Framboise 314, le Raspberry Pi à la sauce française....

La référence du Raspberry Pi en France – Par l'auteur du livre "Raspberry Pi 3 et Pi Zero" paru aux Edts. ENI

Rechercher

Publié le 13 février 2017 - par François MOCQ

Le port série du Raspberry Pi 3 : pas simple !

On ne l'avait pas vue venir, celle-là ! La Fondation nous l'a glissée en loucedé sur le Raspberry Pi 3 : pas de page d'information sur leur blog, juste des réponses dans les forums...
L'adjonction du Bluetooth au Raspberry Pi 3 a amené les concepteurs de la framboise à détourner l'UART du BCM2837 précédemment relié aux bornes 8 et 10 du GPIO vers le Bluetooth.

Cliquez pour avoir de l'information sur les niveaux.

Au sommaire : [cacher]

Le port Série du Raspberry Pi 3

Les UART du Raspberry Pi

Un **UART**, pour ***Universal Asynchronous Receiver Transmitter***, est un émetteur-récepteur asynchrone universel. En langage courant, c'est le composant utilisé pour faire la liaison entre l'ordinateur et le port série . L'ordinateur envoie les données en parallèle (autant de fils que de bits de données). Il faut donc transformer ces données pour les faire passer à travers une liaison série qui utilise un seul fil pour faire passer tous les bits de données. (Wikipedia)

UART0 = PL011

Le SoC du Raspberry Pi est toujours basé sur le même hardware, le BCM2835. Seul le microprocesseur a évolué. Le BCM2835 comporte deux UART, pour les liaisons série. Le premier, le PL011 est un « vrai » UART :

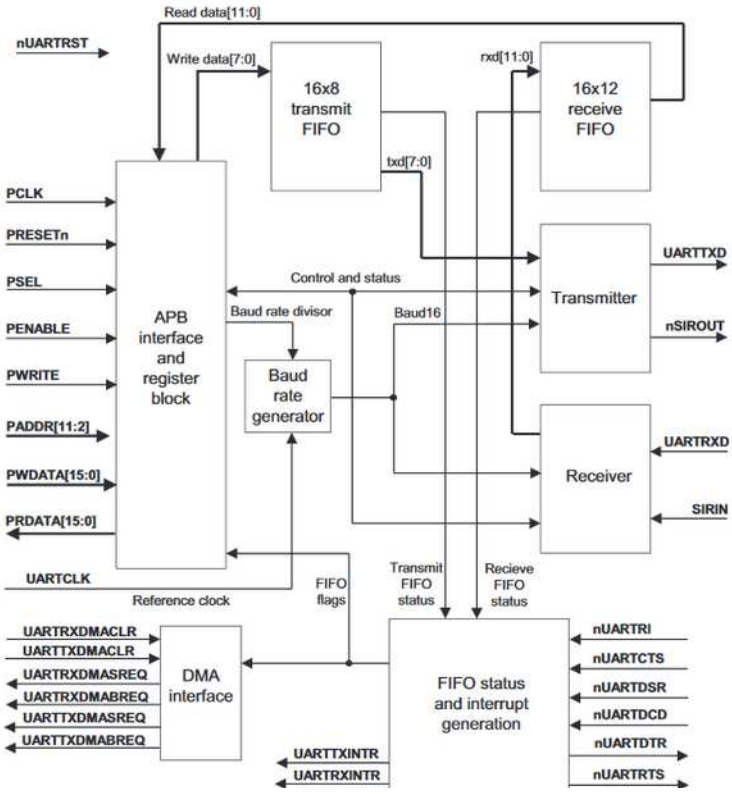


Figure 2-1 UART block diagram

C'est à dire qu'il est autonome, doté de son propre générateur de Baud Rate, et de tous les circuits nécessaires à son fonctionnement.

UART1 = « mini » UART

Le second UART est quand à lui un « mini » UART :

BROADCOM. **BCM2835 ARM Peripherals**

2.2 Mini UART

The mini UART is a secondary low throughput[4] UART intended to be used as a console. It needs to be enabled before it can be used. It is also recommended that the correct GPIO function mode is selected before enabling the mini Uart.

