

# How to Control Tabor AWGs with Python

## Getting started

Python programming language is widely used to control and program various test and measurement instruments, be it a single instrument or a system with various instruments. The reasons Python programming is gaining such popularity are:

- It is an interpreted programming language meaning you don't need to compile your code each time you are doing some changes in order to run it.
- Easy to read & write code with, with a lot of libraries for pretty much anything.
- It's free

This series of tutorials, "How to Control Tabor AWGs with Python", will provide step by step instructions and various examples of how to use Python in conjunction with Tabor Arbitrary Waveform Generators.

This first tutorial of the series will explain how to get started and what needs to be installed in order to control the Tabor AWG with Python. This series of tutorials demonstrates how to program using Python 2.7 (Anaconda) to control **Tabor's WX2184C** model. For connecting with other Tabor models, such as the WW series, just follow the same instructions with the relevant changes (downloading the matching drivers for the specific model).

This set of tutorials assumes you successfully established connection with the Tabor unit using your preferred remote interface method (LAN, GPIB or USB). Please note that if you have a unit which is not one of the WX series model, you will need to first go over our [connectivity tutorials](#) to make sure you are all set (for establishing a USB connection using one of our Wonder Wave series of AWGs, you'll need to **download & install the Tabor USB driver**).

There are two ways to control an instrument. The first is using the Standard Commands for Programmable Instruments (SCPI), which are an ASCII-based set of commands for reading and writing instrument settings. The second is to use the IVI driver functions of the instrument. The IVI driver provides a higher level of programming that doesn't require any knowledge of the instrument's SCPI commands. This tutorial will explain how to install all that is needed regardless of the method chosen to control the instrument.

### IMPORTANT NOTE BEFORE PROCEEDING

It is possible to control the AWG without NI-VISA installation. This is done using Python libraries and instrument specific SCPI commands. For those interested in this method please proceed to the next tutorial “How to Control Tabor AWGs With Python – Using SCPI Commands” where you will find an example attached demonstrating how to do so.

### ➔ To connect and control the Tabor AWG with Python

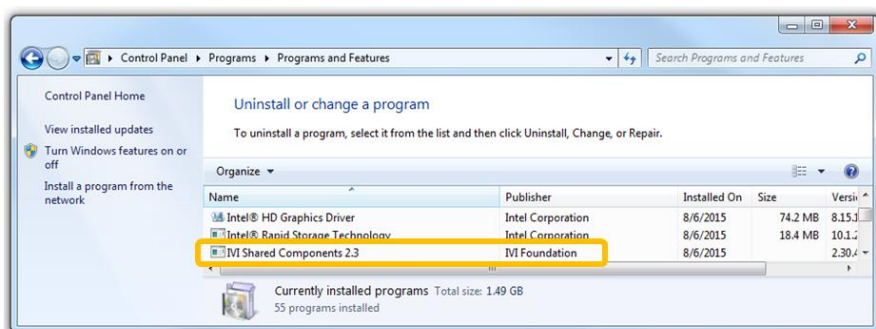
1. In order to control the instrument through VISA, first you will need to install National Instruments latest VISA. Please visit [NI VISA](#) download page and install the latest version according to your OS (32bit & 64bit).

**Skip to step 8 if you only intend to use SCPI commands in your code without the help of our IVI functions:**

2. If you intend to use the Tabor IVI driver functions, please download the NI-IVI Compliance Package. Please visit [IVI Compliance Package](#) download page and install the latest version.
3. Next, please download and install the latest [IVI shared Components from the IVI foundation webpage](#). Please make sure to choose the right version according to your OS.

IviSharedComponents_2.3.0.exe	This file is an executable installer that installs the IVI Shared Components on a 32-bit system. This executable installer installs the same components as the MSI package.
IviSharedComponents64_2.3.0.exe	This file is an executable installer that installs the IVI Shared Components on a 64-bit system. This executable installer installs the same components as the MSI package.

4. To verify that it installed correctly go to *Control Panel\Programs\Programs and Features*:





5. To install the Tabor IVI driver, go to the [downloads](#) page on Tabor's website. Click on the Downloads menu entry, select the model from the Model Number drop-down box, and select "Drivers" from the Download Type drop-down box. Click on the Search button. Download the IVI driver that matches your Python 2.7 version (64bit or 32bit).

