- `amplitude:` Amplitude of the signal (optional)
- `phase:` Initial phase of the signal (optional)
- `samples:` Number of samples in the waveform
- `startFreq:` Start frequency of the signal
- `stopFreq:` Stop Frequency of the signal

Returns:

resulting waveform

wave rrc(const samples, const amplitude=1.0, const position, const beta, const width)

Root raised cosine function with arbitrary amplitude (a), position (p), roll-off factor (\beta) and width (w) and number of samples (N).

$$h(y) = a\frac{\sin(y\pi(1-\beta)) + 4y\beta\cos(y\pi(1+\beta))}{y\pi(1-(4y\beta)^2)} \text{ with } y(x) = 2w\frac{x-p}{N}$$

Args:

- `amplitude:` Amplitude of the signal
- `beta:` Roll-off factor
- `position:` Center point position of the pulse
- `samples:` Number of samples in the waveform
- `width:` Width of the pulse

Returns:

Resulting waveform

wave vect(const value,...)

Specify a waveform sample by sample. Each sample is defined by one of an arbitrary number of input arguments. Only recommended for short waveforms that consist of less than 100 samples. Larger waveforms may be defined in a CSV file.

Args:

- `value:` Waveform amplitude at the respective sample

Returns:

resulting waveform

wave placeholder(const samples, const marker0=false, const marker1=false)

Creates space for a single-channel waveform, optionally with markers, without actually generating any waveform data when compiling the sequence program. Actual waveform data needs to be uploaded separately via the "<dev>/AWGS/<n>/WAVEFORM/WAVES/<index>" API nodes after the sequence compilation and upload. The waveform index can be explicitly assigned to the generated placeholder wave using the assignWaveIndex instruction.

Args:

- `marker0:` true if marker bit 0 must be used (default false)
- `marker1:` true if marker bit 1 must be used (default false)
- `samples:` Number of samples in the waveform

Returns:

waveform object

The following table contains the definition of functions for waveform editing.

wave join(wave wave1, wave wave2, const interpolLength=0)

Connect two or more waveforms with optional linear interpolation between the waveforms.

Args:

- `interpolLength:` Number of samples to interpolate between waveforms (optional, default 0)
- `wave1:` Input waveform
- `wave2:` Input waveform

Returns:

joined waveform

wave join(wave wave1, wave wave2,...)

Connect two or more waveforms.

Args:

- `wave1:` Input waveform
- `wave2:` Input waveform

Returns:

joined waveform

wave interleave(wave wave1, wave wave2,...)

Interleave two or more waveforms sample by sample.

Args:

- `wave1:` Input waveform
- `wave2:` Input waveform

Returns:

interleaved waveform

wave add(wave wave1, wave wave2,...)

Add two or more waveforms sample by sample. Alternatively, the "+" operator may be used for waveform addition.

Args:

- `wave1:` Input waveform
- `wave2:` Input waveform

Returns:

sum waveform

wave multiply(wave wave1, wave wave2,...)

Multiply two or more waveforms sample by sample. Alternatively, the "*" operator may be used for waveform multiplication.

Args:

- `wave1:` Input waveform
- `wave2:` Input waveform

Returns:

product waveform

wave scale(wave waveform, const factor)

Scale the input waveform with the factor and return the scaled waveform. The input waveform remains unchanged.

Args:

- `factor:` Scaling factor
- `waveform:` Input waveform

Returns:

scaled waveform

wave flip(wave waveform)

Flip the input waveform back to front and return the flipped waveform. The input waveform remains unchanged.

Args:

- `waveform:` Input waveform

Returns:

flipped waveform

wave cut(wave waveform, const from, const to)

Cuts a segment out of the input waveform and returns it. The input waveform remains unchanged. The segment is flipped in case that "from" is larger than "to".

Args:

- `from:` First sample of the cut waveform
- `to:` Last sample of the cut waveform
- `waveform:` Input waveform

Returns:

cut waveform

wave filter(wave b, wave a, wave x)

Filter generates a rational transfer function with the waveforms a and b as numerator and denominator coefficients. The transfer function is normalized by the first element of a, which has to be non-zero. The filter is applied to the input waveform x and returns the filtered waveform.

$$y(n) = \frac{1}{a_0}\left(! \sum_{i=0}^{M} b_i x_{n-i} - \sum_{i=1}^{N} a_i y_{n-i} !\right) \text{ with } M = \text{length}(b) - 1 \text{ and } N = \text{length}(a) - 1$$

Args:

- `a:` Denominator coefficients
- `b:` Numerator coefficients
- `x:` Input waveform

Returns:

filtered waveform

wave circshift(wave a, const n)

Circularly shifts a 1D waveform and returns it.

Args:

- `n:` Number of elements to shift
- `waveform:` Input waveform

Returns:

circularly shifted waveform

Waveform Playback and Predefined Functions

The following table contains the definition of functions for waveform playback and other purposes.

void setDIO(var value)

Writes the value as a 32-bit value to the DIO bus.

The value can be either a const or a var value. Configure the Mode setting in the DIO tab when using this command. The DIO interface speed of 50 MHz limits the rate at which the DIO output value is updated.

Args:

- `value:` The value to write to the DIO (const or var)

var getDIO()

Reads a 32-bit value from the DIO bus.

Returns:

var containing the read value

var getDIOTriggered()

Reads a 32-bit value from the DIO bus as recorded at the last DIO trigger position.

Returns:

var containing the read value

void setTrigger(var value)

Sets the AWG Trigger output signals.

The state of all four AWG Trigger output signals is represented by the bits in the binary representation of the integer value. Binary notation of the form 0b0000 is recommended for readability.

Args:

- `value:` to be written to the trigger output lines

void wait(var cycles)

Waits for the given number of Sequencer clock cycles (4.44 ns per cycle).

Args:

- `cycles:` number of cycles to wait

void waitDIOTrigger()

Waits until the DIO interface trigger is active. The trigger is specified by the Strobe Index and Strobe Slope settings in the AWG Sequencer tab.

var getDigTrigger(const index)

Gets the state of the indexed Digital Trigger input (1 or 2 on UHF, 1-8 on HDAWG).

The physical signal connected to the AWG Digital Trigger input is to be configured in the Trigger sub-tab of the AWG tab.

Args:

- `index:` index of the Digital Trigger input to be read; can be either 1 or 2 on UHF, or 1-8 on HDAWG

Returns:

trigger state, either 0 or 1

void error(string msg,...)

Throws the given error message when reached.

Args:

- `msg:` Message to be displayed

void info(string msg,...)

Returns the specified message when reached.

Args:

- `msg:` Message to be displayed

void setInt(string path, var value)

Writes an integer value to one of the nodes in the device.

If the path does not start with a device identifier, then the current device is assumed.

Args:

- `path:` The node path to be written to
- `value:` The integer value to be written

void setDouble(string path, var value)

Writes a floating point value to one of the nodes in the device.

If the path does not start with a device identifier, then the current device is assumed.

Args:

- `path:` The node path to be written to
- `value:` The integer or floating point value to be written

void setDouble(string path, var value, const scale)

Writes a floating point value to one of the nodes in the device.

If the path does not start with a device identifier, then the current device is assumed.

Args:

- `path:` The node path to be written to
- `scale:` Scaling value to be applied to the value before writing to the node
- `value:` The integer or floating point value to be written

void setID(var id)

Sets the ID value that is attached to data streamed from the device to the host PC. The ID value is useful for synchronizing the data acquisition process in combination with the Sweeper or the Software Trigger. The ID value is denoted AWG Seq Index in the tree of tools like the plotter.

Args:

- `id:` The new ID to be attached to streaming data of the device

void setSeqIndex(var id)

Sets the ID value that is attached to data streamed from the device to the host PC. The ID value is useful for synchronizing the data acquisition process in combination with the Sweeper or the Software Trigger. The ID value is denoted AWG Seq Index in the tree of tools like the plotter. The setSeqIndex function is identical to the setID function.

Args:

- `id:` The new ID to be attached to streaming data of the device

void sync()

Perform Multi-Device synchronization command for all devices at this point. Leader/Follower assignment is automatic.

Only for programs running on multiply synchronized instruments.

void waitWave()

Waits until the AWG is done playing the current waveform.

void randomSeed()

Generate a new seed for the subsequent random vector commands.

void assignWaveIndex(const output, wave waveform, const index)

void assignWaveIndex(wave waveform, const index)

void playWave(const output, wave waveform, const rate=AWG_RATE_DEFAULT)

Starts to play the given waveforms on the defined output channels. The playback begins as soon as the previous waveform playback is finished.

Args:

- `output:` defines on which output the following waveform is played
- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playWave(const output, wave waveform,...)

Starts to play the given waveforms on the defined output channels. It can contain multiple waveforms with an output definition. The playback begins as soon as the previous waveform playback is finished.

Args:

- `output:` defines on which output the following waveform is played
- `waveform:` waveform to be played

void playWave(wave waveform, const rate=AWG_RATE_DEFAULT)

Starts to play the given waveforms, output channels are assigned automatically depending on the number of input waveforms. The playback begins as soon as the previous waveform playback is finished.

Args:

- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playWave(wave waveform,...)

Starts to play the given waveforms, output channels are assigned automatically depending on the number of input waveforms. The playback begins as soon as the previous waveform playback is finished.

Args:

- `waveform:` waveform to be played

void setUserReg(const register, var value)

Writes a value to one of the User Registers (indexed 0 to 15).

The User Registers may be used for communicating information to the LabOne User Interface or a running API program.

Args:

- `register:` The register index (0 to 15) to be written to
- `value:` The integer value to be written

var getUserReg(const register)

Reads the value from one of the User Registers (indexed 0 to 15). The User Registers may be used for communicating information to the LabOne User Interface or a running API program.

Args:

- `register:` The register to be read (0 to 15)

Returns:

current register value

void playZero(const samples)

Starts to play zeros on all channels for the specified number of samples. Behaves as if same length all-zeros waveform is played using playWave, but without consuming waveform memory.

Args:

- `samples:` Number of samples to be played. The same min length and granularity applies as for regular waveforms.

void playZero(const samples, const rate)

Starts to play zeros on all channels for the specified number of samples. Behaves as if same length all-zeros waveform is played using playWave, but without consuming waveform memory.

Args:

- `rate:` Sample rate with which the AWG plays zeros (default set in the user interface).
- `samples:` Number of samples to be played. The same min length and granularity applies as for regular waveforms.

void setRate(const rate)

Overwrites the default Sampling Rate for the following playWave commands.

Args:

- `rate:` New default sampling rate

void playWaveNow(const output, wave waveform, const rate=AWG_RATE_DEFAULT)

Starts to play the given waveforms on the defined output channels. It starts immediately even if the previous waveform playback is still in progress.

Args:

- `output:` defines on which output the following waveform is played
- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playWaveNow(const output, wave waveform,...)

Starts to play the given waveforms on the defined output channels. It can contain multiple waveforms with an output definition. It starts immediately even if the previous waveform playback is still in progress.

Args:

- `output:` defines on which output the following waveform is played
- `waveform:` waveform to be played

void playWaveNow(wave waveform, const rate=AWG_RATE_DEFAULT)

Starts to play the given waveforms, channels are assigned automatically depending on the number of input waveforms. It starts immediately even if the previous waveform playback is still in progress.

Args:

- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playWaveNow(wave waveform,...)

Starts to play the given waveforms, channels are assigned automatically depending on the number of input waveforms. It starts immediately even if the previous waveform playback is still in progress.

Args:

- `waveform:` waveform to be played

void playWaveIndexed(const output, wave waveform, var offset, const length, const rate=AWG_RATE_DEFAULT)

Starts to play the specified part of the given waveforms on the defined output channels. It can contain multiple waveforms with an output definition. The playback begins as soon as the previous waveform playback is finished.

Args:

- `length:` number of samples to be played from this waveform
- `offset:` offset in samples from the start of the waveform
- `output:` defines on which output the following waveform is played
- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playWaveIndexed(wave waveform, var offset, const length, const rate=AWG_RATE_DEFAULT)

Starts to play the specified part of the given waveforms, channels are assigned automatically depending on the number of input waveforms. The playback begins as soon as the previous waveform playback is finished.

Args:

- `length:` number of samples to be played from this waveform
- `offset:` offset in samples from the start of the waveform
- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playWaveIndexedNow(const output, wave waveform, var offset, const length, const rate=AWG_RATE_DEFAULT)

Starts to play the specified part of the given waveforms on the defined output channels. It can contain multiple waveforms with an output definition. It starts immediately even if the previous waveform playback is still in progress.

Args:

- `length:` number of samples to be played from this waveform
- `offset:` offset in samples from the start of the waveform
- `output:` defines on which output the following waveform is played
- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playWaveIndexedNow(wave waveform, var offset, const length, const rate=AWG_RATE_DEFAULT)

Starts to play the specified part of the given waveforms, channels are assigned automatically depending on the number of input waveforms. It starts immediately even if the previous waveform playback is still in progress.

Args:

- `length:` number of samples to be played from this waveform
- `offset:` offset in samples from the start of the waveform
- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playDIOWave(wave waveform, const rate=AWG_RATE_DEFAULT)

Starts to play the given waveforms, channels are assigned automatically depending on the number of input waveforms, with enabled 4-channel-mode. Configure the Signal and Channel settings in the Aux tab in combination with this function. The playback begins as soon as the previous waveform playback is finished.

Args:

- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playDIOWave(wave waveform,...)

Starts to play the given waveforms, channels are assigned automatically depending on the number of input waveforms, with enabled 4-channel-mode. Configure the Signal and Channel settings in the Aux tab in combination with this function. The playback begins as soon as the previous waveform playback is finished.

Args:

- `waveform:` waveform to be played

void playAuxWave(const output, wave waveform, const rate=AWG_RATE_DEFAULT)

Starts to play the given waveforms on the defined output channels with enabled 4-channel-mode. Configure the Signal and Channel settings in the Aux tab in combination with this function. The playback begins as soon as the previous waveform playback is finished.

Args:

- `output:` defines on which output the following waveform is played
- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playAuxWave(const output, wave waveform,...)

Starts to play the given waveforms on the defined output channels with enabled 4-channel-mode. It can contain multiple waveforms with an output definition. Configure the Signal and Channel settings in the Aux tab in combination with this function. The playback begins as soon as the previous waveform playback is finished.

Args:

- `output:` defines on which output the following waveform is played
- `waveform:` waveform to be played

void playAuxWave(wave waveform, const rate=AWG_RATE_DEFAULT)

Starts to play the given waveforms, channels are assigned automatically depending on the number of input waveforms, with enabled 4-channel-mode. Configure the Signal and Channel settings in the Aux tab in combination with this function. The playback begins as soon as the previous waveform playback is finished.

Args:

- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playAuxWave(wave waveform,...)

Starts to play the given waveforms, channels are assigned automatically depending on the number of input waveforms, with enabled 4-channel-mode. Configure the Signal and Channel settings in the Aux tab in combination with this function. The playback begins as soon as the previous waveform playback is finished.

Args:

- `waveform:` waveform to be played

void playAuxWaveIndexed(const output, wave waveform, var offset, const length, const rate=AWG_RATE_DEFAULT)

Starts to play the specified part of the given waveforms on the defined output channels with enabled 4-channel-mode. Configure the Signal and Channel settings in the Aux tab in combination with this function. The playback begins as soon as the previous waveform playback is finished.

Args:

- `length:` number of samples to be played from this waveform
- `offset:` offset in samples from the start of the waveform
- `output:` defines on which output the following waveform is played
- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void playAuxWaveIndexed(wave waveform, var offset, const length, const rate=AWG_RATE_DEFAULT)

Starts to play the specified part of the given waveforms, channels are assigned automatically depending on the number of input waveforms, with enabled 4-channel-mode. Configure the Signal and Channel settings in the Aux tab in combination with this function. The playback begins as soon as the previous waveform playback is finished.

Args:

- `length:` number of samples to be played from this waveform
- `offset:` offset in samples from the start of the waveform
- `rate:` sample rate with which the AWG plays the waveforms (default set in the user interface).
- `waveform:` waveform to be played

void waitAnaTrigger(const index, const value)

Waits until the indexed Analog Trigger input (1 or 2) is equal to the given value (0 or 1). The physical signal connected to the AWG Analog Trigger inputs as well as the trigger level is to be configured in the Trigger sub-tab of the AWG tab.

Args:

- `index:` index of the analog trigger input to be waited on; can be either 1 or 2 on UHF, or 1 to 8 on HDAWG
- `value:` value to be compared with the Analog Trigger input, can be either 0 or 1

var getAnaTrigger(const index)

Gets the state of the indexed Analog Trigger input (1 or 2 on UHF, 1-8 on HDAWG).

The physical signal connected to the AWG Analog Trigger inputs as well as the trigger level is to be configured in the Trigger sub-tab of the AWG tab.

Args:

- `index:` index of the Analog Trigger input to be read; an be either 1 or 2 on UHF, or 1-8 on HDAWG

Returns:

trigger state, either 0 or 1

var getSweeperLength(const index)

Reads the sweep Length as configured in the Sweeper tab. The length is only valid when AWG Control is enabled in the Sweeper tab.

Args:

- `index:` The index of the Sweeper parameter. Currently only the value of 1 is accepted.

Returns:

length configured by the Sweeper

void now()

Resets the local timer.

void at(var time)

Waits until the local timer reaches the given value.

Args:

- `time:` value to wait for

void lock(wave waveform)

Ensures that the waveform is kept in the cache memory until the unlock command is used.

Args:

- `waveform:` The waveform to lock

void unlock(wave waveform)

Allow the compiler to use this memory block again to cache other waveforms.

Only valid after the waveform was previously locked using the lock command.

Args:

- `waveform:` The waveform to unlock

void waitTrigger(const mask, const value)

Waits until the masked trigger input is equal to the given value.

Args:

- `mask:` mask to be applied to the input signal
- `value:` value to be compared with the trigger input

void waitDigTrigger(const index, const value)

Waits until the indexed Digital Trigger (1 or 2) is equal to the given value (0 or 1). The physical signals connected to the two AWG Digital Triggers are to be configured in the Trigger sub-tab of the AWG tab.

Args:

- `index:` index of the digital trigger input to be waited on; can be either 1 or 2
- `value:` value to be compared with the digital trigger input, can be either 0 or 1

void waitDemodOscPhase(const demod)

Waits until the oscillator phase of the indexed demodulator (1-8) reaches the defined value.

Args:

- `demod:` index of the demodulator to be waited on, can be between 1 and 8

void waitDemodSample(const demod)

Waits until the indexed demodulator (1-8) delivers a new sample.

This command is used to synchronize the timing of the AWG signal with the demodulator readout e.g. in measurements with the Sweeper or the Software Trigger.

Args:

- `demod:` index of the demodulator to be waited on, can be between 1 and 8

Accessing Instrument Settings

Using the sequencer instructions `setInt` and `setDouble`, a large number of instrument settings may be accessed directly from the sequencer with a much shorter latency than when accessed via the LabOne API. The nodes accessible with `setInt setDouble` are a subset of the full list of device nodes accessible via the LabOne API (see Device Node Tree) and are listed in the table below together with their data type, latency class, and timing determinicity characteristics.

There are 3 levels of latency performance depending on how the node settings are processed on the instrument. Nodes that are classified with latency "medium" are processed with the instrument firmware by a non-realtime processor. These nodes are set typically within 100 microseconds, however not with deterministic timing. For some of the nodes, additional time is needed to take effect due to the time scale of the related hardware settings, e.g. changing the settings of the analog signal path. This needs to be taken into account by including sufficient waiting time by a `wait` sequencer instruction after setting the node.

Nodes classified with latency "low" are processed with deterministic timing on a dedicated bus inside the FPGA. The latency is of the order of 100 nanoseconds. Nodes classified with latency "ultra-low", such as timing-critical phase changes, are accessed via their dedicated signal path on the FPGA, and offer similarly low and deterministic latency as e.g. `playWave` instructions.

Providing the node as a character string is less flexible from the AWG sequencer than from the API: wildcards (*) are not supported, the node cannot start with a dash (/), and the device ID cannot be specified since it is excluded that the sequencer accesses other devices. This code example illustrates these restrictions:

```
setDouble("oscs/1/freq", 1e6); // permitted
setDouble("oscs/*/freq", 1e6); // not permitted
setDouble("dev8000/oscs/1/freq", 1e6); // not permitted
setDouble("/dev8000/oscs/1/freq", 1e6); // not permitted
setDouble("/oscs/1/freq", 1e6); // not permitted
```

In the table, the sequence [i-j] indicate a numeric range of valid indices from i to j

Node	Data type	Latency	Deterministic timing
demods/[0-7]/adcselect	Integer	medium	no
demods/[0-7]/order	Integer	medium	no
demods/[0-7]/rate	Float	medium	no
demods/[0-7]/oscselect	Integer	medium	no
demods/[0-7]/harmonic	Integer	medium	no
demods/[0-7]/phaseshift	Float	medium	no
demods/[0-7]/sinc	Integer	medium	no
demods/[0-7]/bypass	Integer	medium	no
demods/[0-7]/timeconstant	Float	medium	no
demods/[0-7]/enable	Integer	medium	no
scopes/0/enable	Integer	medium	no
scopes/0/time	Integer	medium	no
scopes/0/trigenable	Integer	medium	no
scopes/0/trigrising	Integer	medium	no
scopes/0/trigfalling	Integer	medium	no
scopes/0/triglevel	Float	medium	no
scopes/0/length	Integer	medium	no
scopes/0/trigholdoff	Float	medium	no
scopes/0/trigchannel	Integer	medium	no
scopes/0/channels/[0-1]/inputselect	Integer	medium	no
scopes/0/channels/[0-1]/bwlimit	Integer	medium	no
scopes/0/segments/count	Integer	medium	no
scopes/0/channel	Integer	medium	no
scopes/0/single	Integer	medium	no
scopes/0/trighysteresis/absolute	Float	medium	no
scopes/0/cont	Integer	medium	no
scopes/0/trighysteresis/relative	Float	medium	no
scopes/0/trighysteresis/mode	Integer	medium	no
scopes/0/trigforce	Integer	medium	no
scopes/0/segments/counts/enable	Integer	medium	no
scopes/0/trigstate	Integer	medium	no

Node	Data type	Latency	Deterministic timing
scopes/0/triggate/enable	Integer	medium	no
scopes/0/triggate/inputselect	Integer	medium	no
scopes/0/trigreference	Float	medium	no
scopes/0/trigdelay	Float	medium	no
scopes/0/trigholdofftriggerenable	Integer	medium	no
scopes/0/trigholdofftrigger	Integer	medium	no
scopes/0/channels/[0-1]/limitlower	Float	medium	no
scopes/0/channels/[0-1]/limitupper	Float	medium	no
sigins/[0-1]/ac	Integer	medium	no
sigins/[0-1]/imp50	Integer	medium	no
sigins/[0-1]/bw	Integer	medium	no
sigins/[0-1]/on	Integer	medium	no
sigins/[0-1]/range	Integer	medium	no
sigins/[0-1]/diff	Integer	medium	no
sigins/[0-1]/autorange	Integer	medium	no
sigins/[0-1]/scaling	Float	medium	no
sigouts/0/on	Integer	medium	no
sigouts/0/range	Float	medium	no
sigouts/0/imp50	Integer	medium	no
sigouts/0/autorange	Integer	medium	no
sigouts/[0-1]/on	Integer	medium	no
sigouts/[0-1]/range	Float	medium	no
sigouts/[0-1]/imp50	Integer	medium	no
sigouts/[0-1]/autorange	Integer	medium	no
sigouts/[0-1]/offset	Float	medium	no
sigouts/[0-1]/enables/[0-7]	Integer	medium	no
sigouts/[0-1]/amplitudes/[0-7]	Float	medium	no
mods/[0-1]/enable	Integer	medium	no
mods/[0-1]/output	Integer	medium	no
mods/[0-1]0/mode	Integer	medium	no
mods/[0-1]/sidebands/[0-1]/mode	Integer	medium	no
mods/[0-1]/freqdevenable	Integer	medium	no
mods/[0-1]/freqdev	Float	medium	no
mods/[0-1]/index	Float	medium	no
mods/[0-1]/sampleenable	Integer	medium	no
mods/[0-1]/operation	Integer	medium	no
mods/[0-1]/rate	Float	medium	no
mods/[0-1]/trigger	Integer	medium	no
mods/[0-1]/carrier/inputselect	Integer	medium	no
mods/[0-1]/sidebands/[0-1]/inputselect	Integer	medium	no
mods/[0-1]/carrier/order	Integer	medium	no
mods/[0-1]/sidebands/[0-1]/order	Integer	medium	no
mods/[0-1]/carrier/timeconstant	Float	medium	no

