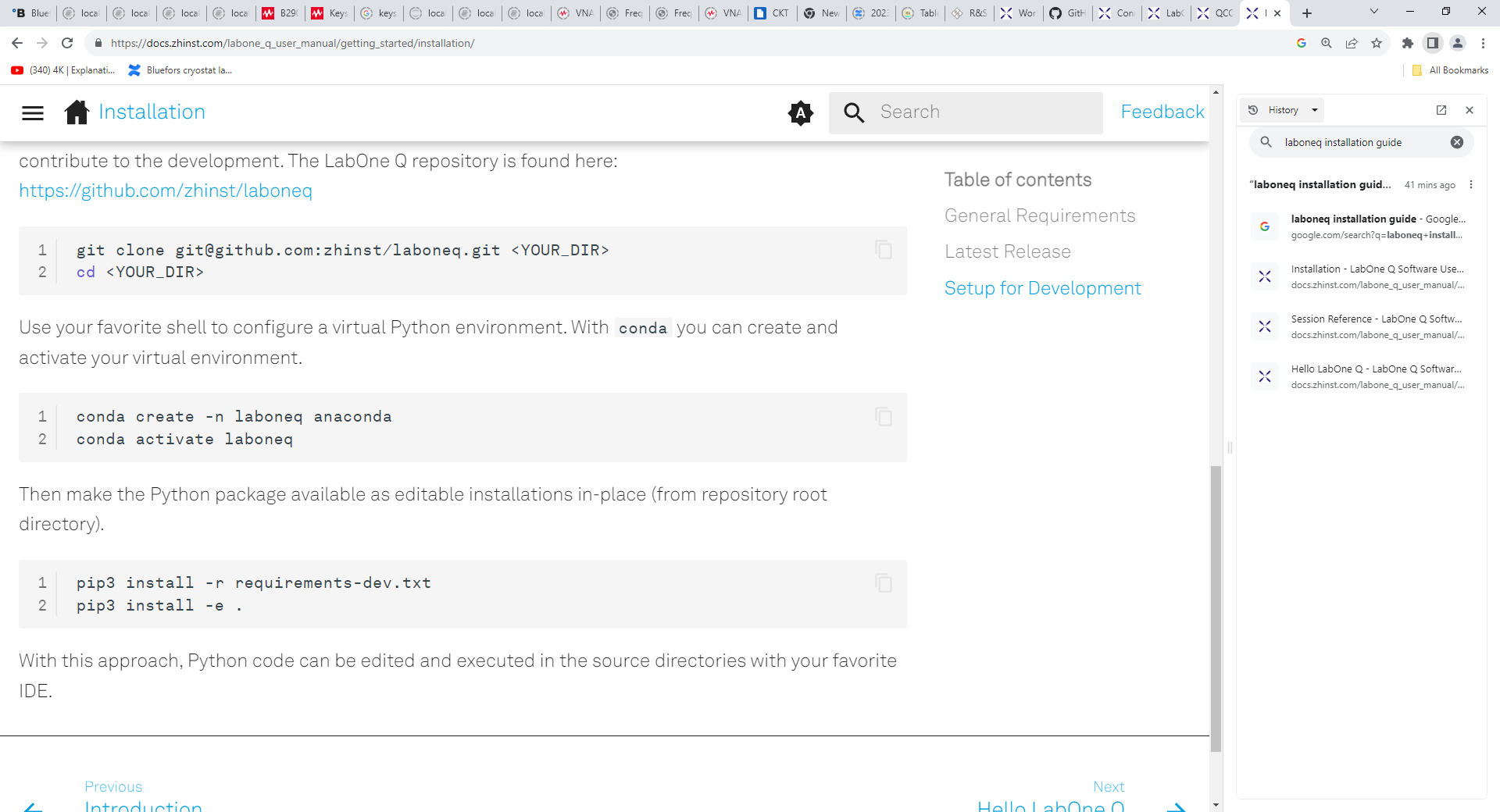
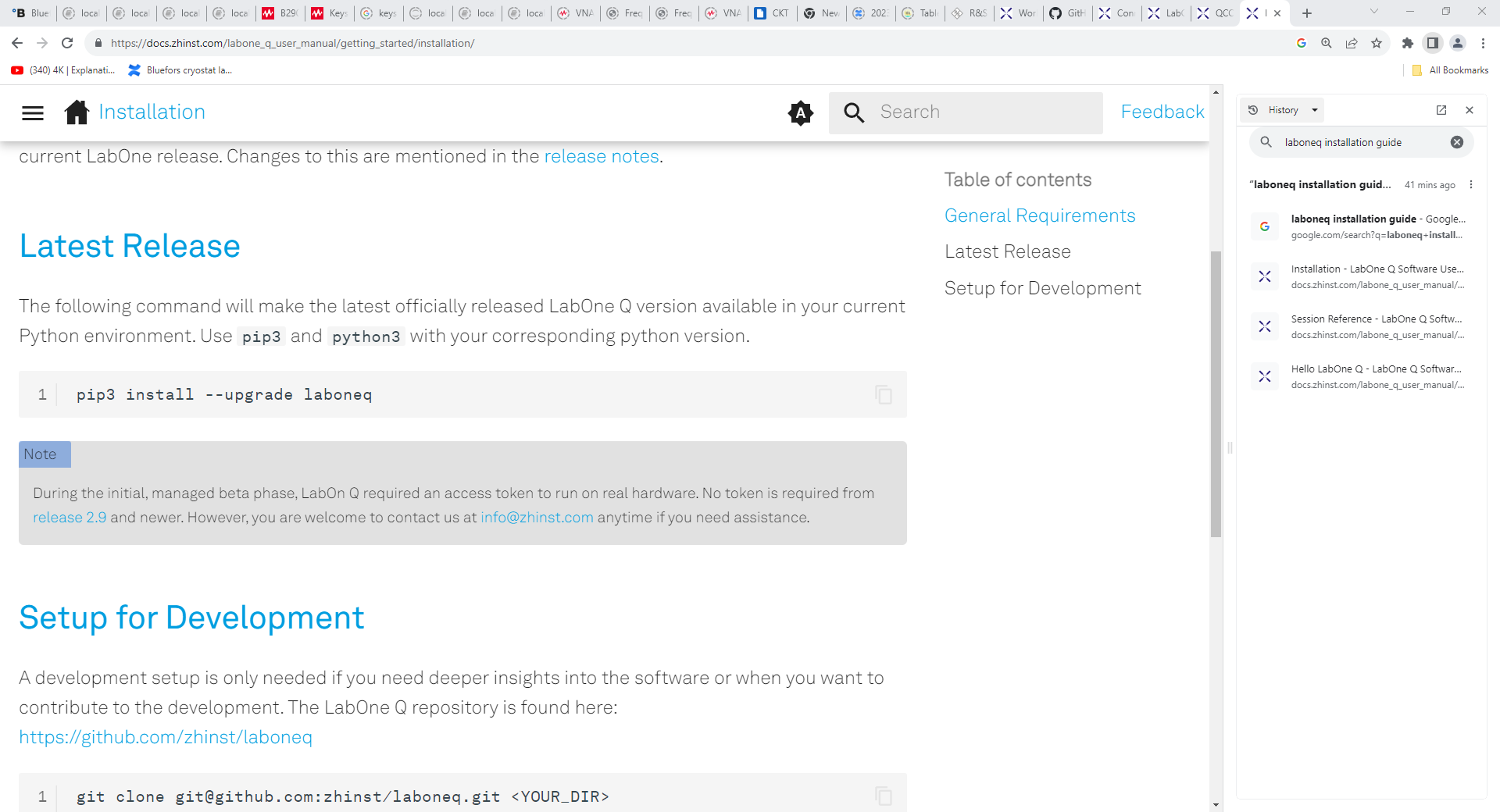
Make sure to install **Git** and **Visual Studio Code**