The mini Uart has the following features:

- 7 or 8 bit operation.
- 1 start and 1 stop bit.
- No parities.
- Break generation.
- 8 symbols deep FIFOs for receive and transmit.
- SW controlled RTS, SW readable CTS.
- Auto flow control with programmable FIFO level.
- 16550 *like* registers.
- Baudrate derived from system clock.

This is a *mini* UART and it does NOT have the following capabilities:

- Break detection
- Framing errors detection.
- Parity bit
- Receive Time-out interrupt
- DCD, DSR, DTR or RI signals.

The mini UART uses 8-times oversampling. The Baudrate can be calculated from:

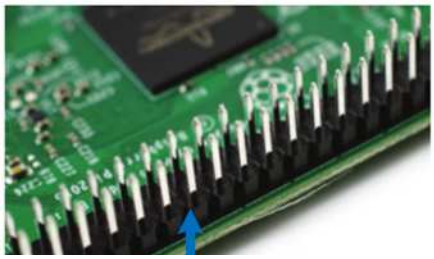
$$baudrate = \frac{system\_clock\_freq}{8*(baudrate\_reg+1)}$$

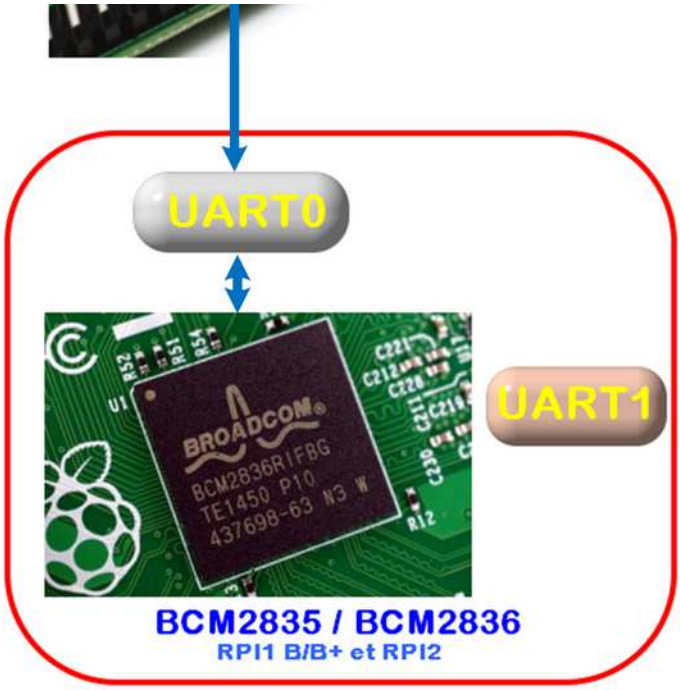
If the system clock is 250 MHz and the baud register is zero the baudrate is 31.25 Mega baud. (25 Mbits/sec or 3.125 Mbytes/sec). The lowest baudrate with a 250 MHz system clock is 476 Baud.

Il ne comporte pas de générateur de Baud Rate et utilise la fréquence du cœur du CPU. Ça pourrait être bien, ***sauf*** que la fréquence du CPU est susceptible de varier en fonction de sa charge 😕

Il ne gère pas non plus la parité.

C'était mieux avant ! Le port série du Raspberry Pi 2





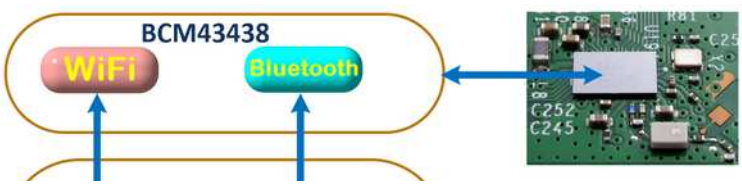
Sur les premières générations de Raspberry Pi (model 1 B, B+ et 2) l'UART0 PL011 est utilisé et il est connecté aux broches 8 et 10 du GPIO. Les messages du système en cours de démarrage sont envoyés par défaut sur ce port. Il suffit de brancher un terminal pour les lire.

