

# TUTORIAL N° 5

## PYTHON TUTORIAL



**PIEZOCONCEPT**  
**15 Rue du Bocage**  
**69008 LYON**  
**France**

The below describes an example of program made under PYTHON. It can be found as part of the python module 'nplab' at:

[https://github.com/nanophotonics/nplab/blob/master/nplab/instrument/stage/Piezoconcept\\_FOC100.py](https://github.com/nanophotonics/nplab/blob/master/nplab/instrument/stage/Piezoconcept_FOC100.py)

For the code to work in its current state, you will need to download and install the entire module. This can be done once the files have been downloaded via setup.py file located in the nplab directory by running:

```
python setup.py install
```

from the command line within the nplab directory.

The program works for Z stage (FOC nanopositioner) but it can be adapted to any other stages.

```
# -*-
coding:
utf-8 -
*_
"""
Created on Thu Oct 01 11:52:44 2015
@author: hera
"""
import serial
import nplab.instrument.serial_instrument as si
class Piezoconcept(si.SerialInstrument):
    '''A simple class for the Piezo concept FOC100 nanopositioning system'''

    def __init__(self, port=None):
        self.termination_character = '\n'
        self.port_settings = {
            'baudrate':115200,
            'bytesize':serial.EIGHTBITS,
            'parity':serial.PARITY_NONE,
            'stopbits':serial.STOPBITS_ONE,
            'timeout':1, #wait at most one second for a response
            # 'writeTimeout':1, #similarly, fail if writing takes >1s
            # 'xonxoff':False, 'rtscts':False, 'dsrdtr':False,
        }
        si.SerialInstrument.__init__(self,port=port)
        self.recenter()

    def move_rel(self,value,unit="n"):
        '''A command for relative movement, where the default units is nm'''
        if unit == "n":
            multiplier=1
        if unit == "u":
            multiplier=1E3
```

```
if (value*multiplier+self.position) > 1E5 or (value*multiplier+self.position) < 0:
    print "The value is out of range! 0-100 um (0-1E8 nm) (Z)"
elif (value*multiplier+self.position) < 1E5 and (value*multiplier+self.position) >= 0:
    self.write("MOVRX "+str(value)+unit)
    self.position=(value*multiplier+self.position)

def move(self,value,unit="n"):
    '''An absolute movement command, will print an error to the console
    if you moveoutside of the range(100um) default unit is nm'''
    if unit == "n":
        multiplier=1
    if unit == "u":
        multiplier=1E3

    if value*multiplier >1E5 or value*multiplier <0:
        print "The value is out of range! 0-100 um (0-1E8 nm) (Z)"

    elif value*multiplier < 1E5 and value*multiplier >=0:
        self.write("MOVEX "+str(value)+unit)
        self.position = value*multiplier

def move_step(self,direction):
    self.move_rel(direction*self.stepsize)

def recenter(self):
    ''' Moves the stage to the center position'''
    self.move(50,unit = "u")
    self.position = 50E3

def INFO(self):
    return self.query("INFOS",multiline=True,termination_line= "\n \n \n \n",timeout=.1,)

if __name__ == "__main__":
    '''Basic test, should open the Z stage and print its info before closing.
    Obvisouly the comport has to be correct!'''
    Z = Piezoconcept(port = "COM9")
    print Z.INFO()
    Z.close()
```