Please choose the type of content that you need

Model Number  Download Type  [SEARCH >](#)

Model WX2184C

Drivers	IVI Driver for models WS8351/2, WX2181/2, WX1281/2B, WX2181/2B, WX1281/2/4C and WX2181/2/4C (32 Bit OS)	14/12/2014 Ver. 3.0.2	 <a href="#">Download (7.2 Mb)</a>
Drivers	IVI Driver for models WS8351/2, WX2181/2, WX1281/2B, WX2181/2B, WX1281/2/4C and WX2181/2/4C (64 Bit OS)	14/12/2014 Ver. 3.0.2	 <a href="#">Download (10.5 Mb)</a>

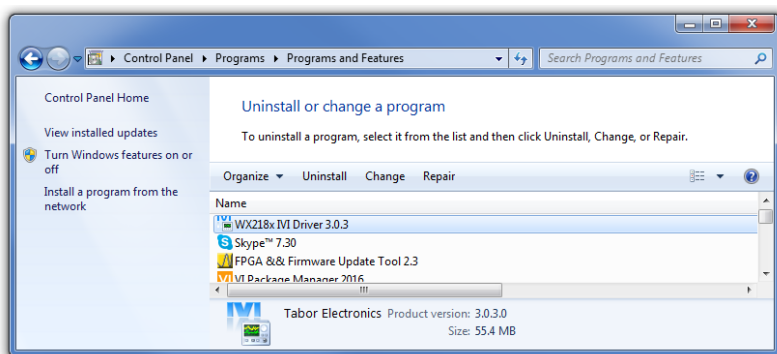
#### NOTE

In order to download the IVI driver, you must be registered to Tabor's website using an email and a password.

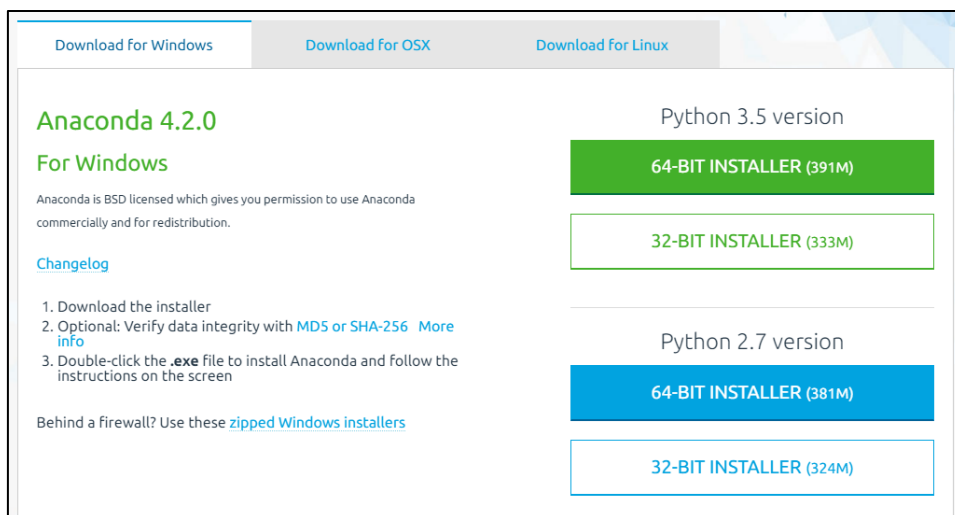
6. After the download is complete execute the installation file and follow the on screen instructions.



7. Once the installation is complete go to *Control Panel\Programs\Programs and Features* and check that the IVI driver has been installed properly:

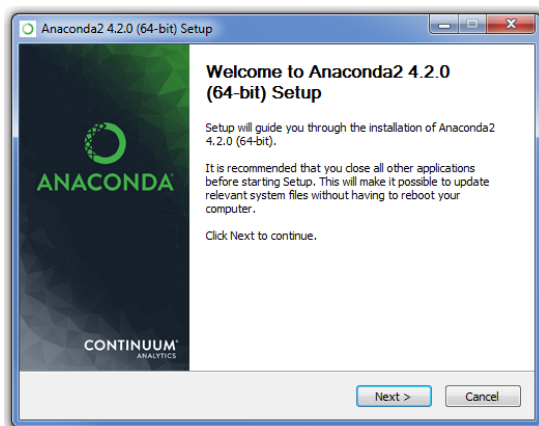
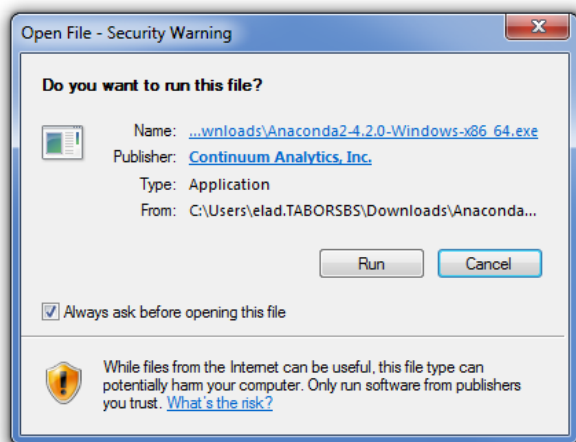


8. Next, download the latest version of [Anaconda](#)'s Python installer from the Continuum Analytics website according to your OS (32bit or 64bit):

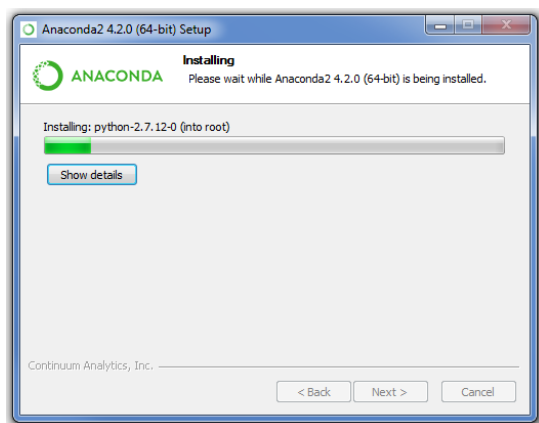


We recommend to use the Python 2.7.x with the Tabor instruments. For this tutorial we choose to demonstrate the Python 2.7 64bit installer on a 64bit Win 7 OS. If you choose differently, please make sure that your NI-VISA + Python (+ IVI driver) share the same bitness (32bit & 64bit).

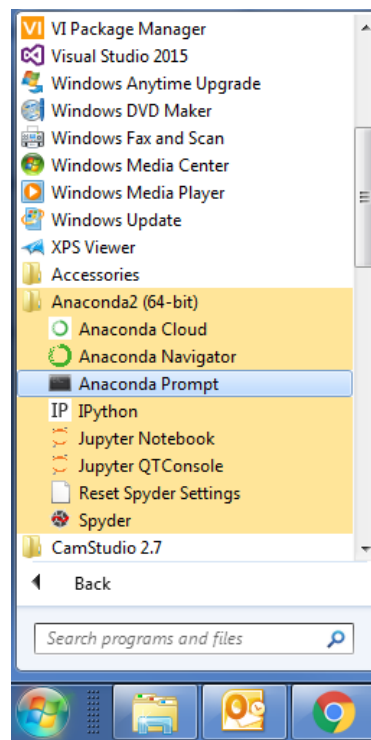
9. To start the download, press the 'Run' button then 'Next':



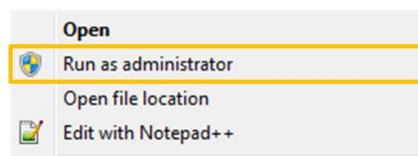
10. Wait for the installation to complete. Once finished, you may be prompted to perform a reboot:



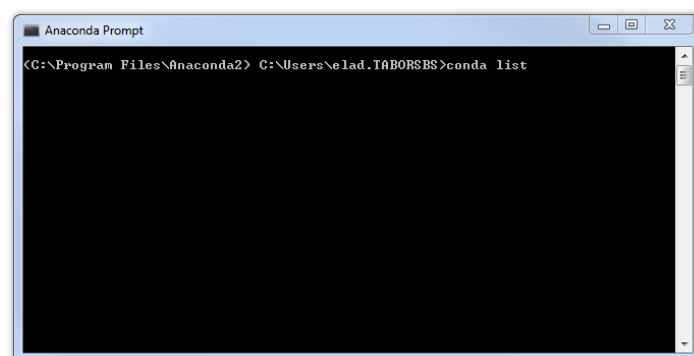
11. Go to Start -> All programs -> Anaconda2:



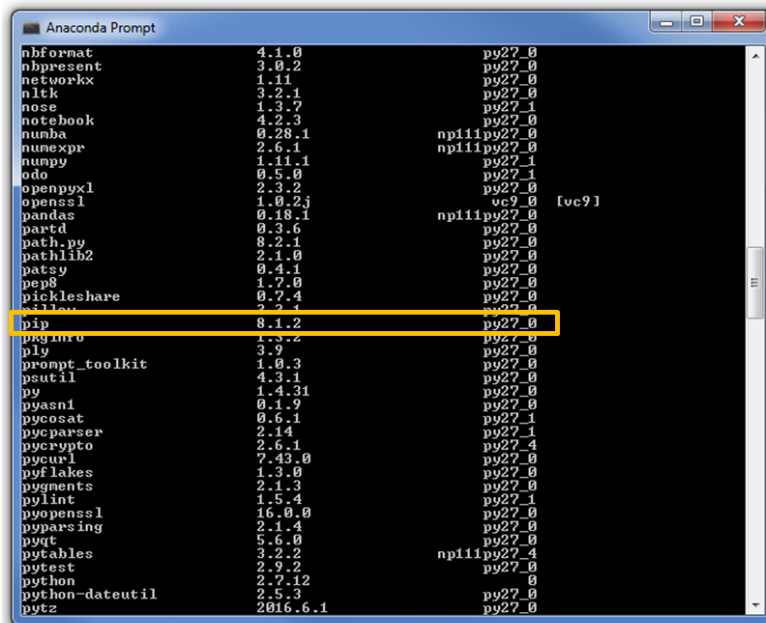
Right click on the 'Anaconda Prompt'. Choose to Run as administrator:



12. Type "conda list" as can be seen below:



It will open a list of all python packages already installed on your computer. Make sure you have 'pip' installed:

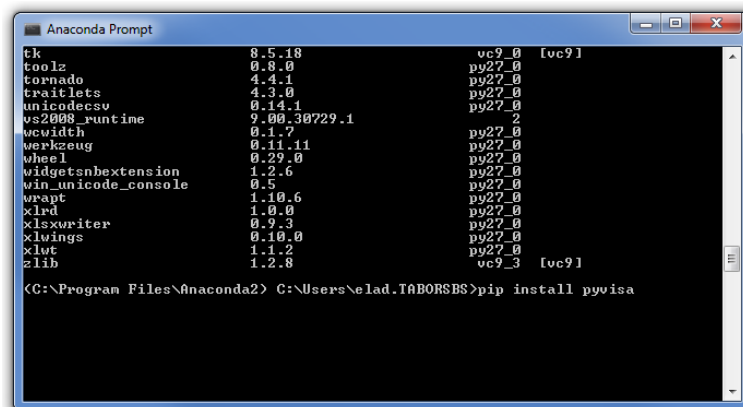


```

Anaconda Prompt
nbformat 4.1.0 py27_0
nbpresent 3.0.2 py27_0
networkx 1.11 py27_0
nlTK 3.2.1 py27_0
nose 1.3.7 py27_1
notebook 4.2.3 py27_0
numba 0.28.1 np111py27_0
numexpr 2.6.1 np111py27_0
numpy 1.11.1 py27_1
odo 0.5.0 py27_1
openpyxl 2.3.2 py27_0
openssl 1.0.2j vc9_0 [vc9]
pandas 0.18.1 np111py27_0
partd 0.3.6 py27_0
path.py 8.2.1 py27_0
pathlib2 2.1.0 py27_0
patsy 0.4.1 py27_0
pep8 1.7.0 py27_0
pickleshare 0.7.4 py27_0
pillow 3.2.0 py27_0
pip 8.1.2 py27_0
pgmq 1.3.2 py27_0
ply 3.9 py27_0
prompt_toolkit 1.0.3 py27_0
psutil 4.3.1 py27_0
py 1.4.31 py27_0
pyasn1 0.1.9 py27_0
pycosat 0.6.1 py27_1
pyparser 2.14 py27_1
pycrypto 2.6.1 py27_4
pycurl 7.43.0 py27_0
pyflakes 1.3.0 py27_0
pygments 2.1.3 py27_0
pylint 1.5.4 py27_1
pyopenssl 16.0.0 py27_0
pyparsing 2.1.4 py27_0
pyqt 5.6.0 py27_0
pytables 3.2.2 np111py27_4
pytest 2.9.2 py27_0
python 2.7.12 py27_0
python-dateutil 2.5.3 py27_0
pytz 2016.6.1 py27_0

