

Notice : version mars 2017
Rédaction du document : Benoit CAPITAINE
Contact mail : benoit.capitaine@ens-lyon.fr

Procédure d'installation pour l'acquisition sous Python carte N.I.

1/Téléchargement de **NIDAQmx** support pour les cartes d'acquisition.
<http://www.ni.com/fr-fr/support.html>, il contient également : NI Measurement & Automation qui permet de détecter les cartes NI. V16.0.1 mais la 9.0 convient également pour nos vieille carte....

Drivers de matériel NI | [Tout afficher](#)

[NI-DAQmx 8.3](#)
[NI-DAQmx 14.1](#)
[NI-DAQmx 16.0.1](#)
[NI-DAQmx 9.0](#)
[NI-DAQmx 15.5](#)

NI-DAQmx 16.0.1

5 Ratings | 4.00 out of 5 | [Print](#)

Available Downloads:

Download Options:

NI Recommended

1. NI Downloader: [NIDAQ16010_downloader.exe](#) (1867.73 MB)
Checksum (MD5): fa5423899e72eabd5a5a4610dct1a85d3
Using the NI Downloader.

Tout afficher sur le site ni.com

Téléchargement de janvier 2017

2/Téléchargement de **Python XY**

Python(x,y) est un logiciel libre scientifique de calcul numérique basé sur le langage Python, les interfaces graphiques Qt (et le cadre de développement associé) et l'environnement de développement scientifique interactif Spyder. V 2.7.9.0

[Non testé : pour l'utilisation des connecteurs GPIB/USB avec les GBF Agilent il faut également l'architecture logicielle NI-488.2 pour le développement d'applications GPIB.
Et peut- être aussi l'architecture pour instrument virtuel (VISA) norme pour la configuration, la programmation et le dépannage des systèmes d'instrumentation comprenant les interfaces GPIB, VXI, PXI, série (RS232 / RS485), Ethernet / LXI et / ou USB.]

Installation de la PyDAQmx

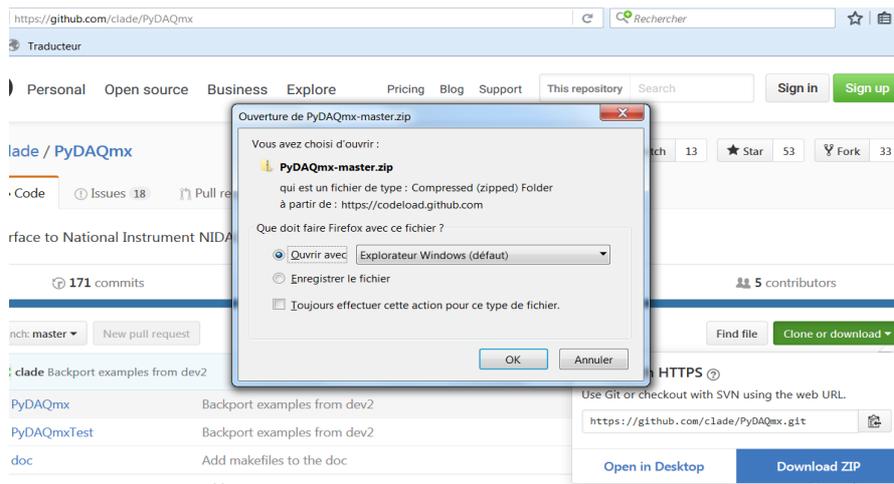
Déjà installé au département de physique (en administrateur)

PC 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14

Portables 2, 3, 4, 5, 13, 14

Procédure d'installation **Julien Salord** :

On télécharge le premier dossier /PyDAQmx sur le site <https://github.com/clade/PyDAQmx>
(Voir image ci-dessous)



