

Configuration du port série (fonction optionnelle)

Fichier `"/etc/inittab"`

Tout d'abord, il faut dévalider le mode console du raspberry pi. En effet, par défaut le port série peut être utilisé pour si connecté avec un simple terminal (pour les français 😊 tel un minitel).

- Pour un Raspbian Wheezy (version 7) édition le fichier le fichier avec la commande:

```
sudo nano /etc/inittab
```

```
T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

doit être mise en commentaire en ajoutant un dièse (#) en début de ligne

```
#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
```

* Pour un Raspbian Jessie (version 8) (/etc/inittab n'existant plus), vous lancer la commande pour que le service de console serie ne se lance plus au démarrage:

```
sudo systemctl disable serial-getty@ttyAMA0.service
```

Fichier `"/boot/cmdline.txt"`

Lorsque le Raspberry Pi démarre, il envoie des informations de démarrage au port série. Il est préférable de désactiver ce fonctionnement pour que l'équipement qui sera branché sur le port série ne réceptionne pas ces lignes d'information. Vous pouvez désactiver ce fonctionnement en éditant le fichier `/boot/cmdline.txt`

```
sudo nano /boot/cmdline.txt
```

Le contenu du fichier est celui-ci par défaut

```
dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200  
console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline  
dwc_otg.speed=1 rootwait
```

Enlever la partie faisant référence au port série `ttyAMA0`. Cela doit alors ressembler à ceci

```
dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4
```

```
elevator=deadline dwc_otg.speed=1 rootwait
```

Installation de la bibliotheque pyserial

```
sudo apt-get install python-pip  
sudo pip install pyserial
```

Tests de fonctionnements

⚠ Sur la carte SVXLCARD version 1.0, la sérigraphie est incorrecte pour les broches RS232.

- La pin TXD est en réalité RXD
- La pin RXD est en réalité TXD

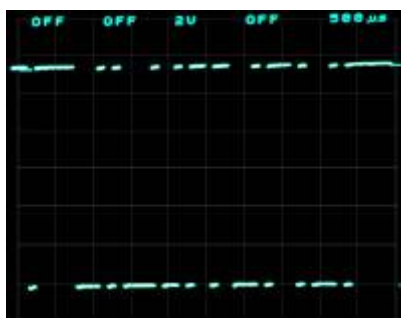
Le code Python suivant va permettre de vérifier à l'oscilloscope que les signaux sont bien générés.

```
#!/usr/bin/python  
import serial  
import RPi.GPIO as GPIO  
import os, time  
  
GPIO.setmode(GPIO.BOARD)  
port = serial.Serial("/dev/ttyAMA0", baudrate=9600, timeout=0.5)  
  
while True:  
    port.write('Un test !')  
    rcv = port.read(10)  
    print rcv
```

Lancer l'exécution du code avec :

```
sudo python test-serial.py
```

L'oscillogramme (mesure entre GND et TXD) ressemble à cela.

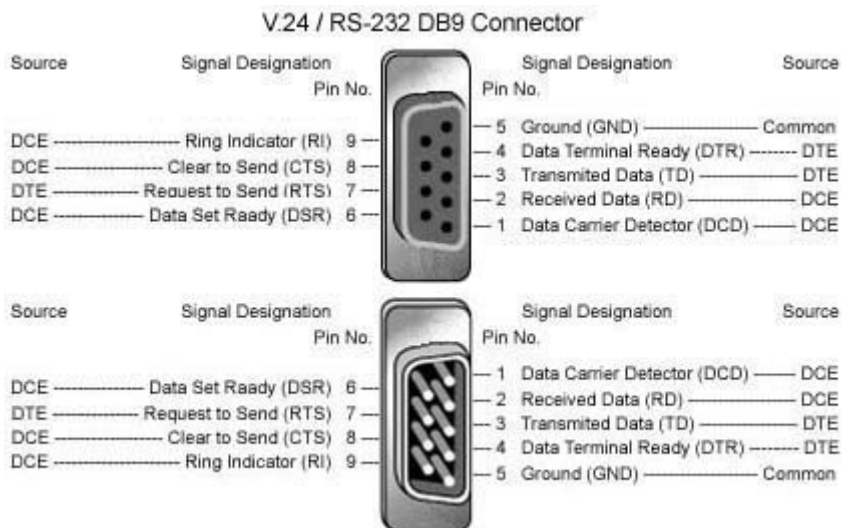


Pour vérifier la fonction de réception et émission en même temps, relier les 2 pins RXD et RXD

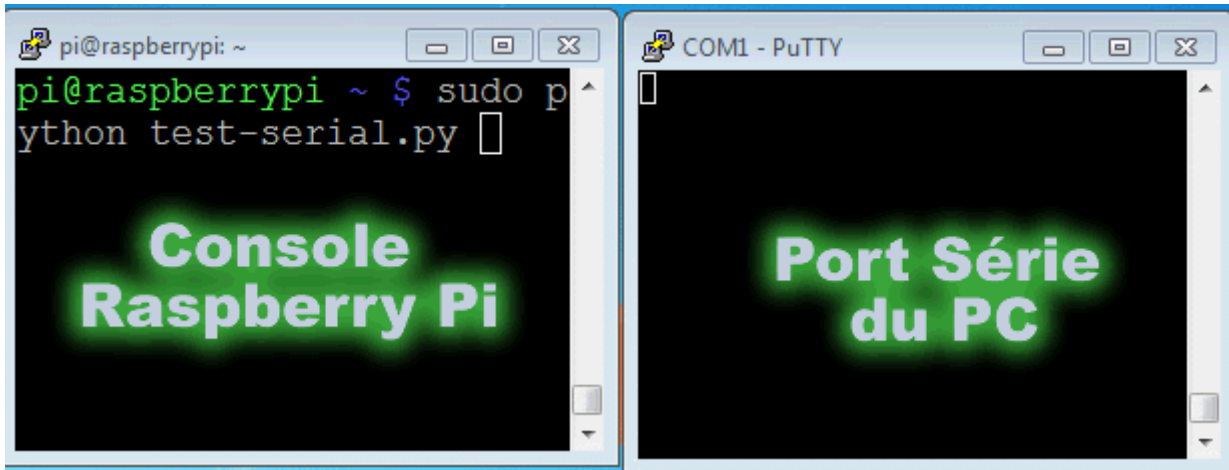
ensemble. Avec le même code python, vous allez voir apparaître à l'écran l'écho (données arrivant sur RXD) de ce qui est émis (par TXD)



Vous pouvez également relier directement le port série de la carte au port série d'un PC. Le brochage d'un port 9 points DB9 est le suivant.



Brancher le RXD sur le TX du PC (broche 3) et le TXD sur le RX du PC (broche 2). Egalement GND sur la broche 7 du PC. Vous voyez ici les échanges bi-directionnels entre la carte SVXCard et le PC (port série COM1)



From:
<http://svxcard.f5uii.net/> - **SVXLink Card**

Permanent link:
http://svxcard.f5uii.net/doku.php?id=fr:configuration_port_serial&rev=1453585129

Last update: **2019/10/28 11:50**

