TECHNICAL OVERVIEW

M5401LxxA

Labber

Overview

The M5401LxxA Labber software options (M5401L01A, M5401L05A, and M5401LENA) all offer a complete, easy-to-use solution for instrument control and lab automation, with particular focus on quantum applications. The software is suitable for an R&D, test, or measurement team that wishes to bring its experimental potential to its maximum, while minimizing the time spent on developing lab software infrastructure.

Labber consists of three core components:

- **An Instrument Server** for instrument communication, supporting many vendors and interfaces, e.g., TCP/IP, GPIB, serial, USB etc.
- **A Measurement Editor** for efficient measurement configuration and execution.
- **A Log Browser** for organized storage of experimental data and configurations.

The M5401LxxA Software Package

This licensed software package for Labber includes a GUI for each component, as well as a Python API that together enable a versatile range of quantum measurements. All three products offer the full set of features listed below, differing only in license type and quantity (see Table 1).

KEYSIGHT TECHNOLOGIES

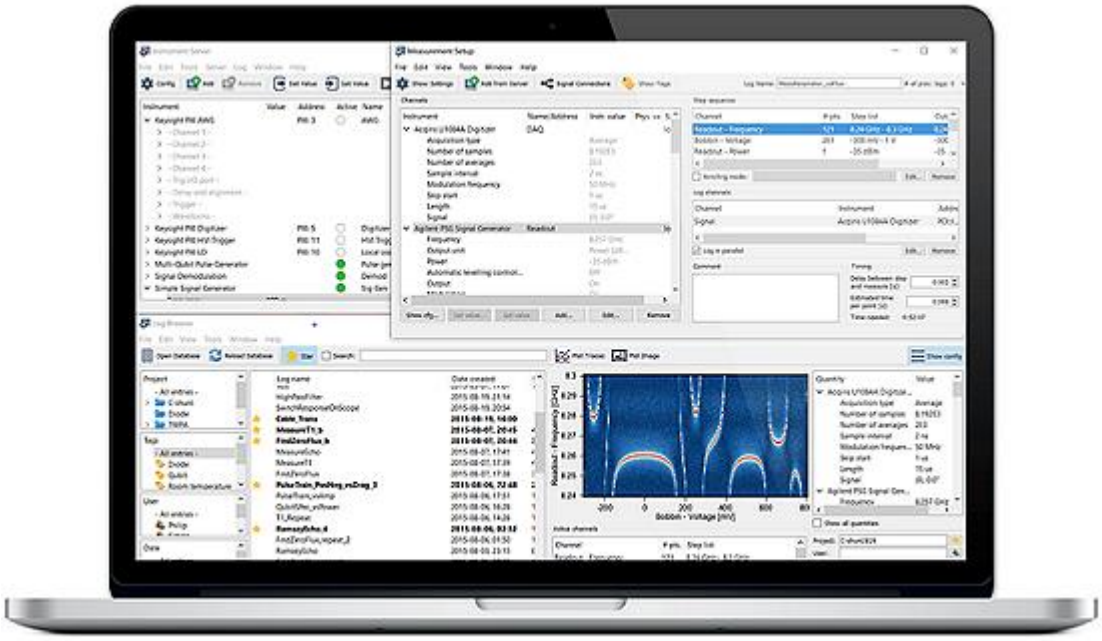


Figure 1. M5401LxxA Software Solution. (Source: Keysight.com/find/labber)

Labber offers a complete, easy-to-use solution for instrument control and lab automation. Researchers and engineers require a flexible and fast software solution enabling the latest breakthrough. They should not have to divert time away from research, writing many lines of code. Instead, they can leverage Labber to accelerate their next breakthrough.

System Requirements

- Refer to the Labber Quantum System Requirements & Installation Guide.

What is included?

