
PyVISA Documentation

Release 1.4

Florian Bauer

February 27, 2017

1	Controlling GPIB, RS232, and USB instruments	1
1.1	PyVISA	1
1.2	General overview	2
1.3	VISA and Python	2
1.4	Links	2

Controlling GPIB, RS232, and USB instruments

Date February 27, 2017

Author Torsten Bronger

Maintainer Florian Bauer <pyvisa-devel@lists.sourceforge.net>

Downloads See the [PyVISA project page](#).

PyVISA

The PyVISA package enables you to control all kinds of measurement equipment through various busses (GPIB, RS232, USB) with Python programs. As an example, reading self-identification from a Keithley Multimeter with GPIB number 12 is as easy as three lines of Python code:

```
import visa
keithley = visa.instrument("GPIB::12")
print keithley.ask("*IDN?")
```

(That's the whole program; really!) It is tailored to work on both Windows and Linux, and with arbitrary adapters (e.g. National Instruments, Agilent, Tektronix, Stanford Research Systems). In order to achieve this, PyVISA relies on an external library file which is bundled with hardware and software of those vendors. (So only in rare cases you have to purchase it separately.)

PyVISA implements convenient and Pythonic programming in two layers:

1. An object-oriented Python module has been created simply called `visa`. It is the recommended way to use PyVISA. See the [PyVISA manual](#) for more information.
2. Additionally, there is the lower level module `vpp43`, which directly calls the VISA functions from Python. See the [PyVISA low-level implementation](#) for more information. This is only for people who need full control or the official VISA functions for some reason.

PyVISA is free open-source software. The [PyVISA project page](#) contains the bug tracker and the download area.

Projects using PyVISA so far:

- [pyvLab](#) – program to control VISA-talking instruments

General overview

The programming of measurement instruments can be real pain. There are many different protocols, sent over many different interfaces and bus systems (GPIB, RS232, USB). For every programming language you want to use, you have to find libraries that support both your device and its bus system.

In order to ease this unfortunate situation, the VISA ¹ specification was defined in the middle of the 90ies. Today VISA is implemented on all significant operating systems. A couple of vendors offer VISA libraries, partly with free download. These libraries work together with arbitrary peripheral devices, although they may be limited to certain interface devices, such as the vendor's GPIB card.

The VISA specification has explicit bindings to Visual Basic, C, and G (LabVIEW's graphical language). However, you can use VISA with any language capable of calling functions in a DLL. Python is such a language.

VISA and Python

Python has a couple of features that make it very interesting for measurement controlling:

- Python is an easy-to-learn scripting language with short development cycles.
- It represents a high abstraction level ², which perfectly blends with the abstraction level of measurement programs.
- It has a very rich set of native libraries, including numerical and plotting modules for data analysis and visualisation.
- A large set of books (in many languages) and on-line publications is available.
- You can download it for free at <http://www.python.org>.

Links

- PyVISA itself:
 - the [PyVISA project page](#) with download area and bug tracker
 - the [PyVISA manual](#)
 - the [PyVISA low-level implementation](#) called vpp43
- The original VISA docs:
 - [VISA specification](#) (scroll down to the end)
 - [VISA library specification](#)
 - [VISA specification for textual languages](#)
- The very good VISA manuals from [National Instruments's VISA pages](#):
 - [NI-VISA User Manual](#)
 - [NI-VISA Programmer Reference Manual](#)
 - [NI-VISA help file in HTML](#)

¹ Virtual Instrument Software Architecture

² For example, you don't need to care about the underlying operating system with all its peculiarities.