

# Quantify

## Software Platform | Open-source Experiment Control

Release November 2021\_V1.4

### Description

Qblox provides a completely open-source software stack, called Quantify, to control experiments on Qblox Cluster and SPI hardware.

Quantify is a Python-based, data acquisition platform focused on providing all the necessary tools for Quantum Computing experiments. It is built on top of QCoDeS, and is the successor of the extensively used PyQED measurement environment.

The simple software framework enables setting-up typical characterisation experiments and advanced experimental procedures with ease-of-use.

The open-source BSD 4 licensed platform is developed by Qblox and Orange Quantum System. All the functionality of the software is well-documented in Read-the-Docs, basic tutorials are given for quick set-up and start of the measurement, whereas advanced sequencing tutorials are also featured for more complex experiments. Quantify is provided together with APIs that can be interfaced at the level of SCPI, Python or QCoDeS. All of the levels are open source, well-documented and tested in a CI pipeline.

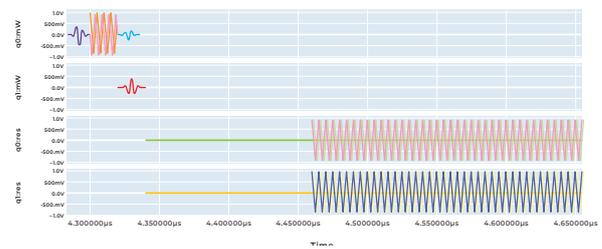
Quantify is compatible with various quantum platforms such as transmons, spin-qubits or NV-centers and can also be used with any equipment from other vendors that supports QCoDeS drivers.

Quantify currently consists of two modules named, Quantify-Core and Quantify-Scheduler.

A typical layered workflow starts at the top layer of the software stack with a high level visualization, where the experiment is expressed with QASM-like gate level instructions, arbitrary pulses or with their combination.

Then, on the second level, instructions become device aware by being converted into pulse shapes, frequencies or timing information corresponding to the quantum processor type (transmon, dot, NV center etc.).

On the third layer, QCoDeS drivers come into play to add electronics-specificity and compile them to the back-end (in this example Qlox Instruments). Finally, the experiment instructions are visualized on the host PC for debugging and further played at the output channels of the modules.

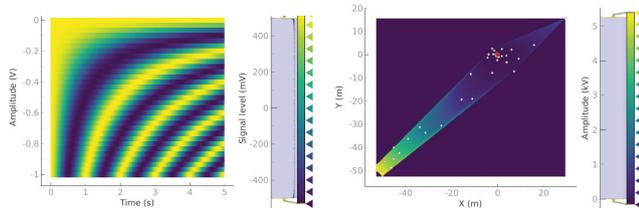


The top two rows show the microwave pulses that will be applied on each qubit, whereas the bottom rows show the readout pulse on each qubit.

### Mastering experiments with the Quantify-Core

The Quantify-Core package is responsible for high-level tasks like:

- Multi-dimensional (unlimited nr of dimensions) software-controlled looping and sweeping of any QCoDeS instrument parameter
- Adaptive Measurements: Adaptive sampling and classical minimization algorithms
- Data saving, using X-array
- Parameter monitoring and live visualization of experiments.
- Data analysis and fitting.



### Calibration Routines

Quantify allows implementing automated calibration routines for typical quantum computing experiments such as qubit and resonator spectroscopy, Rabi and Ramsey oscillation experiments.

Together with the advanced sequence processors, auto-calibration routines bring the coherent control of individual qubits to an advanced level where significant amount of experimental time can be saved.

### Updates & Maintenance

Quantify is professionally maintained and constantly being improved by our software team to provide a growing coverage of the quantum computing experiments.

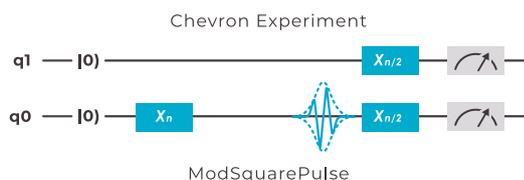
The open-source licensing also allows the researchers and experimentalists to create and possibly share their specific extensions to the main platform, further advancing the capabilities of Quantify.

By subscribing to the Quantify Gitlab repository there is automatic awareness of the latest releases.

Quantify can run on any host PC which runs Python and has been extensively tested with Linux Ubuntu, Mac OS and Windows.

### Hybrid scheduling from the Quantify-Scheduler

Quantify-Scheduler is a module for defining time-domain experiments with deterministic timing at the nano-second level. To make it convenient for variational algorithms (like calibrations) it features a unique hybrid control model allowing quantum gate- and pulse-level descriptions to be combined in a clearly defined and hardware-agnostic way. An example is the Chevron experiment where QASM operations (initialization, single-qubit gates, measurements) are mixed with a classically-defined flux pulse.



Deterministically-timed sequences can be created to make full use of the advanced sequencing capabilities of the Cluster qubit control and readout modules. Quantify-Scheduler is designed to allow experimentalists to easily define complex experiments, and produces synchronized pulse schedules to be distributed to control hardware.

### Links:

Quantify-Core: <https://quantify-quantify-core.readthedocs-hosted.com/en/stable/>

Quantify-Scheduler: <https://quantify-quantify-scheduler.readthedocs-hosted.com/en/stable/>

Qblox Instruments: <https://qblox-qblox-instruments.readthedocs-hosted.com/en/master/>

[www.quantify-opensource.org](http://www.quantify-opensource.org)