Setup (Only Local):

Open Anaconda prompt, then in the prompt:



Afterwards:



Finally: from the Anaconda prompt (with the env selected and all packages installed) run “code” to open VSC

**Env name: laboneq\_dev**

“Global” Installation:

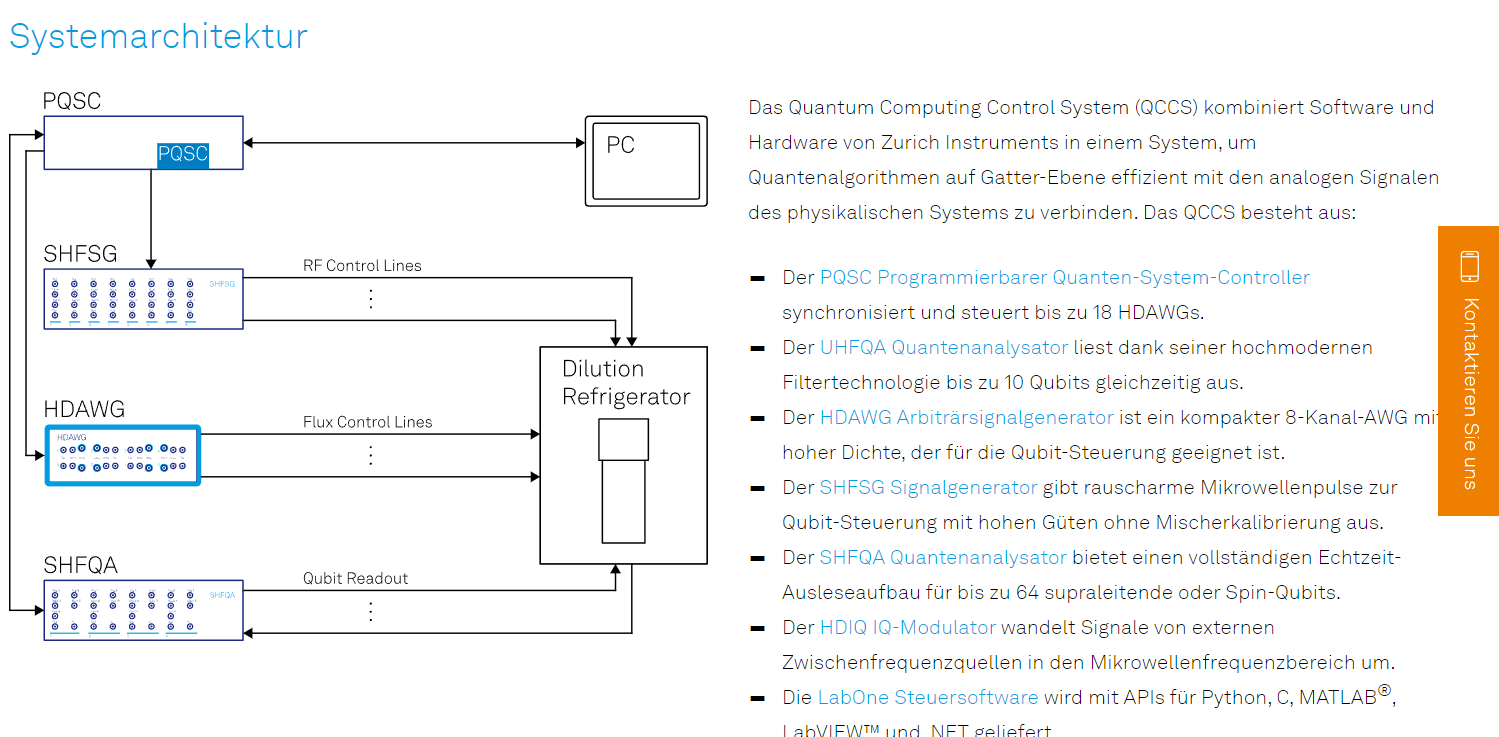
Fork repository into your Github Account and then run all the steps above

Then in VSC create a branch for your changes and commit and push (to forked repository) after changes

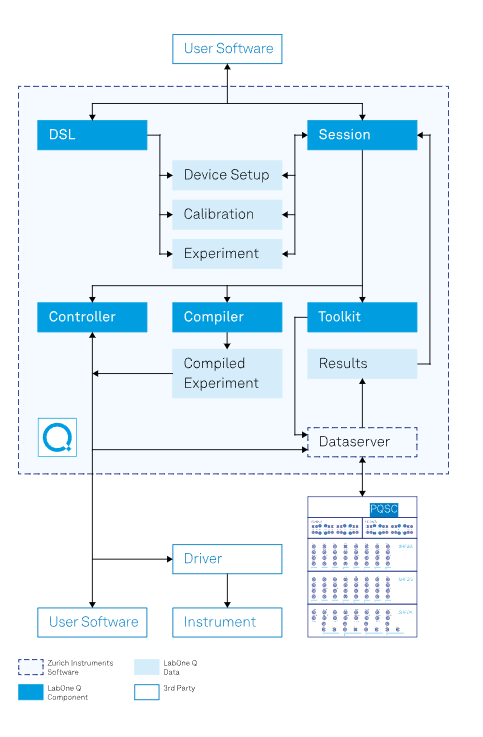
Master can then decide which changes to pull

You should then be able to create your own notebooks and use all the modules from laboneq and to also work with all the examples provided

**System Architecture**

****

**Software Architecture**

****

Dataserver is provided by LabOne Software

The QCCS Monitor can be found under:

T:\MQV Cryo-Team\0 Measurements\1 Measurement scripts\QCCS Monitor\qccs-monitor-windows-x86\_64-23.06.46756

**Starting the server:**

In the terminal and at the according location run:

.\bin\qccs-monitor-server.exe --resdir=.\res -p 9000 .\config\my\_setup.json

With these settings, the webserver can be accessed at http://127.0.0.1:9000, if the monitor server and the browser are running on the same machine. Otherwise, replace 127.0.0.1 with the domain name or IP address of the host.