Table 1. Licenses for M5401L01A, M5401L05A, and M5401LENA

Product number	Product description	Notes	Durations available	License types available
M5401L01A	Individual Labber Quantum software license for lab control and automation	Single Labber software license	6-month 12-month 24-month 36-month	Node-locked Transportable USB Portable Floating
M5401L05A	Group Labber Quantum software license for lab control and automation	Single Labber software license, sold only in sets of five	6-month 12-month 24-month 36-month	Node-locked Transportable USB Portable Floating
M5401LENA	Enterprise Labber Quantum software license for lab control and automation	Single floating Labber license supporting up to 80 concurrent users	36-month	Floating

Highlights

- Easy to use Python interface via a GUI or native API for measurement scripting and data analysis
  - Log Browser that automatically loads and plots data for inspection
- Over 100 existing Keysight and 3rd party hardware (AWG/Digitizer/Signal Generators/Spectrum Analyzers/VNAs) and software drivers
  - There are examples in the Labber User Manual on how to create hardware or software drivers.
- Drivers provide communication via VISA supporting TCP/IP, GPIB, serial, and USB Connections.
- Labber is trusted by universities, research laboratories, and companies worldwide to run their experiments.

Features

- The HDF5 file format saves experiment and instrumentation configuration with run data, enabling drag and drop replication in GUI (Measurement Editor or Log Browser) interchangeable with creation or file load via python API.
- As a convenience feature, users can click and drag an HDF5 file from a folder or an email and drop it onto the Measurement Editor GUI. This enables them to reload the experiment with an instrument configuration to inspect or replicate configurations from collaborators. This feature is also useful to inspect or debug the output of the Python API measurement scripts in the Measurement Editor GUI. Users can similarly drag and drop the HDF5 files to the Log Browser GUI to automatically plot the experiment data.
- Multi-dimensional sweeps with user defined setting relations. Quantities (frequency, amplitude, number of Cliffords) in a step sequence can be:
  - Single value defined by the user or a relational value from other step channels. (For example: $f_s = f_{mqpg}$ – *np.sqrt(2)*25 MHz.)* The user can also call built-in Python or NumPy (np.sqrt in the example) in the calculation.

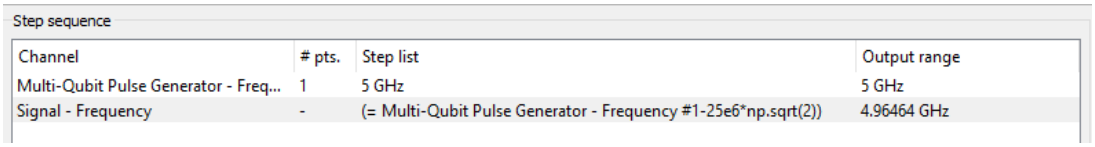


Figure 2. Defining step sequences in the M5401LxxA Software Solution

  - A linear or logarithmic sweep or union of several sweeps. Again, users can define relational quantities same as the single value instance.

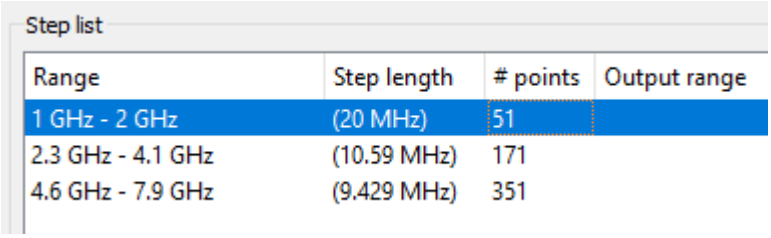


Figure 3. Defining steps in the M5401L Software Solution

For the Keysight M3XXXA AWG and Digitizer series

Labber maintains its functionality of allowing single waveforms to be loaded sequentially for any of the supported Keysight or 3rd party AWGs. Additionally, by using the hardware-loop mode, Labber allows the user to upload a sequence of waveforms to accelerate measurement times. Labber provides further enhancement when integrated with the Keysight Quantum Solution line. The familiar Labber use model of specifying pulse shapes remains. However, all quantities are efficiently parsed into FPGA commands to accelerate measurement times via efficient pulse primitive play as well as on-board FPGA demodulation for the Digitizer. Optimized on the M3202A AWG and M3102A Digitizers are:

- Multi-module and multi-chassis synchronization.

