Paramétrages du Raspberry pour la communication avec AX-12A

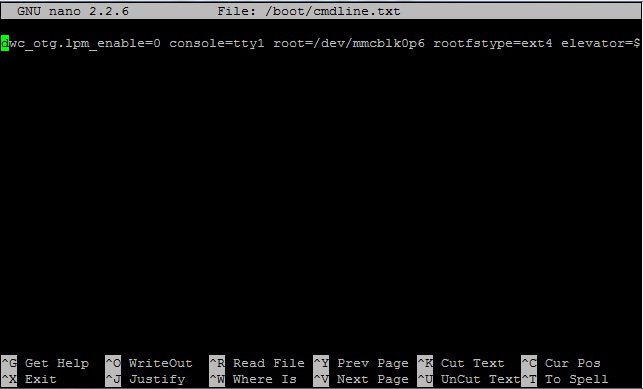
# Configuration de base

Ces paramètres sont utilisés sur un Raspberry Pi 3 avec Raspbian Jessie de Mars 2017.  
Le problème de base est le fait que sur cette version, l’UART est utilisé pour le Bluetooth à la base, il faut donc désactiver le Bluetooth pour pouvoir l’utiliser

# Désactiver le Bluetooth

Il faut désactiver le Bluetooth pour libérer l’UART. Et ensuite il faudra activer l’UART.  
Pour cela, il faut modifier le fichier de config qui se trouve dans le dossier boot :  
sudo nano /boot/config.txt  
et ajouter à la fin ces deux lignes :  
dtoverlay=pi3-disable-bt # Desactive le Bluetooth  
enable\_uart=1 # Active l’UART  
***Sous nano, pour sauvegarder et quitter faites dans l’ordre : Ctrl + O puis Ctrl + X***

Ensuite il faut encore supprimer une ligne dans le fichier cmdline.txt :  
sudo nano /boot/cmdline.txt  
et supprimer les parties contenant  « ttyAMA0 » comme :  
console=ttyAMA0,115200 ou kgdboc=ttyAMA0,115200  
Votre fichier devrait ensuite ressembler à cela :



Vous pouvez redémarrer le Raspberry avec : sudo reboot

# Tester l’UART

Il faudrait ensuite tester l’UART. Pour cela nous allons relier la pin TX à la pin RX du Raspberry pour qu’il communique tout seul. Nous allons ensuite installer minicom pour tester la communication :  
sudo apt-get install minicom

Lancer le programme :  
**!LE PROGRAMME DOIT ETRE LANCER DEPUIS LE RASPBERRY ET NON EN SSH !**  
minicom –b 115200 –o –D /dev/ttyAMA0

Vous pouvez ensuite écrire sur votre clavier. Si les touches écrites apparaissent à l’écran, tout est Ok. Sinon vérifiez bien que la config est en ordre.

Il est possible que cette méthode ne fonctionne pas sur les versions futures de Raspbian

# Installer Kivy

Pour installer Kivy, la méthode décrite sur le site web fonctionne parfaitement, elle contient même la manière de paramétrer l’écran tactile :

<https://kivy.org/docs/installation/installation-rpi.html>

# Lancer le code

Pour lancer le code, il suffit d’aller dans le dossier du code avec la commande suivant :

cd chemin/du/code

puis d’effectuer cette action :

python main.py

# Lancer le code au démarrage

Pour lancer le code au démarrage il faut éditer un fichier spécifique :

sudo nano /etc/rc.local

cela va ouvrir le fichier, dans ce fichier ajouter la commande suivant :

/user/bin/python /chemin/du/code/main.py

Sauvegardez en faisant ctrl+o, puis quittez avec ctrl+x

Et cela devrait lancer le programme au démarrage du Raspberry

# Sources

<https://electrosome.com/uart-raspberry-pi-python/>

<https://raspberrypi.stackexchange.com/questions/45570/how-do-i-make-serial-work-on-the-raspberry-pi3>

<http://www.framboise314.fr/le-port-serie-du-raspberry-pi-3-pas-simple/>

<https://kivy.org/docs/installation/installation-rpi.html>