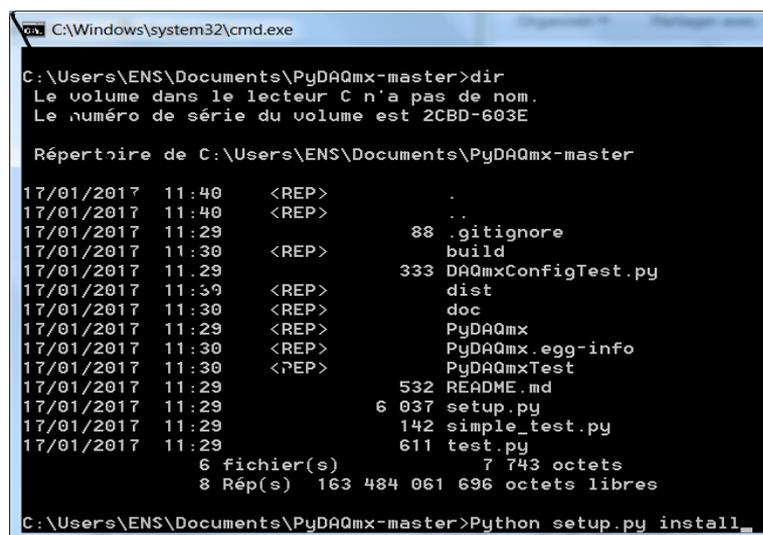
github pour PyDAQmx

Ensuite ouvrir une console (tapez **cmd** dans le champ « rechercher les programmes et fichiers » ou WinE cmd).

On se place dans le répertoire /PyDAQmx-master

On exécute : **Python setup.py install.**

Le dossier doit être dézippé, l'utilisation des commandes Dos `dir` `cd..` peuvent être utiles.



Python setup.py install

(Pensez à sortir du répertoire /PyDAQmx_master `cd..` pour la suite des opérations)

Utilisation de TortoiseHG pour le dépôt devlgi/Fluidlab

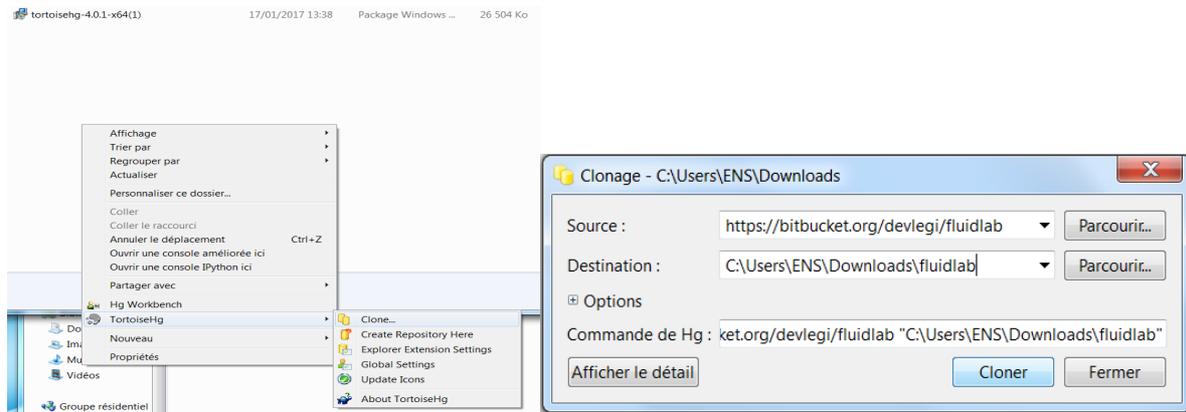


Download TortoiseHG site bitbucket.org

Tortoise HG et Mercurial sont des gestionnaires de versions distribués.

Dans notre cas l'installation de Tortoise HG va nous permettre de télécharger le dépôt situé sur le site <https://bitbucket.org/devlegi/fluidlab> . Une fois Tortoise installé, dans n'importe quel dossier, le

le clic droit contient Tortoise HG/Clone/ il faut alors compléter la source et la destination dans la fenêtre qui apparaît. (voir les images suivantes).