- Leveraging the KS2201A PathWave Test Sync Executive (external license required) allows for nanosecond timing and synchronization for up to four fully loaded M9019A (18-slot PXI) chasses.
- Sequences with parameterized pulses for fast sweeps or updates.
- Leveraging the M5400PLSA Quantum Library Dynamic Pulse Generation (external license required) enables pulse primitives uploaded to FPGA with amplitude, frequency, phase, and time delays between pulses to be set in register values for efficient, gapless operation.

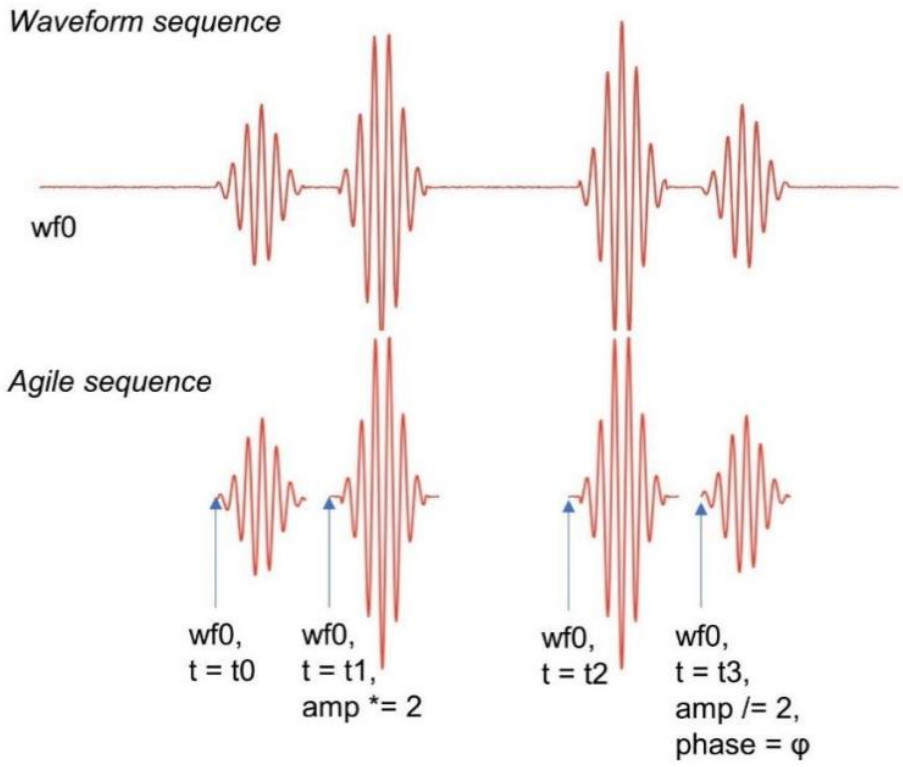
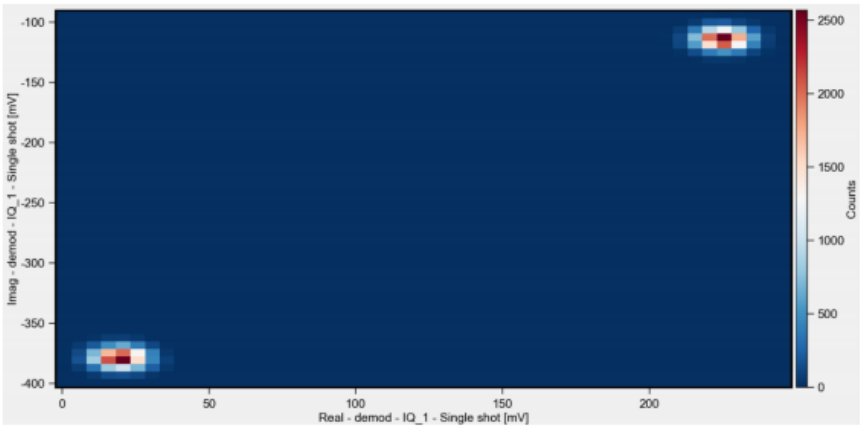


Figure 4. Parameterized pulses

Hardware-accelerated multiplexed readout

Leveraging the M5400DMOA Quantum Library Demodulation (external license required) allows for digital signal processing on the on-board FPGA for up to 10 multiplexed readout tones to be demodulated simultaneously. Users can specify different window shapes such as rectangular or custom. Labber provides an example in the user manual on how to tune-up the readout window.