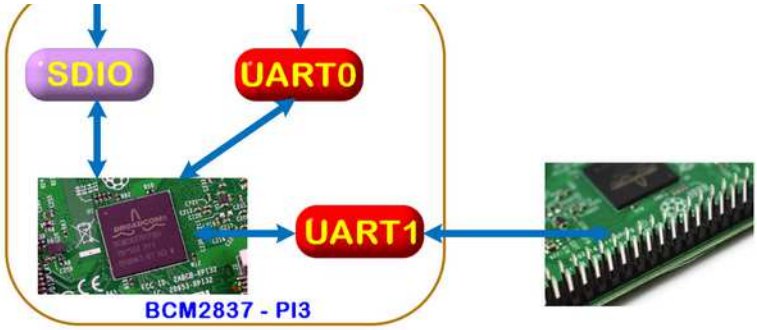
Cette entrée série peut également être connectée à un terminal qui servira alors à se connecter au Raspberry Pi après s'être logué.

Enfin, cette E/S série est utilisée dans des applications industrielles ou domotiques, pour lire des données GPS, relier deux Raspberry Pi entre eux, un Raspberry Pi avec un Arduino etc.

Le port série du Raspberry Pi 3 : la cata !

Le SoC du Raspberry Pi 3 est un BCM2837 SoC. C'est un BCM2836 avec un CPU quad-core ARMv8 qui peut fonctionner en 32 ou en 64 bits. Le mode 32 bits est actuellement sélectionné par défaut la firmware du VideoCore sur le Raspberry Pi 3.





Un autre changement intervient dans l'utilisation des UART. Sur tous les Raspberry Pi précédents, le PL011 était le seul UART en service. Le Raspberry Pi 3 accueille un module Bluetooth qui utilise un UART pour se connecter au SoC. Par défaut, c'est le ***PL011 qui est utilisé pour le Bluetooth*** car il a une pile FIFO plus importante que celle du « mini » UART.

Le ***mini UART est donc relié au port GPIO*** (broches 8 et 10).

Cette ***modification importante*** et non documentée a « cassé » des applications qui tournaient bien avec le port série du Raspberry Pi 2 et qui refusaient de fonctionner sur le Pi3. De nombreux articles de blogs qui fonctionnaient avec les générations précédentes de Raspberry Pi sont devenus inopérants. Les auteurs ne pensent pas forcément à revenir sur ces anciens articles et il faudra être prudent(e) si vous les utilisez.

Un UART pour mon SIGFOX

Dit comme ça ça peut sembler bizarre, mais je vous explique. La SNOC (***S****ociété* ***N****ationale des* ***O****bjets* ***C****onnectés*) située à Saint-Sylvain-d'Anjou a sorti fin janvier une carte de prototypage pour SIGFOX. Cette BRKWS01 distribuée par Yadom, se connecte… sur le port série du Raspberry Pi (ou tout autre port série en 3,3v). Comme je prépare un article sur cette carte j'avais besoin du port série du Raspberry Pi 3.

Voilà, vous avez tout compris.

Pour faire fonctionner dans de bonnes conditions cette carte SIGFOX, je voulais la connecter au port série du GPIO (8 et 10) mais … ça ne fonctionnait pas et du coup bin voilà cet article 🙂

Rendre à César…

La première chose à faire c'est un choix. Est-ce que j'ai besoin du Bluetooth ? Ma réponse est non. Ça veut dire que je peux dévalider le Bluetooth et du coup récupérer l'UART « kivabien » pour ma carte SIGFOX.