Node	Data type	Latency	Deterministic timing
mods/[0-1]/sidebands/[0-1]/timeconstant	Float	medium	no
mods/[0-1]/carrier/oscselect	Integer	medium	no
mods/[0-1]/sidebands/[0-1]/oscselect	Integer	medium	no
mods/[0-1]/carrier/harmonic	Integer	medium	no
mods/[0-1]/sidebands/[0-1]/harmonic	Integer	medium	no
mods/[0-1]/carrier/phaseshift	Float	medium	no
mods/[0-1]/sidebands/[0-1]/phaseshift	Float	medium	no
mods/[0-1]/carrier_enable	Integer	medium	no
mods/[0-1]/sidebands/[0-1]/enable	Integer	medium	no
mods/[0-1]/carrier_gain	Float	medium	no
mods/[0-1]/sideband[0-1]_gain	Float	medium	no
boxcars/[0-1]/enable	Integer	medium	no
boxcars/[0-1]/periods	Integer	medium	no
boxcars/[0-1]/windowstart	Float	medium	no
boxcars/[0-1]/insel	Integer	medium	no
boxcars/[0-1]/oscsel	Integer	medium	no
boxcars/[0-1]/limitrate	Float	medium	no
boxcars/[0-1]/baseline/enable	Integer	medium	no
boxcars/[0-1]/baseline/windowstart	Float	medium	no
boxcars/[0-1]/windowsize	Float	medium	no
outputpwas/[0-1]/enable	Integer	medium	no
outputpwas/[0-1]/single	Integer	medium	no
outputpwas/[0-1]/oscselect	Integer	medium	no
outputpwas/[0-1]/insel	Integer	medium	no
outputpwas/[0-1]/mode	Integer	medium	no
outputpwas/[0-1]/harmonic	Integer	medium	no
outputpwas/[0-1]/shift	Float	medium	no
outputpwas/[0-1]/termination	Integer	medium	no
outputpwas/[0-1]/samplecount	Long	medium	no
outputpwas/[0-1]/holdoff	Float	medium	no
outputpwas/[0-1]/progress	Float	medium	no
outputpwas/[0-1]/status	Integer	medium	no
inputpwas/[0-1]/enable	Integer	medium	no
inputpwas/[0-1]/single	Integer	medium	no
inputpwas/[0-1]/oscselect	Integer	medium	no
inputpwas/[0-1]/insel	Integer	medium	no
inputpwas/[0-1]/mode	Integer	medium	no
inputpwas/[0-1]/harmonic	Integer	medium	no
inputpwas/[0-1]/shift	Float	medium	no
inputpwas/[0-1]/termination	Integer	medium	no
inputpwas/[0-1]/samplecount	Long	medium	no
inputpwas/[0-1]/holdoff	Float	medium	no
pids/[0-3]/enable	Integer	medium	no

Node	Data type	Latency	Deterministic timing
pids/[0-3]/demod/adcselect	Integer	medium	no
pids/[0-3]/demod/order	Integer	medium	no
pids/[0-3]/demod/timeconstant	Float	medium	no
pids/[0-3]/setpoint	Float	medium	no
pids/[0-3]/dlimittimeconstant	Float	medium	no
pids/[0-3]/limitupper	Float	medium	no
pids/[0-3]/limitlower	Float	medium	no
pids/[0-3]/p	Float	medium	no
pids/[0-3]/i	Float	medium	no
pids/[0-3]/d	Float	medium	no
pids/[0-3]/demod/harmonic	Integer	medium	no
pids/[0-3]/outputchannel	Integer	medium	no
pids/[0-3]/output	Integer	medium	no
pids/[0-3]/phaseunwrap	Integer	medium	no
pids/[0-3]/stream/rate	Float	medium	no
pids/[0-3]/rate	Float	medium	no
pids/[0-3]/mode	Integer	medium	no
pids/[0-3]/inputsource	Integer	medium	no
pids/[0-3]/inputchannel	Integer	medium	no
pids/[0-3]/center	Double	medium	no
pids/[0-3]/pll/automode	Integer	medium	no
awgs/0/enable	Integer	medium	no
awgs/0/single	Integer	medium	no
awgs/0/time	Integer	medium	no
awgs/0/userregs/[0-15]]	Integer	medium	no
awgs/0/triggers/[0-1]/level	Float	medium	no
awgs/0/triggers/[0-1]/hysteresis/absolute	Float	medium	no
awgs/0/triggers/[0-1]/hysteresis/relative	Float	medium	no
awgs/0/triggers/[0-1]/hysteresis/mode	Integer	medium	no
awgs/0/triggers/[0-1]/rising	Integer	medium	no
awgs/0/triggers/[0-1]/falling	Integer	medium	no
awgs/0/triggers/[0-1]/channel	Integer	medium	no
awgs/0/triggers/[0-1]/force	Integer	medium	no
awgs/0/triggers/[0-1]/state	Integer	medium	no
awgs/0/triggers/[0-1]/gate/enable	Integer	medium	no
awgs/0/triggers/[0-1]/gate/inputselect	Integer	medium	no
awgs/0/auxtriggers/[0-1]/channel	Integer	medium	no
awgs/0/auxtriggers/[0-1]/rising	Integer	medium	no
awgs/0/auxtriggers/[0-1]/falling	Integer	medium	no
awgs/0/auxtriggers/[0-1]/state	Integer	medium	no
awgs/0/outputs/[0-1]/amplitude	Float	medium	no
awgs/0/outputs/[0-1]/mode	Integer	medium	no
auxouts/[0-3]/preoffset	Double	medium	no

Node	Data type	Latency	Deterministic timing
auxouts/[0-3]/offset	Float	medium	no
auxouts/[0-3]/scale	Float	medium	no
auxouts/[0-3]/limitlower	Float	medium	no
auxouts/[0-3]/limitupper	Float	medium	no
auxouts/[0-3]/offset	Integer	medium	no
auxouts/[0-3]/outputselect	Integer	medium	no
auxouts/[0-3]/demodselect	Integer	medium	no
triggers/in/[0-3]/imp50	Integer	medium	no
triggers/in/[0-3]/level	Integer	medium	no
triggers/in/[0-3]/dcc_fedge	Integer	medium	no
triggers/out/[0-3]/srcsel	Integer	medium	no
triggers/out/[0-3]/dir	Integer	medium	no
triggers/out/[0-3]/static_value	Integer	medium	no
triggers/out/[0-3]/hold	Integer	medium	no
triggers/out/[0-3]/delay	Float	medium	no
aucarts/[0-1]/mode	Integer	medium	no
aucarts/[0-1]/enable	Integer	medium	no
aucarts/[0-1]/rate	Float	medium	no
aucarts/[0-1]/ops/[0-1]/demodselect	Integer	medium	no
aucarts/[0-1]/ops/[0-1]/value	Integer	medium	no
aucarts/[0-1]/ops/[0-1]/coeff	Integer	medium	no
aucarts/[0-1]/ops/[0-1]/scale	Float	medium	no
aupolars/[0-1]/mode	Integer	medium	no
aupolars/[0-1]/enable	Integer	medium	no
aupolars/[0-1]/rate	Float	medium	no
aupolars/[0-1]/reserved1	Integer	medium	no
aupolars/[0-1]/ops/[0-1]/demodselect	Integer	medium	no
aupolars/[0-1]/ops/[0-1]/value	Integer	medium	no
aupolars/[0-1]/ops/[0-1]/coeff	Integer	medium	no
aupolars/[0-1]/ops/[0-1]/scale	Float	medium	no
aupolars/[0-1]/flags	Integer	medium	no
scopes/0/stream/enables/[0-1]	Integer	medium	no
scopes/0/stream/rate	Integer	medium	no
oscs/[0-7]/freq	Frequency	low	yes
demods/[0-7]/oscselect	Integer	ultra-low	yes
demods/[0-7]/phaseshift	Integer	ultra-low	yes

Nodes accessible with `setInt` and `setDouble`

Expressions

Expressions may be used for making computations based on mathematical functions and operators. There are two kinds of expressions: those evaluated at compile time (the moment of clicking "Save" or "Save as..." in the user interface), and those evaluated at run time (after clicking

"Run/Stop" or "Start"). Compile-time evaluated expressions only involve constants (`const`) or compile-time variables (`cvar`) and can be computed at compile time by the host computer. Such expressions can make use of standard mathematical functions and floating point arithmetic. Run-time evaluated expressions involve variables (`var`) and are evaluated by the Sequencer on the instrument. Due to the limited computational capabilities of the Sequencer, these expressions may only operate on integer numbers and there are less operators available than at compile time.

The following table contains the list of mathematical functions supported at compile time.

Table 5.69: Predefined Constants

Name	Value	Description
M_E	2.7182818284590452353602874713526625 0	e
M_LOG2E	1.4426950408889634073599246810018921 4	log2(e)
M_LOG10E	0.4342944819032518276511289189166050 82	log10(e)
M_LN2	0.6931471805599453094172321214581765 68	loge(2)
M_LN10	2.3025850929940456840179914546843642 1	loge(10)
M_PI	3.1415926535897932384626433832795028 8	pi
M_PI_2	1.5707963267948966192313216916397514 4	pi/2
M_PI_4	0.7853981633974483096156608458198757 21	pi/4
M_1_PI	0.3183098861837906715377675267450287 24	1/pi
M_2_PI	0.6366197723675813430755350534900574 48	2/pi
M_2_SQRTPI	1.1283791670955125738961589031215451 7	2/sqrt(pi)
M_SQRT2	1.4142135623730950488016887242096980 8	sqrt(2)
M_SQRT1_2	0.7071067811865475244008443621048490 39	1/sqrt(2)

The following table contains the list of predefined mathematical constants. These can be used for convenience in compile-time evaluated expressions.

Table 5.70: Predefined Constants

Name	Value	Description
M_E	2.7182818284590452353602874713526625 0	e
M_LOG2E	1.4426950408889634073599246810018921 4	log2(e)
M_LOG10E	0.4342944819032518276511289189166050 82	log10(e)
M_LN2	0.6931471805599453094172321214581765 68	loge(2)
M_LN10	2.3025850929940456840179914546843642 1	loge(10)
M_PI	3.1415926535897932384626433832795028 8	pi
M_PI_2	1.5707963267948966192313216916397514 4	pi/2
M_PI_4	0.7853981633974483096156608458198757 21	pi/4
M_1_PI	0.3183098861837906715377675267450287 24	1/pi
M_2_PI	0.6366197723675813430755350534900574 48	2/pi
M_2_SQRTPI	1.1283791670955125738961589031215451 7	2/sqrt(pi)
M_SQRT2	1.4142135623730950488016887242096980 8	sqrt(2)
M_SQRT1_2	0.7071067811865475244008443621048490 39	1/sqrt(2)

Table 5.71: Operators supported at compile time

Operator	Description	Priority
`=`	assignment	-1
`+=, -=, *=, /=, %=, &=, |=, <<=, >>=`	assignment by sum, difference, product, quotient, remainder, AND, OR, left shift, and right shift	-1
`||`	logical OR	1

Operator	Description	Priority
&&	logical AND	2
|	bit-wise logical OR	3
&	bit-wise logical AND	4
!=	not equal	5
==	equal	5
<=	less or equal	6
>=	greater or equal	6
>	greater than	6
<	less than	6
<<	arithmetic left bit shift	7
>>	arithmetic right bit shift	7
+	addition	8
-	subtraction	8
*	multiplication	9
/	division	9
~	bit-wise logical negation	10

Table 5.72: Operators supported at run time

Operator	Description	Priority
=	assignment	-1
+=, -=, *=, /=, %=, &=, |=, <<=, >>=	assignment by sum, difference, product, quotient, remainder, AND, OR, left shift, and right shift	-1
||	logical OR	1
&&	logical AND	2
|	bit-wise logical OR	3
&	bit-wise logical AND	4
==	equal	5
!=	not equal	5
<=	less or equal	6
>=	greater or equal	6
>	greater than	6
<	less than	6
<<	left bit shift	7
>>	right bit shift	7
+	addition	8
-	subtraction	8
~	bit-wise logical negation	9

Control Structures

Functions may be declared using the `var` keyword. **Procedures** may be declared using the `void` keyword. Functions must return a value, which should be specified using the `return` keyword.

Procedures can not return values. Functions and procedures may be declared with an arbitrary number of arguments. The `return` keyword may also be used without arguments to return from an arbitrary point within the function or procedure. Functions and procedures may contain variable and constant declarations. These declarations are local to the scope of the function or procedure.

```
var function_name(argument1, argument2, ...) {
  // Statements to be executed as part of the function.
  return constant-or-variable;
}

void procedure_name(argument1, argument2, ...) {
  // Statements to be executed as part of the procedure.
  // Optional return statement
  return;
}
```

An if-then-else structure is used to create a conditional branching point in a sequencer program.

```
// If-then-else statement syntax
if (expression) {
  // Statements to execute if 'expression' evaluates to 'true'.
} else {
  // Statements to execute if 'expression' evaluates to 'false'.
}

// If-then-else statement short syntax
(expression)?(statement if true):(statement if false)
```

```
// If-then-else statement example
const REQUEST_BIT     = 0x0001;
const ACKNOWLEDGE_BIT = 0x0002;
const IDLE_BIT        = 0x8000;
var dio = getDIO();
if (dio & REQUEST_BIT) {
  dio = dio | ACKNOWLEDGE_BIT;
  setDIO(dio);
} else {
  dio = dio | IDLE_BIT;
  setDIO(dio);
}
```

A switch-case structure serves to define a conditional branching point similarly to the `if-then-else` statement, but is used to split the sequencer thread into more than two branches. Unlike the `if-then-else` structure, the switch statement is synchronous, which means that the execution time is the same for all branches and determined by the execution time of the longest branch. If no default case is provided and no case matches the condition, all cases will be skipped. The case arguments need to be of type `const`.

```
// Switch-case statement syntax
switch (expression) {
  case const-expression:
    expression;
  ...
  default:
    expression;
}
```

```
// Switch-case statement example
switch (getDIO()) {
  case 0:
    playWave(gauss(1024,1.0,512,64));
  case 1:
    playWave(gauss(1024,1.0,512,128));
  case 2:
    playWave(drag(1024,1.0,512,64));
  default:
```

```
    playWave(drag(1024,1.0,512,128));
}
```

The for loop is used to iterate through a code block several times. The `initialization` statement is executed before the loop starts. The `end-expression` is evaluated at the start of each iteration and determines when the loop should stop. The loop is executed as long as this expression is true. The `iteration-expression` is executed at the end of each loop iteration.

Depending on how the `for` loop is set up, it can be either evaluated at compile time or at run time. Run-time evaluation is typically used to play series of waveforms. Compile-time evaluation is typically used for advanced waveform generation, e.g. to generate a series of waveforms with varying amplitude. For a run-time evaluated `for` loop, use the `var` data type as a loop index. To ensure that a loop is evaluated at compile time, use the `cvar` data type as a loop index. Furthermore, the compile-time `for` loop should only contain waveform generation/editing operations and it can't contain any variables of type `var`. The following code example shows both versions of the loop.

```
// For loop syntax
for (initialization; end-expression; iteration-expression) {
  // Statements to execute while end-expression evaluates to true
}
```

```
// FOR loop example to assemble a train of pulses into
// a single waveform (compile-time execution)
cvar gain_factor; // CVAR: integer or float values allowed
wave w_pulse_series;
for (gain_factor = 0; gain_factor < 1.0; gain_factor = gain_factor + 0.1) {
  w_pulse_series = join(w_pulse_series, gain_factor*gauss(1008, 504, 100));
}

// Playback of waveform defined using compile-time FOR loop
playWave(w_pulse_series);

// FOR loop example to vary waiting time between
// waveform playbacks (run-time execution)
var i; // VAR: integer values allowed
for (i = 0; i < 1000; i = i + 100) {
  playWave(gauss(1008, 504, 100));
  waitWave();
  wait(i);
}
```

The while loop is a simplified version of the `for` loop. The `end-expression` is evaluated at the start of each loop iteration. The contents of the loop are executed as long as this expression is true. Like the `for` loop, this loop comes in a compile-time version (if the end-expression involves only `cvar` and `const`) and in a run-time version (if the end-expression involves also `var` data types).

```
// While loop syntax
while (end-expression) {
 // Statements to execute while end-expression evaluates to true
}
```

```
// While loop example
const STOP_BIT = 0x8000;
var run = 1;
var i = 0;
var dio = 0;
while (run) {
  dio = getDIO();
  run = dio & STOP_BIT;

  dio = dio | (i & 0xff);
  setDIO(dio);
  i = i + 1;
}
```

The **repeat loop** is a simplified version of the `for` loop. It repeats the contents of the loop a fixed number of times. In contrast to the `for` loop, the repetition number of the `repeat` loop must be known at compile time, i.e., `const-expression` can only depend on constants and not on variables. Unlike the `for` and the `while` loop, this loop comes only in a run-time version. Thus, no `cvar` data types may be modified in the loop body.

```
// Repeat loop syntax
repeat (constant-expression) {
  // Statements to execute
}
```

```
// Repeat loop example
repeat (100) {
  setDIO(0x1);
  wait(10);
  setDIO(0x0);
  wait(10);
}
```

5.24.4. Functional Elements

Table 5.73: AWG tab: Control sub-tab

Control/Tool	Option/ Range	Description
Start		Runs the AWG.
Sampling Rate	220 kSa/s to 1.8 GSa/s	AWG sampling rate. This value is used by default and can be overridden in the Sequence program. The numeric values are rounded for display purposes. The exact values are equal to the base sampling rate divided by 2^n, where n is an integer between 0 and 13.
Round oscillator frequencies.		Round oscillator frequencies to nearest commensurable with 225 MHz.
Amplitude (FS)	0.0 to 1.0	Amplitude in units of full scale of the given AWG Output. The full scale corresponds to the Range voltage setting of the Signal Outputs.
Mode		Select between plain mode, amplitude modulation, and advanced mode.
	Plain	AWG Output goes directly to Signal Output.
	Modulation	AWG Output is multiplied with a sinusoid carrier signal. On UHFLI instruments, AWG Output 1 (2) is multiplied with oscillator signal of demodulator 4 (8). On UHFQA instruments, AWG Output 1 (2) is multiplied with the in-phase (quadrature) signal of the internal oscillator represented in the In/Out tab.
	Advanced	Output of AWG channel 1 (2) modulates demodulators 1-4 (5-8) with independent envelopes. Option not supported on UHFQA instruments.
Status		Display compiler errors and warnings.
Compile Status	grey/green/ yellow/red	Sequence program compilation status. Grey: No compilation started yet. Green: Compilation successful. Yellow: Compiler warnings (see status field). Red: Compilation failed (see status field).
Upload Progress	0% to 100%	The percentage of the sequencer program already uploaded to the device.
Upload Status	grey/yellow/ green	Indicates the upload status of the compiled AWG sequence. Grey: Nothing has been uploaded. Yellow: Upload in progress. Green: Compiled sequence has been uploaded.
Register selector		Select the number of the user register value to be edited.
Register	0 to 2^32	Integer user register value. The sequencer has reading and writing access to the user register values during run time.

Control/Tool	Option/ Range	Description
Input File		External source code file to be compiled.
Example File		Load pre-installed example sequence program.
New	New	Create a new sequence program.
Revert	Revert	Undo the changes made to the current program and go back to the contents of the original file.
Save (Ctrl+S)	Save	Compile and save the current program displayed in the Sequence Editor. Overwrites the original file.
Save as... (Ctrl+Shift+S)	Save as...	Compile and save the current program displayed in the Sequence Editor under a new name.
Automatic upload	ON / OFF	If enabled, the sequence program is automatically uploaded to the device after clicking Save and if the compilation was successful.
To Device	To Device	Sequence program will be compiled and, if the compilation was successful, uploaded to the device.
Multi-Device	ON / OFF	Compile the program for use with multiple devices. If enabled, the program will be compiled for and uploaded to the devices currently synchronized in the Multi-Device Sync tab.
Sync Status	grey/green/ yellow	Sequence program synchronization status. Grey: No program loaded on device. Green: Program in sync with device. Yellow: Sequence program in editor differs from the one running on the device.

Table 5.74: AWG tab: Waveform sub-tab

Control/ Tool	Option/ Range	Description
Wave Selection		Select wave for display in the waveform viewer. If greyed out, the corresponding wave is too long for display.
Waveforms		Lists all waveforms used by the current sequence program.
Mem Usage (%)	0 to 100	Amount of the used waveform data relative to the device cache memory. The cache memory provides space for 32 kSa of waveform data. Mem Usage > 100% means that waveforms must be loaded from the main memory (128 MSa per channel) during playback, which can lead to delays.

Table 5.75: AWG tab: Trigger sub-tab

Control/ Tool	Option/Range	Description
Force	Force	Enforce a trigger event.
Trigger State	grey/green	State of the Trigger. Grey: No trigger detected. Green: Trigger detected.
Signal		Selects the analog trigger source signal. Navigate through the tree view that appears and click on the required signal.
Slope		Select the signal edge that should activate the trigger. The trigger will be level sensitive when the Level option is selected.
	Level	Level sensitive trigger.
	Rise	Rising edge trigger.
	Fall	Falling edge trigger.
	Both	Rising or falling edge trigger.
Level (V)	numeric value	Defines the analog trigger level.

Control/ Tool	Option/Range	Description
Hysteresis Mode		Selects the mode to define the hysteresis size. The relative mode will work best over the full input range as long as the analog input signal does not suffer from excessive noise.
	Hysteresis (V)	Selects absolute hysteresis.
	Hysteresis (%)	Selects a hysteresis relative to the adjusted full scale signal input range.
Hysteresis (V)	trigger signal range (positive values only)	Defines the voltage the source signal must deviate from the trigger level before the trigger is rearmed again. Set to 0 to turn it off. The sign is defined by the Edge setting.
Hysteresis (%)	numeric percentage value (positive values only)	Hysteresis relative to the adjusted full scale signal input range. A hysteresis value larger than 100% is allowed.
Gating	Trigger In 4	Select the signal source used for trigger gating if gating is enabled.
	Trigger In 3	
Gating enable	ON / OFF	If enabled the trigger will be gated by the trigger gating input signal.
Auxiliary Trigger State	grey/green	State of the Auxiliary Trigger. Grey: No trigger detected. Green: Trigger detected.
Signal		Selects the digital trigger source signal.
DIO/Zsync Trigger state	grey/green	Indicates that triggers are generated from the DIO or ZSync interface to the AWG.
Strobe Index	16 to 31	Selects the index n of the DIO interface bit (notation DIO[n] in the Specification chapter of the User Manual) to be used as a STROBE signal input in connection with the waitDIOTrigger sequencer instruction.
Strobe Slope		Select the signal edge that activates the STROBE trigger in connection with the waitDIOTrigger sequencer instruction.
	None	Off
	Rise	Rising edge trigger
	Fall	Falling edge trigger
	Both	Rising or falling edge trigger
Valid Index	16 to 31	Selects the index n of the DIO interface bit (notation DIO[n] in the Specification chapter of the User Manual) to be used as a VALID signal input, i.e. a qualifier indicating that a valid codeword is available on the DIO interface.
Valid Polarity		Polarity of the VALID bit that indicates that a codeword is available on the DIO interface.
	None	VALID bit is ignored.
	Low	VALID bit must be logical low.
	High	VALID bit must be logical high.
	Both	VALID bit may be logical high or logical low.

Table 5.76: AWG tab: Advanced sub-tab

Control/ Tool	Option/ Range	Description
Sequence Editor		Display and edit the sequence program.
Assembly	Text display	Displays the current sequence program in compiled form. Every line corresponds to one hardware instruction.
AWG Core		Display assembly information.

Control/ Tool	Option/ Range	Description
Counter		Current position in the list of sequence instructions during execution.
Status	Running, Idle, Waiting	Displays the status of the sequencer on the instrument. Off: Ready, not running. Green: Running, not waiting for any trigger event. Yellow: Running, waiting for a trigger event. Red: Not ready (e.g., pending elf download, no elf downloaded)
Rerun	ON / OFF	Reruns the Sequencer program continuously. This way of looping a program results in timing jitter. For a jitter free signal implement a loop directly in the sequence program.
Mem Usage (%)	0 to 100	Size of the current sequence program relative to the device cache memory. The cache memory provides space for a maximum of 1024 instructions.
Status	grey/green/ red	Displays the status of the command table of the selected AWG Core. Grey: no table description uploaded, Green: table description successfully uploaded, Red: Error occurred during uploading of the table description.

5.25. Pulse Counter Tab

The Pulse Counter tab relates to the UHF-CNT Pulse counter option and is only available if this option is installed on the instrument (see Information section in the Device tab).

5.25.1. Features

- 4 counter modules
- 225 MHz maximum count rate
- 4 modes: free running, gated, gated free running, and pulse tagging
- 4 analog signal inputs with adjustable discriminator level
- 32 digital signal inputs
- Background subtraction
- Count integration

5.25.2. Description

The Pulse Counter tab provides access to the pulse counter settings. 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 5.77: App icon and short description

Control/ Tool	Option/ Range	Description
Counter		Configure the Pulse Counters for analysis of pulse trains on the digital signal inputs.

The Pulse Counter tab shown in LabOne UI: Counter tab consists of four side-tabs, one for each Counter module. The Enable button and the Mode selector are the main controls that determine if and how a Counter unit generates an output. The output is displayed in the Value field and is available in the Plotter and Data Acquisition tab.

Figure 5.50: LabOne UI: Counter tab

The counter Input signal is selectable among the Trigger inputs as well as any of the 32 DIO channels on the VHDCI connector on the instrument rear panel. The trigger level of the analog trigger inputs is configurable in the DIO tab. The following operation modes are available.

- Free running: the counter is active during repeated periods defined by the a configurable timer. The timer period is controlled by the Period field. At the beginning of the period the counter is reset, and at the end, the accumulated number of counts is output.
- Gated: the counter is controlled with the Gate Input signal. The counter is enabled at the rising edge of the Gate Input signal and disabled at the falling edge. Pulses are counted as long as the counter is enabled. The accumulated number of counts is output on the falling edge of the Gate Input signal.
- Gated free running: the counter runs on a repetitive time base defined by the Period field. The Gate Input signal controls when the counter is allowed to run. The counter as well as the timer is reset when the Gate Input signal is low. The counter will only deliver new values if the Gate Input signal is high for a time longer than the configured Period.
- Time tagging: every single event is counted and transmitted to the server along with a time tag. The Period defines the minimum hold-off time between the tagging of two subsequent pulses. If more than one pulse occurs within the window defined by the Period, then the pulses are accumulated and output at the end of the window. The Period effectively determines the maximum rate at which pulse information can be transmitted to the host PC.