```

13. Next type “pip install pyvisa” in order to install the Python wrapper for the NI- VISA you already installed. Press Enter:

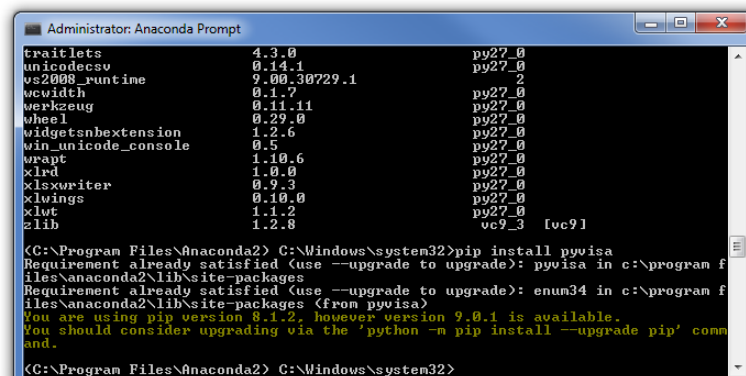


```

Anaconda Prompt
tk 8.5.18 vc9_0 [vc9]
toolz 0.8.0 py27_0
tornado 4.4.1 py27_0
traitlets 4.3.0 py27_0
unicodesv 0.14.1 py27_0
vs2008_runtime 9.00.30729.1 2
wcmid 0.1.7 py27_0
werkzeug 0.11.11 py27_0
wheel 0.29.0 py27_0
widgetsnextension 1.2.6 py27_0
win_unicode_console 0.5 py27_0
wrapt 1.10.6 py27_0
xlrd 1.0.0 py27_0
xlswriter 0.9.3 py27_0
xlwings 0.10.0 py27_0
xlwt 1.1.2 py27_0
zlib 1.2.8 vc9_3 [vc9]

<C:\Program Files\Anaconda2> C:\Users\elad.TABORSBS>pip install pyvisa

```



```

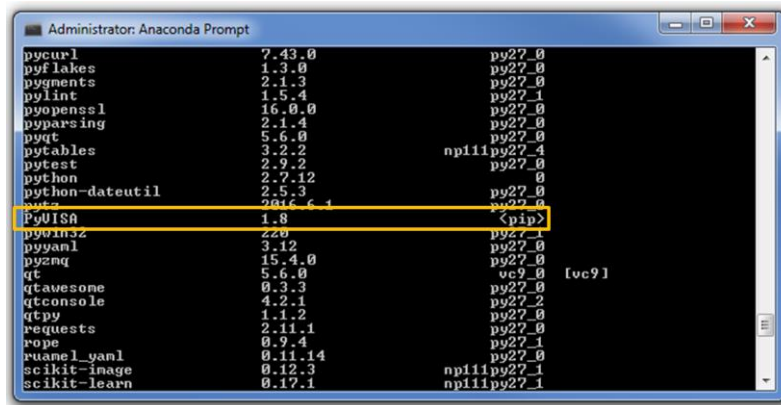
Administrator: Anaconda Prompt
traitlets 4.3.0 py27_0
unicodesv 0.14.1 py27_0
vs2008_runtime 9.00.30729.1 2
wcmid 0.1.7 py27_0
werkzeug 0.11.11 py27_0
wheel 0.29.0 py27_0
widgetsnextension 1.2.6 py27_0
win_unicode_console 0.5 py27_0
wrapt 1.10.6 py27_0
xlrd 1.0.0 py27_0
xlswriter 0.9.3 py27_0
xlwings 0.10.0 py27_0
xlwt 1.1.2 py27_0
zlib 1.2.8 vc9_3 [vc9]

<C:\Program Files\Anaconda2> C:\Windows\system32>pip install pyvisa
Requirement already satisfied (use --upgrade to upgrade): pyvisa in c:\program f
iles\anaconda2\lib\site-packages
Requirement already satisfied (use --upgrade to upgrade): enum34 in c:\program f
iles\anaconda2\lib\site-packages (from pyvisa)
You are using pip version 8.1.2, however version 9.0.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' comm
and.

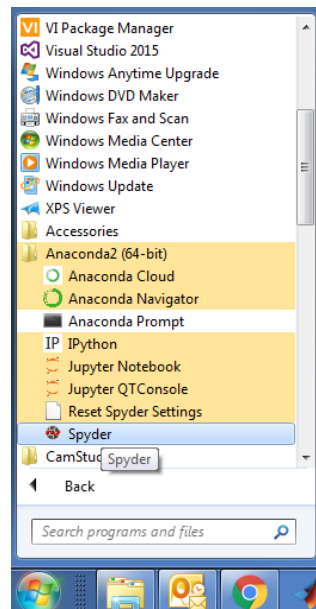
<C:\Program Files\Anaconda2> C:\Windows\system32>