Lorsque le clonage est terminé, même opération pour l'installation des données du dépôt, il faut exécuter : **python setup.py install** du dossier /fluidlab.

```
C:\Windows\system32\cmd.exe
Searching for pyserial>=3.2.1
Best match: pyserial 3.2.1
Processing pyserial-3.2.1-py2.7.egg
pyserial 3.2.1 is already the active version in easy-install.pth
Installing miniterm.py script to C:\Python27\Scripts

Using c:\python27\lib\site-packages\pyserial-3.2.1-py2.7.egg
Searching for psutil>=0.6.1
Best match: psutil 0.6.1
Adding psutil 0.6.1 to easy-install.pth file

Using c:\python27\lib\site-packages
Searching for matplotlib>=1.1.1
Best match: matplotlib 1.1.1
Adding matplotlib 1.1.1 to easy-install.pth file

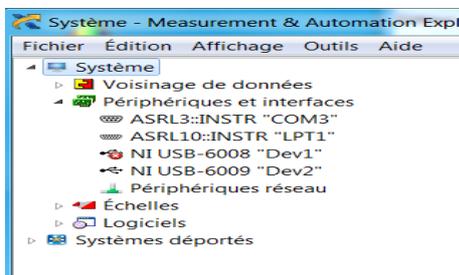
Using c:\python27\lib\site-packages
Searching for numpy>=1.6.2
Best match: numpy 1.6.2
Adding numpy 1.6.2 to easy-install.pth file

Using c:\python27\lib\site-packages
Finished processing dependencies for fluidlab==0.0.2a0
C:\Users\ENS\Documents\fluidlab>python setup.py install
```

python setup.py install

Acquisition des données : fichier EssaiDAQ.py

Pour Tester notre programme dans un premier temps on détecte la présence de la carte grâce à : NI Measurement & Automation (NiMAX) qui nous affiche également l'adresse de communication.



ex ici on relève l'adresse « Dev2 » pour la carte NIUSB6009

Ouvrir le fichier **EssaiDAQ.py** et modifier les paramètres en fonction de votre acquisition :

```
data, = read_analog('Dev2/ai0', terminal_config='Diff', volt_min=-10, volt_max=10,
samples_per_chan=N, sample_rate=sampling, coupling_types='DC')
```

Exemple avec l'interface USB-6008 12Bits 10KS/s

```
read_analog('Dev1/ai0', terminal_config='Diff', volt_min=-10, volt_max=10,  
samples_per_chan=N, sample_rate=sampling, coupling_types='DC')
```

Adresse de la carte : 'Dev1/ai0'

Configuration en mode différentiel : 'Diff'

Terminal config='Diff' ou 'RSE' (Mesures différentielles ou référencées ...)

$volt_min=-10, volt_max=10$

Quantification : permet de calculer $p=plage\ de\ mesure / 2^n$

Pour une plage de mesure de 10V à -10V soit 20V, nombre de bits 12

$P= 20/2^{12} = 4.88mV$

Nombre de point : $N = 1000$ (pas testé la valeur limite ...)

Fréquence d'acquisition sampling (Kilo second) : $10e3$ durée entre chaque point 100ks

Fréquence max échantillonnage de la carte : 10 kÉch/s

Pour $N = 1000$ Sampling = $10e2$ on a 1s

Pour $N = 10\ 000$ Sampling = $10e3$ on a 1s

Pour $N = 100\ 000$ Sampling = $10e4$ on a 1s

Pour $N = 1\ 000\ 000$ Sampling = $10e5$ on a 1s

Pour $N = 1000$ Sampling = $10e3$ on a 0.1s

Pour $N = 1000$ Sampling = $2.10e2$ on a 5s

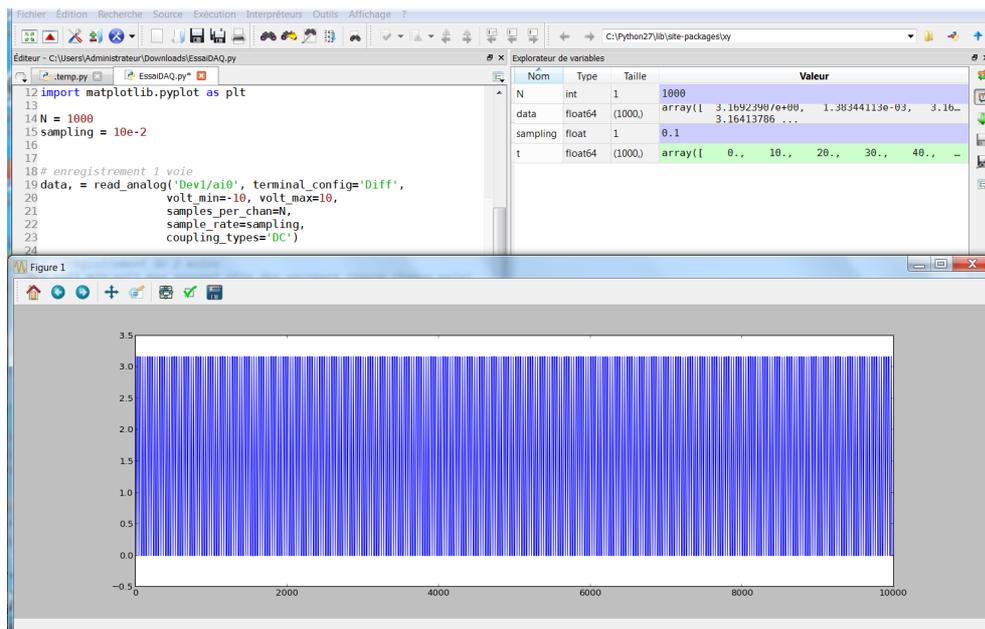
Pour $N = 1000$ Sampling = $10e1$ on a 10s

