

---

# **Instruments.jl Documentation**

*Release 0.1*

**Colm Ryan**

January 17, 2016



|          |                           |          |
|----------|---------------------------|----------|
| <b>1</b> | <b>Installation</b>       | <b>3</b> |
| <b>2</b> | <b>NI-VISA</b>            | <b>5</b> |
| <b>3</b> | <b>Indices and tables</b> | <b>7</b> |



Instruments.jl enables instrument control from Julia. A minimal session might look like:

```
using Instruments

uwSource = GenericInstrument()
connect!(uwSource, "GPIB0::28::INSTR")
query(uwSource, "*IDN?") # prints "Rohde&Schwarz,SMIQ...."
disconnect!(uwSource)
```

The Instrument.jl wraps a VISA library to provide instrument communications over the usual interfaces of GPIB, TCPIP, USB and RS232. In addition it provides types and macros to make it easy to write your own custom driver for a particular instrument.

Contents:



---

## Installation

---

Instruments.jl is available from the Julia package repository and can be installed with:

```
Pkg.add("Instruments")
```



---

## NI-VISA

---

Instruments.jl uses a C VISA library under the hood. The only one it has been tested with is the National Instruments implementation NI-VISA. Fortunately, this is available for all platforms.

- Windows: [NI-VISA 14.0](#)
- Linux (32 bit only <sup>1</sup>): [NI-VISA 14.0](#)
- Mac OS X: [NI-VISA 14.0](#)

---

<sup>1</sup> There is some hope this annoying restriction will go away shortly as NI have a pre-release version with 64 bit support ([forum post](#)).



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`