```

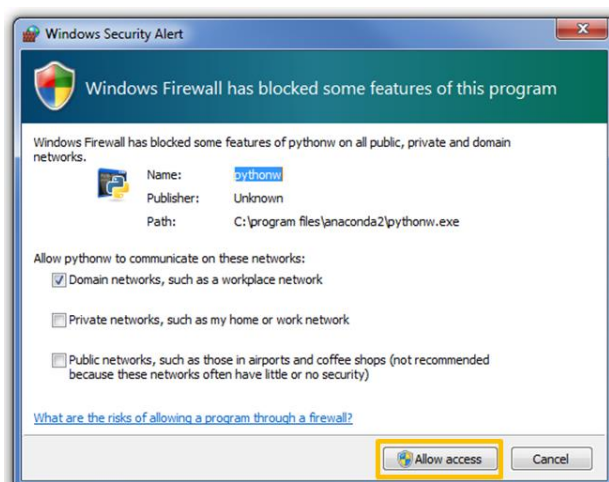
Make sure you can see it in the list of installed packages:



14. Go to 'Start ->> All programs ->> Anaconda2' and run the Spyder IDE:

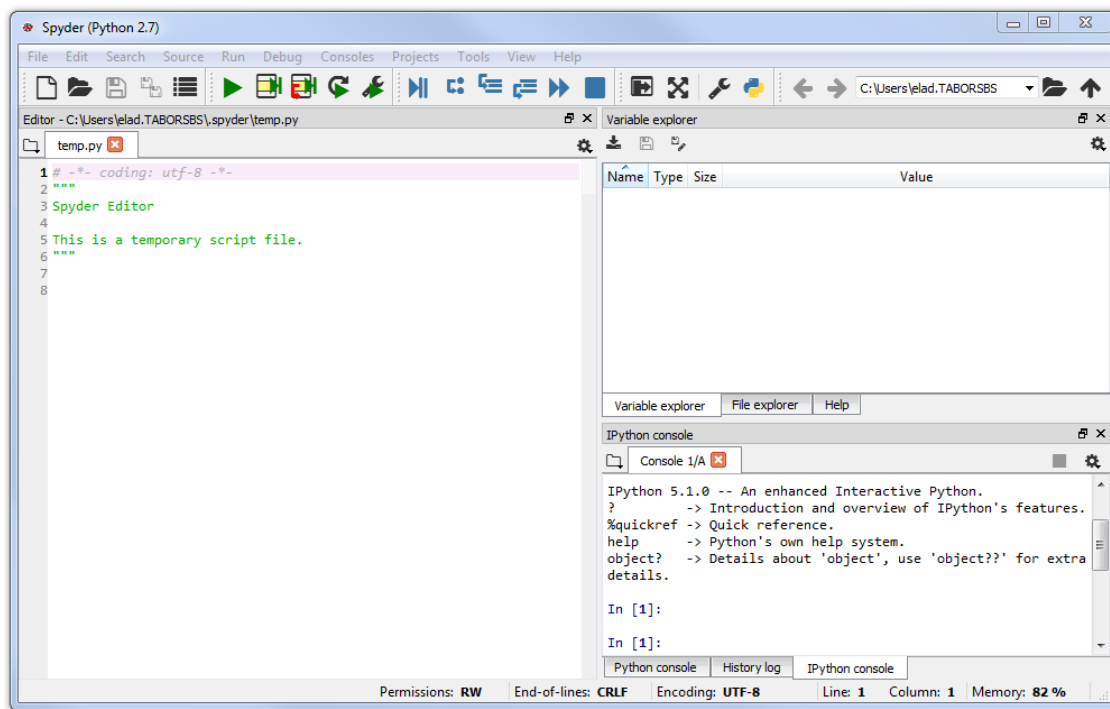


You will be asked to allow access:





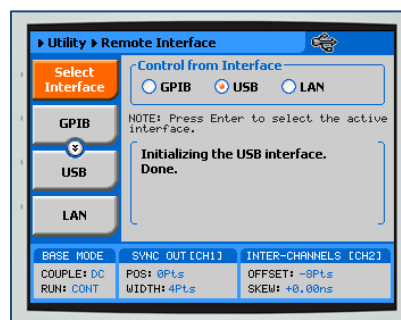
15. Wait for the Spyder IDE to appear:




## ➔ To Test Communication with Tabor AWG using Python

For this example a WX2184C unit will be connected remotely to the computer through the USB port:

1. Set the USB/LAN/GPIB as the remote interface, using the Tabor's front panel buttons. To do so, go to: "Utility"->"Remote Interface"->"Select Interface"->"Control from Interface". Press Enter to select the active Interface you need. Wait for the answer "Done":



2. Visit the following [PyVISA webpage](#), where you will find a detailed information regarding the use of PyVISA. Try the following lines of code as can be seen in the example below to test the communication with the Tabor instrument (for testing a TCPIP connection, skip to the next tutorial: “How to Control Tabor AWGs with Python – Using SCPI commands”).


**PyVISA**  
 Python wrapper for the Virtual Instrument Software Architecture library. You are currently looking at the documentation of version 1.8.

Search docs

Installation  
 Configuring the NI backend  
 Tutorial  
 Reading and Writing values  
 A more complex example  