Background subtraction or summation of data from two counter modules is controlled by the Operation field. For add and subtract operations, counter units 1 is grouped with unit 2, and unit 3 is grouped with unit 4. The Pulse Counter supports integration of counter data over time. The integer timestamp in all recorded data is in units of the instrument data clock, 1.8 GHz.

Note

It is recommended to use the 1GbE interface rather than the USB interface in combination with the Pulse Counter. 1GbE provides a higher stability at high data rates, namely in time tagging mode or in free-running mode with a small Period setting.

5.25.3. Functional Elements

Table 5.78: Pulse Counter tab

Control/ Tool	Option/ Range	Description
Enable	ON / OFF	Enable the pulse counter unit.
Mode		Select the run mode of the counter unit.
	Free Running	The counter runs on a repetitive time base defined by the Period field. At the beginning of each period the counter is reset, and at the end, the accumulated number of counts is output.
	Gated Free Running	The counter runs on a repetitive time base defined by the Period field. The Gate Input signal controls when the unit counter is allowed to run. The counter as well as the timer is reset when the Gate Input signal is low. The counter will only deliver new values if the Gate Input signal is high for a time longer than the configured Period.
	Gated	The counter is controlled with the Gate Input signal. The counter is enabled at the rising edge of the Gate Input signal and disabled at the falling edge. Pulses are counted as long as the counter is enabled. The accumulated number of counts is output on the falling edge of the Gate Input signal.
	Time Tagging	Every pulse is detected individually and tagged with the time of the event. The Period defines the minimum hold-off time between the tagging of two subsequent pulses. If more than one pulse occurs within the window defined by the Period, then the pulses are accumulated and output at the end of the window. The Period effectively determines the maximum rate at which pulse information can be transmitted to the host PC.
Period	44.4 ns to 19 s	Set the period used for the Free Running and Gated Free Running modes. Also sets the hold-off time for the Time Tagging mode.

Control/ Tool	Option/ Range	Description
Input	Ref/Trigger Input 1/2, Trigger Input 3/4, DIO Bit 0-31	Select the counter signal source.
Edge Rise	ON / OFF	Performs a trigger event when the source signal crosses the trigger level from low to high. For dual edge triggering, select also the falling edge.
Edge Fall	ON / OFF	Performs a trigger event when the source signal crosses the trigger level from high to low. For dual edge triggering, select also the rising edge.
Gate Input	Ref/Trigger Input 1/2, Trigger Input 3/4, AWG internal Trigger 1-4	Select the signal source used for enabling the counter in the Gated Free Running and Gated modes.
Operation	Subtract Other Counter	Select the arithmetic operation (addition, subtraction) applied to the counter unit outputs. "Other counter" refers to the grouping of the counter units: 1 with 2, and 3 with 4.
	None	
	Add Other Counter	
Integrate	ON / OFF	Sum up counter values over time.
Value		Displays the counter output value.

5.26. Multi Device Sync Tab

The Multi Device Sync (MDS) tab gives access to the automatic timing synchronization of measurement data from multiple UHF instruments. This functionality and tab is available on all UHF instruments.

5.26.1. Features

- Automatic timing synchronization across instruments
- Periodic check of synchronization
- Selectable instrument subgroup
- Status display

5.26.2. Description

The Multi Device Sync tab contains the controls and status information for synchronized measurements on multiple instruments. 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 5.79: App icon and short description

Control/Tool	Option/Range	Description
MDS		Synchronize multiple instruments.

The Multi Device Sync tab shown in Figure 5.51 consists of the Available Devices section, a Status section, and a wiring diagram.

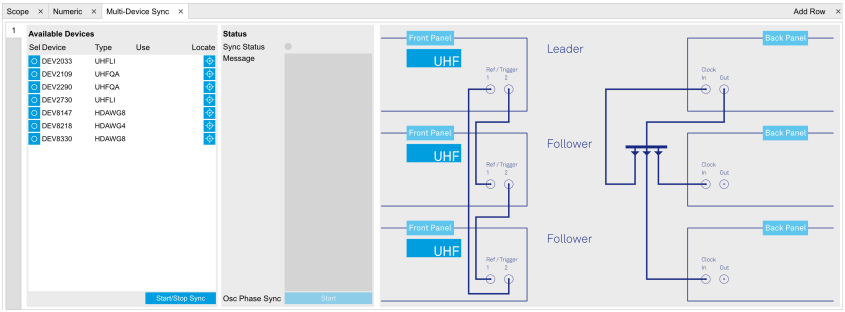


Figure 5.51: LabOne UI: Multi Device Sync tab

The Multi Device Synchronization feature provides an automated functionality to remove the clock offset of separate UHF instruments. This enables a correct simultaneous display of their data in the Plotter Tab and helps when analyzing recorded data. In multi-channel applications that require sub-microsecond timing precision, the user can therefore benefit from having synchronized data from the start, rather than having to manually measure and compensate the clock offset in post-processing.

The first prerequisite for automatic synchronization is that all instruments are connected to the same LabOne Data Server (see Connecting to the Instrument).

To make these connections, click on Session Manager in the Config Tab to open the Device Connection dialog. Go to the Advanced view of this dialog and click on the Enable checkbox next to the corresponding entries in the Available Devices list. Once all instruments are connected, they are selectable in the Tree Selector of a newly opened Plotter tab allowing you to visualize their data simultaneously, though by default these data are not synchronized yet. The settings of multiple instruments can be accessed in parallel by opening a new Web Server session for each of them. This is done by opening a new browser tab and connecting to `localhost:8006` or `127.0.0.1:8006`, respectively, and then double-clicking the respective instrument entry in the Available Devices list. With multiple instruments connected to the same Data Server, tabs that are available for several instruments will feature a device selector as shown in Figure 5.52.

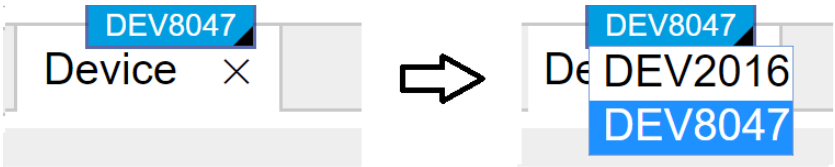


Figure 5.52: Example of the device selector for the Device tab

The second prerequisite for automatic synchronization is correct cabling of the instruments explained in the diagram in Figure 5.53. The instruments should share the same 10 MHz reference clock and communicate via the Ref / Trigger connectors for absolute timing synchronization. The 10 MHz clock signal is distributed in a star arrangement, where the signal of an external clock generator is sent to the Clk 10 MHz In connector of all instruments. On all instruments, the Clock Source in the Device Tab needs to be set to Clk 10 MHz rather than Internal. The bidirectional Ref / Trigger connectors are to be connected in a loop arrangement, where Ref / Trigger 1 of one instrument is connected to Ref / Trigger 2 of the next instrument, and so forth, until the loop is closed.

Note

Alternatively to using an external 10 MHz clock source, it's possible to distribute the Clk 10 MHz Out signal from the leading instrument to the Clk 10 MHz In connector of all following instruments **and** the leading instrument using a 1-to-N power divider and cables of equal length. For a large number of instruments, a clock distribution amplifier rather than a power divider is required.

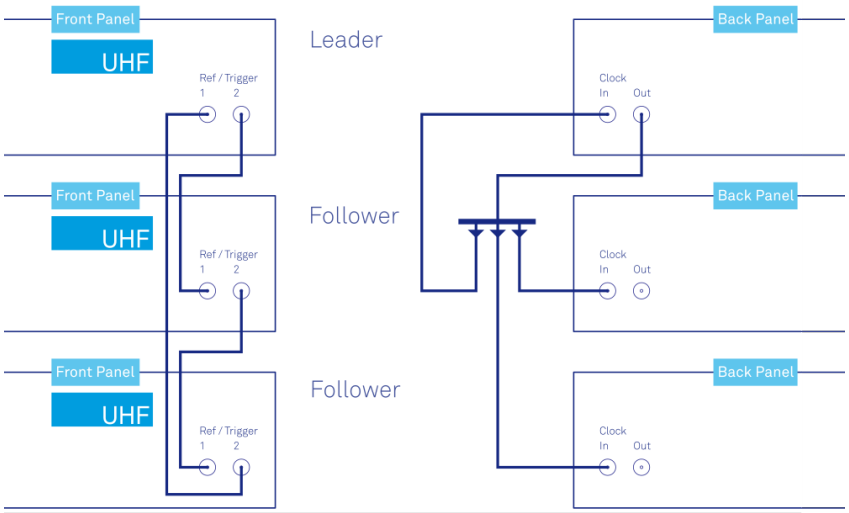


Figure 5.53: Cabling for automatic synchronization of multiple UHF instruments

Once the cabling and the connectivity is set up correctly, automatic synchronization is started in the Multi Device Sync tab by checking the Enable button on the instruments in the Available Devices list, and then clicking on Start/Stop Sync. The sequence assignment of the instruments (Leader, Follower 1, Follower 2,...) can be defined by the order in which the Enable button is clicked. This assignment has to agree with the way the cabling is made. The sequence is also relevant for addressing the instruments in an AWG sequence program. The Message display on the right will then report on the progress, and the Sync Status LED will turn green if the synchronization was successful. In that case, visualizing a time-dependent measurement of multiple instruments in the Plotter will demonstrate the timing synchronization.

5.26.3. Functional Elements

Table 5.80: Multi Device Sync tab

Control/Tool	Option/ Range	Description
Start Sync	Start/Stop Sync	Start the automatic synchronization of the selected devices.
Sync Status		Indicates the status of the synchronization within this group. Green: synchronization successful. Yellow: synchronization in progress. Red: error (see message).
Message		Displays a status message of the synchronization group.
Cabling		This image shows how to connect the devices for device synchronization.
Phase Synchronization		Reset phases of all oscillators on all synchronized devices.
Identify Device		Make device's front LED blink

5.27. 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 5.81: App Icon and short description

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

5.28. Upgrade Tab

The Upgrade tab serves as a source of information about the possible upgrade options for the instrument in use. The tab has no functional purpose but provides the user with a quick link to further information about the upgrade options online.

6. Specifications

Important

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

Important

An internal calibration is performed 10 minutes after powering the instrument. This internal calibration is essential to achieve the specifications of the system. Further it is required to perform the internal calibration after 7 days of instrument use. This automatic calibration is turned on by default and can be configured in the Device tab.

Important

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

6.1. General Specifications

Table 6.1: General and storage

Parameter	min	typ	max
storage temperature	–25 °C	-	65 °C
storage relative humidity (non-condensing)	-	-	95%
operating temperature	5 °C	-	40 °C
operating relative humidity (non-condensing)	-	-	90%
specification temperature	18 °C	-	28 °C
power consumption	-	-	150 W
operating environment	IEC61010, indoor location, installation category II, pollution degree 2		
operating altitude	up to 2000 meters		
power inlet fuses	250 V, 2 A, fast acting, 5 x 20 mm		
power supply AC line	100-240 V (±10%), 50/60 Hz		
dimensions with handles and feet	45.0 x 34.5 x 10.0 cm, 17.7 x 13.6 x 3.9 inch, 19 inch rack compatible		
weight	6.4 kg		
recommended calibration interval	2 years		

Table 6.2: Maximum ratings

Parameter	min	typ	max
damage threshold Signal Input 1 and 2	–5 V	-	+5 V
damage threshold Signal Output 1 and 2	–2.5 V	-	+2.5 V
damage threshold Ref / Trigger 1 and 2	–6 V	-	+6 V
damage threshold Trigger Out 1 and 2	–1 V	-	+6 V

Parameter	min	typ	max
damage threshold Trigger In 1 and 2	−6 V	-	+6 V
damage threshold Aux Output 1, 2, 3, 4	−12 V	-	+12 V
damage threshold Aux In 1 and 2	−12 V	-	+12 V
damage threshold DIO (digital I/O)	−1 V	-	+6 V
damage threshold Clk In and Clk Out	−5 V	-	+5 V

Table 6.3: Host computer requirements

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

Table 6.4: Demodulator output sample rate to host computer.

Host computer connection	Active demodulators	Maximum sample rate per demodulator	Comments
1 GbE	1	1.6 MSa/s	To achieve highest rates, it is advised to remove all other data transfer that loads the LAN/USB interface. It is recommended to check the sample loss flag (in the status tab) from time to time when using high readout rate settings.
	2 - 4	800 kSa/s	
	5 - 8	400 kSa/s	
USB 2.0	1 - 2	400 kSa/s	
	3 - 6	200 kSa/s	
	7 - 8	100 kSa/s	

Note

The sample readout rate is the rate at which demodulated samples are transferred from the Instrument to the host computer. This rate has to be set to at least 2 times the signal bandwidth of the related demodulator in order to satisfy the Nyquist sampling theorem. As the total rate is limited by the USB/LAN interface, the maximum rate becomes smaller when the number of active demodulators is increased. This is summarized in the table above. An up-to-date and performing host computer is required to achieve these rates.

6.2. Analog Interface Specifications

Table 6.5: UHF signal inputs

Parameter	Conditions	min	typ	max
connectors	-	BNC, front panel single-ended		
input impedance	low value	-	50 Ω	-
	high value	-	1 MΩ // 16 pF	-
input frequency range	50 Ω termination	DC	-	600 MHz
input frequency range	1 MΩ termination	DC	-	100 MHz

Parameter	Conditions	min	typ	max
input A/D conversion	-	12 bit, 1.8 GSa/s		
input noise amplitude	> 100 kHz, 10 mV range, 50 Ω termination	-	4 nV/ √Hz	-
input bias current	50 Ω termination	-	10 µA	-
	1 MΩ termination	-	-	1 nA
input full range sensitivity (10 V lock-in amplifier output)	-	1 nV	-	1.5 V
input AC ranges	-	10 mV	-	1.5 V
input range (AC + common mode)	DC coupling	–1.5 V	-	+1.5 V
	AC coupling	–3.5 V	-	+3.5 V
AC coupling cutoff frequency	50Ω termination	-	320 kHz	-
	1 MΩ termination	-	80 Hz	-
input amplitude accuracy	< 100 MHz	-	3 %	-
	> 100 MHz	-	10 %	-
input amplitude stability	-	-	0.1 %/°C	-
input offset amplitude	with respect to range	-	-	5%
input harmonic distortion (HD2/ HD3)	1 Vpp, 50 Ω termination, 10 minutes after manual input calibration < 1 MHz	-	–75 dB	-
	< 10 MHz	-	–70 dB	-
	< 100 MHz	-	–60 dB	-
	> 100 MHz	-	–50 dB	-
dynamic reserve		-	90 dB	100 dB

Table 6.6: UHF signal outputs

Parameter	Conditions	min	typ	max
connectors	-	BNC, front panel single-ended		
output impedance	-	-	50 Ω	-
output frequency range	-	DC	-	600 MHz
output rise time / fall time	10% to 90%	-	750 ps	-
output frequency resolution	-	-	6 µHz	-
output phase range	-	–180 °	-	180 °
output phase resolution	-	-	1.0 µ°	-
output D/A conversion	-	14 bit, 1.8 GSa/s		
output amplitude ranges	-	±150 mV, ±1.5 V		
output DC offset range	-	±150 mV or ±1.5 V, equal to the set output amplitude range		
output power	-	-	-	7.5 dBm
output amplitude accuracy	< 100 MHz	-	2%	-
	> 100 MHz	-	5%	-
output harmonic distortion (HD2/HD3)	1 Vpp, 50 Ω termination, < 1 MHz	-	–70 dB	-

Parameter	Conditions	min	typ	max
	< 10 MHz	-	–70 dB	-
	< 100 MHz	-	–55 dB	-
	> 100 MHz	-	–42 dB	-
output noise amplitude	> 100 kHz	-	25 nV/√Hz	-
output phase noise	10 MHz, BW = 0.67 Hz, offset 100 Hz	-	–120 dBc/Hz	-
	10 MHz, BW = 0.67 Hz, offset 1 kHz	-	–130 dBc/Hz	-
output random jitter (RMS)	100 MHz, 6 dBm sine	-	4.5 ps	-
output offset amplitude	-	–5 mV	-	5 mV
output drive current	-	-	-	100 mA

Table 6.7: Reference signals and reference modes

Parameter	Conditions	min	typ	max
connectors	-	BNC, front panel bidirectional SMA, back panel input SMA, back panel output		
input impedance (front and back panel)	low value	-	50 Ω	-
	high value	-	1 kΩ	-
input level at Ref / Trigger (front panel) and Trigger In (back panel)	low input impedance	–2.5 V	-	+2.5 V
	high input impedance	–5 V	-	+5 V
output impedance (front and back panel)	-	-	50 Ω	-
output level (front and back panel)	-	-	-	3.3 V TTL
input trigger hysteresis	-	-	100 mV	-
internal reference mode, output of reference on UHF outputs	frequency range	1 mHz	-	600 MHz
	reference orthogonality	-	0 °	-
	reference acquisition time, lock time	instantaneous		
internal reference mode, output of reference on Ref / Trigger	frequency range	1 mHz	-	200 MHz
	reference orthogonality	-	0 °	-
	reference acquisition time, lock time	instantaneous		
external reference mode and auto reference mode, reference input at Signal Input 1 and 2	frequency range	10 Hz	-	600 MHz
	amplitude, note: for low-swing input signals the gain should be set to full-swing range to achieve best performance	100 mV	-	-
	amplitude (using UHF-PID option), note: for low-swing input signals the gain should be set to full-swing range to achieve best performance	10 mV	-	-
	reference acquisition time, lock time	-	-	100 reference cycles or 1.2 ms whatever is larger

Parameter	Conditions	min	typ	max
external reference mode, reference input at Ref / Trigger	signal type	arbitrary, active at rising edge		
	frequency range	10 Hz	-	600 MHz
	amplitude	250 mV	-	-
	reference acquisition time, lock time	-	-	100 reference cycles or 1.2 ms, whatever is larger

Note

The UHF Instrument uses the same connectors for reference and trigger signals. This applies to input signals as well as output signals. Overall, the instrument features 2 output, 2 input, and 2 bidirectional connectors for reference and triggering purposes.

Table 6.8: Demodulators

Parameter	Details	min	typ	max
demodulator number	-	8		
demodulator harmonic setting range	-	1	-	1023
demodulator filter time constant	-	30 ns	-	76 s
demodulator measurement bandwidth	-	628 µHz	-	5 MHz
demodulator filter slope / roll-off	-	6, 12, 18, 24, 30, 36, 42, 48 dB/oct, consisting of up to 8 cascaded critical damping filters		
demodulator output resolution	-	X, Y, R, Θ with 64-bit resolution		
demodulator output sample rate (readout rate), for detailed specifications refer to Table 6.4	on auxiliary outputs	-	-	28 MSa/s
	USB 2.0 high speed	-	-	400 kSa/s
	1GbE, 1 Gbit/s LAN	-	-	1.6 MSa/s
demodulator harmonic rejection	-	110 dBc	-	-
group delay (lag time from Signal Input to Aux Output)	30 ns time constant and 1st order filter	-	-	3 µs

Table 6.9: Auxiliary Inputs and Outputs

Parameter	Details	min	typ	max
auxiliary output	connectors	BNC, 4 outputs on front-panel		
	sampling	28 MSa/s, 16-bit		
	bandwidth	-	-	7 MHz
	impedance	-	50 Ω	-
	amplitude	–10 V	-	10 V
	resolution	0.3 mV	-	-
	drive current	-	-	100 mA
auxiliary input	connectors	SMA, 2 inputs on back-panel		
	sampling	400 kSa/s, 16-bit		
	bandwidth	-	-	100 kHz

Parameter	Details	min	typ	max	
	impedance	-	1 MΩ	-	
	amplitude	–10 V	-	10 V	
	resolution	0.3 mV	-	-	

Table 6.10: Oscillator and clocks

Parameter	Details	min	typ	max
internal clock (ovenized crystal)	initial accuracy	-	±0.5 ppm	±1 ppm
	long term accuracy / aging	-	-	±0.4 ppm/ year
	short term stability (1 s)	0.00005 ppm	-	-
	short term stability (100 s)	0.0005 ppm	-	-
	temperature coefficient (23° ± 5°)	-	-	±0.03 ppm/°
	phase noise (at 100 Hz)	-	–130 dBc/Hz	-
	phase noise (at 1 kHz)	-	–140 dBc/Hz	-
	warm-up time	-	-	60 s
UHF-RUB Rubidium clock (option)	initial accuracy at 25°	-	-	±0.0005 ppm
	long term accuracy / aging	-	- a	±5e-6 ppm/ day ±0.0005 ppm/ year
	short term stability, AVAR (1 s)	0.00008 ppm	-	-
	short term stability, AVAR (100 s)	0.000008 ppm	-	-
	temperature coefficient (25° ± 25°)	-	-	±0.0005 ppm/ °
	phase noise (at 100 Hz)	-	-	-
	phase noise (at 1 kHz)	-	–140 dBc/Hz	-
	warm-up time	-	-	300 s @ 25°C
clock input	connector 3.+	SMA, on back-panel		
	impedance	-	50 Ω	-
	amplitude	200 mV	320 mV	1 V
	frequency	9.98 MHz	10 MHz	10.02 MHz
clock output	connector 3.+	SMA, on back-panel		
	impedance	-	50 Ω	-
	amplitude, 50 Ω	250 mV	500 mV	1 V
	frequency	-	10 MHz	-

6.3. Digital Interface Specifications

Table 6.11: Digital interfaces

Parameter	Description
host computer connection	USB 2.0 high-speed, 480 Mbit/s
	1GbE, LAN / Ethernet, 1 Gbit/s
DIO port	4 x 8 bit, general purpose digital input/output port, 5 V TTL specification
ZCtrl peripheral port	2 connectors for ZI proprietary bus to control external peripherals

6.3.1. DIO Port

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

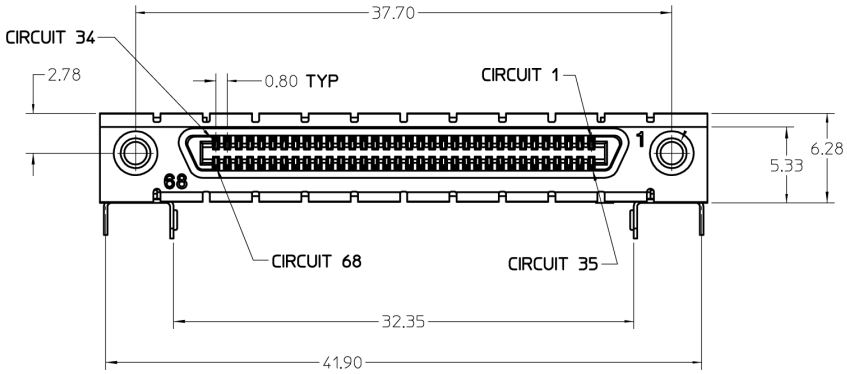


Figure 6.1: DIO HD 68 pin connector

Table 6.12: Electrical Specifications

Parameter	Details	Min	Typ	Max
output series termination	DO	low impedance (CMOS output)		
output series termination	DOL	33 Ω		
input termination	DI, CLKI	high impedance (CMOS input)		
high-level input voltage VIH	DI, CLKI	2.0 V	-	-
low-level input voltage VIL	DI, CLKI	-	-	0.8 V
high-level output voltage VOH	DO, at IOH < 24 mA	4.2 V	-	-
high-level output voltage VOH	DOL, at IOH < 32 mA	3.8 V	-	-
low-level output voltage VOL	DO, at IOL < 24 mA	-	-	0.55 V
low-level output voltage VOL	DOL, at IOL < 32 mA	-	-	0.55 V
high-level output current IOH (sourcing)	DO	-	-	24 mA
high-level output current IOH (sourcing)	DOL	-	-	32 mA
low-level output current IOL (sinking)	DO	-	-	24 mA
low-level output current IOL (sinking)	DOL	-	-	32 mA

Table 6.13: DIO pin assignment

Pin	Name	Description	Range specification
68	CLKI	clock input, used to latch signals at the digital input ports - can also be used to retrieve digital signals from the output port using an external sampling clock	5 V CMOS/TTL
67	DOL	DIO output latch, 56.25 MHz clock signal, the digital outputs are synchronized to the falling edge of this signal	5 V CMOS

Pin	Name	Description	Range specification
66-59	DI[31:24]	digital input or output (set by user)	output CMOS 5 V, input is CMOS/TTL
58-51	DIO[23:16]	digital input or output (set by user)	output CMOS 5 V, input is CMOS/TTL
50-43	DIO[15:8]	digital input or output (set by user)	output CMOS 5 V, input is CMOS/TTL
42-35	DIO[7:0]	digital input or output (set by user)	output CMOS 5 V, input is CMOS/TTL
34-1	GND	digital ground	-

The figure below shows the architecture of the DIO input/output. The DIO port features 32 bits that can be configured byte-wise as inputs or outputs by means of a drive signal. The digital output data is latched synchronously with the falling edge of the internal clock, which is running at 56.25 MHz. The internal sampling clock is available at the DOL pin of the DIO connector. Digital input data can either be sampled by the internal clock or by an external clock provided through the CLKI pin. A decimated version of the input clock is used to sample the input data. The Decimation unit counts the clocks to decimation and then latches the input data. The default decimation is 5625000, corresponding to a digital input sampling rate of 1 sample per second.

