PreciGenome

Microfluidics Flow Control System

Software Development Kit for Python

Version 0.1.1

December 6, 2019

Installation

Install using Python

To install the python package on Windows, use the pip script.

```
pip install Precigenome_mfcs-0.1.0-py2-none-any.whl on Windows,
pip install Precigenome_mfcs-0.1.0.tar.gz on Linux,
```

to uninstall this package, use the pip script

```
pip uninstall Precigenome_mfcs
```

Use without installing

If you wish to use the package on a single project without installing it, extract the Precigenome_mfcs-0.1.0.zip file and place the Precigenome folder in your project directory. As long as you work from that directory, the package will be available as if it were installed, it will be available to use with the same syntax as the examples provided.

Using the SDK

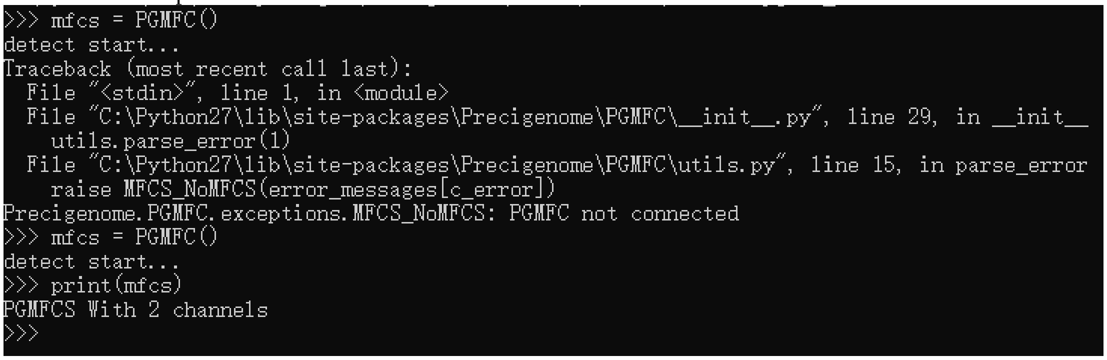
The PGMFC class

The PGMFC class is the main entry point for the SDK. A PGMFC object is initialized with the serial number of a connected PG-MFC-Light and acts as a proxy for the instrument inside the code. The serial number is printed under the instrument. An exception will be raised if the specified instrument is not connected. If the PGMFC class is initialized without an argument, or with 0 as the serial number, it will initialize the first PGMFC it finds. In that case, an exception will be raised if no PGMFC is connected.

```
from Precigenome.PGMFC import PGMFC

mfcs = PGMFC()
print(mfcs)
# start reading sensor data at intervals of 100ms
mfcs.monitor_start(span=100)
mfcs.set_params(channel=1, peak=2.3, runtime=50)
mfcs.purge_on(1)
print (mfc.get_pressure(1))
```

No PGMFC exception.



```
>>> mfcs = PGMFC()
detect start...
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\Python27\lib\site-packages\Precigenome\PGMFC\__init__.py", line 29, in __init__
    utils.parse_error(1)
  File "C:\Python27\lib\site-packages\Precigenome\PGMFC\utils.py", line 15, in parse_error
    raise MFCS_NoMFCS(error_messages[c_error])
Precigenome.PGMFC.exceptions.MFCS_NoMFCS: PGMFC not connected
>>> mfcs = PGMFC()
detect start...
>>> print(mfcs)
PGMFCS With 2 channels
>>>
```

The Channel Class

Although the MFCS class allows you to use all of the functionalities available in the device, an additional Channel class is provide for convenience. It allows the user to specify the channel number only once and obtain a reference to the corresponding PGMFC Channel, which can then be used in the rest of the code.

To obtain a Channel reference, simply index the PGMFC object with the desired channel number. Note that the port number is the same as displayed on the instrument's front panel, i.e., it starts at 1.

```
from Precigenome.PGMFC import PGMFC

mfcs = PGMFC()
# Get a reference to the first channel of the device
channel = mfcs[1]
print(channel)
channel.set_params(peak=2.3, runtime=50)
channel.purge_on(1)
print (channel.get_pressure())
```

Additional help

To view the documentation of the method and attributes of each class in the package, import the package in Python and use the help() command:

```
from Precigenome.PGMFC import PGMFC

# help on the PGMFC package
help(PGMFC)
```