 Resources  
 A frontend for multiple backends  
 PyVISA Shell  
 Architecture  
 VISA resource names  
 Migrating from PyVISA < 1.5  
 Contributing to PyVISA  
 Frequently asked questions  
 NI-VISA Installation  
 API

[Docs](#) » [PyVISA: Control your instruments with Python](#)
[Edit on GitHub](#)

## PyVISA: Control your instruments with Python




PyVISA is a Python package that enables you to control all kinds of measurement devices independently of the interface (e.g. GPIB, RS232, USB, Ethernet). 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
>>> rm = visa.ResourceManager()
>>> rm.list_resources()
('ASRL1::INSTR', 'ASRL2::INSTR', 'GPIB0::12::INSTR')
>>> inst = rm.open_resource('GPIB0::12::INSTR')
>>> print(inst.query("*IDN?"))

```

(That's the whole program; really!) It works on Windows, Linux and Mac; with arbitrary adapters (e.g. National Instruments, Agilent, Tektronix, Stanford Research Systems).


**PyVISA**  
 Python wrapper for the Virtual Instrument Software Architecture library. You are currently looking at the documentation of version 1.8.

Search docs

Installation  
 Configuring the NI backend  
**Tutorial**  
 An example  
 Example for serial (RS232) device  
 Reading and Writing values  
 A more complex example  
 Resources  
 A frontend for multiple backends  
 Read the Docs v: stable

## An example

Let's go *in medias res* and have a look at a simple example:

```

>>> import visa
>>> rm = visa.ResourceManager()
>>> rm.list_resources()
('ASRL1::INSTR', 'ASRL2::INSTR', 'GPIB0::14::INSTR')
>>> my_instrument = rm.open_resource('GPIB0::14::INSTR')
>>> print(my_instrument.query("*IDN?"))

```

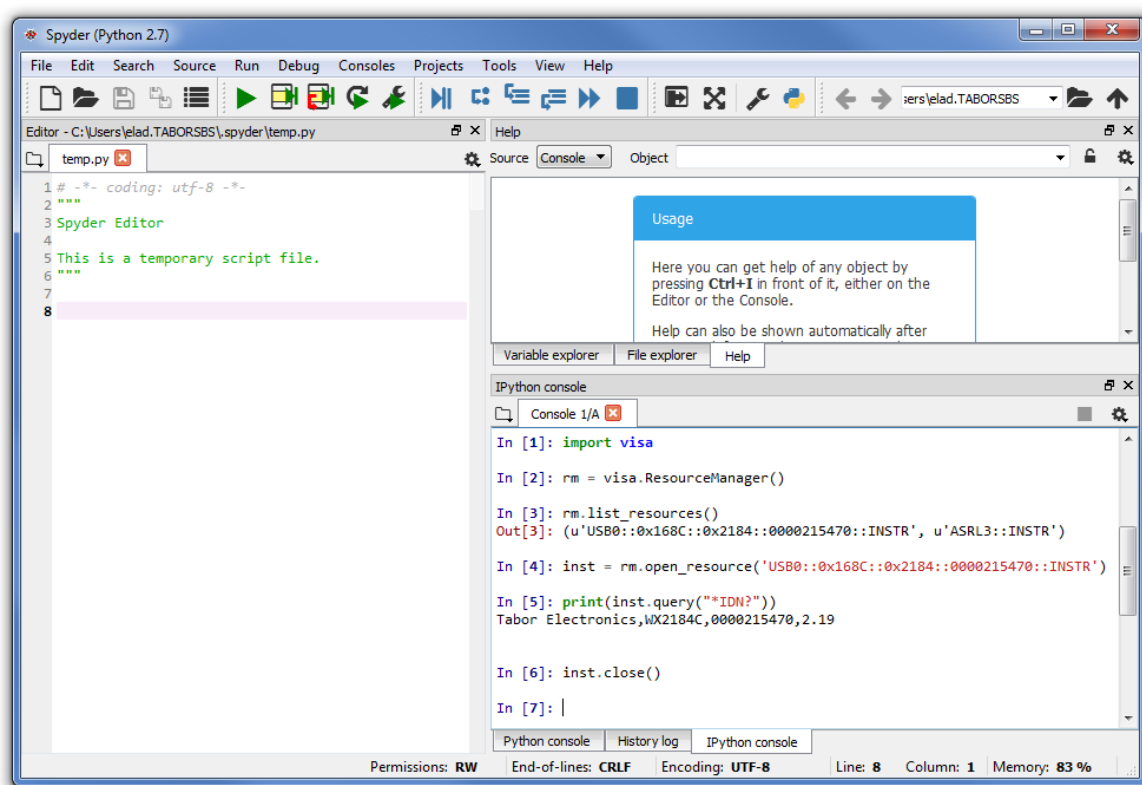
This example already shows the two main design goals of PyVISA: preferring simplicity over generality, and doing it the object-oriented way.

After importing `visa`, we create a `ResourceManager` object. If called without arguments, PyVISA will use the default backend (NI) which tries to find the VISA shared library for you. You can check, the location of the shared library used simply by:

```

>>> print(rm)
<ResourceManager('/path/to/visa.so')>

```



3. Once you'll run the example, a red REMOTE LOCKOUT sentence will pop up on the LCD display of the Tabor instrument, indicating you have established connection with the unit.

Now that all the software and drivers required to work with Python have been installed, please proceed to the next tutorial "How to Control Tabor AWGs with Python – Using SCPI commands".

## For More Information

To learn more about how to remote control Tabor instruments using Python, visit our website Support & Tutorials zone. If you encounter difficulties with connecting to Tabor units using Python, please contact us at [support@taborelec.com](mailto:support@taborelec.com) and our support team will gladly help. For more of Tabor's solutions or to schedule a demo, please contact your local Tabor representative or email your request to [info@tabor.co.il](mailto:info@tabor.co.il). More information can be found at our website at [www.taborelec.com](http://www.taborelec.com)

© Proprietary of Tabor Electronics Ltd.