Pour $N = 1000$ Sampling = $10e0$ on a 100s

Pour $N = 1000$ Sampling = $10e-1$ on a 1000s (16min)

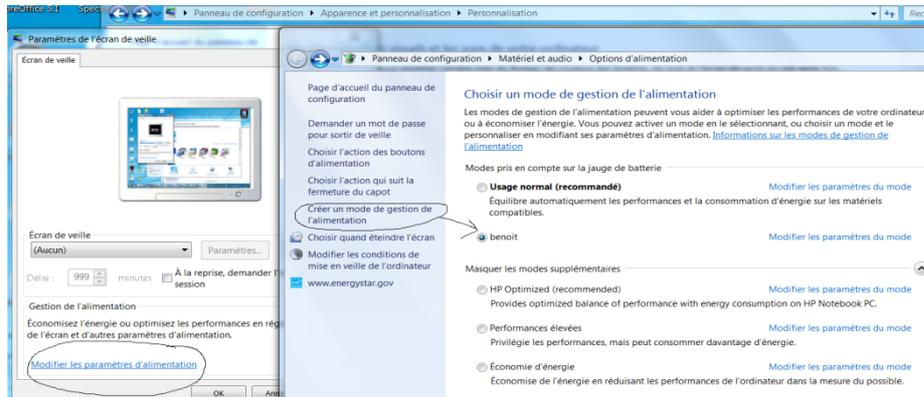
Pour $N = 1000$ Sampling = $10e-2$ on a 10000s (3h) ça marche testé le 28 mars 2017

Pour $N = 5000$ Sampling = $10e-2$ on a 50000s (14h d'enregistrement) testé le 29 mars 2017



Pour ce genre d'enregistrement (3h), il faut désactiver les modes de gestion de l'alimentation et de veille de l'ordinateur.

Sous windows7 / Panneau de configuration / Personnalisation



Modifier les paramètres du mode : benoit

Choisissez les paramètres de mise en veille et d'affichage de votre ordinateur.

	 Sur la batterie	 Sur secteur
Estomper l'affichage :	<input type="text" value="5 minutes"/>	<input type="text" value="10 minutes"/>
Éteindre l'écran :	<input type="text" value="15 minutes"/>	<input type="text" value="15 minutes"/>
Mettre l'ordinateur en veille :	<input type="text" value="Jamais"/>	<input type="text" value="Jamais"/>
Régler la luminosité du plan :	<input type="range" value="50"/>	<input type="range" value="50"/>

Fichier *EssaiDAQ.py*

Codage Python :

```
#On importe les fonctions de la bibliothèque fluidlab DAQmx avec  
from fluidlab.instruments.daq.daqmx import read_analog
```

```
#Les bibliothèque python sciences avec  
import numpy as np  
import matplotlib.pyplot as plt
```

```
# enregistrement 1 voie
```

```
data, = read_analog('Dev2/ai0', terminal_config='Diff',  
                  volt_min=-10, volt_max=10,  
                  samples_per_chan=N,  
                  sample_rate=sampling,  
                  coupling_types='DC')
```

```
# enregistrement de 2 voies
```

```
# volt_min volt_max peuvent etre des vecreurs (poure chaque voie)  
dataA, dataB = read_analog(('Dev2/ai0', 'Dev2/ai1'),  
                           terminal_config='Diff',  
                           volt_min=-10, volt_max=10,  
                           samples_per_chan=N,  
                           sample_rate=sampling,
```

```
coupling_types='DC')
```

```
# affichage des voies dataA, dataB avec base de temps
```