Vous avez 4 options

- **Option 1 :** Utiliser l'UART (le vrai !) en perdant la fonction Bluetooth. Il faut permuter les E/S des deux UART. Pour cela, ajoutez à ***/boot/config.txt*** : ***dtoverlay = pi3-disable-bt***
  Puis supprimer :
  console=serial0,115200 dans ***cmdline.txt***
- **Option 2 :** Faire fonctionner l'interface série et le Bluetooth correctement, mais la vitesse d'horloge du processeur sera fixée (à une vitesse faible [250MHz] ou à une vitesse élevée [500MHz?]). Ajoutez ***enable_uart = 1*** à ***/boot/config.txt***. Cela affectera les performances du processeur car ça contrôle la vitesse du cache L2, et on notera également une réduction de la qualité audio analogique (voyez ici et là).
  Si vous optez pour la vitesse d'horloge élevée (il faudra vraiment prévoir un ventilateur et un radiateur) pour garder la performance du processeur et la qualité audio, ajoutez également ***force_turbo = 1*** à ***/boot/config.txt***.
- **Option 3 :** Avoir une interface série « pourrie » sur le GPIO (vitesse variable) mais avoir un Bluetooth correcte : Ne rien faire. Ce sont les paramètres par défauts
- **Option 4 :** Faire fonctionner correctement l'interface série (UART), avec un Bluetooth qui fonctionne lentement. Permutez les UART : ajoutez ***dtoverlay = pi3-miniuart-bt*** au fichier ***/boot/config.txt***, puis définissez la fréquence à une valeur fixe (faible) en ajoutant, toujours à ***/boot/config.txt*** : ***core_freq = 250***. Cela affectera les performances du processeur. Si vous préférez conserver des performances plus élevées n'ajoutez pas la ligne ***core_freq = 250*** mais plutôt la ligne ***force_turbo = 1*** mais cela nécessite d'utiliser un ventilateur et un radiateur.

J'ai choisi la première.

Les ports série

```
lrwxrwxrwx 1 root root           7 févr. 10 19:11 serial0 -> ttyAMA0
lrwxrwxrwx 1 root root           5 févr. 10 19:11 serial1 -> ttyS0
```

Si tout va bien en faisant un ls -l /dev vous devriez retrouver un port ***serial0*** qui pointe vers ***ttyAMA0***.

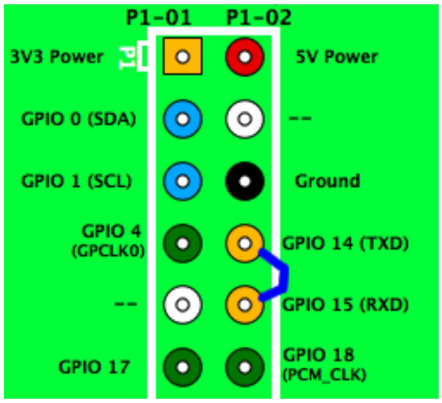
Les tests

Bien entendu pas question de connecter quoi que ce soit derrière le port série sans savoir d'abord si ça fonctionne. Vous me voyez venir ? Alors c'est une question dans les commentaires à la fin de l'article sur la carte SIGFOX (*j'imagine, bien sûr : ça ne se produira pas !* 😀) :
« *Bonjour, j'ai suivi le tuto mais ça ne marche pas !* »
Réponse : « *Est-ce que vous avez testé le port série avant de connecter la*

carte SIGFOX ? »
« *Non, mais j'ai bien suivi le tuto !! ça doit marcher !!*«

Eh bien non, cher(e) lecteur(trice) ! Point ne suffit de suivre à la lettre un tuto ! Tu te dois de vérifier à chaque étape que le résultat attendu est bien au rendez vous…

Certes me diras tu, mais comment qu'je fais moi ? pour tester le port série ?
Fastoche :



Tu vas relier les ports GPIO 8 et 10 correspondant à TXD et RxD (Données émises => Données reçues).
Lorsque tu vas envoyer des données sur le port série TxD, elles vont revenir par le port RxD… Hop là retour à l'envoyeur (ça s'appelle un loopback) ! Et le programme utilisé pour envoyer les données va les recevoir et les afficher 🙂
Suite à une remarque de ***msg*** (voir les commentaires) reliez les bornes 8 et 10 avec une résistance pour éviter la destruction des ports (ou pire) en cas d'erreur. 680 Ω ou 560 Ω fera l'affaire.
Bon avant de sortir la grosse artillerie on va dégainer un petit bout de Python, déjà pour se rendre compte de ce qui se passe.