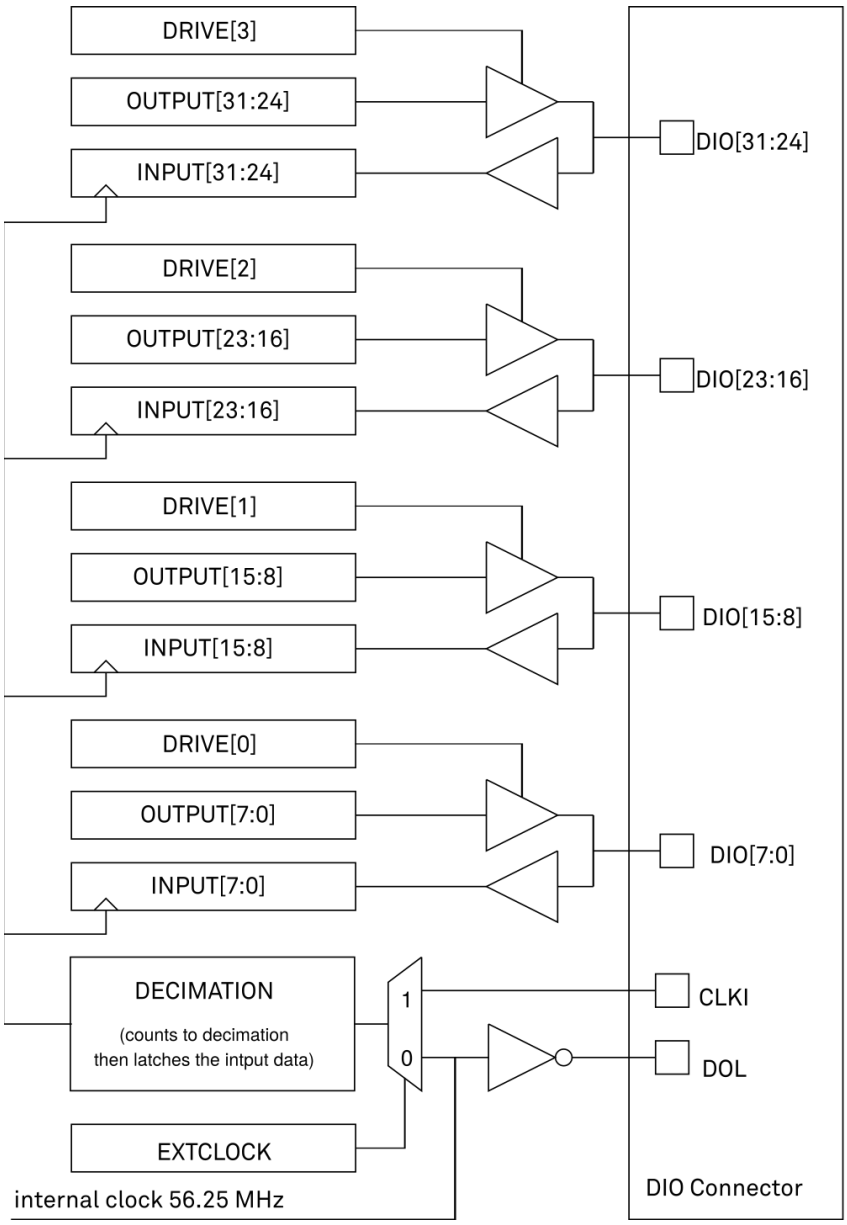


Figure 6.2: DIO input/output architecture

6.3.2. ZCtrl Peripheral Port

The ZCtrl port serves to power and communicate to external equipment, such as pre-amplifiers: the port provides a floating power supply with ±14.5 V and 100 mA per port. After Instrument power-on, the port is not active and must be switched on in order to be used. Two activation methods are supported:

- Manual switch in the user interface
- Manual switch by shorting the ZCtrl_Detect and Device_Ground - these pins should be floating against ZCtrl_GND and ZCtrl_PWR

Th ZCtrl port can be connected with an RJ45 connector, therefore non-crossed Ethernet cables can be used for convenient interfacing.

Warning

Connection to a Ethernet might damage the UHF Instrument.

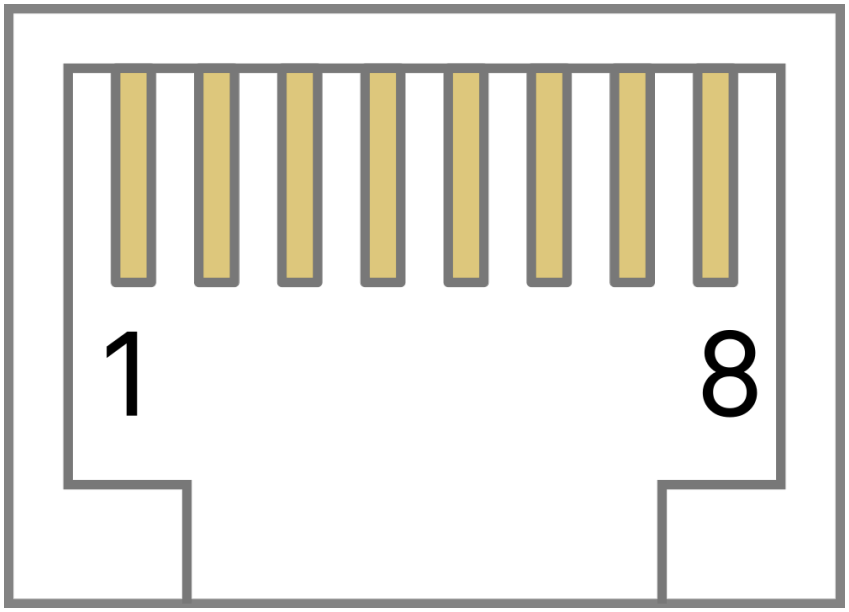


Figure 6.3: The pinout of the ZCtrl port

Table 6.14: DIO port pin assignment

Pin	Name	Description	Range specification
1	ZCtrl_Power+	power pin, for external use	14.5 V, 100 mA
2	ZCtrl_Detect	connection detection	-
3	Device_Ground	ground of UHF Instrument, connected to earth pin	-
4	ZCtrl_Power–	power pin, for external use	–14.5 V, 100 mA
5	ZCtrl_D	proprietary function	-
6	ZCtrl_C	proprietary function	-
7	ZCtrl_GND	floating input	-
8	ZCtrl_GND	reference ground pin for ZCtrl_Power+ and ZCtrl_Power–	-

6.4. Performance Diagrams

Many of the parameters mentioned in Analog Interface Specifications are valid without specific conditions. Other parameters instead are typical specifications, because they depend on several parameters, such as the input range setting, the input termination and/or the frequency. This

section completes the previous chapters with detailed performance diagrams in order to support the validation of applications.

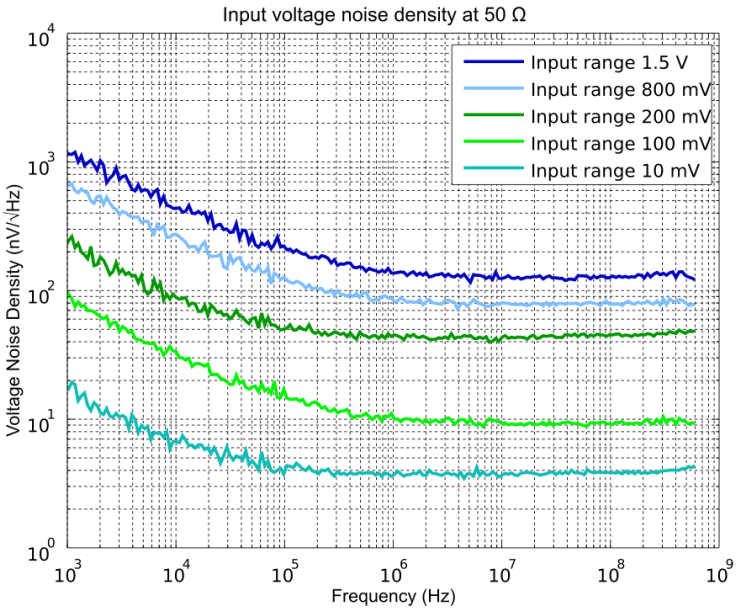


Figure 6.4: UHF Series input noise with 50Ω input impedance

Input noise amplitude depends on several parameters, and in particular on the frequency and the input range setting. The input noise is lower for smaller input ranges, and it is recommended to use small ranges especially for noise measurements. Only the noise with DC input coupling is shown here as the input noise with AC coupling is the same, as long as the frequency is above the AC cutoff frequency (see Table 6.5).

The input noise does depend on the input impedance setting, which can be 50 Ω or 1 MΩ. The performance diagrams for 50 Ω and 1 MΩ input impedance are shown in Figure 6.4 and in Figure 6.5, respectively. For both, the corner frequency of the 1/f noise is in the range of 100 kHz. For 50 Ω input impedance, the white noise floor is around 4 nV/√Hz for the smallest input range. For 1 MΩ input impedance, the white noise floor is below 8 nV/√Hz for the smallest input range.

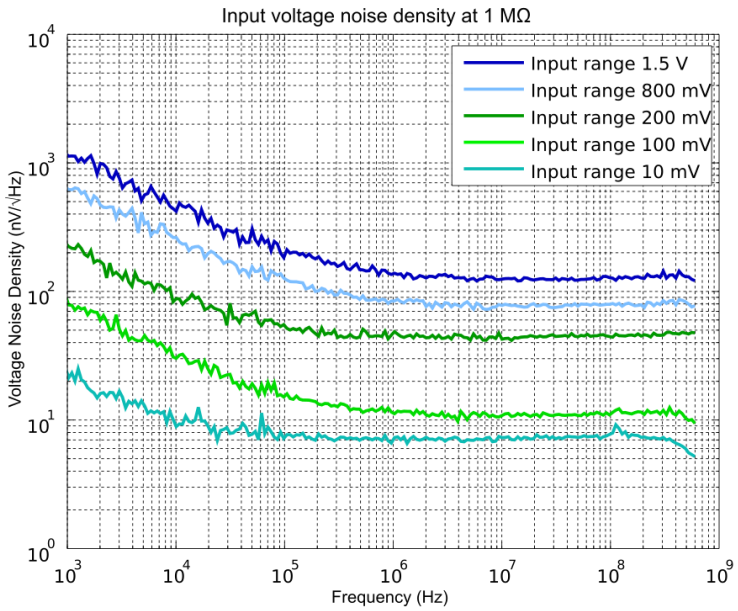


Figure 6.5: UHF Series input noise with 1 MΩ input impedance

Figure 6.6 shows a typical SSB phase noise measured at the signal output. For this measurement, the UHF instrument was connected to a phase noise analyzer and the signal output amplitude was set to 1.5 V. The phase noise at 10 MHz at 10 kHz offset is around -139 dBc/Hz. The phase noise at 100 MHz at 10 kHz offset is around -118 dBc/Hz.

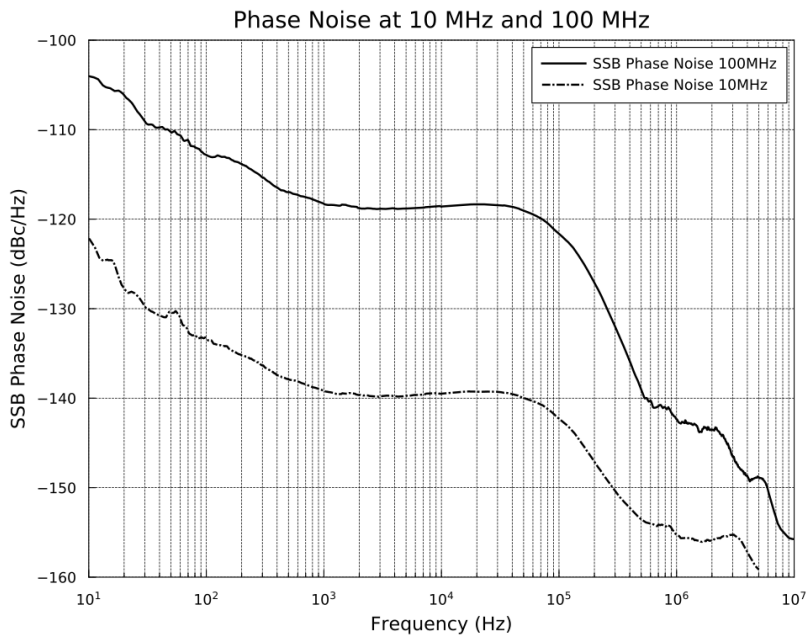


Figure 6.6: UHF phase noise

6.5. Clock 10 MHz

A 10 MHz clock input and output is provided for synchronization with other instruments. The figure explains the internal routing of the different clock signals. An internal clock generation unit receives a 10 MHz clock reference and generates all necessary internal sampling clocks. The clock reference either comes from the internal quartz/Rubidium oscillator or from an external clock source connected to the Clock 10 MHz In connector. The user can define if the clock is taken from the internal or external source. The Clock 10 MHz Out connector always provides the 10 MHz clock of the internal quartz/Rubidium oscillator.

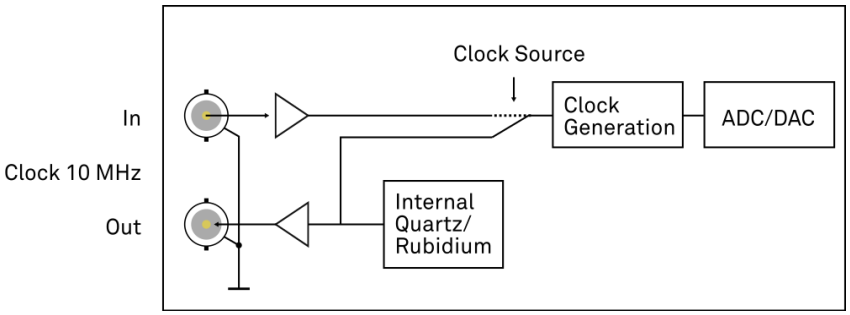


Figure 6.7: Clock routing

6.6. Auto Calibration

The instrument requires a self calibration after a short warm-up period to ensure operation according to specifications. During this self calibration process, components of the sensitive analog front-end are calibrated to account for temperature variations and drift. The self calibration is not to be confused with the Instrument Calibration service by Zurich Instruments. The latter is performed at the manufacturer site. The self calibration lasts about one second and only applies a fine-tuning.

The first self calibration after warm-up is executed automatically. Any further self calibration needs to be manually executed by the user. The self calibration process can be executed by clicking the Run button of the Auto Calibration section in the Device tab of the user interface.

The user can disable the calibration procedure completely if necessary. This can be done by changing the Enable button of the Auto Calibration in the Device tab. If this flag is disabled, no calibration is executed after warm-up time.

The default self calibration procedure can be divided into three different states, which are also indicated by the CAL flag in the footer of the user interface. The CAL flag can be either yellow, gray/off, or red.

- Yellow: The yellow CAL flag indicates that the calibration has not been executed yet. After a warm-up and temperature settling period of approximately 16 minutes, a self calibration is executed and the CAL flag turns gray. If the self calibration is disabled, the CAL flag turns red after the warm-up period to indicate that no calibration was performed.
- Gray/off: The gray CAL flag indicates that the instrument is self calibrated. The CAL flag turns red when the temperature change is larger than a given threshold or the time since the last calibration is longer than a given time interval. The values of these thresholds are indicated in the Device tab.
- Red: The red CAL flag indicates that it is recommended to perform a self calibration. The self calibration is never executed automatically in this state. The CAL flag is red, either, when the instrument experienced a temperature change larger than a given threshold, or when the time since the last calibration is longer than a given time interval. By executing a self calibration, the CAL flag will turn gray.

7. Signal Processing Basics

This chapter provides insights about several lock-in amplifier principles not necessarily linked to a specific instrument from Zurich Instruments. Since the appearance of the first valve-based lock-in amplifiers in the 1930s the physics have not changed, but the implementation and the performance have evolved greatly. Many good lock-in amplifier primers have appeared in the past decades, and some of them appear outdated now because they were written with analog instruments in mind. This section does not aim to replace any existing primer, but to complete them with a preferred emphasis on digital lock-in amplifiers.

The first subsection describes the principles of lock-in amplification, followed by the description of the function of discrete-time filters. After, we discuss the definition of the full range sensitivity, a specification parameter particularly important for analog lock-in amplifiers but with somewhat reduced importance for digital instruments. In the following, we describe the function and use of sinc filtering in particular for low-frequency lock-in measurements. The last section is dedicated to the zoom FFT feature. Innovative in the context of lock-in amplifiers, zoom FFT offers a fast and high-resolution spectral analysis around the lock-in operation frequency.

7.1. Principles of Lock-in Detection

Lock-in demodulation is a technique to measure the amplitude A_s and the phase θ of a periodic signal with the frequency $\omega_s = 2\pi f_s$ by comparing the signal to a reference signal. This technique is also called phase-sensitive detection. By averaging over time the signal-to-noise ratio (SNR) of a signal can be increased by orders of magnitude, allowing very small signals to be detected with a high accuracy making the lock-in amplifier a tool often used for signal recovery. For both signal recovery and phase-sensitive detection, the signal of interest is isolated with narrow band-pass filtering therefore reducing the impact of noise in the measured signal.

Figure 7.1 shows a basic measurement setup: a reference V_r signal is fed to the device under test. This reference signal is modified by the generally non-linear device with attenuation, amplification, phase shifting, and distortion, resulting in a signal $V_s = A_s\cos(\omega_s t + \theta_s)$ plus harmonic components.

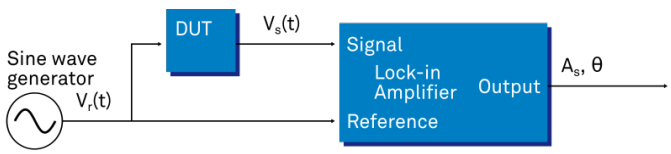


Figure 7.1: Basic measurement setup incorporating a lock-in amplifier

For practical reasons, most lock-in amplifiers implement the band-pass filter with a mixer and a low-pass filter (depicted in Figure 7.2): the mixer shifts the signal of interest into the baseband, ideally to DC, and the low-pass filter cuts all unwanted higher frequencies.

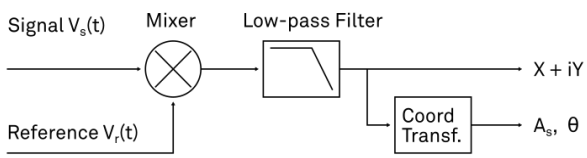


Figure 7.2: Mixing and low-pass filtering performed by the lock-in amplifier

The input signal $V_s(t)$ is multiplied by the reference signal $V_r(t) = \sqrt{2}e^{-i\omega_r t}$, where $\omega_r = 2\pi f_r$ is the demodulation frequency and i is the imaginary unit. This is the complex representation of a sine and cosine signal (phase shift 90°) forming the components of a quadrature demodulator, capable of measuring both the amplitude and the phase of the signal of interest. In principle it is possible to multiply the signal of interest with any frequency, resulting in a heterodyne operation. However the objective of the lock-in amplifier is to shift the signal as close as possible to DC, therefore the frequency of the reference and the signal is chosen similar. In literature this is called homodyne detection, synchrodyne detection, or zero-IF direct conversion.

The result of the multiplication is the signal

$$V_s(t) \cdot V_r(t) = V_s(t \cdot \sqrt{2}e^{-i\omega_r t} = \frac{A_s}{\sqrt{2}}e^{i[(\omega_s-\omega_r)t+\theta]} + \frac{A_s}{\sqrt{2}}e^{-i[(\omega_s+\omega_r)t+\theta]} \quad (1)$$

It consists of a slow component with frequency $\omega_s - \omega_r$ and a fast component with frequency $\omega_s + \omega_r$.

The demodulated signal is then low-pass filtered with an infinite impulse response (IIR) RC filter, indicated by the symbol $\langle\cdot\rangle$. The frequency response of the filter $F(\omega)$ will let pass the low frequencies $F(\omega_s - \omega_r)$ while considerably attenuating the higher frequencies $F(\omega_s + \omega_r)$. Another way to consider the low-pass filter is an averager.

$$X + iY = \langle V_s(t) \cdot \sqrt{2}e^{-i\omega_r t}\rangle \approx F(\omega_s - \omega_r)\frac{A_s}{\sqrt{2}}e^{i[(\omega_s-\omega_r)t+\theta]} \quad (2)$$

The result after the low-pass filter is the demodulated signal $X + iY$, where X is the real and Y is the imaginary part of a signal depicted on the complex plane. These components are also called in-phase and quadrature components. The transformation of X and Y into the amplitude R and phase θ information of $V_s(t)$ can be performed with trigonometric operations.

It is interesting to note that the value of the measured signal corresponds to the RMS value of the signal, which is equivalent to $R = A_s/\sqrt{2}$.

Most lock-in amplifiers output the values (X,Y) and (R, θ) encoded in a range of –10 V to +10 V of the auxiliary output signals.

7.1.1. Lock-in Amplifier Applications

Lock-in amplifiers are employed in a large variety of applications. In some cases the objective is measuring a signal with good signal-to-noise ratio, and then that signal could be measured even with large filter settings. In this context the word phase sensitive detection is appropriate. In other applications, the signal is very weak and overwhelmed by noise, which forces to measure with very narrow filters. In this context the lock-in amplifier is employed for signal recovery. Also, in another context, a signal modulated on a very high frequency (GHz or THz) that cannot be measured with standard approaches, is mixed to a lower frequency that fits into the measurement band of the lock-in amplifier.

One example for measuring a small, stationary or slowly varying signal which is completely buried in the 1/f noise, the power line noise, and slow drifts. For this purpose a weak signal is modulated to a higher frequency, away from these sources of noise. Such signal can be efficiently mixed back and measured in the baseband using a lock-in amplifier. In Figure 7.3 this process is depicted. Many optical applications perform the up-mixing with a chopper, an electro-optical modulator, or an acousto-optical modulator. The advantage of this procedure is that the desired signal is measured in a spectral region with comparatively little noise. This is more efficient than just low-pass filtering the DC signal.

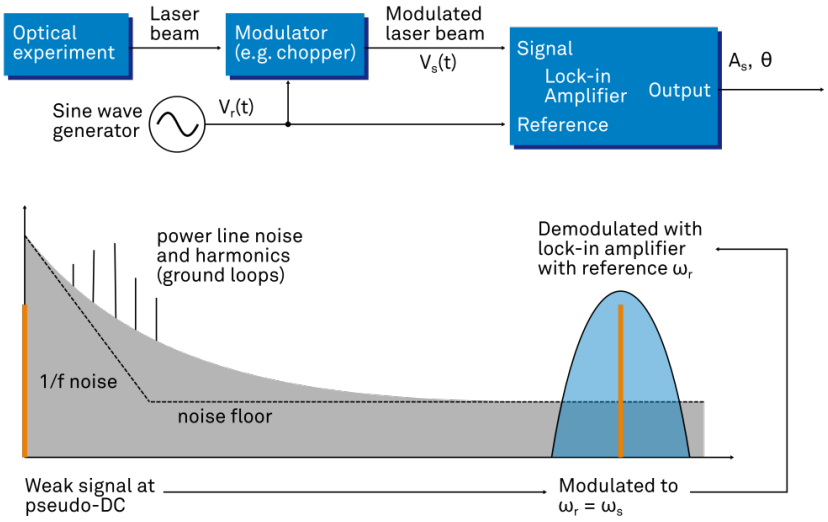


Figure 7.3: Lock-in measurement of a noisy DC signal

7.2. Signal Bandwidth

The signal bandwidth (BW) theoretically corresponds to the highest frequency components of interest in a signal. In practical signals, the bandwidth is usually quantified by the cut-off frequency. It is the frequency at which the transfer function of a system shows 3 dB attenuation relative to DC (BW = $f_{cut\text{-}off}$ = f_{-3dB}); that is, the signal power at f_{-3dB} is half the power at DC. The bandwidth, equivalent to cut-off frequency, is used in the context of dynamic behavior of a signals or separation of different signals. This is for instance the case for fast-changing amplitudes or phase values like in a PLL or in imaging applications, or when signals closely spaced in frequency need to be separated.

The noise equivalent power bandwidth (NEPBW) is also a useful figure, and it is distinct from the signal bandwidth. This unit is typically used for noise measurements: in this case one is interested in the total amount of power that passes through a low-pass filter, equivalent to the area under the solid curve in Figure 7.4. For practical reasons, one defines an ideal brick-wall filter that lets pass the same amount of power under the assumption that the noise has a flat (white) spectral density. This brick-wall filter has transmission 1 from DC to f_{NEPBW}. The orange and blue areas in Figure 7.4 then are exactly equal in a linear scale.

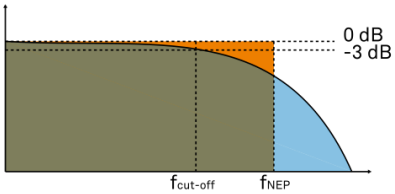


Figure 7.4: Signal bandwidth and noise equivalent power bandwidth

It is possible to establish a simple relation between the $f_{cut\text{-}off}$ and the f_{NEPBW} that only depends on the slope (or roll-off) of the filter. As the filter slope actually depends on the time constant (TC) defined for the filter, it is possible to establish the relation also to the time constant. It is intuitive to understand that for higher filter orders, the $f_{cut\text{-}off}$ is closer to the f_{NEPBW} than for smaller orders.