Rectangular:



Custom:

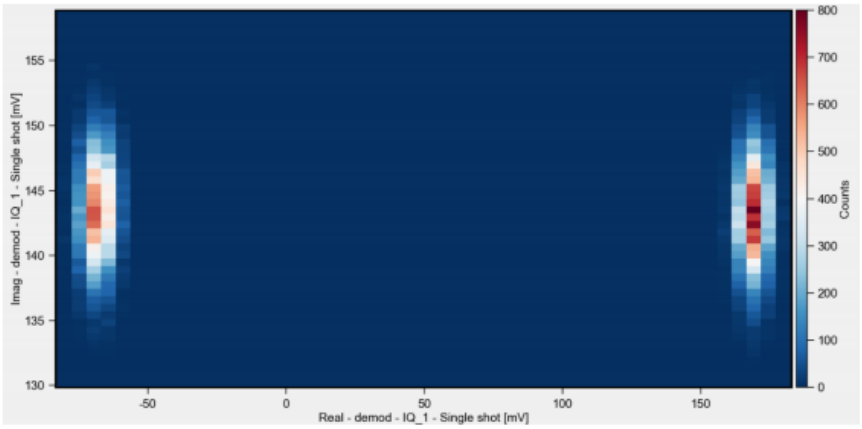
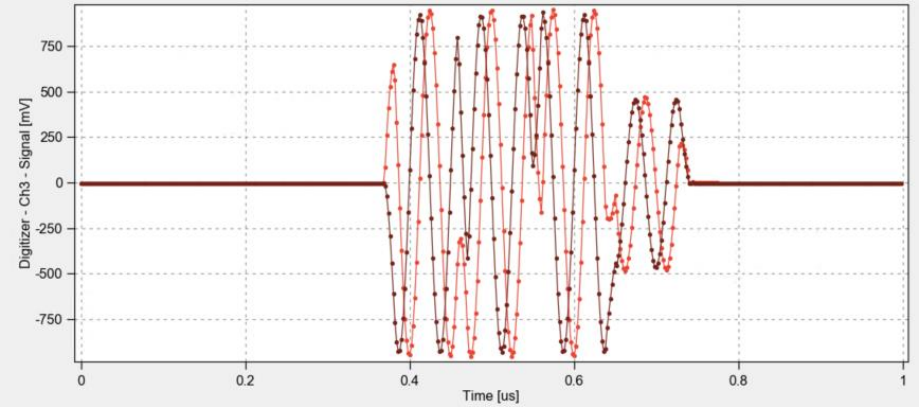


Figure 5. Multiplexed readouts

Multi-tone coherent control

- Leveraging the M5400PLSA Quantum Library Dynamic Pulse Generation (external license required) enables up to four coherent tones on one I/Q pair as well as two additional coherent tones on a second I/Q pair.