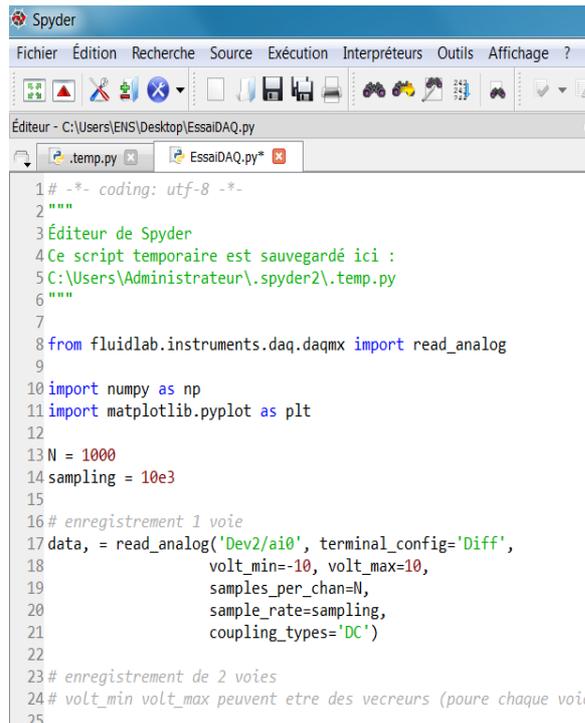
```
plt.figure()
```

```
plt.clf()
```

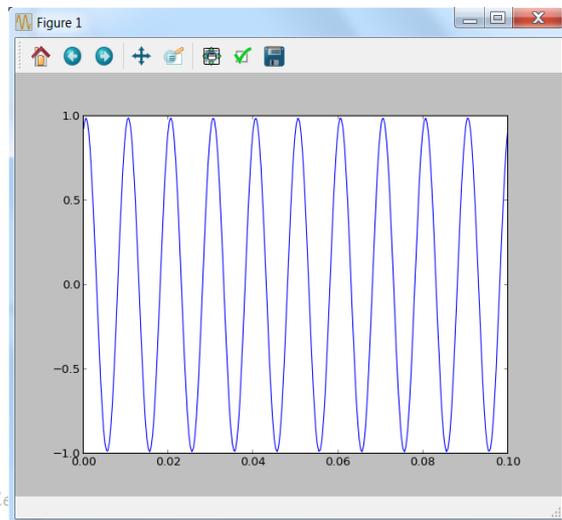
```
t = np.arange(N)/sampling
```

```
plt.plot(t, dataA)
```

```
plt.show()
```



```
1# -*- coding: utf-8 -*-
2"""
3 Éditeur de Spyder
4 Ce script temporaire est sauvegardé ici :
5 C:\Users\Administrateur\.spyder2\.temp.py
6 """
7
8 from fluidlab.instruments.daq.daqmx import read_analog
9
10 import numpy as np
11 import matplotlib.pyplot as plt
12
13 N = 1000
14 sampling = 10e3
15
16 # enregistrement 1 voie
17 data, = read_analog('Dev2/ai0', terminal_config='Diff',
18                   volt_min=-10, volt_max=10,
19                   samples_per_chan=N,
20                   sample_rate=sampling,
21                   coupling_types='DC')
22
23 # enregistrement de 2 voies
24 # volt_min volt_max peuvent être des vecteurs (pour chaque voie)
25
```



```
# essai commande GBF Agilent
```

```
from fluidlab.instruments.daq.daqmx import read_analog
```

```
from fluidlab.instruments.funcgen.agilent_33220a import Agilent33220a
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
gbf = Agilent33220a['GPI::10']
```

```
gbf.frequency.set(500)
```

```
gbf.vrms.set(2.0)
```

Documentations et responsable du TP: julien-Salord et stephane-Roux