

## Utiliser Python pour contrôler son Arduino

Il vous faut suivre la procédure suivante (pour Pyzo, même version que celle du lycée) :

### **1) INSTALLATION TimerOne-r11.zip pour Arduino :**

Dans le logiciel *Arduino*, cliquer sur *Croquis/Inclure une bibliothèque / Ajouter la bibliothèque Zip*. Choisir alors la bibliothèque *TimerOne-r11.zip* pour l'installer.

### **2) INSTALLATION py2duino.py pour Python :**

Placer le programme *py2duino.py* dans le répertoire *Lib* de votre distribution *Pyzo*.

### **3) INSTALLATION pyserial :**

Voir la vidéo <https://www.youtube.com/watch?v=Pf-cGzOQmXU> très bien faite.

## Utiliser le fichier Pyserial **3.4**

### Autre solution :

Dans le répertoire contenant l'exécutable "python.exe", pour installer la version de pyserial, il faut créer un fichier .bat et copier les deux commandes suivantes :

```
python -m pip install pyserial==2.7  
pause
```

Après avoir enregistré le fichier, il faut lancer l'installation en cliquant sur le fichier .bat. Une invite de commande s'ouvre et le script s'exécute.

Remarque : Pour installer un package python, il est recommandé d'utiliser pip. C'est un module qui télécharge et installe le package souhaité. Il est présent par défaut dans la librairie Python.

### **4) Téléverser *toolbox\_arduino\_v4.ino* dans l'Arduino. Il n'y aura plus rien à téléverser dans Arduino, tout se fera ensuite sous Python 😊**

Je vous mets également le site des collègues de SII qui ont fait ce super programme Arduino/Python, avec les explications pour les fonctions qui sont directement codées (servomoteur, encodeur, moteurCC, ...)

[https://github.com/xpessoles/Cycle\\_01\\_DecouverteSII/blob/master/Chaine\\_Fonctionnelle/02\\_Fonction\\_Traiter/TP\\_Traiter\\_Python\\_Arduino/Librairie%20py2duino%20v4/py2duino.py](https://github.com/xpessoles/Cycle_01_DecouverteSII/blob/master/Chaine_Fonctionnelle/02_Fonction_Traiter/TP_Traiter_Python_Arduino/Librairie%20py2duino%20v4/py2duino.py)

## Exemple d'utilisation

Un exemple également de code python (**MESURES\_SERVO.py**) utilisant une technique pour tracer en temps réel les courbes. Il s'agit d'un code pour mesurer les angles de rotation d'un servomoteur. Il y a différentes parties.

- Au début on initialise les variables et les pins de connexions sur la carte Arduino.
- Une fonction « attente » permet d'attendre que l'utilisateur installe le servo-moteur.
- La fonction « graphique » permet de tracer **EN TEMPS REEL** ( !!! ) la courbe sous python.
- La fonction « donnee » permet d'enregistrer les données reçues de l'Arduino, de les stocker, ici dans un fichier sur l'ordinateur. Il est également possible dans cette fonction de donner des ordres tels qu'une commande en rotation du servo-moteur par exemple.

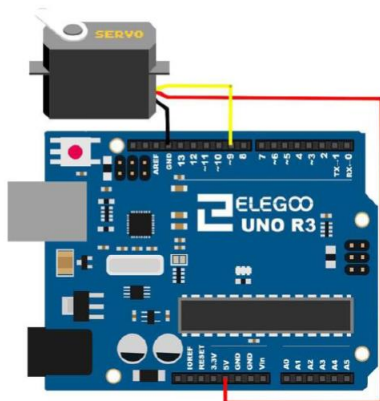
Puis les dernières lignes avec le THREAD, lancent toutes ces fonctions automatiquement.

Un exemple de traitement des données obtenues est également donné (**Traitement-Mesures\_SERVO.py**). Il permet de tracer (à partir des mesures réalisées) la position, la vitesse instantanée et la vitesse moyenne du servomoteur.

## Montage du servomoteur sur la carte Arduino :

### Montage servomoteur

Un **servomoteur** (souvent abrégé en « servo ») est un **moteur électrique intégré dans une boucle d'asservissement en position**, il s'agit donc d'un système asservi. Sur ce type de moteur, la position est définie avec une limite de débattement d'angle de 180°.



- Fil rouge du servomoteur sur la broche **5V**.
- Fil noir sur la broche **GND**.
- Fil jaune (ou blanc en fonction des constructeurs) sur la broche **D9**.