Example with two coherent tones



FFT

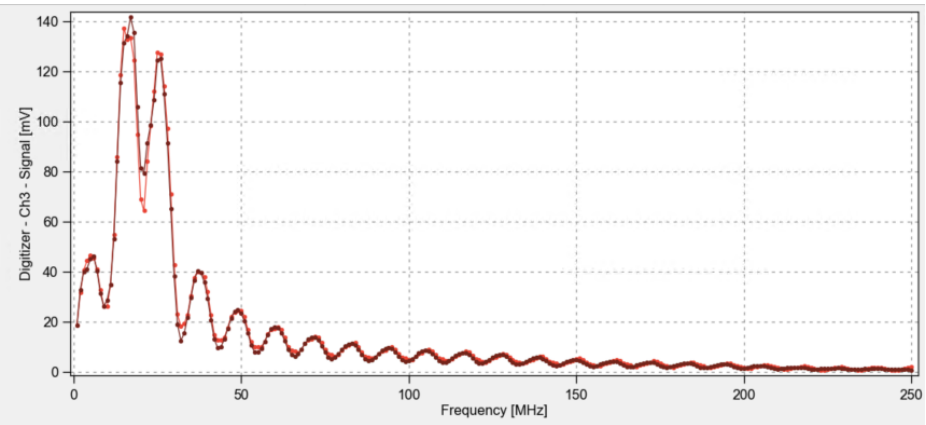


Figure 6. Multi-toned coherence control

Scheduling of experiments

Measurements can be scheduled from the user interface, or from the Python API. The user can set the scheduled date and time to run experiments. There is a priority flag that can be set to prioritize one experiment over the other if both are scheduled for the same time. Experiments can be scheduled to run just once or to run at fixed, repeated user defined intervals.

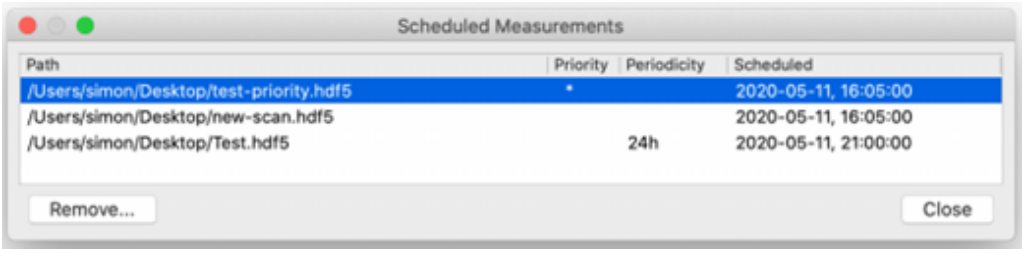


Figure 7. Measurement scheduling

Pulse sequencer supporting up to 64 qubits, with examples (e.g., randomized benchmarking)

- Pre-built Multi-Qubit Pulse Generator is for 16 qubits assuming a nearest-neighbor linear topology. Can easily rebuild from command line *'python generate_mqpg.py --n_qubits 64 (replace with desired qubit number)'* with topology specified in a JSON file. Pre-built single qubit randomized benchmarking and two qubit randomized benchmarking. Other built-in sequences are Rabi, Ramsey, CP/CPMG, as well as the option to write custom sequences via Python calling standard gates (Hadamard) or rotations (pi pulse or pi/2).

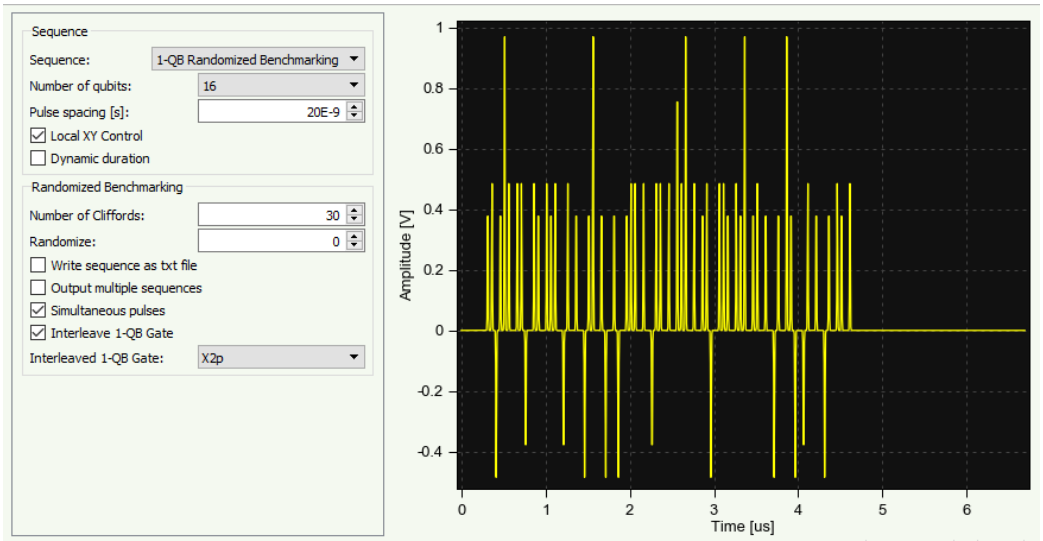
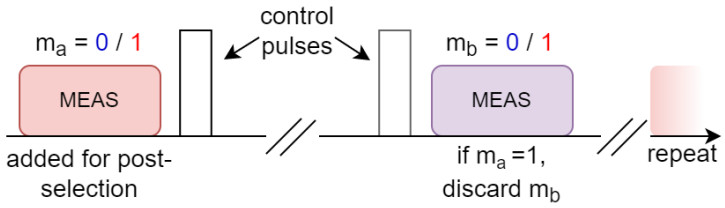


