



**OptiInstrument**

Instruments Communication and Control Tool

2.0

## New Features

---

OptiInstrument 2.0 is a new released software by Optiwave that addresses the needs of researchers, scientists, photonic engineers, professors and students who are working with instruments. OptiInstrument software satisfies the demands of users who are searching for a powerful yet easy tool to physically or remotely communicate and control instruments.





## OptiInstrument Software Overview

The user-friendly graphical user interface (GUI) of **OptiInstrument 2.0** Software is shown in Fig. 1. It is a standalone tool that can be used to communicate and control different kinds of instruments. OptiInstrument uses the standard commands for programmable instruments (**SCPI**) to communicate **physically** or **remotely** with instruments. The tool uses standard communication interfaces such as **TCP/IP**, **USB**, **GPIB**, or a serial port (**RS232/RS485**). Users can load lists of SCPI commands from **XML files** or write individual commands to control the instrument(s). The commands appear in a **tree configuration**. A single command or a sequence of commands can be executed by OptiInstrument. A **Python script** can be generated for the SCPI commands, saved, loaded and executed by OptiInstrument or in a Python environment. OptiInstrument GUI has a built-in viewer and CSV file analysis window. The GUI supports dockable windows that can be split of the main GUI or placed anywhere in the GUI. OptiInstrument is ideal for automated testing and characterization.

OptiInstrument software version 2.0 does not require the user to install python software on his computer. A directory with required python and plotting directories are embedded and offered as part of the installer of OptiInstrument software.

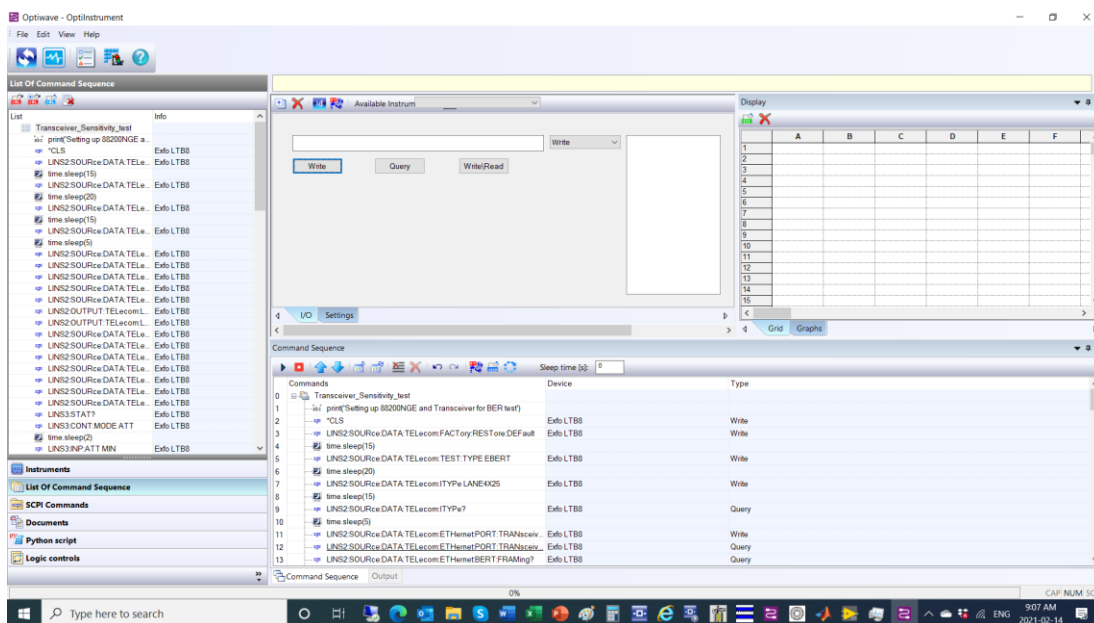


Fig. 1 OptiInstrument GUI

## Key Features of OptiInstrument 2.0

- User friendly GUI for efficient and intelligent testing and characterization.
- Embedded python installation files.



# OptiInstrument

- Execute single or sequence of SCPI commands.
- Load XML files and all other file formats into GUI panels.
- Drag and drop commands with flexible sequence ordering.
- Generate Python script for sequence of commands.
- Support logic statement such as while (while loop), if, else, elif (else if), for (for loop).
- Independent delay time (sleep) for each individual SCPI command.
- Offer “Basic helpers” statements such as print, sleep, break from a loop, continue and variables to build complex SCPI command sequences.
- Error handling support for identifying the type of error in the sequence.
- Addon GUI for data post processing, graphing and saving.
- Save output results in a Jason file format.
- Live display of output results for each SCPI command in the sequence.
- Built-in signal viewer and CSV file analysis page.
- Built-in full Python script editor.
- Remote operation and control of instruments.
- Supports dockable windows.