The time constant is a parameter used to interpret the filter response in the time domain, and relates to the time it takes to reach a defined percentage of the final value. The time constant of a low-pass filter relates to the bandwidth according to the formula

$$TC = \frac{FO}{2\pi f_{cut-off}} \tag{3}$$

where FO is said factor that depends on the filter slope. This factor, along with other useful conversion factors between different filter parameters, can be read from the following table.

Table 7.1: Summary of conversion factors for bandwidth definitions

filter order	filter roll-off	FO	$f_{cut\text{-}off}$	f_{NEPBW}	f_{NEPBW} / $f_{cut\text{-}off}$
1st	6 dB/oct	1.0000	0.1592 / TC	0.2500 / TC	1.5708
2nd	12 dB/oct	0.6436	0.1024 / TC	0.1250 / TC	1.2203
3rd	18 dB/oct	0.5098	0.0811 / TC	0.0937 / TC	1.1554
4th	24 dB/oct	0.4350	0.0692 / TC	0.0781 / TC	1.1285
5th	30 dB/oct	0.3856	0.0614 / TC	0.0684 / TC	1.1138
6th	36 dB/oct	0.3499	0.0557 / TC	0.0615 / TC	1.1046
7th	42 dB/oct	0.3226	0.0513 / TC	0.0564 / TC	1.0983
8th	48 dB/oct	0.3008	0.0479 / TC	0.0524 / TC	1.0937

7.3. Discrete-Time Filters

7.3.1. Discrete-Time RC Filter

There are many options how to implement digital low-pass filters. One common filter type is the exponential running average filter. Its characteristics are very close to those of an analog resistor-capacitor RC filter, which is why this filter is sometimes called a discrete-time RC filter. The exponential running average filter has the time constant $TC = \tau_N$ as its only adjustable parameter. It operates on an input signal $X_{in}[n,]$ defined at discrete times $nT_s, (n+1)T_s, (n+2)T_s$, etc., spaced at the sampling time T_s . Its output $X_{out}[n, T_s]$ can be calculated using the following recursive formula,

$$X_{out}[n, T_s] = e^{-T_s/\tau_n} X_{out}[n-1, T_s] + (1 - e^{-T_s/\tau_N}) X_{in}[n, T_s] \tag{4}$$

The response of that filter in the frequency domain is well approximated by the formula

$$H_1(\omega) = \frac{1}{1 + i \cdot \omega \cdot \tau_n} \tag{5}$$

The exponential filter is a first-order filter. Higher-order filters can easily be implemented by cascading several filters. For instance the 4th order filter is implemented by chaining 4 filters with the same time constant $TC = \tau_n$ one after the other so that the output of one filter stage is the input of the next one. The transfer function of such a cascaded filter is simply the product of the transfer functions of the individual filter stages. For an n-th order filter, we therefore have

$$H_n(\omega) = \frac{1}{(1 + i \cdot \omega \cdot \tau_n)^n} \tag{6}$$

The attenuation and phase shift of the filters can be obtained from this formula. Namely, the filter attenuation is given by the absolute value squared $|H_n(\omega)|^2$. The filter transmission phase is given by the complex argument $\arg[H_n(\omega)]$.

7.3.2. Filter Settling Time

The low-pass filters after the demodulator cause a delay to measured signals depending on the filter order and time constant $TC = \tau_n$. After a change in the signal, it will therefore take some time before the lock-in output reaches the correct measurement value. This is depicted in Figure 7.5 where the response of cascaded filters to a step input signal is shown.

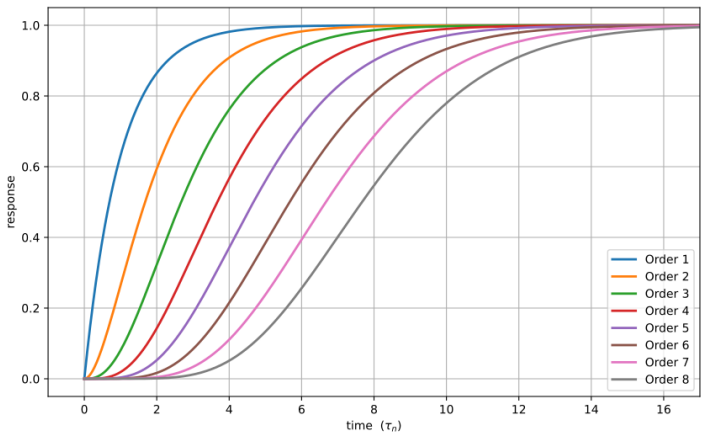


Figure 7.5: Time-domain step response of the demodulator low-pass filter for orders from 1 to 8.

More quantitative information on the settling time can be obtained from Table 7.2. In this table, you find settling times in units of the 1st-order filter's time constant (TC) for all filter orders available with the UHF Lock-in Amplifier. The values tell the time you need to wait for the filtered demodulator signal to reach 50%, 63%, 95% and 99% of the final value. This can help in making a quantitatively correct choice of filter parameters for example in a measurement involving a parameter sweep.

Table 7.2: Summary of Filter Settling Times

Filter order	50%	63% (1-1/e)	90%	95%	99%
1st	0.7 · TC	1.0 · TC	2.3 · TC	3.0 · TC	4.6 · TC
2nd	1.7 · TC	2.1 · TC	3.9 · TC	4.7 · TC	6.6 · TC
3rd	2.7 · TC	3.3 · TC	5.3 · TC	6.3 · TC	8.4 · TC
4th	3.7 · TC	4.4 · TC	6.7 · TC	7.8 · TC	10.0 · TC
5th	4.7 · TC	5.4 · TC	8.0 · TC	9.2 · TC	11.6 · TC
6th	5.7 · TC	6.5 · TC	9.3 · TC	10.5 · TC	13.1 · TC
7th	6.7 · TC	7.6 · TC	10.5 · TC	11.8 · TC	14.6 · TC
8th	7.7 · TC	8.6 · TC	11.8 · TC	13.1 · TC	16.0 · TC

7.4. Full Range Sensitivity

The sensitivity of the lock-in amplifier is the RMS value of an input sine that is demodulated and results in a full scale analog output. Traditionally the X, Y, or R components are mapped onto the 10 V full scale analog output. In such a case, the overall gain from input to output of the lock-in amplifier is composed of the input and output amplifier stages. Many lock-in amplifiers specify a sensitivity between 1 nV and 1 V. In other words the instrument permits an input signal between 1 nV and 1 V to be amplified to the 10 V full range output.

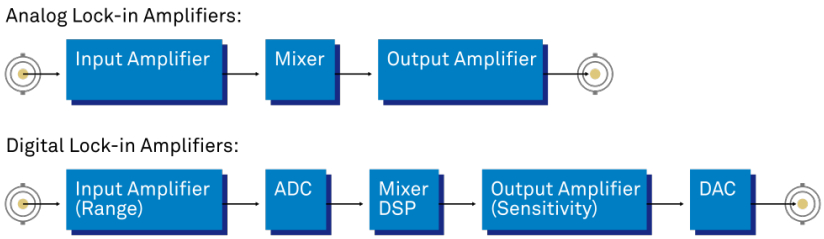


Figure 7.6: Sensitivity from signal input to signal output

In analog lock-in amplifiers the sensitivity is simple to understand. It is the sum of the analog amplification stages between in the input and the output of the instrument: in particular the input amplifier and the output amplifier.

In digital lock-in amplifiers the sensitivity less straightforward to understand. Analog-to-digital converters (ADC) operate with a fixed input range (e.g. 1 V) and thus require a variable-gain amplifier to amplify the input signal to the range given by the ADC. This variable-gain amplifier must be in the analog domain and its capability determines the minimum input range of the instrument. A practical analog input amplifier provides a factor 1000 amplification, thus 1 V divided by 1000 is the minimum input range of the instrument.

The input range is the maximum signal amplitude that is permitted for a given range setting. The signal is internally amplified with the suited factor, e.g. (1 mV)·1000 to result in a full swing signal at the ADC. For signals larger than the range, the ADC saturates and the signal is distorted – the measurement result becomes useless. Thus the signal should never exceed the range setting.

But the input range is not the same as the sensitivity. In digital lock-in amplifiers the sensitivity is only determined by the output amplifier, which is an entirely digital signal processing unit which performs a numerical multiplication of the demodulator output with the scaling factor. The digital output of this unit is then fed to the output digital-to-analog converter (DAC) with a fixed range of 10 V. It is this scaling factor that can be retrofitted to specify a sensitivity as known from the analog lock-in amplifiers. A large scaling factor, and thus a high sensitivity, comes at a relatively small expense for digital amplification.

One interesting aspect of digital lock-in amplifiers is the connection between input resolution and sensitivity. As the ADC operates with a finite resolution, for instance 14 bits, the minimum signal that can be detected and digitized is for instance 1 mV divided by the resolution of the ADC. With 14 bits the minimum level that can be digitized would be 122 nV. How is it possible to reach 1 nV sensitivity without using a 21 bit analog-to-digital converter? In a world without noise it is not possible.

Inversely, thanks to noise and current digital technology it is possible to achieve a sensitivity even below 1 nV.

Most sources of broadband noise, including the input amplifier, can be considered as Gaussian noise sources. Gaussian noise is equally distributed in a signal, and thus generates equally distributed disturbances. The noise itself can be filtered by the lock-in amplifier down to a level where it does not impact the measurement. Still, in the interplay with the signal, the noise does have an effect on the measurement. The input of the ADC is the sum of the noise and the signal amplitude. Every now and then, the signal amplitude on top of the large noise will be able to toggle the least significant bits even for very small signals, as low as 1 nV and below. The resulting digital signal has a component at the signal frequency and can be detected by the lock-in amplifier.

There is a similar example from biology. Rod cells in the human eye permit humans to see in very low light conditions. The sensitivity of rod cells in the human eye is as low as a single photon. This sensitivity is achieved in low light conditions by a sort of pre-charging of the cell to be sensitive to the single photon that triggers the cell to fire an impulse. In a condition with more surround light, rod cells are less sensitive and need more photons to fire.

To summarize, in digital lock-in amplifiers the full range sensitivity is only determined by the scaling factor capability of the digital output amplifier. As the scaling can be arbitrary big, 1 nV minimum full range sensitivity is achievable without a problem. Further, digital lock-in amplifiers exploit the input noise to heavily increase the sensitivity without impacting the accuracy of the measurement.

7.5. Sinc Filtering

As explained in Principles of Lock-in Detection, the demodulated signal in an ideal lock-in amplifier has a signal component at DC and a spurious component at twice the demodulation frequency. The components at twice the demodulation frequency (called the 2ω component) is effectively removed by regular low-pass filtering. By selecting filters with small bandwidth and faster roll-offs, the 2ω component can easily be attenuated by 100 dB or more. The problem arises at low demodulation frequencies, because this forces the user to select long integration times (e.g. >60 ms for a demodulation frequency of 20 Hz) in order to achieve the same level of 2ω attenuation.

In practice, the lock-in amplifier will modulate DC offsets and non-linearities at the signal input with the demodulation frequency, resulting in a signal at the demodulation frequency (called ω component). This component is also effectively removed by the regular low-pass filters at frequencies higher than 1 kHz.

At low demodulation frequencies, and especially for applications with demodulation frequencies close to the filter bandwidth, the ω and 2ω components can affect the measurement result. Sinc filtering allows for strong attenuation of the ω and 2ω components. Technically the sinc filter is a comb filter with notches at integer multiples of the demodulation frequency (ω, 2ω, 3ω, etc.). It removes the ω component with a suppression factor of around 80 dB. The amount of 2ω component that gets removed depends on the input signal. It can vary from entirely (e.g. 80 dB) to slightly (e.g. 5 dB). This variation is not due to the sinc filter performance but depends on the bandwidth of the input signal.

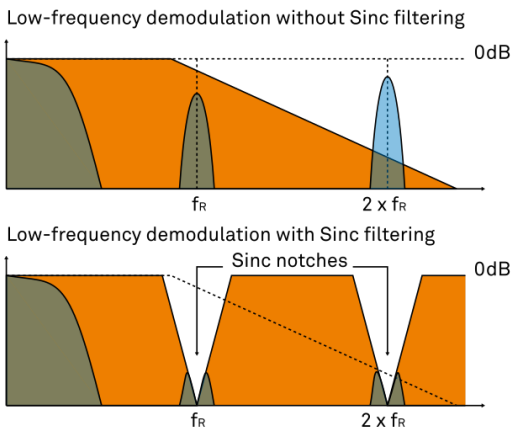


Figure 7.7: Effect of sinc filtering

Table 7.3: Artifacts in the demodulation signal

Input signal	Demodulation result before low-pass filter	Result
Signal at ω	DC component	Amplitude and phase information (wanted signal)
	2ω component	Unwanted component (can additionally be attenuated by sinc filter)
DC offset	ω component	Unwanted component (can additionally be attenuated by sinc filter)

We can observe the effect of the sinc filter by using the Spectrum Analyzer Tool of the UHF Lock-in Amplifier. As an example, consider a 30 Hz signal with an amplitude of 0.1 V that demodulated using a filter bandwidth of 100 Hz and a filter order 8. In addition 0.1 V offset is added to the signal so that we get a significant ω component.

Figure 7.8 shows a spectrum with the sinc filter disabled, whereas for Figure 7.9 the sinc filter is enabled. The comparison of the two clearly shows how the sinc options dampens both the ω and 2ω components by about 100 dB.

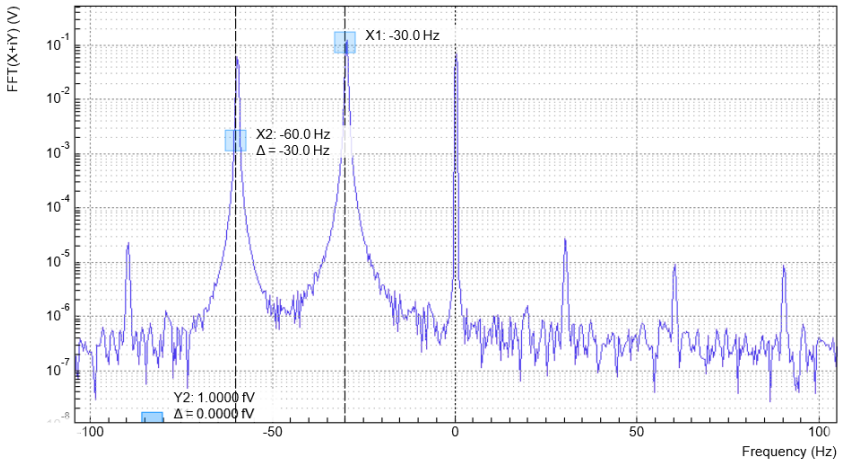


Figure 7.8: Spectrum of a demodulated 30 Hz signal without sinc filter

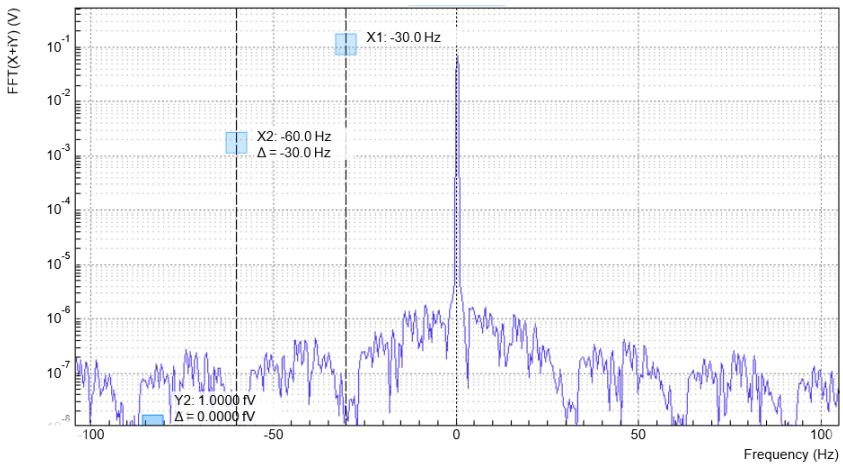


Figure 7.9: Spectrum of a demodulated 30 Hz signal with sinc filter

Note

In order to put the notches of the digital filter to ω and 2ω, the sampling rate of the filter would have to be precisely adjusted to the signal frequency. As this is technically not feasible, the generated signal frequency is adjusted instead by a very small amount.

7.6. Zoom FFT

The concept of zoom FFT allows the user to analyze the spectrum of the input signal around a particular frequency by zooming in on a narrow frequency portion of the spectrum. This is done by performing a Fourier transform of the demodulated in-phase and quadrature (X and Y) components or more precisely, on the complex quantity X+iY, where i is the imaginary unit. In the LabOne user interface, this functionality is available in the Spectrum tab.

In normal FFT, the sampling rate determines the frequency span and the total acquisition time determines the frequency resolution. Having a large span and a fine resolution at the same time then requires long acquisition times at high sample rates. This means that a lot of data needs to be acquired, stored, and processed, only to retain a small portion of the spectrum and discard most of it in the end. In zoom FFT, the lock-in demodulation is used to down-shift the signal frequency, thereby allowing one to use both a much lower sampling rate and sample number to achieve the same frequency resolution. Typically, to achieve a 1 Hz frequency resolution at 1 MHz, FFT would require to collect and process approximately 10^6 points, while zoom FFT only processes 10^3 points. (Of course the high rate sampling is done by the lock-in during the demodulation stage, so the zoom FFT still needs to implicitly rely on a fast ADC.)

In order to illustrate why this is so and what benefits this measurement tool brings to the user, it is useful to remind that at the end of the demodulation of the input signal $V_s(t) = A_s\cos(\omega_s t + \tau)$, the output signal is $X + iY = F(\omega_s - \omega_r)(A_s/\sqrt{2})e^{i[(\omega_s-\omega_r)t+\tau]}$ where F(ω) is the frequency response of the filters.

Since the demodulated signal has only one component at frequency $\omega_s-\omega_r$, its power spectrum (Fourier transform modulus squared) has a peak of height $(|A_s|^2/2) \cdot |F(\omega_s - \omega_r)|^2$ at $\omega_s-\omega_r$: this tells us the spectral power distribution of the input signal at frequencies close to ω_r within the demodulation bandwidth set by the filters F(ω).

Note that:

- the ability of distinguish between positive and negative frequencies works only if the Fourier transform is done on X+iY. Had we taken X for instance, the positive and negative frequencies of its power spectrum would be equal. The symmetry relation G(−ω)=G*(ω) holds for the Fourier transform G(ω) of a real function g(t) and two identical peaks would appear at $\pm|\omega_s-\omega_r|$.
- one can extract the amplitude of the input signal by diving the power spectrum by $|F(\omega)|^2$, the operation being limited by the numerical precision. This is implemented in LabOne and is activated by the Filter Compensation button: with the Filter Compensation enabled, the background noise appears white; without it, the effect of the filter roll-off becomes apparent.

The case of an input signal containing a single frequency component can be generalized to the case of multiple frequencies. In that case the power spectrum would display all the frequency components weighted by the filter transfer function, or normalized if the Filter Compensation is enabled.

When dealing with discrete-time signal processing, one has to be careful about aliasing which occurs when the signal frequencies higher than the sampling rate ω are not sufficiently suppressed. Remember that ω is the user settable readout rate, not the 2 GSa/s sampling rate of the GHFLI input. Since the discrete-time Fourier transform extends between −ω/2 and +ω/2, the user has to make sure that at ±ω/2 the filters provide the desired attenuation: this can be done either by increasing the sampling rate or resolving to measure a smaller frequency spectrum (i.e. with a smaller filter bandwidth).

Similarly to the continuous case, in which the acquisition time determines the maximum frequency resolution ($2\pi/T$ if T is the acquisition time), the resolution of the zoom FFT can be increased by increasing the number of recorded data points. If N data points are collected at a sampling rate ω, the discrete Fourier transform has a frequency resolution of ω/N.

8. Device Node Tree

This chapter contains reference documentation for the settings and measurement data available on UHF Instruments. Whilst Functional Description LabOne User Interface 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 UHF Instruments, organized by branch in the Node Tree.

8.1. 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 "dev2006" 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.

8.1.1. Node Properties and Data Types

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

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

A node may contain data of the following types:

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

8.1.2. 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 8.1 shows how the equivalent MATLAB command is displayed after modifying the value of the auxiliary output 1's offset. The format of the LabOne UI's command history can be configured in the Config Tab (MATLAB, Python and .NET are available). The entire history generated in the current UI session can be viewed by clicking the "Show Log" button.

