

Un microcontrôleur comme stroboscope

Fiche étudiant

Christophe Caire

20 juin 2019

I Commande du microcontrôleur par usage du port série

Pour jouer à ces jeux, il est nécessaire que votre distribution python soit équipée d'une librairie de gestion du port série, typiquement il s'agit de **pyserial**.

A installer par votre méthode préférée : pip, conda, anaconda navigator.

En cas de doute la documentation en ligne de pyserial se consulte à l'adresse suivante :

<https://pyserial.readthedocs.io/en/latest/index.html>

```
In [ ]: import serial # gestion du port série
import time # module de gestion du temps
import matplotlib.pyplot as plt # module graphique
import numpy as np # module de calcul scientifique

# permet d'afficher les graphiques directement dans le notebook
%matplotlib inline
```

En cas de doute sur le port ouvert, il est aisé d'en avoir confirmation avec la commande python ci-après.

Rappel : dans un environnement Ipython, le "!" est un moyen de lancer des commandes systèmes.

```
In [ ]: portId=!python -m serial.tools.list_ports
print(portId)
```

Il est alors aisé d'ouvrir le port et de gérer la communication.

Important :

Veillez à ce que le code Arduino C et celui de votre script python soit réglé sur les mêmes débits attendus au niveau du port série (Baud Rates).

```

In [ ]: #connexion du PC au port série
        # Supported baud rates 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 31250, 38400
        serial_port = serial.Serial( port = portId[0] , baudrate =115200) # Ouverture du port
        serial_port.setDTR(False) # Initialisation de la carte Arduino
        time.sleep(1) # temps de stabilisation - temporisation
        serial_port.setDTR(True) # Activation de la carte
        serial_port.reset_input_buffer() # Vide le tampon de stockage existant

        # les mesures
        N=2000 # Nombre d'acquisitions
        mesure = np.zeros(N) # Tableau d'acquisition des valeurs de mesure
        temps = np.empty_like(mesure) # Tableau d'acquisition des instants de mesure
        try:
            for i in range(N) :
                val = serial_port.readline().split() # ligne une ligne avec EOL + partition
                temps[i]= int(val[1])
                mesure[i]=int(val[0])
        except:
            print("defaut de fonctionnement")
            pass

        # fermeture du port série
        serial_port.close()

```

Un ligne d'interprétation pour vérifier le processus en cas de dysfonctionnement.

Si le code plante en cours de saisie, pensez à fermer le port pour contrôler de nouveau l'Arduino :

```
serial_port.close()
```

Normalement le try & except est fait pour cela mais...

```
In [ ]: val
```

Si tout s'est bien passé, *temps* et *mesure* sont des tableaux 1D contenant vos données.

Il suffit maintenant de procéder à l'affichage des résultats

```

In [ ]: # On peut éviter les effets de bord en éliminant
        # les valeurs de début et de fin de transmission
        plt.plot ( temps, mesure)
        plt.xlabel ("Temps (s)")
        plt.ylabel ("Intensité")
        plt.grid ( )
        plt.show( )

```

II Sauvegarde des données acquises

Si votre acquisition vous Satisfait, il peut être souhaitable de sauvegarder les données dans un fichier, de façon à les récupérer selon vos envies.

Nous utiliserons à cette fin le module **pickle**.

Il permet l'enregistrement, dans un fichier de type **file**, de vos données au format binaire. Ici nous préserverons *temps* et *mesure*.

```
In [ ]: import pickle

        fname='donnees_test.pickle' # nom du fichier - libre de choix

In [ ]: # Ouverture du fichier binaire et sauvegarde
        with open(fname,'wb') as ftemp:
            pickle.dump(temps,ftemp,pickle.HIGHEST_PROTOCOL)
            pickle.dump(mesure,ftemp,pickle.HIGHEST_PROTOCOL)
```

La structure *with ... as* permet d'ouvrir et de fermer le fichier. Si vous voulez recharger vos données, il suffira maintenant de valider la cellule suivante :

```
In [ ]: with open(fname,'rb') as ftemp:
        temps=pickle.load(ftemp)
        mesure=pickle.load(ftemp)
```

III Traitement des données

A vous d'imaginer un moyen de mesurer la période de ce signal.

Vos enregistrements n'apparaîtront probablement pas périodiques en raison des défauts de vos mesures (vibration du portable, éclairage périphérique). Il faudra essayer d'imposer le caractère périodique sur la base d'un critère.

Nous attendons vos suggestions.

.....