> ⚠️ **Attention !**
>
> Le loopback fonctionne bien lorsque l'UART est connecté aux broches 8 et 10. Pensez à débrancher le court-circuit entre ces deux bornes après la fin des tests, sinon vous risquez d'endommager votre SoC !

Un programme en Python

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  #
4  # Test du port série
5  import serial
6  test_string = "Je teste le port série 1 2 3 4 5"
7  port_list = ["/dev/ttyAMA0", "/dev/ttyAMA0", "/dev/tty
8  for port in port_list:
9    try:
10     serialPort = serial.Serial(port, 9600, timeout = 2
11     print "Port Série ", port, " ouvert pour le test :
12     bytes_sent = serialPort.write(test_string)
13     print "Envoyé ", bytes_sent, " octets"
```

```
      if loopback == test_string:
        print "Reçu ", len(loopback), "octets identiques
      else:
        print "Reçu des données incorrectes : ", loopbac
      serialPort.close()
    except IOError:
      print "Erreur sur ", port, "\n"
```

Ce programme (*adapté d'un prog du forum raspberrypi.org*) va envoyer des données en sortie sur le port série, puis les récupérer sur l'entrée. Ici le test qui nous intéresse est celui de /dev/ttyAMA0. Vous pouvez mettre les ports que vous voulez tester dans la liste.

Lancez le programme de test et vous devriez obtenir :

```
pi@raspberrypi:~ $ python tesTxRx.py
Port Série  /dev/ttyAMA0  ouvert pour le test :
Envoyé  33  octets
Reçu  33 octets identiques. Le port /dev/ttyAMA0 foncti
Port Série  /dev/ttyAMA0  ouvert pour le test :
Envoyé  33  octets
Reçu  33 octets identiques. Le port /dev/ttyAMA0 foncti
Erreur sur  /dev/ttyS0
Erreur sur  /dev/ttyS0
```

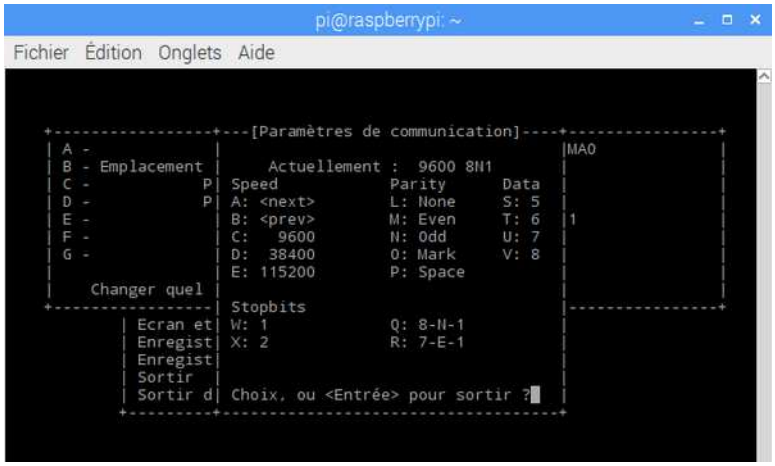
Si vous n'avez pas la confirmation que le port ttyAMA0 fonctionne correctement, inutile de continuer. Il faut d'abord faire fonctionner ce port pour pouvoir l'utiliser.

Minicom un mini émulateur de terminal sous Linux

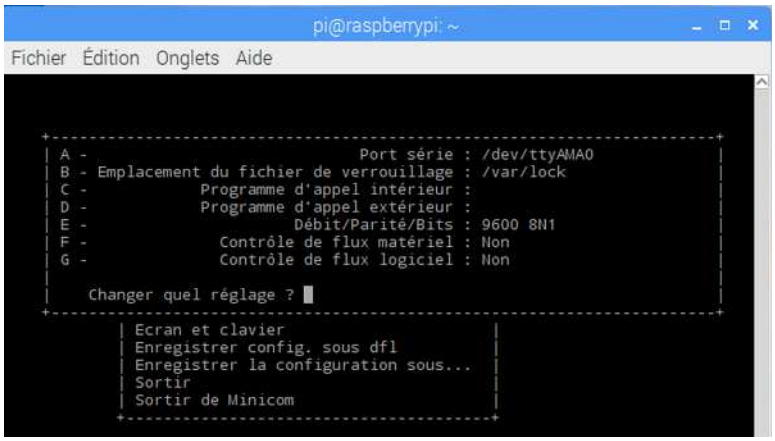
Pour utiliser Minicom, commencez par l'installer sur votre Raspberry Pi.