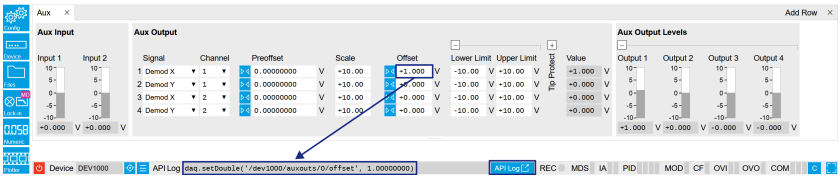


Figure 8.1: When a device's configuration is modified in the LabOne User Interface, the Status Bar displays the equivalent command to perform the same configuration via a LabOne programming interface. Here, the MATLAB code to modify auxiliary output 1's offset value is provided. When "Show Log" is clicked 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('/dev2006/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
```

8.1.3. 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

8.2. Reference Node Documentation

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

8.2.1. AUCARTS

/dev..../aucarts/n/enable

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

Enables the streaming of results to the host computer.

0	"on": ON: The arithmetic unit is operative and results are streamed to the host computer.
1	"off": OFF: The arithmetic unit is operative but results are not streamed to the host computer.

/dev..../aucarts/n/mode

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

Selects the operation mode of the arithmetic unit

0	"add": Add: The arithmetic unit is in add mode: two independent demodulator outputs can be added together.
1	"divide": Divide: The arithmetic unit is in divide mode: two independent demodulator outputs can be divided by each other.
2	"multiply": Multiply: The arithmetic unit is in multiply mode: two independent demodulator outputs can be multiplied with each other.

/dev..../aucarts/n/ops/n/coeff

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

Select a coefficient to be applied to the selected Signal. Default: 0.

0	"mult_by_1": A coefficient of 1 is used (default).
1	"auxin0", "auxiliary_input0": The signal on Aux In 1 is used as coefficient.
2	"auxin1", "auxiliary_input1": The signal on Aux In 2 is used as coefficient.
3	"cartesian_au0": Output of Cartesian AU 1 (C1) is used as coefficient (for Cartesian AU only).
4	"cartesian_au1": Output of Cartesian AU 2 (C2) is used as coefficient (for Cartesian AU only).
5	"polar_au0": Output of Polar AU 1 (P1) is used as coefficient (for Polar AU only).
6	"polar_au1": Output of Polar AU 2 (P2) is used as coefficient (for Polar AU only).

/dev..../aucarts/n/ops/n/demodselect

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

Select demodulator and/or Boxcar channel number.

/dev..../aucarts/n/ops/n/scale

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

Custom scaling factor.

/dev..../aucarts/n/ops/n/value

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

Select the arithmetic unit input signal.

0	"demod_x": Use demodulator X (for Cartesian AU only).
1	"demod_y": Use demodulator Y (for Cartesian AU only).
2	"boxcar": Use Boxcar (for Cartesian AU only).
3	"demod_r": Use demodulator R (for polar AU only).
4	"demod_theta": Use demodulator Θ (for polar AU only).
5	"magnitude": Use the magnitude of C1 + iC2 (for polar AU only).
6	"angle": Use the angle of C1 + iC2 (for polar AU only).

/dev..../aucarts/n/rate

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

Defines the number of arithmetic unit result samples that are sent to the host computer per second.

/dev..../aucarts/n/sample

Properties: Read, Stream
Type: Double
Unit: Dependent

Streaming node giving the output samples of the arithmetic unit.

/dev..../aucarts/n/value

Properties: Read
Type: Double
Unit: Dependent

Gives the result of the arithmetic unit at a low sampling rate (10 per second).

8.2.2. AUPOLARS

/dev..../aupolars/n/enable

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

Enables the streaming of results to the host computer.

/dev..../aupolars/n/mode

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

Selects the operation mode of the arithmetic unit.

0	"add": Add: The arithmetic unit is in add mode: two independent demodulator outputs can be added together.
1	"divide": Divide: The arithmetic unit is in divide mode: two independent demodulator outputs can be divided by each other.
2	"multiply": Multiply: The arithmetic unit is in multiply mode: two independent demodulator outputs can be multiplied with each other.

/dev..../aupolars/n/ops/n/coeff

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

Select a coefficient to be applied to the selected Signal. Default: 0.

0	"mult_by_1": A coefficient of 1 is used (default).
1	"auxin0", "auxiliary_input0": The signal on Aux In 1 is used as coefficient.
2	"auxin1", "auxiliary_input1": The signal on Aux In 2 is used as coefficient.
3	"cartesian_au0": Output of Cartesian AU 1 (C1) is used as coefficient (for Cartesian AU only).
4	"cartesian_au1": Output of Cartesian AU 2 (C2) is used as coefficient (for Cartesian AU only).
5	"polar_au0": Output of Polar AU 1 (P1) is used as coefficient (for Polar AU only).
6	"polar_au1": Output of Polar AU 2 (P2) is used as coefficient (for Polar AU only).

/dev..../aupolars/n/ops/n/demodselect

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

Select demodulator channel number.

/dev..../aupolars/n/ops/n/scale

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

Custom scaling factor.

/dev..../aupolars/n/ops/n/value

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

Select the arithmetic unit input signal

0	"demod_x": Use demodulator X (for Cartesian AU only).
1	"demod_y": Use demodulator Y (for Cartesian AU only).
2	"boxcar": Use Boxcar (for Cartesian AU only).
3	"demod_r": Use demodulator R (for polar AU only).
4	"demod_theta": Use demodulator Θ (for polar AU only).
5	"magnitude": Use the magnitude of C1 + iC2 (for polar AU only).
6	"angle": Use the angle of C1 + iC2 (for polar AU only).

/dev..../aupolars/n/rate

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

Defines the number of arithmetic unit result samples that are sent to the host computer per second.

/dev..../aupolars/n/sample

Properties: Read, Stream
Type: Double
Unit: Dependent

Streaming node giving the output samples of the arithmetic unit.

/dev..../aupolars/n/value

Properties: Read
Type: Double
Unit: Dependent

Gives the result of the arithmetic unit at a low sampling rate (10 per second).

8.2.3. AUXINS

/dev..../auxins/n/averaging

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

Defines the number of samples on the input to average as a power of two. Possible values are in the range [0, 16]. A value of 0 corresponds to the sampling rate of the auxiliary input's ADC.

/dev..../auxins/n/sample

Properties: Read, Stream
Type: ZIAuxInSample
Unit: V

Voltage measured at the Auxiliary Input after averaging. The data rate depends on the averaging value. Note, if the instrument has demodulator functionality, the auxiliary input values are available as fields in a demodulator sample and are aligned by timestamp with the demodulator output.

/dev..../auxins/n/values/n

Properties: Read
Type: Double
Unit: V

Voltage measured at the Auxiliary Input after averaging. The value of this node is updated at a low rate (50 Hz); the streaming node auxins/n/sample is updated at a high rate defined by the averaging.

8.2.4. AUXOUTS

/dev..../auxouts/n/demodselect

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

Select the channel number of the selected signal source.

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

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

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

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

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

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

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

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

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

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

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

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

-1	"manual": Select Manual as the output option.
0	"demod_x": Select Demod X as the output option.
1	"demod_y": Select Demod Y as the output option.
2	"demod_r": Select Demod R as the output option.
3	"demod_theta": Select Demod Theta as the output option.
4	"awg": Select one of the AWG Outputs for auxiliary output when running the AWG in four-channel mode. The AWG option needs to be installed.
5	"pid": Select PID Out as the output option.
6	"boxcar": Select Boxcar as the output option.
7	"au_cartesian": Select AU Cartesian as the output option.
8	"au_polar": Select AU Polar as the output option.
9	"pid_shift": Select PID Shift as the output option.
10	"pid_error": Select PID Error as the output option.
12	"pulse_counter": Select Pulse Counter as the output option.

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

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

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

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

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

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

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

Properties:	Read
Type:	Double
Unit:	V

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

8.2.5. AWGS

/dev..../awgs/n/auxtriggers/n/channel

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

Selects the digital trigger source signal.

0	"trigin0", "trigger_input0": Trigger In 1
1	"trigin1", "trigger_input1": Trigger In 2
2	"trigin2", "trigger_input2": Trigger In 3
3	"trigin3", "trigger_input3": Trigger In 4
4	"trigout0", "trigger_output0": Trigger Out 1
5	"trigout1", "trigger_output1": Trigger Out 2
6	"trigout2", "trigger_output2": Trigger Out 3
7	"trigout3", "trigger_output3": Trigger Out 4

/dev..../awgs/n/auxtriggers/n/slope

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

Select the signal edge that should activate the trigger. The trigger will be level sensitive when the Level option is selected.

0	"level_sensitive": Level sensitive trigger
1	"rising_edge": Rising edge trigger
2	"falling_edge": Falling edge trigger
3	"both_edges": Rising or falling edge trigger

/dev..../awgs/n/auxtriggers/n/state

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

State of the Auxiliary Trigger: No trigger detected/trigger detected.

/dev..../awgs/n/dio/delay/index

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

Index of the bit on the DIO interface for which the delay should be changed.

/dev..../awgs/n/dio/delay/value

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

Corresponding delay value to apply to the given bit of the DIO interface in units of 450 MHz clock cycles. Valid values are 0 to 3.

/dev..../awgs/n/dio/strobe/index

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

Select the DIO bit to use as the STROBE signal.

/dev..../awgs/n/dio/strobe/slope

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

Select the signal edge of the STROBE signal for use in timing alignment.

0	"off": Off
1	"rising_edge": Rising edge trigger
2	"falling_edge": Falling edge trigger
3	"both_edges": Rising or falling edge trigger

/dev..../awgs/n/dio/valid/index

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

Select the DIO bit to use as the VALID signal to indicate a valid input is available.

/dev..../awgs/n/dio/valid/polarity

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

Polarity of the VALID bit that indicates that a valid input is available.

0	"none": None: VALID bit is ignored.
1	"low": Low: VALID bit must be logical zero.
2	"high": High: VALID bit must be logical high.
3	"both": Both: VALID bit may be logical high or zero.

/dev..../awgs/n/elf/checksum

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

Checksum of the uploaded ELF file.

/dev..../awgs/n/elf/data

Properties: Write
Type: ZIVectorData
Unit: None

Accepts the data of the sequencer ELF file.

/dev..../awgs/n/elf/length

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

Length of the compiled ELF file.

/dev..../awgs/n/elf/memoryusage

Properties: Read
Type: Double
Unit: None

Size of the uploaded ELF file relative to the size of the main memory.

/dev..../awgs/n/elf/name

Properties: Read
Type: ZIVectorData
Unit: None

The name of the uploaded ELF file.

/dev..../awgs/n/elf/progress

Properties: Read
Type: Double
Unit: %

The percentage of the sequencer program already uploaded to the device.

/dev..../awgs/n/enable

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

Activates the AWG.

/dev..../awgs/n/outputs/n/amplitude

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

Amplitude in units of full scale of the given AWG Output. The full scale corresponds to the Range voltage setting of the Signal Outputs.

/dev..../awgs/n/outputs/n/enables/n

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

Indicates the routing of the AWG signal (k index) to the wave output or to the digital mixer input (m index).

/dev..../awgs/n/outputs/n/mode

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

Select between plain mode, amplitude modulation, and advanced mode.

0	"plain": Plain: AWG Output goes directly to Signal Output.
1	"modulation": Modulation: AWG Output 1 (2) is multiplied with oscillator signal of demodulator 4 (8).
2	"advanced": Advanced: Output of AWG channel 1 (2) modulates demodulators 1-4 (5-8) with independent envelopes.

/dev..../awgs/n/ready

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

AWG has a compiled wave form and is ready to be enabled.

/dev..../awgs/n/sequencer/assembly

Properties:	Read
Type:	ZIVectorData
Unit:	None

Displays the current sequence program in compiled form. Every line corresponds to one hardware instruction.

/dev..../awgs/n/sequencer/continue

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

Reserved for future use.

/dev..../awgs/n/sequencer/memoryusage

Properties:	Read
Type:	Double
Unit:	None

Size of the current Sequencer program relative to the available instruction memory of 1 kInstructions (1'024 instructions).

/dev..../awgs/n/sequencer/next

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

Reserved for future use.

/dev..../awgs/n/sequencer/pc

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

Current position in the list of sequence instructions during execution.

/dev..../awgs/n/sequencer/program

Properties: Read
Type: ZIVectorData
Unit: None

Displays the source code of the current sequence program.

/dev..../awgs/n/sequencer/status

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

Status of the sequencer on the instrument. Bit 0: sequencer is running; Bit 1: reserved; Bit 2: sequencer is waiting for a trigger to arrive; Bit 3: AWG has detected an error; Bit 4: sequencer is waiting for synchronization with other channels.

/dev..../awgs/n/single

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

Puts the AWG into single shot mode.

/dev..../awgs/n/sweep/awgtrigs/n

Properties: Read, Write
Type: Double
Unit: Dependent

Node used by the sweeper module for fast index sweeps. When selected as sweep grid the sweeper module will switch into a fast index based scan mode. This mode can be up to 1000 times faster than conventional node sweeps. The sequencer program must support this functionality. See section 'AWG Index Sweep' of the UHF user manual for more information.

/dev..../awgs/n/time

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

AWG sampling rate. The numeric values are rounded for display purposes. The exact values are equal to the base sampling rate (1.8 GHz) divided by 2^n, where n is the node value. This value is used by default and can be overridden in the Sequence program.

0	"1.80_GHz": 1.80 GHz
1	"900_MHz": 900 MHz
2	"450_MHz": 450 MHz
3	"225_MHz": 225 MHz
4	"113_MHz": 112.5 MHz
5	"56.3_MHz": 56.25 MHz
6	"28.1_MHz": 28.12 MHz
7	"14.1_MHz": 14.06 MHz
8	"7.03_MHz": 7.03 MHz
9	"3.52_MHz": 3.52 MHz
10	"1.76M_Hz": 1.76 MHz
11	"878.91_kHz": 878.91 kHz
12	"439_kHz": 439.45 kHz
13	"220_kHz": 219.73 kHz

/dev..../awgs/n/triggers/n/channel

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

Selects the signal source for the analogue trigger.

0	"sigin0", "signal_input0": Signal Input 1
1	"sigin1", "signal_input1": Signal Input 2
2	"trigin0", "trigger_input0": Trigger Input 1
3	"trigin1", "trigger_input1": Trigger Input 2
4	"auxout0", "auxiliary_output0": Aux Output 1. Requires an installed digitizer (DIG) option.
5	"auxout1", "auxiliary_output1": Aux Output 2. Requires an installed digitizer (DIG) option.
6	"auxout2", "auxiliary_output2": Aux Output 3. Requires an installed digitizer (DIG) option.
7	"auxout3", "auxiliary_output3": Aux Output 4. Requires an installed digitizer (DIG) option.
8	"auxin0", "auxiliary_input0": Aux Input 1
9	"auxin1", "auxiliary_input1": Aux Input 2
10	"demod3": Osc φ Demod 4
11	"demod7": Osc φ Demod 8
16	"demod0_x": Demod 1 X. Requires an installed digitizer (DIG) option.
17	"demod1_x": Demod 2 X. Requires an installed digitizer (DIG) option.
18	"demod2_x": Demod 3 X. Requires an installed digitizer (DIG) option.
19	"demod3_x": Demod 4 X. Requires an installed digitizer (DIG) option.
20	"demod4_x": Demod 5 X. Requires an installed digitizer (DIG) option.
21	"demod5_x": Demod 6 X. Requires an installed digitizer (DIG) option.
22	"demod6_x": Demod 7 X. Requires an installed digitizer (DIG) option.
23	"demod7_x": Demod 8 X. Requires an installed digitizer (DIG) option.
32	"demod0_y": Demod 1 Y. Requires an installed digitizer (DIG) option.
33	"demod1_y": Demod 2 Y. Requires an installed digitizer (DIG) option.
34	"demod2_y": Demod 3 Y. Requires an installed digitizer (DIG) option.
35	"demod3_y": Demod 4 Y. Requires an installed digitizer (DIG) option.
36	"demod4_y": Demod 5 Y. Requires an installed digitizer (DIG) option.
37	"demod5_y": Demod 6 Y. Requires an installed digitizer (DIG) option.
38	"demod6_y": Demod 7 Y. Requires an installed digitizer (DIG) option.
39	"demod7_y": Demod 8 Y. Requires an installed digitizer (DIG) option.
48	"demod0_r": Demod 1 R. Requires an installed digitizer (DIG) option.
49	"demod1_r": Demod 2 R. Requires an installed digitizer (DIG) option.
50	"demod2_r": Demod 3 R. Requires an installed digitizer (DIG) option.
51	"demod3_r": Demod 4 R. Requires an installed digitizer (DIG) option.
52	"demod4_r": Demod 5 R. Requires an installed digitizer (DIG) option.
53	"demod5_r": Demod 6 R. Requires an installed digitizer (DIG) option.
54	"demod6_r": Demod 7 R. Requires an installed digitizer (DIG) option.
55	"demod7_r": Demod 8 R. Requires an installed digitizer (DIG) option.
64	"demod0_theta": Demod 1 Θ. Requires an installed digitizer (DIG) option.
65	"demod1_theta": Demod 2 Θ. Requires an installed digitizer (DIG) option.
66	"demod2_theta": Demod 3 Θ. Requires an installed digitizer (DIG) option.
67	"demod3_theta": Demod 4 Θ. Requires an installed digitizer (DIG) option.
68	"demod4_theta": Demod 5 Θ. Requires an installed digitizer (DIG) option.
69	"demod5_theta": Demod 6 Θ. Requires an installed digitizer (DIG) option.
70	"demod6_theta": Demod 7 Θ. Requires an installed digitizer (DIG) option.
71	"demod7_theta": Demod 8 Θ. Requires an installed digitizer (DIG) option.
80	"pid0_value": PID 1 value. Requires an installed digitizer (DIG) option.
81	"pid1_value": PID 2 value. Requires an installed digitizer (DIG) option.
82	"pid2_value": PID 3 value. Requires an installed digitizer (DIG) option.
83	"pid3_value": PID 4 value. Requires an installed digitizer (DIG) option.
96	"boxcar0": Boxcar 1. Requires an installed digitizer (DIG) option.
97	"boxcar1": Boxcar 2. Requires an installed digitizer (DIG) option.
112	"au_cartesian0": AU Cartesian 1. Requires an installed digitizer (DIG) option.
113	"au_cartesian1": AU Cartesian 2. Requires an installed digitizer (DIG) option.
128	"au_polar1": Au Polar 2. Requires an installed digitizer (DIG) option.
144	"pid0_shift": PID 1 Shift. Requires an installed digitizer (DIG) option.
145	"pid1_shift": PID 2 Shift. Requires an installed digitizer (DIG) option.
146	"pid2_shift": PID 3 Shift. Requires an installed digitizer (DIG) option.
147	"pid3_shift": PID 4 Shift. Requires an installed digitizer (DIG) option.
176	"awg_marker0": AWG Marker 1. Requires an installed digitizer (DIG) option.
177	"awg_marker1": AWG Marker 2. Requires an installed digitizer (DIG) option.
178	"awg_marker2": AWG Marker 3. Requires an installed digitizer (DIG) option.

179	"awg_marker3": AWG Marker 4. Requires an installed digitizer (DIG) option.
192	"awg_trigger0": AWG Trigger 1. Requires an installed digitizer (DIG) option.
193	"awg_trigger1": AWG Trigger 2. Requires an installed digitizer (DIG) option.
194	"awg_trigger2": AWG Trigger 3. Requires an installed digitizer (DIG) option.
195	"awg_trigger3": AWG Trigger 4. Requires an installed digitizer (DIG) option.
208	"pid0_error": PID 1 Error. Requires an installed digitizer (DIG) option.
209	"pid1_error": PID 2 Error. Requires an installed digitizer (DIG) option.
210	"pid2_error": PID 3 Error. Requires an installed digitizer (DIG) option.
211	"pid3_error": PID 4 Error. Requires an installed digitizer (DIG) option.

/dev..../awgs/n/triggers/n/falling

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

Sets a falling edge trigger.

/dev..../awgs/n/triggers/n/force

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

Allows to manually force a trigger.

/dev..../awgs/n/triggers/n/gate/enable

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

If enabled the trigger will be gated by the trigger gating input signal.

/dev..../awgs/n/triggers/n/gate/inputselect

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

Select the signal source used for trigger gating if gating is enabled.

0	"trigin2", "trigger_input2": Trigger In 3
1	"trigin3", "trigger_input3": Trigger In 4

/dev..../awgs/n/triggers/n/hysteresis/absolute

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

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

/dev..../awgs/n/triggers/n/hysteresis/mode

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

Selects the mode to define the hysteresis strength. The relative mode will work best over the full input range as long as the analog input signal does not suffer from excessive noise.

0	"absolute": Selects absolute hysteresis (V).
1	"relative": Selects a hysteresis relative to the adjusted full scale signal input range (%).

/dev..../awgs/n/triggers/n/hysteresis/relative

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

Hysteresis relative to the adjusted full scale signal input range. A hysteresis value larger than 100% is allowed.

/dev..../awgs/n/triggers/n/level

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

Defines the analog trigger level.

/dev..../awgs/n/triggers/n/rising

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

Sets a rising edge trigger.

/dev..../awgs/n/triggers/n/slope

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

Select the signal edge that should activate the trigger. The trigger will be level sensitive when the Level option is selected.

0	"level_sensitive": Level sensitive trigger
1	"rising_edge": Rising edge trigger
2	"falling_edge": Falling edge trigger
3	"both_edegs": Rising or falling edge trigger

/dev..../awgs/n/triggers/n/state

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

State of the Trigger: No trigger detected/trigger detected.

/dev..../awgs/n/userregs/n

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

Integer user register value. The sequencer has reading and writing access to the user register values during run time.

/dev..../awgs/n/waveform/descriptors

Properties: Read
Type: ZIVectorData
Unit: None

JSON-formatted string containing a dictionary of various properties of the current waveform: name, filename, function, channels, marker bits, length, timestamp.

/dev..../awgs/n/waveform/memoryusage

Properties: Read
Type: Double
Unit: %

Amount of the used waveform data relative to the device cache memory. The cache memory provides space for 32 kSa (32'768 Sa) per-channel of waveform data. Memory Usage over 100% means that waveforms must be loaded from the main memory of 64 MSa (67'108'864 Sa) per-channel during playback, which can lead to delays.

/dev..../awgs/n/waveform/waves/n

Properties: Read, Write
Type: ZIVectorData
Unit: None

The waveform data in the instrument's native format for the given playWave waveform index. This node will not work with subscribe as it does not push updates. For short vectors get may be used. For long vectors (causing get to time out) getAsEvent and poll can be used. The index of the waveform to be replaced can be determined using the Waveform sub-tab in the AWG tab of the LabOne User Interface.

8.2.6. BOXCARS

/dev..../boxcars/n/averagerbandwidth

Properties: Read
Type: Double
Unit: Hz

The 3 dB signal bandwidth of the Boxcar Averager is determined by the oscillation frequency and the Number of Averaging Periods set. Note: internally the boxcar signal is sampled at a rate of 14 MSa/s and the signal bandwidth of the auxiliary output is 7 MHz.

/dev..../boxcars/n/baseline/enable

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

Enable Baseline Suppression.

/dev..../boxcars/n/baseline/windowstart

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

Boxcar baseline suppression gate opening start in degrees based on one oscillator frequency period equals 360 degrees.

/dev..../boxcars/n/decimation

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

Indicates, in powers of 2, the number of averager outputs sent to the PC while Averaging Periods Boxcar integrations are obtained. Positive integer values indicate oversampling. Negative integer values indicate undersampling. Examples for oversampling values: 0 : 2^0 = 1 averager output is sent to the PC during Averaging Periods Boxcar integrations. 2 : 2^2 = 4 averager outputs are sent to the PC during Averaging Periods Boxcar integrations. -1 : 2^-1 = 0.5, only every other Averaging Periods Boxcar integrations an averager output is sent to the PC.

/dev..../boxcars/n/enable

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

Enable the BOXCAR unit.

/dev..../boxcars/n/fifooverflow

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

Data lost during streaming to PC. Sticky flag cleared by restarting the boxcar.

/dev..../boxcars/n/freqoverflow

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

Frequency for the boxcar is above or equal 450 MHz. The boxcar output may not be reliable any more. Sticky flag cleared by restarting the boxcar.

/dev..../boxcars/n/inputselect

Properties: Read, Write, Setting
Type: Integer (64 bit)
Unit: V

Select Signal Input used for the boxcar analysis.

/dev..../boxcars/n/limitrate

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

Rate Limit for Boxcar output data sent to PC. This value does not affect the Aux Output for which the effective rate is given by min(14 MSa/s , Frequency / max(1, Averaging Periods/512)).

/dev..../boxcars/n/oscselect

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

Selection of the oscillator used for the boxcar analysis.

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

/dev..../boxcars/n/overflow

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

Overflow detected. The boxcar output may not be reliable any more. Sticky flag cleared by restarting the boxcar.

/dev..../boxcars/n/periodoverflowevents

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

The boxcar averaging gate opening width is more than one cycle of the signal and should be reduced.

/dev..../boxcars/n/periods

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

Number of periods to average. This setting has no effect on Output PWAs.

/dev..../boxcars/n/rate

Properties:	Read
Type:	Double
Unit:	1/s

Current data transfer rate to the PC given by min(14 MSa/s , Frequency / max(1, Averaging Periods/512)). This value is read-only.

/dev..../boxcars/n/sample

Properties:	Read, Stream
Type:	Double
Unit:	V

Streaming node containing the output data of the boxcar.

/dev..../boxcars/n/value

Properties:	Read
Type:	Double
Unit:	Dependent

The current boxcar output.

/dev..../boxcars/n/windowsize

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

Boxcar averaging gate opening width in seconds. It can be converted to phase assuming 360 equals to a full period of the driving oscillator.

/dev..../boxcars/n/windowstart

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

Boxcar averaging gate opening start in degrees. It can be converted to time assuming 360 equals to a full period of the driving oscillator.

8.2.7. CLOCKBASE

/dev..../clockbase

Properties:	Read
Type:	Double
Unit:	Hz

Returns the internal clock frequency of the device.

8.2.8. CNTS

/dev..../cnts/n/enable

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

Enable the pulse counter unit.

/dev..../cnts/n/gateselect

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

Select the signal source used for enabling the counter in the Gated Free Running and Gated modes.

0	"trigin0", "trigger_input0": Trigger/Ref Input 1 (front panel).
1	"trigin1", "trigger_input1": Trigger/Ref Input 2 (front panel).
2	"trigin2", "trigger_input2": Trigger Input 3 (rear panel).
3	"trigin3", "trigger_input3": Trigger Input 4 (rear panel).
4	"awg_trigger0": AWG Trigger 1.
5	"awg_trigger1": AWG Trigger 2.
6	"awg_trigger2": AWG Trigger 3.
7	"awg_trigger3": AWG Trigger 4.

/dev..../cnts/n/inputselect

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

Select the counter signal source.

0	DIO Bit 0.
1	DIO Bit 1.
2	DIO Bit 2.
3	DIO Bit 3.
4	DIO Bit 4.
5	DIO Bit 5.
6	DIO Bit 6.
7	DIO Bit 7.
8	DIO Bit 8.
9	DIO Bit 9.
10	DIO Bit 10.
11	DIO Bit 11.
12	DIO Bit 12.
13	DIO Bit 13.
14	DIO Bit 14.
15	DIO Bit 15.
16	DIO Bit 16.
17	DIO Bit 17.
18	DIO Bit 18.
19	DIO Bit 19.
20	DIO Bit 20.
21	DIO Bit 21.
22	DIO Bit 22.
23	DIO Bit 23.
24	DIO Bit 24.
25	DIO Bit 25.
26	DIO Bit 26.
27	DIO Bit 27.
28	DIO Bit 28.
29	DIO Bit 29.
30	DIO Bit 30.
31	DIO Bit 31.
32	"trigin0", "trigger_input0": Trigger/Ref Input 1 (front panel).
33	"trigin1", "trigger_input1": Trigger/Ref Input 2 (front panel).
34	"trigin2", "trigger_input2": Trigger Input 3 (rear panel).
35	"trigin3", "trigger_input3": Trigger Input 4 (rear panel).

/dev..../cnts/n/integrate

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

Sum up counter values over time.

/dev..../cnts/n/mode

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

Select the run mode of the counter unit.

1	"free_running": Free Running: The counter runs on a repetitive time base defined by the Period field. At the beginning of each period the counter is reset, and at the end, the accumulated number of counts is output.
2	"gated_free_running": Gated Free Running: The counter runs on a repetitive time base defined by the Period field. The Gate Input signal controls when the unit counter is allowed to run. The counter as well as the timer is reset when the Gate Input signal is low. The counter will only deliver new values if the Gate Input signal is high for a time longer than the configured Period.
3	"gated": Gated: The counter is controlled with the Gate Input signal. The counter is enabled at the rising edge of the Gate Input signal and disabled at the falling edge. Pulses are counted as long as the counter is enabled. The accumulated number of counts is output on the falling edge of the Gate Input signal.
4	"time_tagging": Time Tagging: Every pulse is detected individually and tagged with the time of the event. The Period defines the minimum hold-off time between the tagging of two subsequent pulses. If more than one pulse occurs within the window defined by the Period, then the pulses are accumulated and output at the end of the window. The Period effectively determines the maximum rate at which pulse information can be transmitted to the host PC.

/dev..../cnts/n/operation

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

Select the arithmetic operation (addition, subtraction) applied to the counter unit outputs. 'Other counter' refers to the grouping of the counter units: 1 with 2, and 3 with 4.

0	"none": None
1	"add_other_counter": Add Other Counter
2	"subtract_other_counter": Subtract Other Counter

/dev..../cnts/n/period

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

Set the period used for the Free Running and Gated Free Running modes. Also sets the hold-off time for the Time Tagging mode.

/dev..../cnts/n/sample

Properties: Read, Stream
Type: ZICntSample
Unit: None

Streaming node containing counter values.

/dev..../cnts/n/trigfalling

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

Performs a trigger event when the source signal crosses the trigger level from high to low. For dual edge triggering, select also the rising edge.

/dev..../cnts/n/trigrising

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

Performs a trigger event when the source signal crosses the trigger level from low to high. For dual edge triggering, select also the falling edge.

/dev..../cnts/n/value

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

Counter output value.

8.2.9. DEMODS

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