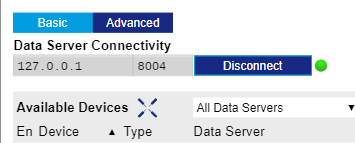
Overview of commands: .\bin\qccs-monitor-server.exe –help

Terminate the monitor server by pressing: CTRL+C

**Configuring the server:**

If you want to change the **configuration** of the monitor edit the my\_setup.json file in the config folder

Connect to the server in LabOne and remember the **Host address**

****

Have the same address in your setup JSON File

Then run this code while **replacing 8004 with your port address if it differs and the device IDs:**

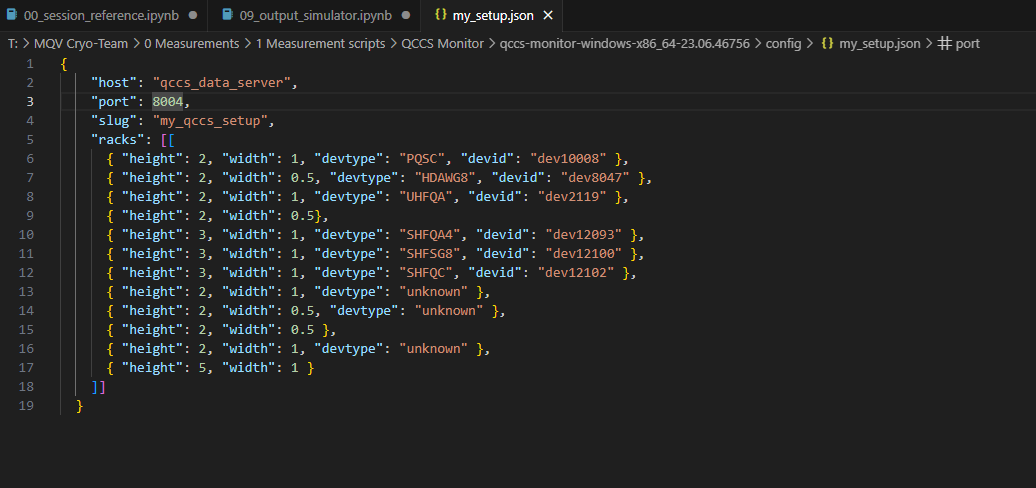
**.\bin\qccs-monitor-server.exe --resdir=.\res -p 9000 .\config\my\_setup.json**

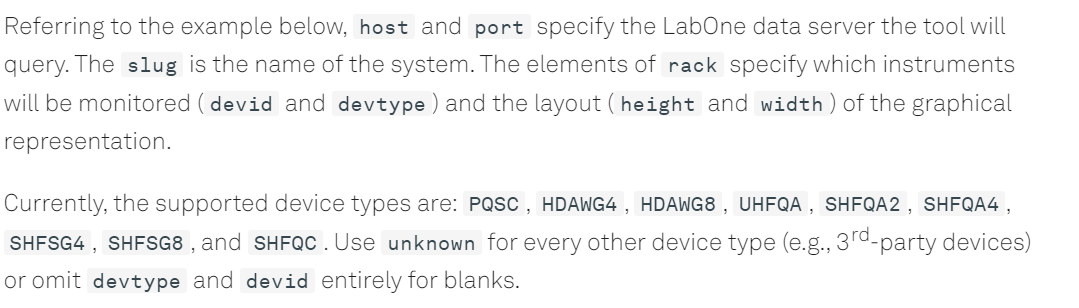
**Since the host addr is not 127.0.0.1 we have to change the command for the LAB PC to:**

**(You can find all IP addr in cmd with ipconfig /all)**

**Command: .\bin\qccs-monitor-server.exe --resdir=.\res -p 8008 .\config\my\_setup.json --addr 127.0.0.1**

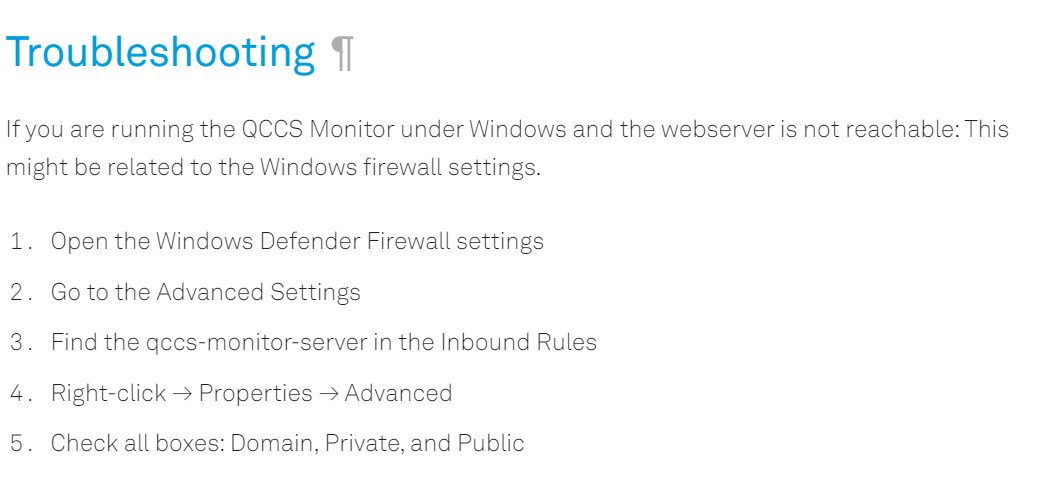
**Then go to this web address: http://127.0.0.1:8008**





You can terminate the monitor server by pressing CTRL+C.

Note:



QA and SG are being combined

SG has IQ Modulation inside to get the appropriate signal (instead of just combining rec with sinus)

Try to get the QA to do a signal sweep for the resonators like the VNA does

**Synchronizing PQSC with other devices**

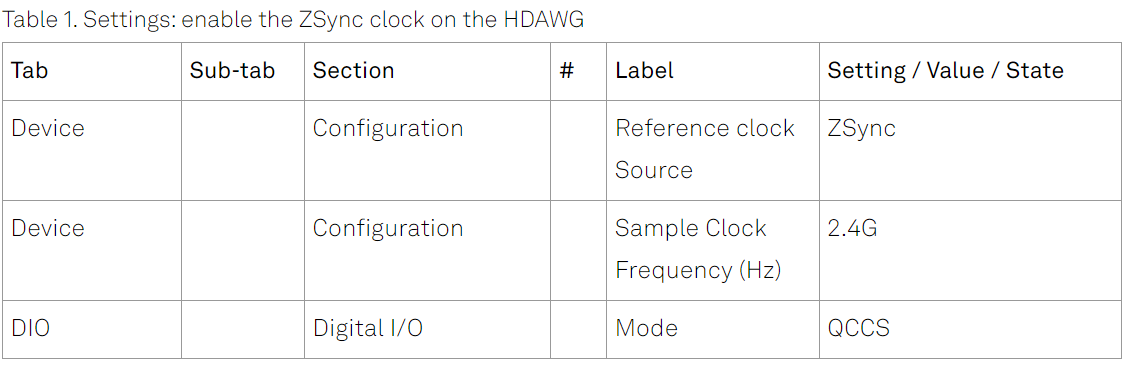
30 min warm up until status LED in Lab One turns Green:



Bottom right corner in PQSC

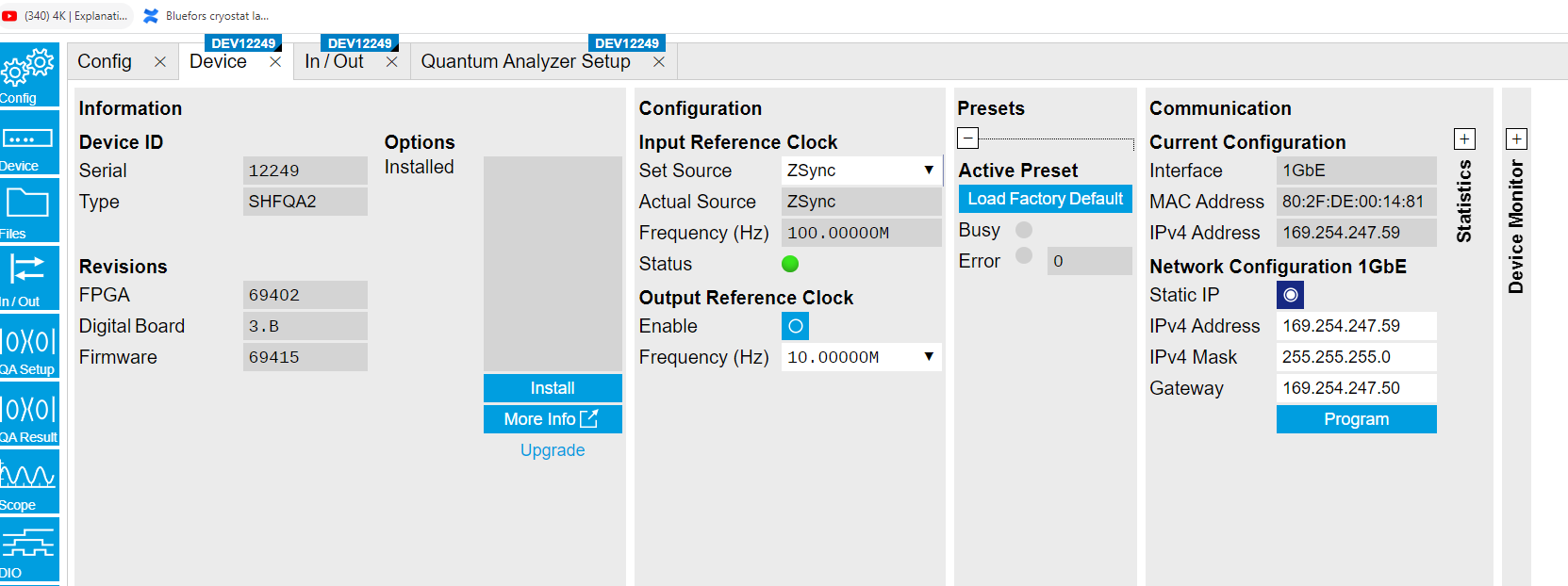
**Connect ZSync Cables**

Then following setting for the devices to be connected:

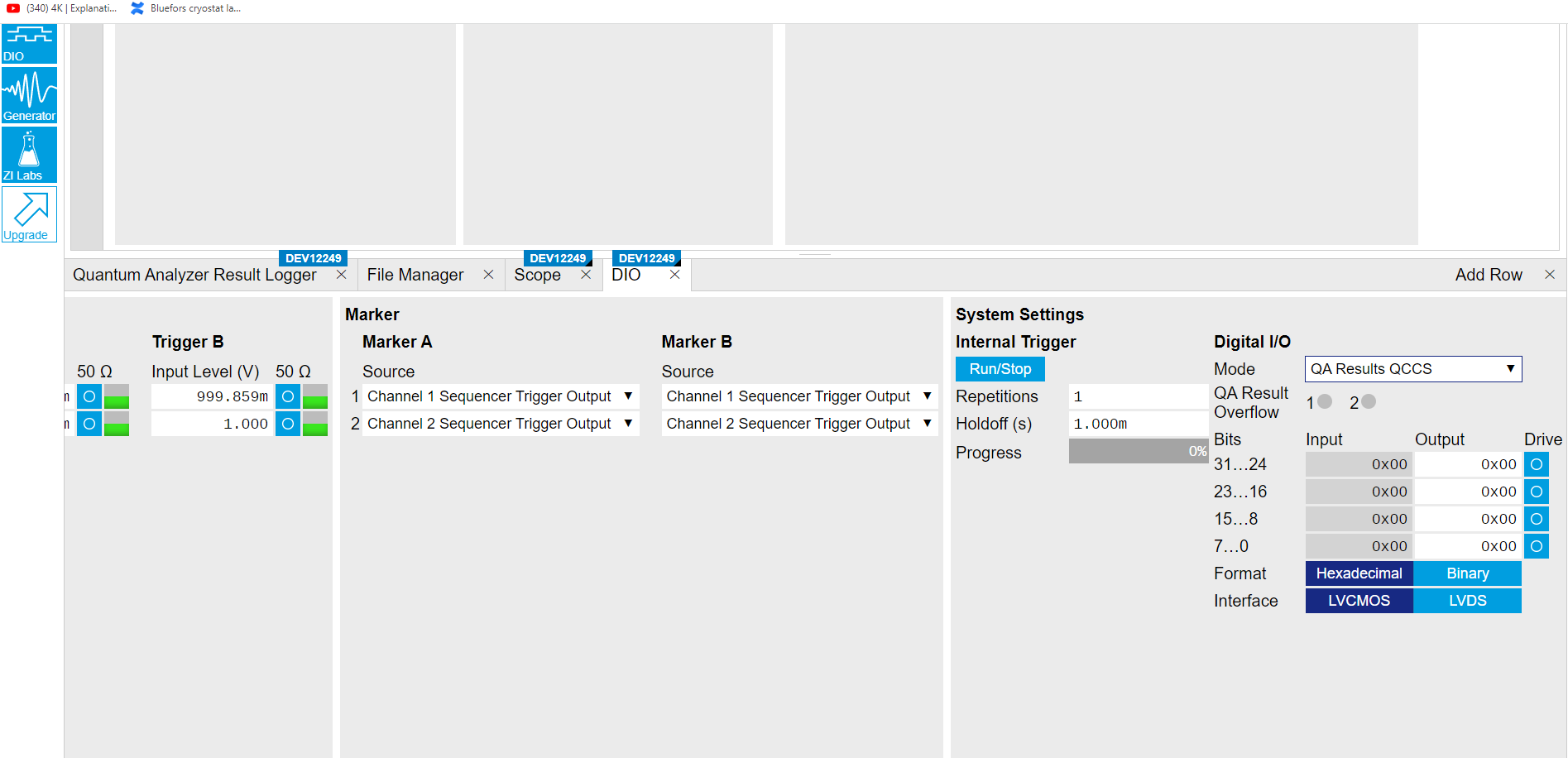


**For SHFQA 12249:**

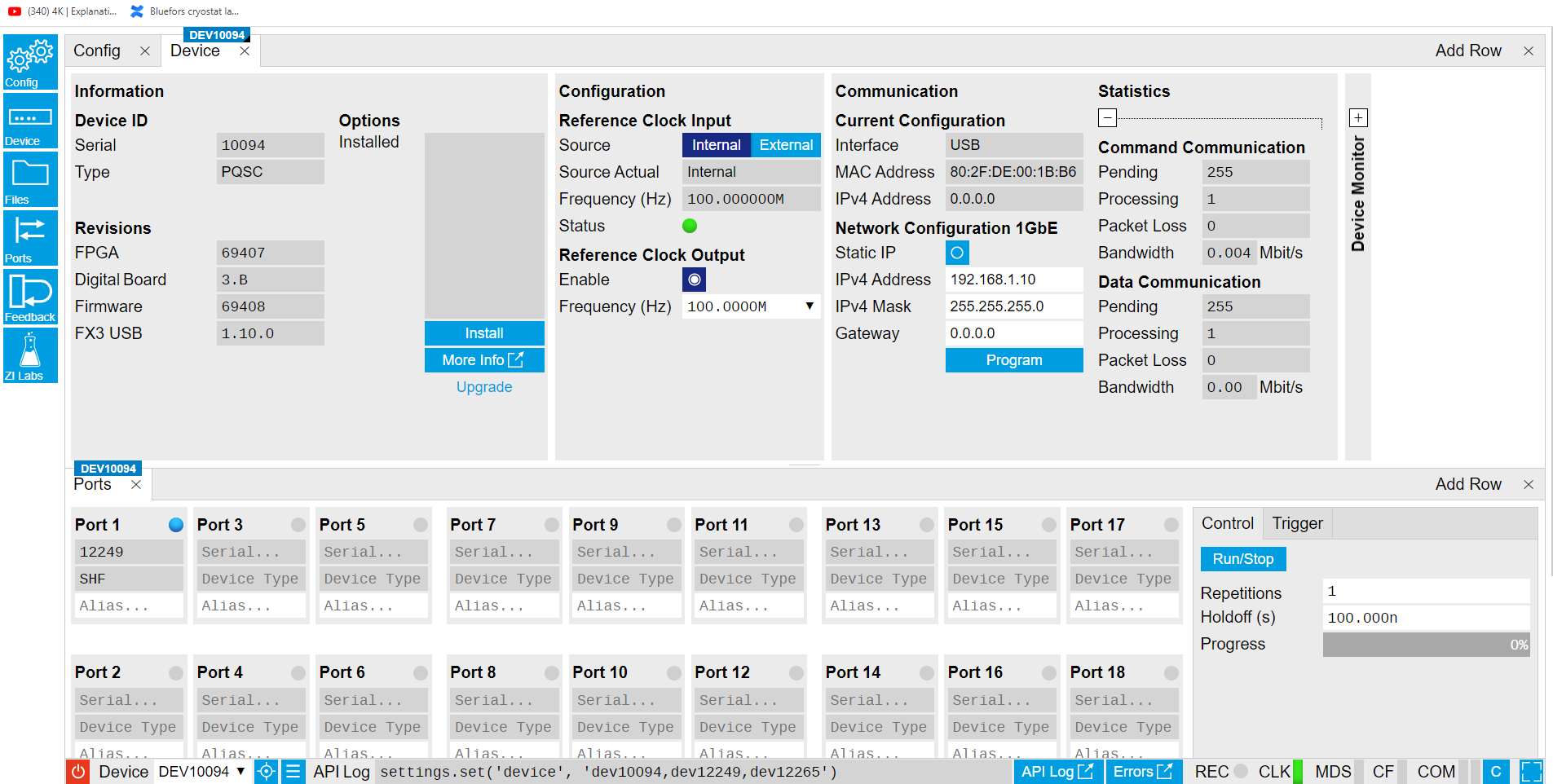
Input Reference Clock to ZSync



Change the digital I/O to QCCS

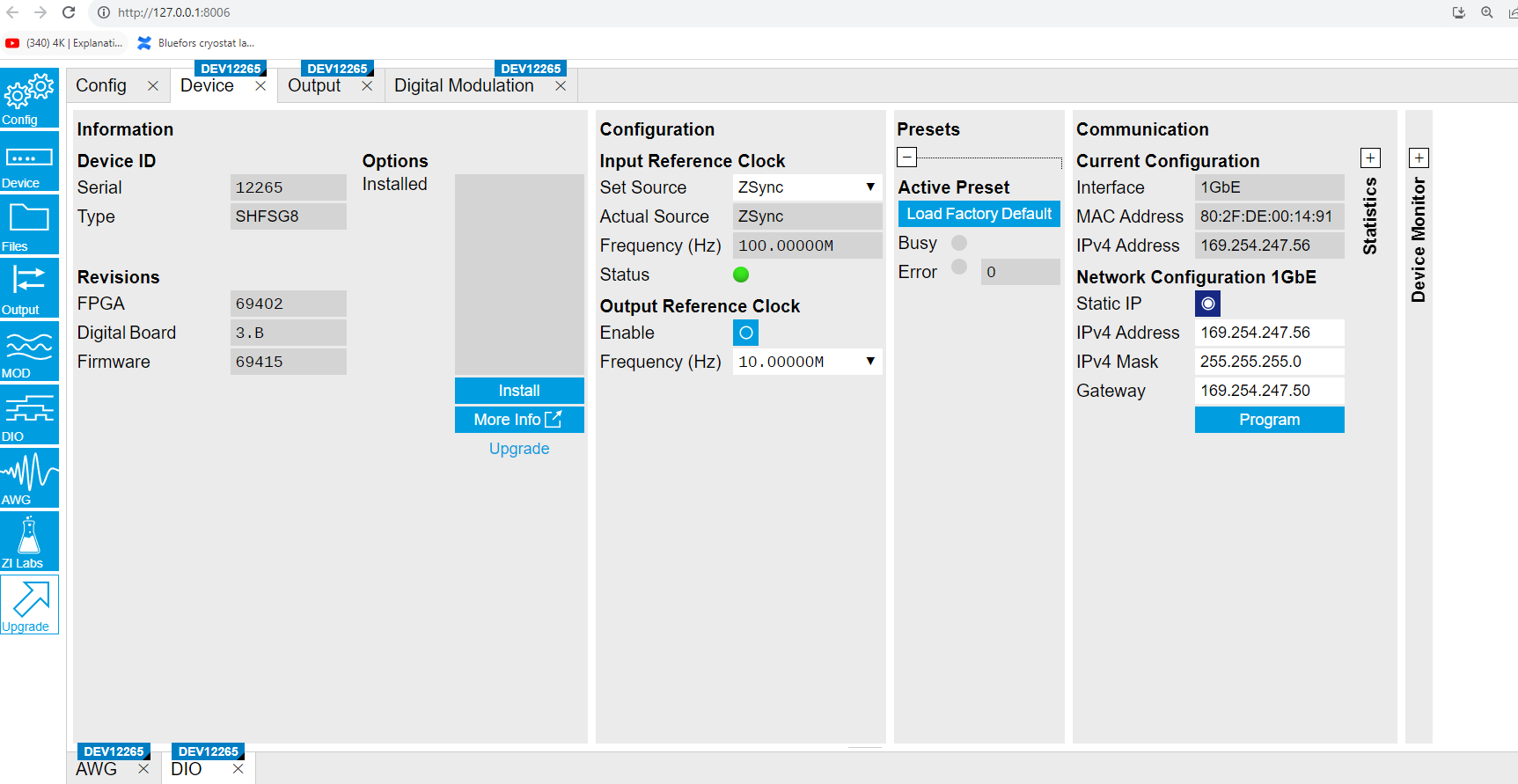


**PQSC10094:**



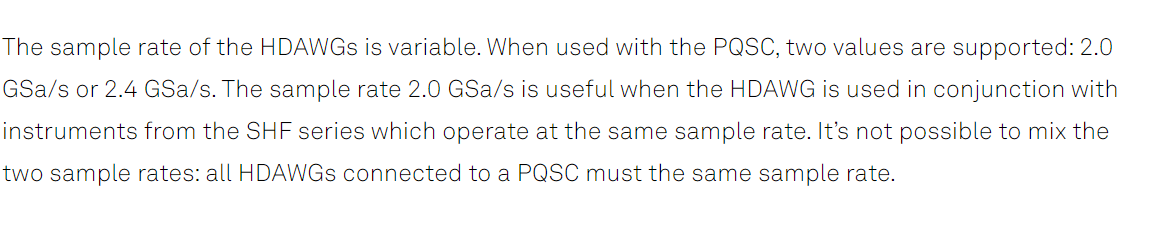
Then we can see the device being connected to port on**e**