```
pi@raspberrypi:~ $ sudo apt-get install minicom
```

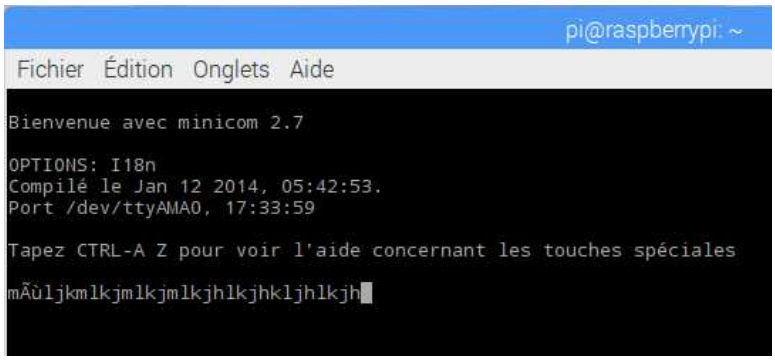
Configurez le :



Pour adapter le fonctionnement à la carte SIGFOX qui rejoindra ce Raspberry Pi, il faut régler les paramètres du port série à 9600 bits par seconde, 8 bits de données, pas de parité et un bit de stop. Ceci se traduit par 9600 8N1.



Choisissez également le port série pour qu'il corresponde au port raccordé au GPIO : ***/dev/ttyAMA0***.



Il ne reste plus qu'à tester : tapez… n'importe quoi sur le clavier et ça doit apparaître sur l'écran. Si rien n'apparait… C'est que le port série ne fonctionne pas ou que quelque chose est mal configuré.

Conclusion

Ce n'est pas la première fois que la Fondation Raspberry Pi nous fait le coup. On avait déjà connu ça avec l'introduction de systemd, peu documentée. Les habitués de Linux s'en sortaient tant bien que mal, mais cela avais mis de nombreux débutants en mauvaise posture. Pas ou peu d'infos, pas de doc… Système D (😀) obligatoire.

Eh bien là c'est rebelotte. On modifie les ports série, pas ou peu d'infos… Des surprises à la clé !

Eh les gars pensez aux utilisateurs… Le Raspberry Pi est fait pour l'éducation (aussi) et ceux/celles qui le mettent en œuvre ne sont pas forcément des vieux linuxiens barbus ! Alors s'il vous plait un peu d'infos et de doc, juste un peu 😉

Sources

- https://www.element14.com/community/thread/55627/l/how-to-use-serial-port-in-raspberry-pi-3?displayFullThread=true

- https://www.element14.com/community/message/195438/l/re-raspberry-pi-3-und-enocean-pi-kompatibilit%C3%A4tsproblem#195438
- https://www.raspberrypi.org/forums/viewtopic.php?f=28&t=153514

Le contenu a été bloqué, car il n'a pas été signé à l'aide d'un certificat de sécurité valide.

Pour plus d'informations, voir « À propos des erreurs de certificat », dans l'aide d'Internet Explorer.

Raspberry Pi 3 + UART/Bluetooth issues from **yeokm1**

À propos François MOCQ

Électronicien d'origine, devenu informaticien, et passionné de nouvelles technologies, formateur en maintenance informatique puis en Réseau et Télécommunications. Dès son arrivée sur le marché, le potentiel offert par Raspberry Pi m'a enthousiasmé j'ai rapidement créé un blog dédié à ce nano-ordinateur (www.framboise314.fr) pour partager cette passion. Auteur de plusieurs livres sur le Raspberry Pi publiés aux Editions ENI.
Voir tous les articles de François MOCQ →

21 réflexions au sujet de « Le port série du Raspberry Pi 3 : pas simple ! »

F6EEQ
13 février 2017 à 8 h 51 min

Merci pour cet article.

J'avais déjà +/- débrouillé le bazar pour brancher un GPS en port série, mais les tutos ou forums apportent toujours des réponses partielles.
Au moins ici on a les explications précises de l'origine du problème et les remèdes à apporter su un seul article!

Gérard

François MOCQ Auteur de l'article
13 février 2017 à 9 h 26 min

Merci Gérard
c'est une compilation d'infos trouvées à droite et à gauche 🙂
73's
François / F1GYT

S.POURRE
13 février 2017 à 10 h 52 min

Bonjour,

C'est bien plus qu'une simple compilation.