## OptiInstrument Software Applications

- Remotely communicate with instruments
- Setup parameters of equipment
- Automate testing and characterization
- View generated signals
- Extract & save the data of generated signals for post processing
- Integrate instruments with photonics and systems simulation tools

## Data Post Processing Addon GUI

OptiInstrument software Post Processing popup GUI is used for graphing, organizing and saving of the output data obtained from executing the sequence of commands. The post processing GUI is shown in Fig 2.



# OptiInstrument



Fig. 2 OptiInstrument Post Processing GUI

## OptiInstrument & Python Script

OptiInstrument software supports Python scripting. A Python script is generated for a single SCPI command or a list of commands using the tool. The generated script can be saved into a file. The generated script can be executed from OptiInstrument GUI or in **command prompt** or **Windows PowerShell**. A Python script can be loaded into OptiInstrument GUI and executed by the GUI. Fig. 3 shows a Python script generated for a sequence of SCPI commands and displayed in the GUI Python script pane. This capability allows users to execute features that are not supported by OptiInstrument GUI such as logic control and looping options.

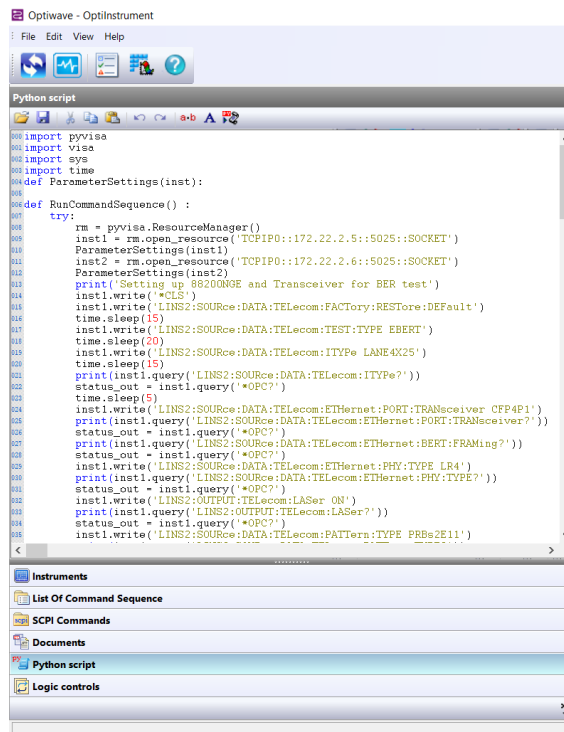


Fig. 3 Generated Python script for a sequence displayed in the Python script pane



## OptiInstrument 2.0 Example Library

OptiInstrument 2.0 Software has many examples that are created using commercial instruments from Rigol and EXFO. The examples are organized in subdirectories for each vendor. Each example has a readme file that describes the setup and the instrument(s)/card(s) used in each example as well as the result file(s). The **Samples** directory has also a subdirectory (**EXFO\_General SCPI Commands**) for all SCPI command offered by EXFO for their different equipment. These commands are saved in XML files that can be loaded into OptiInstrument List of Command Sequence pane and used to build the desired SCPI command sequences.

### 1. EXFO Samples

- a. CFP4 Longterm Sensitivity Test
- b. CFP4 Transceiver Sensitivity Setup-I
- c. CFP4 Transceiver Sensitivity Setup-II
- d. CFP4 Transceiver Sensitivity Setup-III
- e. EXFO OTDR card
- f. Long Term Stability\_LTB-8 cards
- g. Double Nested Loops\_LTB-8 cards
- h. PowerBalzer\_CFP4\_EBERT
- i. PowerMeter\_VOA\_CW Source
- j. PowerMeter\_2 CW Sources
- k. Switch\_OSA\_2 CW Sources
- l. Switch\_OSA\_VOA\_4 CW Sources
- m. Switch\_OSA\_VOA\_CW Source

### 2. EXFO\_General SCPI Commands

### 3. RIGOL Samples

- a. AM waveform
- b. Arbitrary waveform
- c. Burt waveform
- d. Harmonic waveform
- e. PSK waveform
- f. Pulse waveform
- g. Ramp waveform
- h. Sinewave
- i. Square waveform