

Instrbuilder: A Python package for electrical instrument control

Lucas J. Koerner¹

¹ Lucas J. Koerner, Department of Electrical and Computer Engineering, University of St. Thomas

DOI: [10.21105/joss.01172](https://doi.org/10.21105/joss.01172)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 06 January 2019

Published: 15 April 2019

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Background

instrbuilder is an open-source Python package for control of electrical instruments. This package eases the development of high-level “drivers” to interface with oscilloscopes, power supplies, function generators, multimeters, or any instrument that uses ASCII textual string communication, such as SCPI (SCPI Consortium, 1999). *instrbuilder* is particularly suited for prototyping and automating experiments in research laboratory setups within an IPython terminal. Our goals are to accelerate the development of automated data collection and improve the reproducibility of laboratory experiments.

Summary

instrbuilder reads and writes data from instruments over a serial interface by using PySerial (Liechti, 2017) or over a USB interface by using PyVISA (PyVISA Authors, 2018). *instrbuilder* leverages the consistency of SCPI commands so that new instrument “drivers” can be created with a simple CSV file (example command files are included in the package). An example write command to configure the offset of a function generator:

```
fig.set('offset', 0.5)
```

and an example read command:

```
offset = fig.get('offset')
```

Nonstandard instrument commands may also be generated by writing Python code. Each instrument object is comprised of a communication handle and a list of commands. Each command may be automatically tested for communication errors or unexpected return values and stores a help string. To support experimental reproducibility, commands that access a configuration state are specified so that a complete archive of the instrument configuration may be automatically read and logged. Instrument addresses and other environment properties may be stored in a YAML system configuration file so that the user may name each instrument and simply open instruments by name. *instrbuilder* prompts the user through the steps to generate the YAML system configuration file.

Documentation is online <https://lucask07.github.io/instrbuilder/build/html/> and [examples](#) are available in the [GitHub repository](#). These minimal working examples (MWEs) include control of a function generator, power supply, oscilloscope, lock-in amplifier, and

an integrated circuit (IC) interfaced with an Aardvark I2C/SPI controller (Total Phase, Santa Clara, CA).

instrbuilder supports a Register class that targets registers of an IC. The IC class objects include a dictionary of the registers and have methods for reading and writing these registers via SPI or I2C.

instrbuilder is part of a Python software suite that uses Bluesky (Arkilic et al., 2017) and associated modules created for x-ray science by a team at the NSLS-II (see <https://nsls-ii.github.io/>). This combination generates an instrument and experiment control environment that facilitates reproducible experiments. The use of *instrbuilder* and *Bluesky* is [documented online](#) and a [MWE is available](#). *instrbuilder* is available at (<https://pypi.org/project/instrbuilder/>) and can be installed with pip.

Acknowledgements

L. Koerner acknowledges laboratory start-up funds from the University of St. Thomas School of Engineering.

References

- Arkilic, A., Allan, D. B., Caswell, T., Li, L., Lauer, K., & Abeykoon, S. (2017). Towards integrated facility-wide data acquisition and analysis at NSLS-II. *Synchrotron Radiation News*, 30(2), 44–45. doi:[10.1080/08940886.2017.1289810](https://doi.org/10.1080/08940886.2017.1289810)
- Liechti, C. (2017). PySerial documentation. Retrieved from <http://pyserial.readthedocs.io>
- PyVISA Authors. (2018). PyVISA documentation. Retrieved from <https://pyvisa.readthedocs.io>
- SCPI Consortium. (1999). *Standard commands for programmable instruments (SCPI), volume 1: Syntax and style*. USA, May.