Figure 8. Pulse sequencing

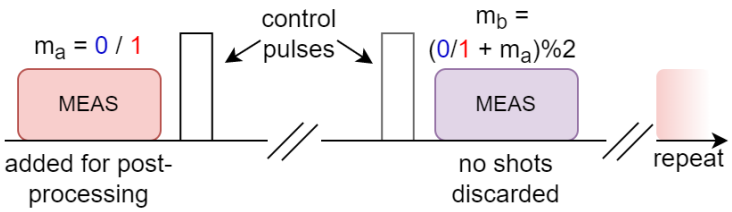
Software-based qubit state assignment and post-processing

- User defined demodulation voltage threshold to assign qubit state (0, 1, etc.). Able to save conditional sequences for effective qubit state reset as well as conditional mid-circuit measurement for parity monitoring.

Post Selection



Post Processing



Restless (uses last measurement in sequence)

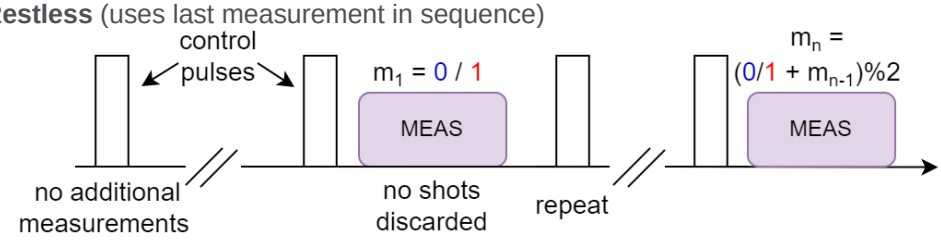


Figure 9. Qubit state assignment and post-processing

API toolkit with fitting functions

- Suite of pre-defined common fitting functions (Lorentzian, Gaussian, Exponential, and an Exponential convolved with a cosine) that can be auto fit from the GUI or API. Additionally, from the API, can call fit values from the fit log file to update for a sequential measurement enabling conditional scripting based on measurement results.

```
# # Use addFit to include the log channel name as well as fit function type
res_spec_meas_obj.addFit('Resonator Spectroscopy - Signal Frequency',
                                                'Lorentzian')

# # Perform measurement set perform_fit = True
res_spec_meas_obj.performMeasurement(return_data = True, use_scheduler = False,
                                                        perform_fit = True
                                            );

# # Create a measurement object for qubit spectroscopy
qubit_spec_meas_obj = Labber.ScriptTools.MeasurementObject(
                                            'Qubit Spectroscopy.labber',
                                            'Qubit Spectroscopy.hdf5'
                                                        )

# # Update the readout resonator LO based on the fit
qubit_spec_meas_obj.updateValueFromFit('Readout LO - Frequency',
                        res_spec_meas_obj.sCfgFileFits, 'f0'
                                            )

# # Use addFit to include the log channel name as well as fit function type
qubit_spec_meas_obj.addFit('Qubit Spectroscopy - Signal Frequency',
                                                'Lorentzian'
                                                            )

# # Perform measurement set perform_fit = True
qubit_spec_meas_obj.performMeasurement(return_data = True, use_scheduler = False,
                                                        perform_fit = True);
```

Labber Use Model

The software package consists of three separate programs:

1. The Instrument Server handles communication with instruments.
2. The Measurement program allows instrument values to be controlled and measured in user-defined sequences.
3. The Log Browser is used to organize and analyze the acquired data.