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

Selects the input signal for the demodulator.

0	"sigin0", "signal_input0": Sig In 1
1	"sigin1", "signal_input1": Sig In 2
2	"trigin0", "trigger_input0": Trigger Input 1
3	"trigin1", "trigger_input1": Trigger Input 2
4	"auxout0", "auxiliary_output0": Aux Out 1
5	"auxout1", "auxiliary_output1": Aux Out 2
6	"auxout2", "auxiliary_output2": Aux Out 3
7	"auxout3", "auxiliary_output3": Aux Out 4
8	"auxin0", "auxiliary_input0": Aux In 1
9	"auxin1", "auxiliary_input1": Aux In 2
10	"demod3": φ Demod 4
11	"demod7": φ Demod 8

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

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

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

/dev..../demods/n/enable

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

Enables the data acquisition for the corresponding demodulator. Note: increasing number of active demodulators increases load on the physical connection to the host computer.

0	"off": OFF: demodulator inactive
1	"on": ON: demodulator active

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

Properties: Read
Type: Double
Unit: Hz

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

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

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

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

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

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

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

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

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

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

Connects the demodulator with the supplied oscillator. Number of available oscillators depends on the installed options.

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

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

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

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

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

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

Phase shift applied to the reference input of the demodulator.

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

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

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

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

Properties: Read, Stream
Type: ZIDemodSample
Unit: Dependent

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

/dev..../demods/n/sinc

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

Enables the sinc filter. When the filter bandwidth is comparable to or larger than the demodulation frequency, the demodulator output may contain frequency components at the frequency of demodulation and its higher harmonics. The sinc is an additional filter that attenuates these unwanted components in the demodulator output.

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

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

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

/dev..../demods/n/trigger

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

Selects the acquisition mode (i.e. triggering) of the demodulator.

0	"continuous": Continuous: demodulator data is continuously streamed to the host computer.
1	"trigin2_rising", "trigger_input2_rising": Trigger Input 3: rising edge triggered.
2	"trigin2_falling", "trigger_input2_falling": Trigger Input 3: falling edge triggered.
3	"trigin2_both", "trigger_input2_both": Trigger Input 3: triggering on both rising and falling edge.
4	"trigin3_rising", "trigger_input3_rising": Trigger Input 4: rising edge triggered.
5	"trigin2or3_rising", "trigger_input2or3_rising": Trigger Input 3 or 4: rising edge triggered on either input.
8	"trigin3_falling", "trigger_input3_falling": Trigger Input 4: falling edge triggered.
10	"trigin2or3_falling", "trigger_input2or3_falling": Trigger Input 3 or 4: falling edge triggered on either input.
12	"trigin3_both", "trigger_input3_both": Trigger Input 4: triggering on both rising and falling edge.
15	"trigin2or3_both", "trigger_input2or3_both": Trigger Input 3 or 4: triggering on both rising and falling edge or either trigger input.
16	"trigin2_low", "trigger_input2_low": Trigger Input 3: demodulator data is streamed to the host computer when the level is low (TTL).
32	"trigin2_high", "trigger_input2_high": Trigger Input 3: demodulator data is streamed to the host computer when the level is high (TTL).
64	"trigin3_low", "trigger_input3_low": Trigger Input 4: demodulator data is streamed to the host computer when the level is low (TTL).
80	"trigin2or3_low", "trigger_input2or3_low": Trigger Input 3 or 4: demodulator data is streamed to the host computer when either level is low (TTL).
128	"trigin3_high", "trigger_input3_high": Trigger Input 4: demodulator data is streamed to the host computer when the level is high (TTL).
160	"trigin2or3_high", "trigger_input2or3_high": Trigger Input 3 or 4: demodulator data is streamed to the host computer when either level is high (TTL).
1048576	"awg_trigger0_rising": AWG Trigger 1: rising edge triggered. Requires an installed AWG option.
2097152	"awg_trigger1_rising": AWG Trigger 2: rising edge triggered. Requires an installed AWG option.
4194304	"awg_trigger2_rising": AWG Trigger 3: rising edge triggered. Requires an installed AWG option.
8388608	"awg_trigger3_rising": AWG Trigger 4: rising edge triggered. Requires an installed AWG option.
16777216	"awg_trigger0_high": AWG Trigger 1: demodulator data is streamed to the host computer when the level is high. Requires an installed AWG option.
33554432	"awg_trigger1_high": AWG Trigger 2: demodulator data is streamed to the host computer when the level is high. Requires an installed AWG option.
67108864	"awg_trigger2_high": AWG Trigger 3: demodulator data is streamed to the host computer when the level is high. Requires an installed AWG option.
134217728	"awg_trigger3_high": AWG Trigger 4: demodulator data is streamed to the host computer when the level is high. Requires an installed AWG option.

8.2.10. DIOS

/dev..../dios/n/auxdrive

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

Not used. Reserved for future use.

/dev..../dios/n/decimation

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

Sets the decimation factor for DIO data streamed to the host computer.

/dev..../dios/n/drive

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

When on (1), the corresponding 8-bit bus is in output mode. When off (0), it is in input mode. Bit 0 corresponds to the least significant byte. For example, the value 1 drives the least significant byte, the value 8 drives the most significant byte.

/dev..../dios/n/extclk

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

Select DIO internal or external clocking.

0	"internal": The DIO is internally clocked with a frequency of 56.25 MHz.
1	"external": The DIO is externally clocked with a clock signal connected to DIO Pin 68. The available range is from 1 Hz up to the internal clock frequency.

/dev..../dios/n/input

Properties: Read, Stream
Type: ZIDIOSample
Unit: None

Gives the value of the DIO input for those bytes where drive is disabled.

/dev..../dios/n/mode

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

Select DIO mode

0	"manual": Enables manual control of the DIO output bits.
1	"awg_sequencer_commands": Enables setting of DIO output values by AWG sequencer commands.

/dev..../dios/n/output

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

Sets the value of the DIO output for those bytes where 'drive' is enabled.

8.2.11. EXTREFS

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

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

Indicates the input signal selection for the selected demodulator.

0	"sigin0", "signal_input0": Signal Input 1 is connected to the corresponding demodulator.
1	"sigin1", "signal_input1": Signal Input 2 is connected to the corresponding demodulator.
2	"trigin0", "trigger_input0": Trigger Input 1 is connected to the corresponding demodulator.
3	"trigin1", "trigger_input1": Trigger Input 2 is connected to the corresponding demodulator.
4	"auxout0", "auxiliary_output0": Auxiliary Output 1 is connected to the corresponding demodulator.
5	"auxout1", "auxiliary_output1": Auxiliary Output 2 is connected to the corresponding demodulator.
6	"auxout2", "auxiliary_output2": Auxiliary Output 3 is connected to the corresponding demodulator.
7	"auxout3", "auxiliary_output3": Auxiliary Output 4 is connected to the corresponding demodulator.
8	"auxin0", "auxiliary_input0": Auxiliary Input 1 is connected to the corresponding demodulator.
9	"auxin1", "auxiliary_input1": Auxiliary Input 2 is connected to the corresponding demodulator.
10	"demod3": Oscillator Phase of Demod 4 is connected to the corresponding demodulator. This selection combined with the demodulator's ExtRef Mode can be used to phase-lock two internal oscillators.
11	"demod7": Oscillator Phase of Demod 8 is connected to the corresponding demodulator. This selection combined with the demodulator's ExtRef Mode can be used to phase-lock two internal oscillators.

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

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

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

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

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

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

Indicates the demodulator connected to the extref channel.

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

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

Enables the external reference.

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

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

Indicates whether the external reference is locked.

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

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

Indicates which oscillator is being locked to the external reference.

8.2.12. 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.

8.2.13. INPUTPWAS

/dev..../inputpwas/n/acquisitiontime

Properties: Read
Type: Double
Unit: s

Estimated time needed for recording of the specified number of samples.

/dev..../inputpwas/n/enable

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

Enables the PWA for data acquisition. Depending on the value of the 'single' node, this may be continuous, or a one-shot acquisition.

/dev..../inputpwas/n/harmonic

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

Defines the width of the PWA acquisition window in terms of harmonics of the selected reference oscillator frequency. e.g. A value of 1 corresponds to a complete cycle of the reference frequency. A value of 2 corresponds to half of a cycle (180 deg). A value of 4 corresponds to a quarter of a cycle (90 deg).

/dev..../inputpwas/n/holdoff

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

Not used. Reserved for future use.

/dev..../inputpwas/n/inputinterlock

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

Interlock PWA and Boxcar Input settings. If on, the input signal has to be selected via the boxcar inputselect node.

/dev..../inputpwas/n/inputselect

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

Select PWA input signal. Has no effect if the inputinterlock node is set active.

0	"sigin0", "signal_input0": Signal Input 1
1	"sigin1", "signal_input1": Signal Input 2
2	"trigin0", "trigger_input0": Trigger 1 (Front)
3	"trigin1", "trigger_input1": Trigger 2 (Front)
4	"auxout0", "auxiliary_output0": Aux Output 1. Requires an installed digitizer (DIG) option.
5	"auxout1", "auxiliary_output1": Aux Output 2. Requires an installed digitizer (DIG) option.
6	"auxout2", "auxiliary_output2": Aux Output 3. Requires an installed digitizer (DIG) option.
7	"auxout3", "auxiliary_output3": Aux Output 4. Requires an installed digitizer (DIG) option.
8	"auxin0", "auxiliary_input0": Aux Input 1
9	"auxin1", "auxiliary_input1": Aux Input 2
10	"demod3": Osc φ Demod 4
11	"demod7": Osc φ Demod 8

/dev..../inputpwas/n/looptime

Properties:	Read
Type:	Double
Unit:	None

Not used. Reserved for future use.

/dev..../inputpwas/n/mode

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

Sets the data interpretation mode. Measurement data can be interpreted in four different modes and displayed over either phase (native), time, frequency (FFT) or harmonics of the base frequency (FFT).

0	"phase": Phase
1	"time": Time
2	"frequency": Freq Domain (FFT)
3	"harmonics": Harmonics (FFT)

/dev..../inputpwas/n/oscselect

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

Select reference oscillator for PWA signal acquisition. Has no effect if the inputinterlock node is set active.

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

/dev..../inputpwas/n/progress

Properties: Read
Type: Double
Unit: %

State of the PWA acquisition in percent.

/dev..../inputpwas/n/samplecount

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

Defines the number of samples acquired of each PWA data set (450 MSa/s).

/dev..../inputpwas/n/shift

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

Defines the start of the PWA range in terms of the phase of the reference frequency.

/dev..../inputpwas/n/single

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

If set to 1, a single data acquisition is executed when the PWA is enabled.

/dev..../inputpwas/n/status

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

Indicates whether the PWA is acquiring data. 0 = PWA is inactive, 1 = acquiring data, 2 = terminating acquisition.

/dev..../inputpwas/n/termination

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

Defines the condition under which an acquisition is prematurely terminated (before acquiring the specified number of samples). Normally has the value 1.

/dev..../inputpwas/n/wave

Properties: Read, Stream
Type: ZIPWAWave
Unit: Dependent

Streaming node that delivers the acquired data from the PWA.

8.2.14. MODS

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

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

Set the carrier amplitude

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

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

Enable the modulation

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

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

Set the harmonic of the carrier frequency. 1 = Fundamental

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

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

Select Signal Input for the carrier demodulation

0	"sigin0", "signal_input0": Sig In 1
1	"sigin1", "signal_input1": Sig In 2
2	"trigin0", "trigger_input0": Trigger Input 1
3	"trigin1", "trigger_input1": Trigger Input 2
4	"auxout0", "auxiliary_output0": Aux Out 1
5	"auxout1", "auxiliary_output1": Aux Out 2
6	"auxout2", "auxiliary_output2": Aux Out 3
7	"auxout3", "auxiliary_output3": Aux Out 4
8	"auxin0", "auxiliary_input0": Aux In 1
9	"auxin1", "auxiliary_input1": Aux In 2
10	"demod3": φ Demod 4
11	"demod7": φ Demod 8

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

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

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

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

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

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

Select the oscillator for the carrier signal.

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

/dev..../mods/n/carrier/phaseadjust

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

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

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

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

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

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

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

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

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

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

Enables the modulation.

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

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

FM mode peak deviation value.

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

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

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

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

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

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

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

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

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

Select the modulation mode.

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

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

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

Select Signal Output.

0	"none": None
1	"sigout0", "signal_output0": Signal Output 1
2	"sigout1", "signal_output1": Signal Output 2
3	"sigout0and1", "signal_output0and1": Signal Output 1 and 2

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

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

Set the amplitude of the sideband components.

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

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

Enable the signal generation for the respective sideband

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

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

Set harmonic of the sideband frequencies. 1 = fundamental

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

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

Select Signal Input for the sideband demodulation

0	"sigin0", "signal_input0": Sig In 1
1	"sigin1", "signal_input1": Sig In 2
2	"trigin0", "trigger_input0": Trigger Input 1
3	"trigin1", "trigger_input1": Trigger Input 2
4	"auxout0", "auxiliary_output0": Aux Out 1
5	"auxout1", "auxiliary_output1": Aux Out 2
6	"auxout2", "auxiliary_output2": Aux Out 3
7	"auxout3", "auxiliary_output3": Aux Out 4
8	"auxin0", "auxiliary_input0": Aux In 1
9	"auxin1", "auxiliary_input1": Aux In 2
10	"demod3": φ Demod 4
11	"demod7": φ Demod 8

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

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

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

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

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

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

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

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

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

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

Select the oscillator for the second sideband.

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

/dev..../mods/n/sidebands/n/phaseadjust

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

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

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

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

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

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

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

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

8.2.15. OSCS

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

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

Frequency control for each oscillator.

8.2.16. OUTPUTPWAS

/dev..../outputpwas/n/acquisitiontime

Properties: Read
Type: Double
Unit: s

Estimated time needed for recording of the specified number of samples.

/dev..../outputpwas/n/enable

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

Enables the Output PWA for data acquisition. Depending on the value of the 'single' node, this may be continuous, or a one-shot acquisition.

/dev..../outputpwas/n/harmonic

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

Defines the width of the PWA acquisition window in terms of harmonics of the selected reference oscillator frequency. e.g. A value of 1 corresponds to a complete cycle of the reference frequency. A value of 2 corresponds to half of a cycle (180 deg). A value of 4 corresponds to a quarter of a cycle (90 deg).

/dev..../outputpwas/n/holdoff

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

Not used. Reserved for future use.

/dev..../outputpwas/n/inputselect

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

Select PWA input signal.

0	"boxcar0": Boxcar 1
1	"boxcar1": Boxcar 2

/dev..../outputpwas/n/looptime

Properties: Read
Type: Double
Unit: None

Not used. Reserved for future use.

/dev..../outputpwas/n/mode

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

Sets the data interpretation mode. Measurement data can be interpreted in four different modes and displayed over either phase (native), time, frequency (FFT) or harmonics of the base frequency (FFT).

0	"phase": Phase
1	"time": Time
2	"frequency": Freq Domain (FFT)
3	"harmonics": Harmonics (FFT)

/dev..../outputpwas/n/oscselect

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

Select reference oscillator for Output PWA signal acquisition.

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

/dev..../outputpwas/n/progress

Properties:	Read
Type:	Double
Unit:	%

State of the Output PWA acquisition in percent.

/dev..../outputpwas/n/samplecount

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

Defines the number of samples acquired of each Output PWA data set (450 MSa/s).

/dev..../outputpwas/n/shift

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

Defines the start of the Output PWA range in terms of the phase of the reference frequency.

/dev..../outputpwas/n/single

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

If set to 1, a single data acquisition is executed when the Output PWA is enabled.

/dev..../outputpwas/n/status

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

Indicates whether the Output PWA is acquiring data. 0 = Output PWA is inactive, 1 = acquiring data, 2 = terminating acquisition.

/dev..../outputpwas/n/termination

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

Defines the condition under which an acquisition is prematurely terminated (before acquiring the specified number of samples). Normally has the value 1.

/dev..../outputpwas/n/wave

Properties: Read, Stream
Type: ZIPWAWave
Unit: Dependent

Streaming node that delivers the acquired data from the Output PWA.

8.2.17. PIDS

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

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

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

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

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

PID derivative gain.

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

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

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

0	"sigin0", "signal_input0": Signal Input 1 is connected to the corresponding demodulator.
1	"sigin1", "signal_input1": Signal Input 2 is connected to the corresponding demodulator.
2	"trigin0", "trigger_input0": Trigger Input 1 is connected to the corresponding demodulator.
3	"trigin1", "trigger_input1": Trigger Input 2 is connected to the corresponding demodulator.
4	"auxout0", "auxiliary_output0": Auxiliary Output 1 is connected to the corresponding demodulator.
5	"auxout1", "auxiliary_output1": Auxiliary Output 2 is connected to the corresponding demodulator.
6	"auxout2", "auxiliary_output2": Auxiliary Output 3 is connected to the corresponding demodulator.
7	"auxout3", "auxiliary_output3": Auxiliary Output 4 is connected to the corresponding demodulator.
8	"auxin0", "auxiliary_input0": Auxiliary Input 1 is connected to the corresponding demodulator.
9	"auxin1", "auxiliary_input1": Auxiliary Input 2 is connected to the corresponding demodulator.
10	"demod3": Oscillator Phase of Demod 4 is connected to the corresponding demodulator. This selection combined with the demodulator's ExtRef Mode can be used to phase-lock two internal oscillators.
11	"demod7": Oscillator Phase of Demod 8 is connected to the corresponding demodulator. This selection combined with the demodulator's ExtRef Mode can be used to phase-lock two internal oscillators.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Enable the PID controller.

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

Properties: Read
Type: Double
Unit: Dependent

Error = Set point - PID Input.

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

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

PID integral gain I.

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

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

Select input source of PID controller.

0	"demod_x": Demodulator X
1	"demod_y": Demodulator Y
2	"demod_r": Demodulator R
3	"demod_theta": Demodulator Theta
4	"auxin", "auxiliary_input": Aux Input
5	"auxout", "auxiliary_output": Aux Output
6	"au_cartesian": Arithmetic Unit Cartesian
7	"au_polar": Arithmetic Unit Polar

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

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

Select input channel of PID controller.

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

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

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

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

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

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

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

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

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

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

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

Sets the operation mode of the PID module.

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

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

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

Select output of the PID controller.

0	"sigout0_amp", "signal_output0_amplitude": Driving Signal Output 1 amplitudes
1	"sigout1_amp", "signal_output1_amplitude": Driving Signal Output 2 amplitudes
2	"oscillator_frequency": Controlling any of the internal oscillator frequencies
3	"demod_phase": Controlling any of the demodulator phase set points
5	"auxout_offset", "auxiliary_output_offset": Driving any of the 4 Auxiliary Outputs' offset
7	"sigout_offset", "signal_output_offset": Driving the main Signal Output's offset

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

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

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

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

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

PID Proportional gain P.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

PID controller setpoint

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

Properties: Read
Type: Double
Unit: Dependent

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

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

Properties: Read
Type: Double
Unit: 1/s

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

/dev..../pids/n/stream/error

Properties: Read, Stream
Type: Double
Unit: Dependent

PID Error = Set point - PID Input.

/dev..../pids/n/stream/overflow

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

Indicates the streaming fifo overflow state. 0 = OK, 1 = overflow.

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

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

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

/dev..../pids/n/stream/shift

Properties: Read, Stream
Type: Double
Unit: Dependent

Gives the difference between the current output value and the center value. Shift = PError + IInt(Error, dt) + D*dError/dt

/dev..../pids/n/stream/value

Properties:	Read, Stream
Type:	Double
Unit:	Dependent

Gives the current PID output value.

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

Properties:	Read
Type:	Double
Unit:	Dependent

Gives the current PID output value.

8.2.18. SCOPES

/dev..../scopes/n/channel

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

Activates the scope channels.

1	Only channel 1 is active.
2	Only channel 2 is active.
3	"both": Both, channel 1 and 2 are active.

/dev..../scopes/n/channels/n/bwlimit

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

Selects between sample decimation and sample averaging. Averaging avoids aliasing, but may conceal signal peaks.

0	"on": On: Selects sample averaging for sample rates lower than the maximal available sampling rate.
1	"off": OFF: Selects sample decimation for sample rates lower than the maximal available sampling rate.

/dev..../scopes/n/channels/n/fullscale

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

Indicates the full scale value of the scope channel.

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

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

Selects the scope input signal.

0	"sigin0", "signal_input0": Signal Input 1
1	"sigin1", "signal_input1": Signal Input 2
2	"trigin0", "trigger_input0": Trigger Input 1
3	"trigin1", "trigger_input1": Trigger Input 2
4	"auxout0", "auxiliary_output0": Aux Output 1. Requires an installed digitizer (DIG) option.
5	"auxout1", "auxiliary_output1": Aux Output 2. Requires an installed digitizer (DIG) option.
6	"auxout2", "auxiliary_output2": Aux Output 3. Requires an installed digitizer (DIG) option.
7	"auxout3", "auxiliary_output3": Aux Output 4. Requires an installed digitizer (DIG) option.
8	"auxin0", "auxiliary_input0": Aux Input 1
9	"auxin1", "auxiliary_input1": Aux Input 2
10	"demod3": Osc φ Demod 4
11	"demod7": Osc φ Demod 8
16	"demod0_x": Demod 1 X. Requires an installed digitizer (DIG) option.
17	"demod1_x": Demod 2 X. Requires an installed digitizer (DIG) option.
18	"demod2_x": Demod 3 X. Requires an installed digitizer (DIG) option.
19	"demod3_x": Demod 4 X. Requires an installed digitizer (DIG) option.
20	"demod4_x": Demod 5 X. Requires an installed digitizer (DIG) option.
21	"demod5_x": Demod 6 X. Requires an installed digitizer (DIG) option.
22	"demod6_x": Demod 7 X. Requires an installed digitizer (DIG) option.
23	"demod7_x": Demod 8 X. Requires an installed digitizer (DIG) option.
32	"demod0_y": Demod 1 Y. Requires an installed digitizer (DIG) option.
33	"demod1_y": Demod 2 Y. Requires an installed digitizer (DIG) option.
34	"demod2_y": Demod 3 Y. Requires an installed digitizer (DIG) option.
35	"demod3_y": Demod 4 Y. Requires an installed digitizer (DIG) option.
36	"demod4_y": Demod 5 Y. Requires an installed digitizer (DIG) option.
37	"demod5_y": Demod 6 Y. Requires an installed digitizer (DIG) option.
38	"demod6_y": Demod 7 Y. Requires an installed digitizer (DIG) option.
39	"demod7_y": Demod 8 Y. Requires an installed digitizer (DIG) option.
48	"demod0_r": Demod 1 R. Requires an installed digitizer (DIG) option.
49	"demod1_r": Demod 2 R. Requires an installed digitizer (DIG) option.
50	"demod2_r": Demod 3 R. Requires an installed digitizer (DIG) option.
51	"demod3_r": Demod 4 R. Requires an installed digitizer (DIG) option.
52	"demod4_r": Demod 5 R. Requires an installed digitizer (DIG) option.
53	"demod5_r": Demod 6 R. Requires an installed digitizer (DIG) option.
54	"demod6_r": Demod 7 R. Requires an installed digitizer (DIG) option.
55	"demod7_r": Demod 8 R. Requires an installed digitizer (DIG) option.
64	"demod0_theta": Demod 1 Θ. Requires an installed digitizer (DIG) option.
65	"demod1_theta": Demod 2 Θ. Requires an installed digitizer (DIG) option.
66	"demod2_theta": Demod 3 Θ. Requires an installed digitizer (DIG) option.
67	"demod3_theta": Demod 4 Θ. Requires an installed digitizer (DIG) option.
68	"demod4_theta": Demod 5 Θ. Requires an installed digitizer (DIG) option.
69	"demod5_theta": Demod 6 Θ. Requires an installed digitizer (DIG) option.
70	"demod6_theta": Demod 7 Θ. Requires an installed digitizer (DIG) option.
71	"demod7_theta": Demod 8 Θ. Requires an installed digitizer (DIG) option.
80	"pid0_value": PID 1 value. Requires an installed digitizer (DIG) option.
81	"pid1_value": PID 2 value. Requires an installed digitizer (DIG) option.
82	"pid2_value": PID 3 value. Requires an installed digitizer (DIG) option.
83	"pid3_value": PID 4 value. Requires an installed digitizer (DIG) option.
96	"boxcar0": Boxcar 1. Requires an installed digitizer (DIG) option.
97	"boxcar1": Boxcar 2. Requires an installed digitizer (DIG) option.
112	"au_cartesian0": AU Cartesian 1. Requires an installed digitizer (DIG) option.
113	"au_cartesian1": AU Cartesian 2. Requires an installed digitizer (DIG) option.
128	"au_polar1": Au Polar 2. Requires an installed digitizer (DIG) option.
