PROGRAMMATION DE LA SORTIE BINAIRE D'ORPHY GTS EN PYTHON

1 – Logiciels à installer

Sur http://www.python.org/.

1.1 – Installation sous Windows

Il vous faudra installer les fichiers (dans l'ordre) : python-2.4.2.msi puis pywin32-207.win32-py2.4.exe et enfin pyserial-2.2.win32.exe (ce sont les versions que j'ai installé).

1.2 – Installation sous Linux

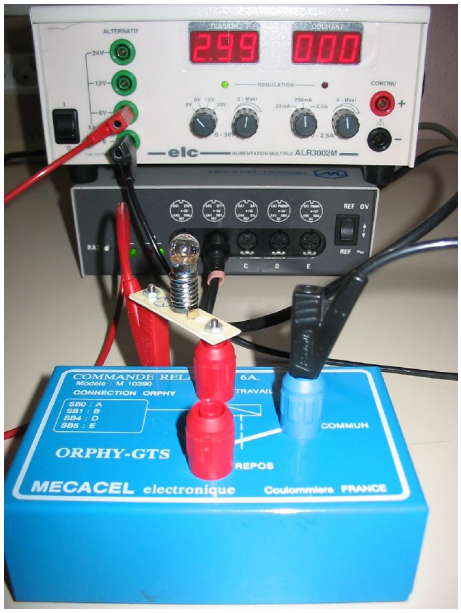
En général, sous Linux, Python est installé en « standard » sinon il faut télécharger python-2.4.2.tgz et pyserial-2.2.zip. L'installation se fait « comme d'habitude sous Linux ». Avec Mandriva Linux (ce que j'utilise), on peut trouver les fichiers RPM pour l'installation.

1.3 – Installation sous Mac

Il faut télécharger macpython sur http://www.python.org/.
Mais le module pySerial (à ma connaissance) n'est pas transposé sur Macinstosh.

2- Le montage

L'ampoule est reliée au générateur de tension variable et sur la borne travail de la commande relais qui est reliée à la voie A ou B d'Orphy-Gts placée en Réf 0 V.



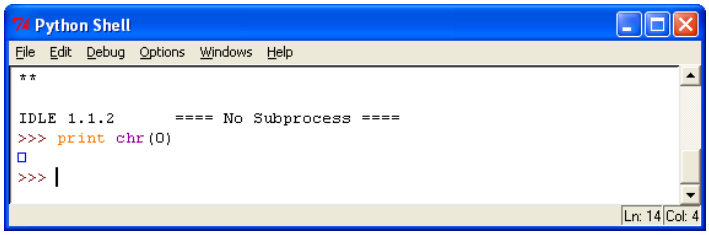
3 – Le programme

1) Description du programme :

1) Il faut importer les modules Tkinter pour pouvoir réaliser des fenêtres graphiques et Serial pour pouvoir communiquer avec la voie série.
2) Les fonctions à définir :

- ouvrirserie : ouverture du port série
- allume : mise à 1 de la sortie binaire
- éteint : mise à 0 de la sortie binaire
- quitte : quitter le programme et SURTOUT fermeture de la voie série (IL FAUDRA DONC UTILISER LE BOUTON QUITTER POUR FERMER LA FENETRE et NON PAS LA « CROIX » DE LA FENETRE SINON LE PORT SERIE NE SERA PAS FERME ET LE SHELL INDIQUERA ALORS UNE ERREUR.

3) Ensuite on crée les modules graphiques : fen1 est le nom de la fenêtre utilisée, bou0, bou1 et bou2 sont les bouttons créés.
4) Remarque : pour obtenir le caractère □ dans le programme, faire ceci : print chr(0) dans le shell et faire un « copier-coller » dans l'éditeur Python



2) Copie d'écran du programme :

sortie binaire lampe2.py - G:\MPI\CommandeAmpoule\sortie binaire lampe2.py

File Edit Format Run Options Windows Help

```
# -*- coding:Latin-1 -*-
from Tkinter import *
import serial
def ouvrirserie():          #fonction qui définit l'ouverture du port série
    global se               #se en variable globale (déclarée pour tout le programme)
    se = serial.Serial()    #se variable du port série
    se.baudrate = 9600      #vitesse de transmission
    se.bytesize=8           #nombre de bits transmis
    se.parities=0           #pas de parité
    se.stopbits=1           #bit de stop
    se.xonxoff=0            #pas de contrôle logiciel
    se.rtscts=0             #pas de contrôle RTS/CTS
    se.timeout=1            #timeout (important)
    se.port = 0             #le port 0 est ici COM1, COM2 serait le port 1
    se.open()               #ouvre la voie série sélectionnée
def allume():               #fonction pour l'allumage
    se.write("!")           #on met la sortie binaire 1 (ou 0) à 1
def eteint():               #fonction pour l'extinction
    se.write("□")           #on met la sortie binaire 1 (ou 0) à 0
def quitter():              #fonction pour fermer le port série et quitter
    se.close()
    fen1.quit()
fen1 = Tk()                 #appel d'une fenêtre graphique nommée fen1
ouvrirserie()               #ouverture du port série
fen1.title("Commande ampoule")                                  #titre de la fenêtre
tex0=Label(fen1, text="Choisissez un bouton", font=("Arial",12))    #création d'un texte
tex0.pack()
bou0=Button(fen1, text="Allumer", command=allume)               #création d'un bouton
bou0.pack()
bou1=Button (fen1, text="Eteindre", command=eteint)
bou1.pack()
bou2 = Button(fen1, text='Quitter', command=quitter)
bou2.pack()
fen1.mainloop()                                                 #activation de la fenêtre
fen1.destroy()
```

Ln: 38 Col: 0

3) Fonctionnement du programme

Appuyer sur F5 pour lancer le programme.
Ce programme n'a été testé que sur Orphy-GTS.