**SHFSG12265**:



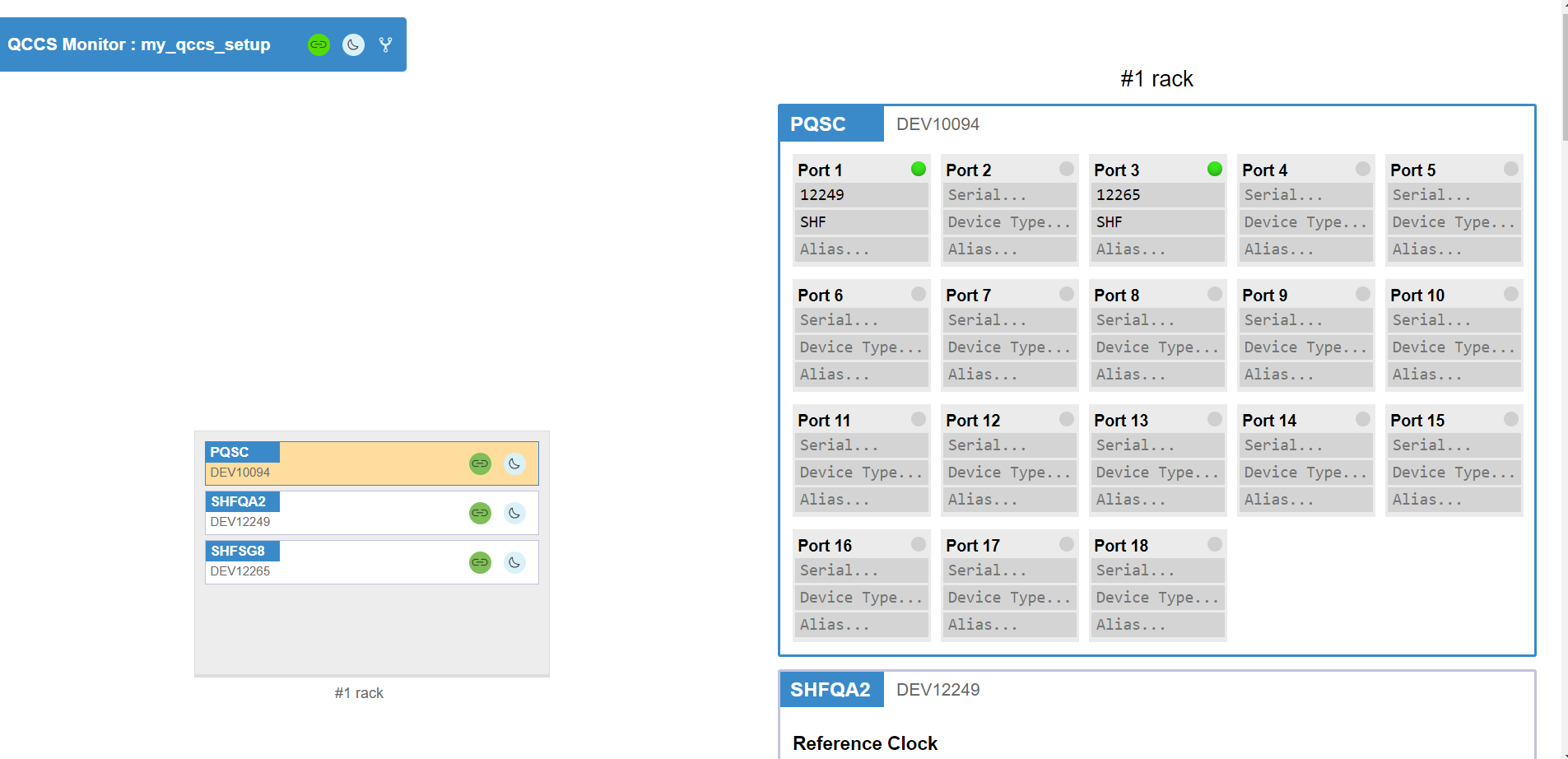
Set to ZSync

**HDAWG:**

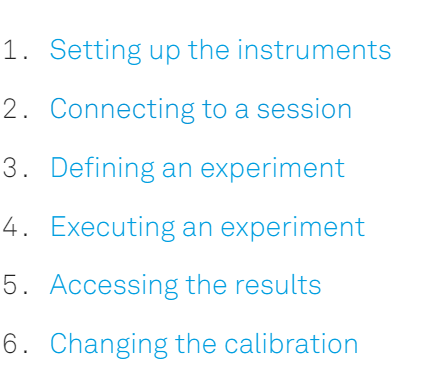


Sample rate for HDAWG has to be set to 2.0GSa/s

**QCCS:**

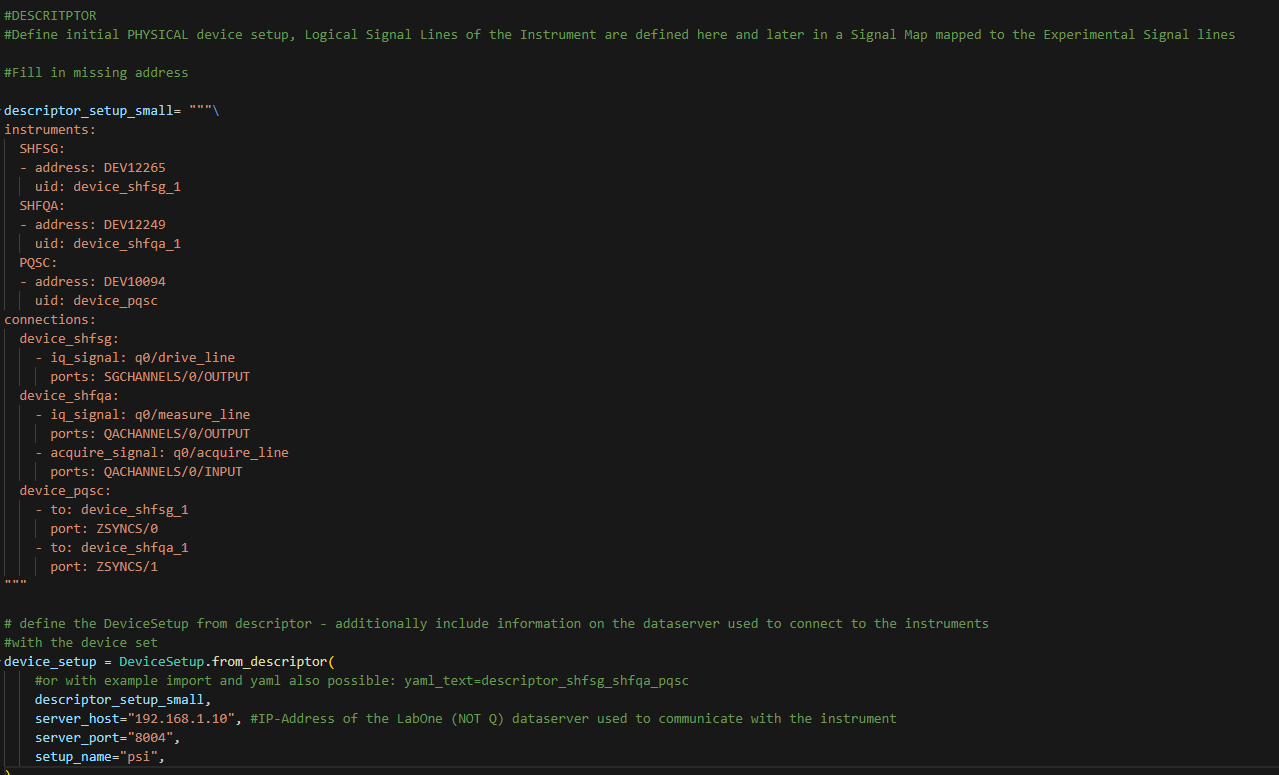
****

**Code Workflow**



1. --- Descriptor, Logical Signal Lines, Calibrating Device and Signal Lines (Baseline and Exp Calibration)