144	"pid0_shift": PID 1 Shift. Requires an installed digitizer (DIG) option.
145	"pid1_shift": PID 2 Shift. Requires an installed digitizer (DIG) option.
146	"pid2_shift": PID 3 Shift. Requires an installed digitizer (DIG) option.
147	"pid3_shift": PID 4 Shift. Requires an installed digitizer (DIG) option.
160	Reserved for future use.
161	Reserved for future use.
176	"awg_marker0": AWG Marker 1. Requires an installed digitizer (DIG) option.

177	"awg_marker1": AWG Marker 2. Requires an installed digitizer (DIG) option.
178	"awg_marker2": AWG Marker 3. Requires an installed digitizer (DIG) option.
179	"awg_marker3": AWG Marker 4. Requires an installed digitizer (DIG) option.
192	"awg_trigger0": AWG Trigger 1. Requires an installed digitizer (DIG) option.
193	"awg_trigger1": AWG Trigger 2. Requires an installed digitizer (DIG) option.
194	"awg_trigger2": AWG Trigger 3. Requires an installed digitizer (DIG) option.
195	"awg_trigger3": AWG Trigger 4. Requires an installed digitizer (DIG) option.
208	"pid0_error": PID 1 Error. Requires an installed digitizer (DIG) option.
209	"pid1_error": PID 2 Error. Requires an installed digitizer (DIG) option.
210	"pid2_error": PID 3 Error. Requires an installed digitizer (DIG) option.
211	"pid3_error": PID 4 Error. Requires an installed digitizer (DIG) option.

/dev..../scopes/n/channels/n/limitlower

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

Lower limit of the scope full scale range. For demodulator, PID, Boxcar, and AU signals the limit should be adjusted so that the signal covers the specified range to achieve optimal resolution.

/dev..../scopes/n/channels/n/limitupper

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

Upper limit of the scope full scale range. For demodulator, PID, Boxcar, and AU signals the limit should be adjusted so that the signal covers the specified range to achieve optimal resolution.

/dev..../scopes/n/channels/n/offset

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

Indicates the offset value of the scope channel.

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

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

Enables the acquisition of scope shots.

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

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

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

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

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

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

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

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

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

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

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

Puts the Scope into single shot mode.

/dev..../scopes/n/stream/enables/n

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

Enable scope streaming for the specified channel. This allows for continuous recording of scope data on the plotter and streaming to disk. Note: scope streaming requires the DIG option.

/dev..../scopes/n/stream/rate

Properties: Read, Write, Setting
Type: Integer (enumerated)
Unit: Hz

Streaming Rate of the scope channels. The streaming rate can be adjusted independent from the scope sampling rate. The maximum rate depends on the interface used for transfer. Note: scope streaming requires the DIG option.

6	"28.1_MHz": 28.1 MHz
7	"14_MHz": 14.0 MHz
8	"7.03_MHz": 7.03 MHz
9	"3.5_MHz": 3.50 MHz
10	"1.75_MHz": 1.75 MHz
11	"880_kHz": 880 kHz
12	"440_kHz": 440 kHz
13	"220_kHz": 220 kHz
14	"110_kHz": 110 kHz
15	"54.9_kHz": 54.9 kHz
16	"27.5_kHz": 27.5 kHz

/dev..../scopes/n/stream/sample

Properties:	Read, Stream
Type:	ZIScopeWave
Unit:	None

Streaming node containing scope sample data. Note: scope streaming requires the DIG option.

/dev..../scopes/n/time

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

Defines the time base of the scope from the divider exponent of the instrument's clock base. The resulting sampling time is 2^n/clockbase.

0	1.80 GHz
1	900 MHz
2	450 MHz
3	225 MHz
4	113 MHz
5	56.2 MHz
6	28.1 MHz
7	14.0 MHz
8	7.03 MHz
9	3.50 MHz
10	1.75 MHz
11	880 kHz
12	440 kHz
13	220 kHz
14	110 kHz
15	54.9 kHz
16	27.5 kHz

/dev..../scopes/n/trigchannel

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

Selects the trigger source signal.

0	"sigin0", "signal_input0": Signal Input 1
1	"sigin1", "signal_input1": Signal Input 2
2	"trigin0", "trigger_input0": Trigger Input 1
3	"trigin1", "trigger_input1": Trigger Input 2
4	"auxout0", "auxiliary_output0": Aux Output 1. Requires an installed digitizer (DIG) option.
5	"auxout1", "auxiliary_output1": Aux Output 2. Requires an installed digitizer (DIG) option.
6	"auxout2", "auxiliary_output2": Aux Output 3. Requires an installed digitizer (DIG) option.
7	"auxout3", "auxiliary_output3": Aux Output 4. Requires an installed digitizer (DIG) option.
8	"auxin0", "auxiliary_input0": Aux Input 1
9	"auxin1", "auxiliary_input1": Aux Input 2
10	"demod3": Osc φ Demod 4
11	"demod7": Osc φ Demod 8
16	"demod0_x": Demod 1 X. Requires an installed digitizer (DIG) option.
17	"demod1_x": Demod 2 X. Requires an installed digitizer (DIG) option.
18	"demod2_x": Demod 3 X. Requires an installed digitizer (DIG) option.
19	"demod3_x": Demod 4 X. Requires an installed digitizer (DIG) option.
20	"demod4_x": Demod 5 X. Requires an installed digitizer (DIG) option.
21	"demod5_x": Demod 6 X. Requires an installed digitizer (DIG) option.
22	"demod6_x": Demod 7 X. Requires an installed digitizer (DIG) option.
23	"demod7_x": Demod 8 X. Requires an installed digitizer (DIG) option.
32	"demod0_y": Demod 1 Y. Requires an installed digitizer (DIG) option.
33	"demod1_y": Demod 2 Y. Requires an installed digitizer (DIG) option.
34	"demod2_y": Demod 3 Y. Requires an installed digitizer (DIG) option.
35	"demod3_y": Demod 4 Y. Requires an installed digitizer (DIG) option.
36	"demod4_y": Demod 5 Y. Requires an installed digitizer (DIG) option.
37	"demod5_y": Demod 6 Y. Requires an installed digitizer (DIG) option.
38	"demod6_y": Demod 7 Y. Requires an installed digitizer (DIG) option.
39	"demod7_y": Demod 8 Y. Requires an installed digitizer (DIG) option.
48	"demod0_r": Demod 1 R. Requires an installed digitizer (DIG) option.
49	"demod1_r": Demod 2 R. Requires an installed digitizer (DIG) option.
50	"demod2_r": Demod 3 R. Requires an installed digitizer (DIG) option.
51	"demod3_r": Demod 4 R. Requires an installed digitizer (DIG) option.
52	"demod4_r": Demod 5 R. Requires an installed digitizer (DIG) option.
53	"demod5_r": Demod 6 R. Requires an installed digitizer (DIG) option.
54	"demod6_r": Demod 7 R. Requires an installed digitizer (DIG) option.
55	"demod7_r": Demod 8 R. Requires an installed digitizer (DIG) option.
64	"demod0_theta": Demod 1 Θ. Requires an installed digitizer (DIG) option.
65	"demod1_theta": Demod 2 Θ. Requires an installed digitizer (DIG) option.
66	"demod2_theta": Demod 3 Θ. Requires an installed digitizer (DIG) option.
67	"demod3_theta": Demod 4 Θ. Requires an installed digitizer (DIG) option.
68	"demod4_theta": Demod 5 Θ. Requires an installed digitizer (DIG) option.
69	"demod5_theta": Demod 6 Θ. Requires an installed digitizer (DIG) option.
70	"demod6_theta": Demod 7 Θ. Requires an installed digitizer (DIG) option.
71	"demod7_theta": Demod 8 Θ. Requires an installed digitizer (DIG) option.
80	"pid0_value": PID 1 value. Requires an installed digitizer (DIG) option.
81	"pid1_value": PID 2 value. Requires an installed digitizer (DIG) option.
82	"pid2_value": PID 3 value. Requires an installed digitizer (DIG) option.
83	"pid3_value": PID 4 value. Requires an installed digitizer (DIG) option.
96	"boxcar0": Boxcar 1. Requires an installed digitizer (DIG) option.
97	"boxcar1": Boxcar 2. Requires an installed digitizer (DIG) option.
112	"au_cartesian0": AU Cartesian 1. Requires an installed digitizer (DIG) option.
113	"au_cartesian1": AU Cartesian 2. Requires an installed digitizer (DIG) option.
128	"au_polar0": AU Polar 1. Requires an installed digitizer (DIG) option.
129	"au_polar1": Au Polar 2. Requires an installed digitizer (DIG) option.
144	"pid0_shift": PID 1 Shift. Requires an installed digitizer (DIG) option.
145	"pid1_shift": PID 2 Shift. Requires an installed digitizer (DIG) option.
146	"pid2_shift": PID 3 Shift. Requires an installed digitizer (DIG) option.
147	"pid3_shift": PID 4 Shift. Requires an installed digitizer (DIG) option.
176	"awg_marker0": AWG Marker 1. Requires an installed digitizer (DIG) option.
177	"awg_marker1": AWG Marker 2. Requires an installed digitizer (DIG) option.

178	"awg_marker2": AWG Marker 3. Requires an installed digitizer (DIG) option.
179	"awg_marker3": AWG Marker 4. Requires an installed digitizer (DIG) option.
192	"awg_trigger0": AWG Trigger 1. Requires an installed digitizer (DIG) option.
193	"awg_trigger1": AWG Trigger 2. Requires an installed digitizer (DIG) option.
194	"awg_trigger2": AWG Trigger 3. Requires an installed digitizer (DIG) option.
195	"awg_trigger3": AWG Trigger 4. Requires an installed digitizer (DIG) option.
208	"pid0_error": PID 1 Error. Requires an installed digitizer (DIG) option.
209	"pid1_error": PID 2 Error. Requires an installed digitizer (DIG) option.
210	"pid2_error": PID 3 Error. Requires an installed digitizer (DIG) option.
211	"pid3_error": PID 4 Error. Requires an installed digitizer (DIG) option.

/dev..../scopes/n/trigdelay

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

Trigger position relative to reference. A positive delay results in less data being acquired before the trigger point, a negative delay results in more data being acquired before the trigger point.

/dev..../scopes/n/trigenable

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

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

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

/dev..../scopes/n/trigfalling

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

When set (1), enables falling edge triggering. This settings is synchronized with the settings done in the /TRIGSLOPE node.

/dev..../scopes/n/trigforce

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

Forces a trigger event.

/dev..../scopes/n/triggate/enable

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

If enabled the trigger will be gated by the trigger gating input signal. This feature requires the DIG option.

/dev..../scopes/n/triggate/inputselect

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

Select the signal source used for trigger gating if gating is enabled. This feature requires the DIG option.

0	"trigin2_high", "trigger_input2_high": Only trigger if the Trigger Input 3 is at high level.
1	"trigin2_low", "trigger_input2_low": Only trigger if the Trigger Input 3 is at low level.
2	"trigin3_high", "trigger_input3_high": Only trigger if the Trigger Input 4 is at high level.
3	"trigin3_low", "trigger_input3_low": Only trigger if the Trigger Input 4 is at low level.

/dev..../scopes/n/trigholdoff

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

Defines the time before the trigger is rearmed after a recording event.

/dev..../scopes/n/trigholdoffcount

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

Defines the trigger event number that will trigger the next recording after a recording event. A value of '1' will start a recording for each trigger event.

/dev..../scopes/n/trigholdoffmode

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

Selects the holdoff mode.

0	"time": Holdoff is defined as time (s).
1	"number_of_events": Holdoff is defined as number of events.

/dev..../scopes/n/trighysteresis/absolute

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

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

/dev..../scopes/n/trighysteresis/mode

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

Selects the mode to define the hysteresis strength. The relative mode will work best over the full input range as long as the analog input signal does not suffer from excessive noise.

0	"absolute": Selects absolute hysteresis.
1	"relative": Selects a hysteresis relative to the adjusted full scale signal input range.

/dev..../scopes/n/trighysteresis/relative

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

Hysteresis relative to the adjusted full scale signal input range. A hysteresis value larger than 1 (100%) is allowed.

/dev..../scopes/n/triglevel

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

Defines the trigger level.

/dev..../scopes/n/trigreference

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

Trigger reference position relative to the acquired data. Default is 50% (0.5) which results in a reference point in the middle of the acquired data.

/dev..../scopes/n/trigrising

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

When set (1), enables rising edge triggering. This settings is synchronized with the settings done in the /TRIGFALLING node.

/dev..../scopes/n/trigslope

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

Sets on which slope of the trigger signal the scope should trigger. This setting is synchronized with the settings done in the /TRIGFALLING and /TRIGRISING nodes.

0	"none": None
1	"rising": Rising edge triggered.
2	"falling": Falling edge triggered.
3	"both": Triggers on both the rising and falling edge.

/dev..../scopes/n/trigstate

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

When 1, indicates that the trigger signal satisfies the conditions of the trigger.

/dev..../scopes/n/wave

Properties: Read, Stream
Type: ZIScopeWave
Unit: None

Contains the scope shot data.

8.2.19. SIGINS

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

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

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

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

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

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

Automatic adjustment of the Range to about two times the maximum signal input amplitude measured over about 100 ms.

/dev..../sigins/n/bw

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

Bandwidth of input aliasing filter at 600 MHz (ON) or 900 MHz (OFF) cut off frequency.

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

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

Switch input mode between normal (OFF), inverted, and differential. The differential modes are implemented digitally and are not suited for analog common-mode rejection. When using the differential modes, the user is responsible for keeping the configuration (range, coupling, termination) of both channels in sync, the device provides no control mechanisms to force that.

0	"off": Off
1	"inverted": Inverted

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

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

Switches between 50 Ohm (ON) and 1 M Ohm (OFF).

0	"1_MOhm": OFF: 1 M Ohm
1	"50_Ohm": ON: 50 Ohm

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

Properties: Read, Write
Type: Double
Unit: None

Indicates the maximum measured value at the input normalized to input range.

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

Properties: Read, Write
Type: Double
Unit: None

Indicates the minimum measured value at the input normalized to input range.

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

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

Enables the signal input.

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

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

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

/dev..../sigins/n/scaling

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

Applies the given scaling factor to the input signal.

8.2.20. SIGOUTS

/dev..../sigouts/n/amplitudes/n

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

Sets the peak amplitude that the oscillator assigned to the given demodulation channel contributes to the signal output.

/dev..../sigouts/n/autorange

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

If enabled, selects the most suited output range automatically.

/dev..../sigouts/n/enables/n

Properties: Read, Write, Setting
Type: Integer (64 bit)
Unit: V

Enables individual output signal amplitude. When the MD option is used, it is possible to generate signals being the linear combination of the available demodulator frequencies.

/dev..../sigouts/n/imp50

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

Select the load impedance between 50 Ohm and HiZ. The impedance of the output is always 50 Ohm. For a load impedance of 50 Ohm the displayed voltage is half the output voltage to reflect the voltage seen at the load.

0	"high_impedance": HiZ
1	"50_Ohm": 50 Ohm

/dev..../sigouts/n/offset

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

Defines the DC voltage that is added to the dynamic part of the output signal.

/dev..../sigouts/n/on

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

Enabling/Disabling the Signal Output. Corresponds to the blue LED indicator on the instrument front panel.

/dev..../sigouts/n/over

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

Indicates that the signal output is overloaded.

/dev..../sigouts/n/range

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

Sets the output voltage range. The instrument selects the next higher available range.

8.2.21. STATS

/dev..../stats/cmdstream/bandwidth

Properties: Read
Type: Double
Unit: Mbit/s

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

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

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

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

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

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

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

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

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

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

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

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

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

/dev..../stats/cmdstream/packetssent

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

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

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

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

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

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

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

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

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

Properties: Read
Type: Double
Unit: Mbit/s

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

/dev..../stats/physical/currents/n

Properties:	Read
Type:	Double
Unit:	mA

Internal current measurements.

/dev..../stats/physical/fanspeed

Properties:	Read
Type:	Double
Unit:	RPM

Speed of the internal cooling fan.

/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.

/dev..../stats/physical/temperatures/n

Properties: Read
Type: Double
Unit: °C

Internal temperature measurements.

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

Properties: Read
Type: Double
Unit: V

Internal voltage measurements.

8.2.22. STATUS

/dev..../status/adc0max

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

The maximum value on Signal Input 1 (ADC0) during 100 ms.

/dev..../status/adc0min

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

The minimum value on Signal Input 1 (ADC0) during 100 ms

/dev..../status/adc1max

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

The maximum value on Signal Input 2 (ADC1) during 100 ms.

/dev..../status/adc1min

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

The minimum value on Signal Input 2 (ADC1) during 100 ms

/dev..../status/fifolevel

Properties: Read
Type: Double
Unit: None

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

/dev..../status/flags/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. Bit 0: Signal Input 1 overflow, Bit 1: Signal Input 2 overflow, Bit 2: Analog PLL fail, Bit 3: Output 1 DAC OK, Bit 4: Output 2 DAC OK, Bit 5: Signal Output 1 clipping, Bit 6: Signal Output 2 clipping, Bit 7: Ext Ref 1 Locked, Bit 8: Ext Ref 2 Locked, Bit 9:Ext Ref 3 Locked, Bit 10:Ext Ref 4 Locked, Bit 11: Sample Loss, Bits 12 - 13: Trigger In 1, Bits 14 - 15: Trigger In 2, Bits 16 - 17: Trigger In 3, Bits 18 - 19: Trigger In 4, Bit 20: PLL 1 locked, Bit 21: PLL 2 locked, Bit 22: PLL 3 locked, Bit 23: PLL 4 locked, Bit 24: Rubidium clock locked, Bit 25: AU Cartesian 1 Overflow, Bit 26: AU Cartesian 2 Overflow, Bit 27: AU Polar 1 Overflow, Bit 28: AU Polar 2 Overflow.

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

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

Flag indicating if tcp packages have been lost.

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

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

Flag indicating if udp packages have been lost.

/dev..../status/time

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

The current timestamp.

8.2.23. SYSTEM

/dev..../system/activeinterface

Properties: Read
Type: String
Unit: None

Currently active interface of the device.

/dev..../system/awg/channelgrouping

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

Sets the channel grouping mode of the device.

0	"groups_of_2": Use the outputs in groups of 2. One sequencer program controls 2 outputs (use /dev..../awgs/0..4/).
1	"groups_of_4": Use the outputs in groups of 4. One sequencer program controls 4 outputs (use /dev..../awgs/0/ and /dev..../awgs/2/)
2	"groups_of_8": Use the outputs in groups of 8. One sequencer program controls 8 outputs (use /dev..../awgs/0/). Requires 8 channel device.

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

Properties: Read
Type: String
Unit: None

Hardware revision of the FPGA base board

/dev..../system/calib/auto

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

Enables an automatic instrument self calibration about 16 min after start up. In order to guarantee the full specification, it is recommended to perform a self calibration after warm-up of the device.

/dev..../system/calib/calibrate

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

Initiates self calibration to improve input digitizer linearity.

/dev..../system/calib/next

Properties: Read
Type: Integer (64 bit)
Unit: s

Remaining seconds until the first calibration is executed or a recalibration is requested.

/dev..../system/calib/required

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

State of device self calibration.

0	Device is warmed-up and self calibrated.
1	It is recommended to manually execute a self calibration to assure operation according to specifications.
2	Device is warming up and will automatically execute a self calibration after 16 minutes.

/dev..../system/calib/tempthreshold

Properties: Read, Write, Setting
Type: Double
Unit: °C

When the temperature changes by the specified amount, it is recommended to rerun the self calibration.

/dev..../system/calib/timeinterval

Properties: Read, Write, Setting
Type: Integer (64 bit)
Unit: s

Time interval for which the self calibration is valid. After this time it is recommended to rerun the auto calibration.

/dev..../system/compdelay/calibrate

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

Perform automatic calibration of the input delay compensation.

/dev..../system/compdelay/delays/n

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

Current values of the compensation delays. 0: Signal Input 0, 1: Signal Input 1, 2: Aux Inputs, 3: Trigger Inputs, 4: Loopbacks

/dev..../system/extclk

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

10 MHz reference clock source.

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

/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/fx2revision

Properties: Read
Type: String
Unit: None

USB firmware revision.

/dev..../system/identify

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

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

/dev..../system/interfacespeed

Properties: Read
Type: String
Unit: None

Speed of the currently active interface (USB only).

/dev..../system/jumbo

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

Enables jumbo frames (4k) on the TCP/IP interface. This will reduce the load on the PC and is required to achieve maximal throughput. Make sure that jumbo frames (4k) are enabled on the network card as well. If one of the devices on the network is not able to work with jumbo frames, the connection will fail.

/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/jumbo

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

Stored setting for jumbo frames. Will be applied after reboot.

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

Properties: Read
Type: String
Unit: None

Current MAC address of the device network interface.

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

Properties: Read
Type: String
Unit: None

Current network mask.

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

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

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

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

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

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

/dev..../system/owner

Properties: Read
Type: String
Unit: None

Returns the current owner of the device (IP).

/dev..../system/porttcp

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

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

/dev..../system/portudp

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

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

/dev..../system/powerconfigdate

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

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

/dev..../system/preampenable

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

Enables the preamplifier.

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

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

Indicates if presets are currently loaded.

/dev..../system/preset/default

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

Indicates the preset which is used as default preset at start-up of the device.

0	"factory": Select factory preset as default preset.
1	Select the preset stored in internal flash memory at position 1 as default preset.
2	Select the preset stored in internal flash memory at position 2 as default preset.
3	Select the preset stored in internal flash memory at position 3 as default preset.
4	Select the preset stored in internal flash memory at position 4 as default preset.
5	Select the preset stored in internal flash memory at position 5 as default preset.
6	Select the preset stored in internal flash memory at position 6 as default preset.

/dev..../system/preset/erase

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

Erase the selected preset.

/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/index

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

Select between factory preset or presets stored in internal flash memory.

0	"factory": Select factory preset.
1	Select the preset stored in internal flash memory at position 1.
2	Select the preset stored in internal flash memory at position 2.
3	Select the preset stored in internal flash memory at position 3.
4	Select the preset stored in internal flash memory at position 4.
5	Select the preset stored in internal flash memory at position 5.
6	Select the preset stored in internal flash memory at position 6.

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

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

Load the selected preset.

/dev..../system/preset/records/n/features

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

Properties of the preset.

/dev..../system/preset/records/n/label

Properties:	Read, Write
Type:	String
Unit:	None

Name of this preset.

/dev..../system/preset/records/n/timestamp

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

Not used.

/dev..../system/preset/records/n/valid

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

True if a valid preset is stored.

/dev..../system/preset/save

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

Save the actual setting as preset.

/dev..../system/properties/freqresolution

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

The number of bits used to represent a frequency.

/dev..../system/properties/freqscaling

Properties: Read
Type: Double
Unit: None

The scale factor to use to convert a frequency represented as a freqresolution-bit integer to a floating point value.

/dev..../system/properties/maxfreq

Properties: Read
Type: Double
Unit: None

The maximum oscillator frequency that can be set.

/dev..../system/properties/maxtimeconstant

Properties: Read
Type: Double
Unit: s

The maximum demodulator time constant that can be set. Only relevant for lock-in amplifiers.

/dev..../system/properties/minfreq

Properties: Read
Type: Double
Unit: None

The minimum oscillator frequency that can be set.

/dev..../system/properties/mintimeconstant

Properties: Read
Type: Double
Unit: s

The minimum demodulator time constant that can be set. Only relevant for lock-in amplifiers.

/dev..../system/properties/negativefreq

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

Indicates whether negative frequencies are supported.

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

Properties: Read
Type: Double
Unit: s

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

/dev..../system/saveports

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

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

/dev..../system/stall

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

Indicates if the network connection is stalled.

/dev..../system/update

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

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

/dev..../system/xenpakenable

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

Indicates whether the 10 gigabit ethernet option is installed and enabled.

8.2.24. TRIGGERS

/dev..../triggers/in/n/autothreshold

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

Automatically adjust the trigger threshold. The level is adjusted to fall in the center of the applied transitions.

/dev..../triggers/in/n/imp50

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

Trigger input impedance: When on, the trigger input impedance is 50 Ohm, when off 1 k Ohm.

/dev..../triggers/in/n/level

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

Trigger voltage level at which the trigger input toggles between low and high. Use 50% amplitude for digital input and consider the trigger hysteresis.

/dev..../triggers/out/n/delay

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

Trigger delay, controls the fine delay of the trigger output. The resolution is 78 ps.

/dev..../triggers/out/n/drive

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

When on, the bidirectional trigger on the front panel is in output mode. When off, the trigger is in input mode.

/dev..../triggers/out/n/pulsewidth

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

Defines the minimal pulse width for the case of Scope and AWG Trigger/Active events written to the trigger outputs of the device.

/dev..../triggers/out/n/source

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

Select the signal assigned to the trigger output.

0	"disabled": The output trigger is disabled.
1	"demod3_phase": Oscillator phase of demod 4 (trigger output channel 1) or demod 8 (trigger output channel 2). Trigger event is output for each zero crossing of the oscillator phase.
2	"scope_trigger": Scope Trigger. Requires the DIG Option.
3	"scope_not_trigger": Scope /Trigger. Requires the DIG Option.
4	"scope_armed": Scope Armed. Requires the DIG Option.
5	"scope_not_armed": Scope /Armed. Requires the DIG Option.
6	"scope_active": Scope Active. Requires the DIG Option.
7	"scope_not_active": Scope /Active. Requires the DIG Option.
8	"awg_marker0": AWG Marker 1. Requires the AWG Option.
9	"awg_marker1": AWG Marker 2. Requires the AWG Option.
10	"awg_marker2": AWG Marker 3. Requires the AWG Option.
11	"awg_marker3": AWG Marker 4. Requires the AWG Option.
20	"awg_active": AWG Active. Requires the AWG Option.
21	"awg_waiting": AWG Waiting. Requires the AWG Option.
22	"awg_fetching": AWG Fetching. Requires the AWG Option.
23	"awg_playing": AWG Playing. Requires the AWG Option.
32	"awg_trigger0": AWG Trigger 1. Requires the AWG Option.
33	"awg_trigger1": AWG Trigger 2. Requires the AWG Option.
34	"awg_trigger2": AWG Trigger 3. Requires the AWG Option.
35	"awg_trigger3": AWG Trigger 4. Requires the AWG Option.
51	"mds_clock_out": MDS Clock Out.
52	"mds_sync_out": MDS Sync Out.