Derrière, il y a un travail d'analyse, de tests, de rédaction. chapeau l'artiste 😉

Sylvain

François MOCQ Auteur de l'article
13 février 2017 à 19 h 17 min

merci Sylvain ! fais attention à mes chevilles 😀

gUI
13 février 2017 à 17 h 32 min

Merci pour ces explications très complètes !

Je me sers encore de l'UART souvent en tout dernier dépannage. J'ai pas (encore) de RPi3, mais j'aurais pas mal galéré 🙂

François MOCQ Auteur de l'article
13 février 2017 à 19 h 17 min

🙂 merci

msg
15 février 2017 à 10 h 14 min

Bonjour François ,

Petite remarque pour le test :
Ne peut-on pas remplacer le court circuit par une résistance tampon ?
ça serait moins dangereux pour le Pi en cas de fausse manœuvre ou d'oubli .

Je ne veux pas dire de bêtises , mais je crois que l'UART , comme les GPIO en entrée , fonctionnent en tension et non en courant ,
Une résistance de pas trop grosse valeur devrait faire l'affaire .

la 1KΩ est parfaite pour les signaux en 5V , ça limite le courant à 5mA en cas de court jus .
Testé sur un montage où j'avais des « GPIO » en mode bi-directionnel (lecture depuis un composant puis écriture vers un autre en mode parallèle sur la même ligne de données) , la résistance tampon m'évitait de faire des court-jus lorsque les bits reçus étaient de valeur contraire aux bits envoyés .

Pour le test , vu que le GPIO tourne en 3,3V , une 680 Ω ou une 560 Ω devrait faire l'affaire pour un courant équivalent .

François MOCQ Auteur de l'article
15 février 2017 à 18 h 22 min

Bonjour
oui tout à fait d'accord!
c'est vrai que j'ai gardé l'habitude du court-circuit qu'on faisait...

avant sur les prises DB25 et DB 9 😀
merci je modifie l'article en conséquence !
cordialement
François

Namu
25 mars 2017 à 8 h 55 min

Salut,

La dernière révision du Pi 2 qui utilise le même SoC que le Pi 3 mais sans wifi, ni BT. Logiquement, elle ne devrait pas avoir de soucis mais si quelqu'un pouvait le confirmer ça m'arrangerait 🙂

Je dois remplacer le B+ qui gère ma domotique et j'utilise l'UART pour la Téléinfo EDF…

marc
12 avril 2017 à 21 h 26 min

bonjour,

J'ai choisi l'option pour redirigé les port serial, mais cela ne fonctionne pas.

hajji
13 avril 2017 à 11 h 29 min

bonjour,
j'ai un projet qui nécessite d'envoyer des données : par trois équipements liée en série
-Envoyer des données du raspberry pi 3 par un adaptateur USB/RS232 vers un adaptateur RS232/ethernet (MOXA 5210 A) et ensuite vers un data concentrateur via un câble RJ 45.
les données sont envoyées bidirectionnels
l'adresse IP du raspberry pi :169.254.244.62
l'adresse IP du MOXA :192.168.0.3
l'adresse IP du Data concentrateur : 192.168.0.1
comment créer un script shell qui envoie les données du raspberry

pi vers le data concentrateur ?
et merci

G DE LA RUE
13 avril 2017 à 18 h 57 min

Merci pour cet article François ! Je l'avait fait sur tous les raspi avant le 3, et j'ai été bloqué 2h cet après-midi sur rpi3 à cause de ça...
Pour moi également j'ai fait le choix de supprimer le BT et utiliser l'UART0 en full speed.
Je fais une chose en plus : désactiver le service BT qui ne sert plus à rien.