The relation between the parts is illustrated in Figure 10.

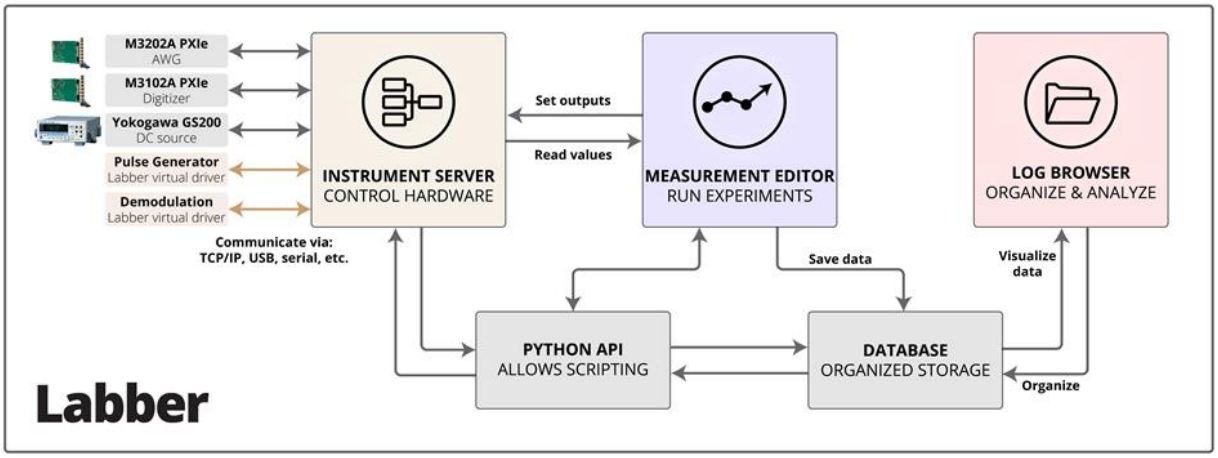


Figure 10. Labber use model

In a typical experimental setup, the Instrument Server keeps track of and communicates with all instruments and equipment available in the setup. The communication can be over GPIB, serial, USB, TCPIP, or any other interface supported by VISA. During an experiment, the Measurement program will connect to an Instrument Server to output values to one specific instrument, or to read data from another one. Note that the Measurement program only talks to the Instrument Server, and not directly with the instruments. This modular approach allows the same generic procedure to be used for setting/reading values, regardless of the instrument type or the communication interface.

The Measurement program saves the experimental configuration, the instrument settings, and the acquired data into a central log database. The Log Browser provides a fast and efficient method for browsing, visualizing, and organizing the measured data. Finally, the Log Viewer provides functionality for data analysis and for generating high-quality plots and figures.

In addition to the Instrument Server, Measurement, and Log Browser programs, there is a Python API that allows all functionality to be accessed programmatically for scripting purposes or for writing custom applications.

Licensing terminologies

Table 2. Licensing terminologies

Terminology name	Description
Subscription	Subscription licenses can be used through the term of the license only (6, 12, 24, or 36 months).

Licensing types

Table 3. Licensing types

License type	Description
Node-locked	License can be used on one specified instrument/computer.
Transportable	License can be used on one instrument/computer at a time but may be transferred to another using Keysight Software Manager (internet connection required).
USB Portable	License can be used on one instrument/computer at a time but may be transferred to another using a certified USB dongle (available for additional purchase with Keysight part number E8900-D10).
Floating (single site, regional, or worldwide)	Networked instruments/computers can access a license from a server one at a time. Multiple licenses can be purchased for concurrent usage.

For more information on Quantum Solutions, visit here.

Learn more at: www.keysight.com

For more information on Keysight Technologies products, applications, or services, please contact your local Keysight office. The complete list is available at www.keysight.com/find/contactus.

KEYSIGHT TECHNOLOGIES