Descriptor for single qubit characterization (HDAWG missing):

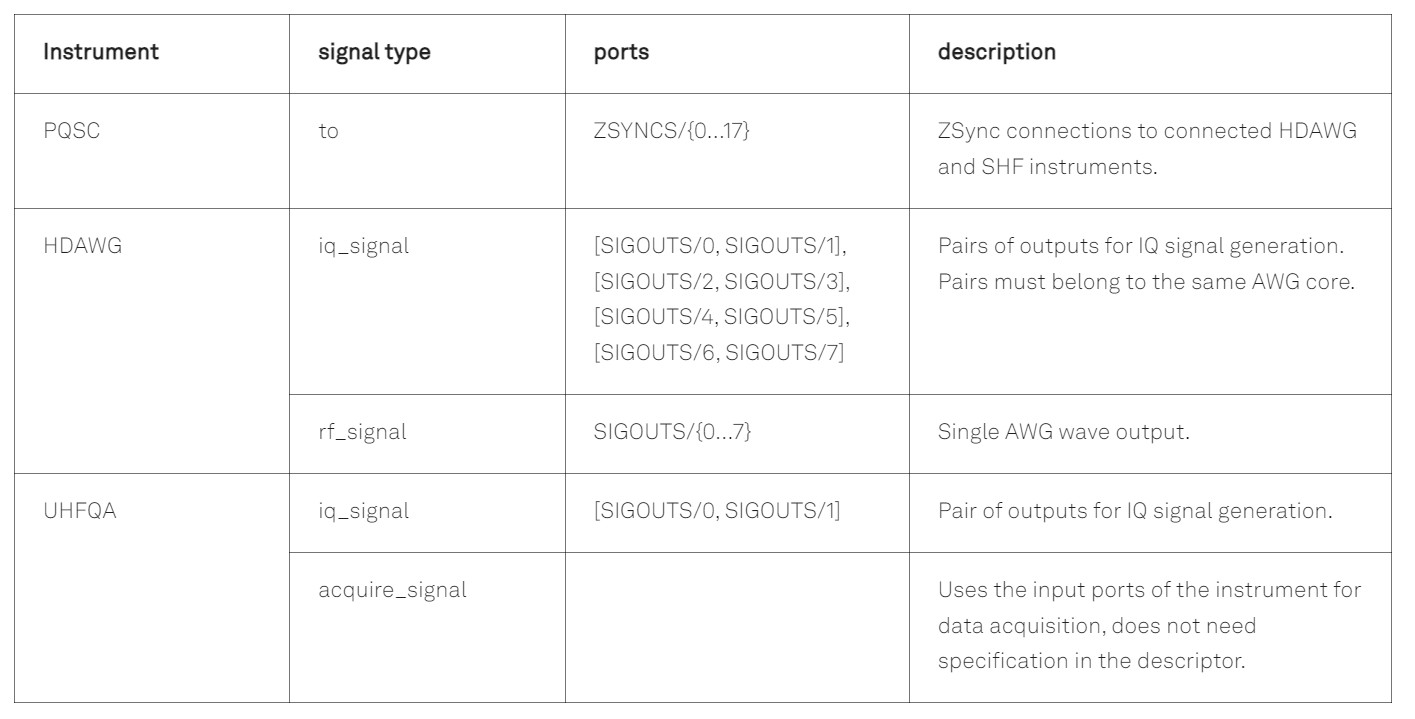


Logical Signal Lines

In a qubit experiment, you need to both control and measure your qubits. To control your qubits, you need a combination of drive lines and flux lines, and to measure, you need to send a pulse down a measurement line and evaluate the response in an acquire line. For our example experiment, we have four logical signal types: drive, flux, measure, and acquire, which form a logical signal group for each qubit.

The signal type names, as well as the group names, are entirely arbitrary and determined by the user. However, the associated type of signal on the device is not arbitrary and will be either acquire\_signal, rf\_signal, or iq\_signal, see details [here](https://docs.zhinst.com/labone_q_user_manual/concepts/set_up_equipment/#tab_setup_descriptor_signal_types).

Within a logical signal group, we can have more than one line of a single type, two drive lines, for example. Each group will have its own sets of signal lines, and they are programmatically distinct from one another. Logical signal lines retain this distinction even if the physical lines used are the same. For example, if we have our first qubit, q0, and our second qubit, q1, on the same physical multiplexed acquisition line, they will still have separate logical measure and acquire lines associated with them.



LSL are defined in the Descriptor e.g.:

my\_descriptor = """\

instrument\_list:

HDAWG:

- address: DEV8xyz

uid: device\_hdawg

SHFQA:

- address: DEV12yxz

uid: device\_shfqa

SHFSG:

- address: DEV12xyz

uid: device\_shfsg

PQSC:

- address: DEV10xyz

uid: device\_pqsc

connections:

device\_hdawg:

- iq\_signal: q0/drive\_line #type,LSG,LSL name

ports: [SIGOUTS/0, SIGOUTS/1]

#### to specify both the I and Q components of the signal, we need two physical outputs of the HDAWG, given by the first (SIGOUTS/0) and second (SIGOUTS/1) wave outputs on the HDAWG.

- rf\_signal: q0/flux\_line

ports: [SIGOUTS/4]

- rf\_signal: q1/flux\_line

ports: [SIGOUTS/6]

device\_shfsg:

- iq\_signal: q1/drive\_line

ports: SGCHANNELS/0/OUTPUT

device\_shfqa:

- iq\_signal: q0/measure\_line

ports: QACHANNELS/0/OUTPUT

- acquire\_signal: q0/acquire\_line

ports: QACHANNELS/0/INPUT

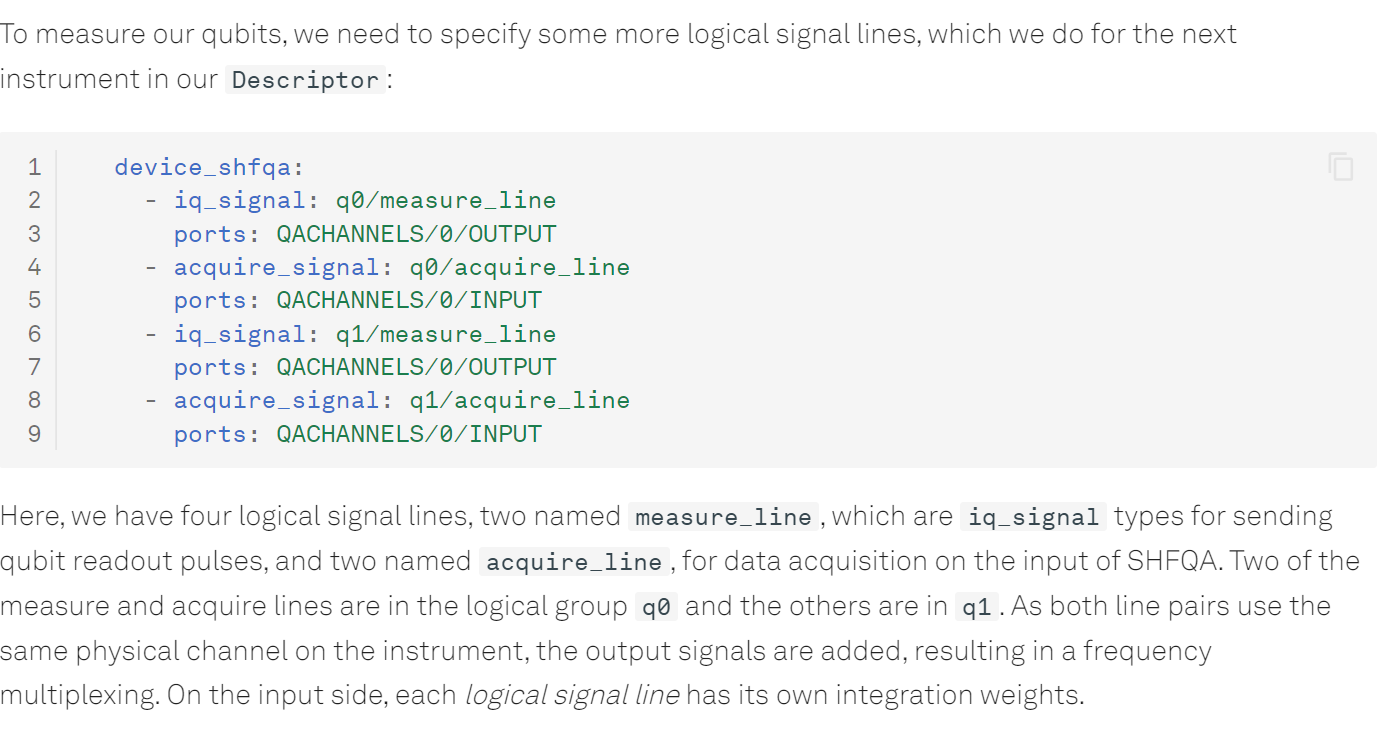
- iq\_signal: q1/measure\_line

ports: QACHANNELS/0/OUTPUT

- acquire\_signal: q1/acquire\_line

ports: QACHANNELS/0/INPUT

#### Multiplexing Signals (using the same channel for two qubit readout pulses (measure line) and to acquire their data (aquire line)



device\_pqsc:

- to: device\_hdawg

port: ZSYNCS/0

- to: device\_shfsg

port: ZSYNCS/1

- to: device\_shfqa

port: ZSYNCS/2

"""

1. Defining an Experiment
2. exp\_qubit\_spec = Experiment(
3. uid="Qubit Spectroscopy",
4. signals=[
5. ExperimentSignal("drive"),
6. ExperimentSignal("measure"),
7. ExperimentSignal("acquire"),
8. ]
9. )

They are independent of the physical/logical signal lines for reusability

Later we map the signals:

## signal map for qubit 0

q0\_map = {"drive": "/logical\_signal\_groups/q0/drive\_line",

"measure": "/logical\_signal\_groups/q0/measure\_line",

"acquire": "/logical\_signal\_groups/q0/acquire\_line",

}

## signal map for qubit 1

q1\_map = {"drive": "/logical\_signal\_groups/q1/drive\_line\_ge",

"measure": "/logical\_signal\_groups/q1/measure\_line",

"acquire": "/logical\_signal\_groups/q1/acquire\_line",

}

## apply qubit 0 signal map and run the experiment

exp\_qubit\_spec.set\_signal\_map(q0\_map)

my\_results\_q0 = my\_session.run(exp\_qubit\_spec)

## apply qubit 1 signal map and run the experiment

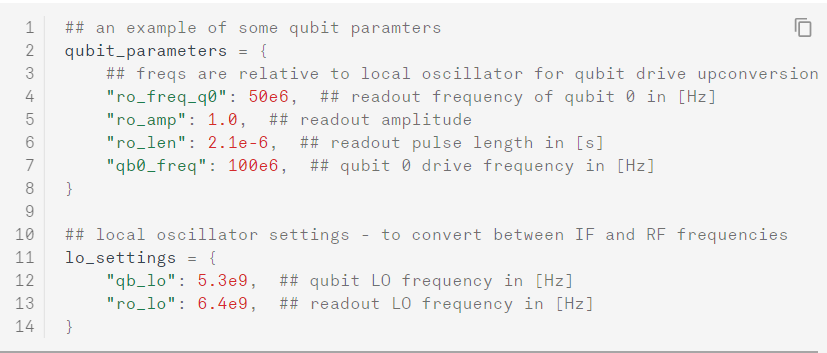
exp\_qubit\_spec.set\_signal\_map(q1\_map)

my\_results\_q1 = my\_session.run(exp\_qubit\_spec)

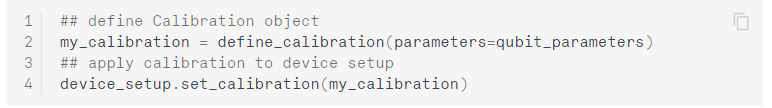
Calibrating devices and signal lines

In a typical workflow, you might begin with a baseline [calibration for your instruments](https://docs.zhinst.com/labone_q_user_manual/concepts/instrument_calibration/), along with some guesses of your qubit parameters. After acquiring data, you may want to update your calibration and use an [experiment calibration](https://docs.zhinst.com/labone_q_user_manual/concepts/experiment_calibration/) kept separate from your baseline one.

Baseline Calibration:







Add this: [Experiment Calibration - LabOne Q Software User Manual (zhinst.com)](https://docs.zhinst.com/labone_q_user_manual/concepts/experiment_calibration/)

1. Connecting to a session (See beginning of Developer Diary)
2. Defining an Experiment (Single Qubit)

[Single Qubit Calibration - LabOne Q Software User Manual (zhinst.com)](https://docs.zhinst.com/labone_q_user_manual/tutorials/single_qubit/)

We will start the qubit calibration with a resonator spectroscopy experiment, determining the resonance frequency of the readout resonator coupled to each of the qubits. Next will be a first measurement of the qubit resonance frequency through qubit spectroscopy, which will then allow us to calibrate qubit excitation pulses in an amplitude Rabi experiment. The calibrated pulse amplitudes will then be used to run a first characterization of qubit parameters, measuring the qubit lifetime in a T1 experiment and its dephasing time in a Ramsey experiment.

[laboneq/examples/03\_superconducting\_qubits/00\_qubit\_tuneup\_shfsg\_shfqa\_shfqc.ipynb at main · zhinst/laboneq (github.com)](https://github.com/zhinst/laboneq/blob/main/examples/03_superconducting_qubits/00_qubit_tuneup_shfsg_shfqa_shfqc.ipynb)

…. This will now lead to our code

**One Qubit Time-domain Characterization @EMFTgit**