```
sudo vi /boot/cmdline.txt

dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2
rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait


echo "Disable bluetooth service"
sudo systemctl disable hciuart


sudo vi /boot/config.txt

# Disable BT and enable UART0 on /dev/ttyAMA0
dtoverlay=pi3-disable-bt
enable_uart=1
```

Thierry
9 mai 2017 à 11 h 07 min

Bonjour François,
Merci pour ce tuto c'est exactement ce que je cherchais, malgrés tout, lorsque je lance le script python j'ai ca en retour :
Erreur sur /dev/ttyAMA0
Erreur sur /dev/ttyAMA0
Erreur sur /dev/ttyS0
Erreur sur /dev/ttyS0
Il y a donc un dysfonctionnement quelque part ???
Par contre je suis sous Ubuntu Mate 16.04, peut être est ce cela mon

problème ???
Thierry

François MOCQ Auteur de l'article
9 mai 2017 à 11 h 25 min

Bonjour Thierry
c'est possible je n'utilse pas Ubuntu
il faudrait analyser en détail les messages
et la façon dont Ubuntu gère les ports RS232
cordialement
François

Thierry
9 mai 2017 à 16 h 23 min

Merci pour cette réponse, je viens de faire le test avec la dernière RaspBian et cela fonctionne.
Il ne me reste plus qu'a adapter ROS sur la Debian.
Merci François.

thierry76
13 mai 2017 à 19 h 48 min

Merci François pour toutes ces explications, malheureusement pour moi, ça ne fonctionne pas.
J'ai choisi l'option 1 en respectant la syntaxe et après reboot, la Raspi est plantée.
Reformatage et écriture de la dernière image Raspbian Jessie.
Y-a-t-il un endroit particulier dans le fichier config.txt où ecrire la modif dtoverlay = pi3-disable-bt

Thierry76

Thierry
14 mai 2017 à 20 h 46 min

Non, tu le met à la fin du fichier, c'est tout

boubou
30 mai 2017 à 12 h 38 min

Bonjour,
J'ai désactivé la fonction bluetooth parce que j'e n'en pas besoin pour l'instant
lorsque je tape cette commande ls -l / dev j'ai : » serial1 -> ttyAMA0 »
J'ai mis le même programme que vous avez mis en dessus pour testear le bon fonctionnement du port mais ça bloque dans la première phrase du programme : » Port Série /dev/ttyAMA0 ouvert pour le test : »
et ne rien fait après, je ne vois aucune émission/réception de données
Avez vous une idée ?
merci d'avance

S.DECORME
22 août 2017 à 11 h 13 min

Bonjour
Merci pour cette article complet .
Je cherche juste le baudrate Max que peut atteindre l'UART0 , j'ai besoin de monter à 460800bauds
Merci

François MOCQ Auteur de l'article
22 août 2017 à 11 h 43 min

Bonjour
à ma connaissance c'est 115200 mais avec les exemples ici vous pouvez tester d'autres valeurs
http://elinux.org/RPi_Serial_Connection
au dela je pense qu'il faut aller plus avant dans le paramétrage
https://www.raspberrypi.org/forums/viewtopic.php?t=17559
ou encore ici : https://www.raspberrypi.org/forums/viewtopic.php?t=63057

cordialement
François

Hendrik
14 septembre 2017 à 17 h 29 min

Bonjour,
Je suis novice en Linux et je développe une application qui nécessite le port série complet.
Cet article est, semble-t-il, très complet.
Mais mon Pi3 me refuse le droit de modifier les fichiers config.txt et cmdline.txt.
Comment faire pour modifier les droits de ces fichiers?
Merci pour votre réponse