



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

simple !

Le port série du Raspberry Pi 3 : pas

Rechercher



Le port série du Raspberry Pi 3 : pas simple !

Novice

Débutant

Avancé

Confirmé

Expert

Niveau

Au sommaire : [[cacher](#)]

- [1 Le port Série du Raspberry Pi 3](#)
 - [1.1 Les UART du Raspberry Pi](#)
 - [1.1.1 UART0 = PL011](#)
 - [1.1.2 UART1 = « mini » UART](#)
 - [1.2 C'était mieux avant ! Le port série du Raspberry Pi 2](#)
 - [1.3 Le port série du Raspberry Pi 3 : la cata !](#)
- [2 Un UART pour mon SIGFOX](#)
- [3 Rendre à César...](#)
 - [3.1 Vous avez 4 options](#)
 - [3.2 Les ports série](#)
- [4 Les tests](#)
 - [4.1 Un programme en Python](#)
 - [4.2 Minicom un mini émulateur de terminal sous Linux](#)
 - [4.3 En Python](#)
- [5 Conclusion](#)
- [6 Le port série du Raspberry Pi 4](#)
- [7 Sources](#)

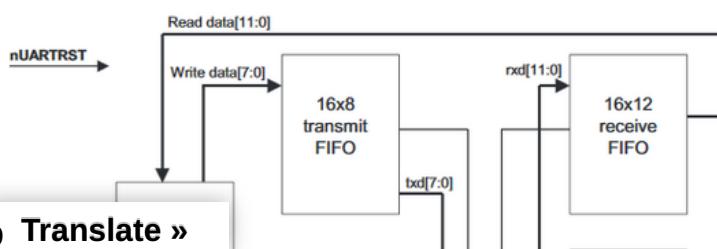
Le port Série du Raspberry Pi 3

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 :



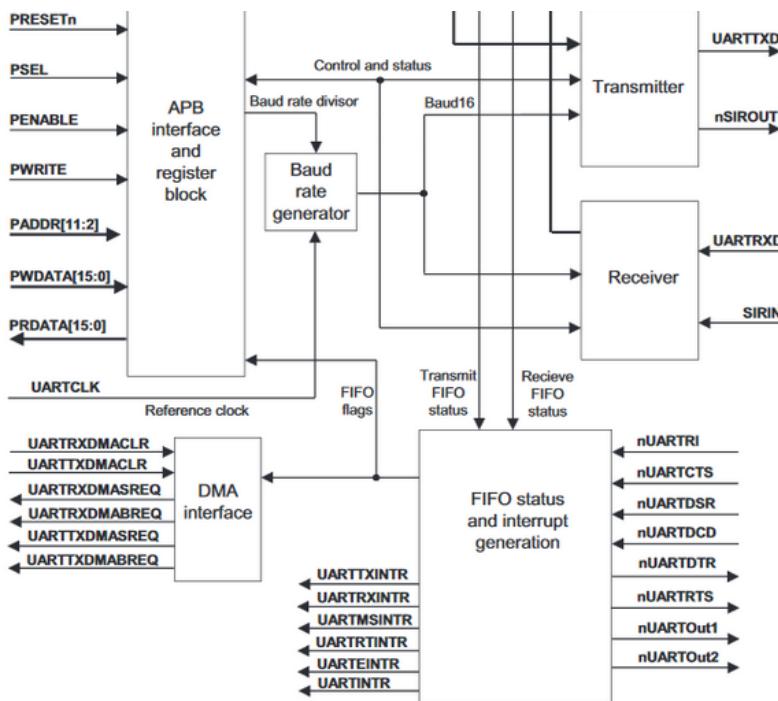


Figure 2-1 UART block diagram

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 throughput⁴ 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:

$$\text{baudrate} = \frac{\text{system_clock_freq}}{8 * (\text{baudrate_reg} + 1)}$$

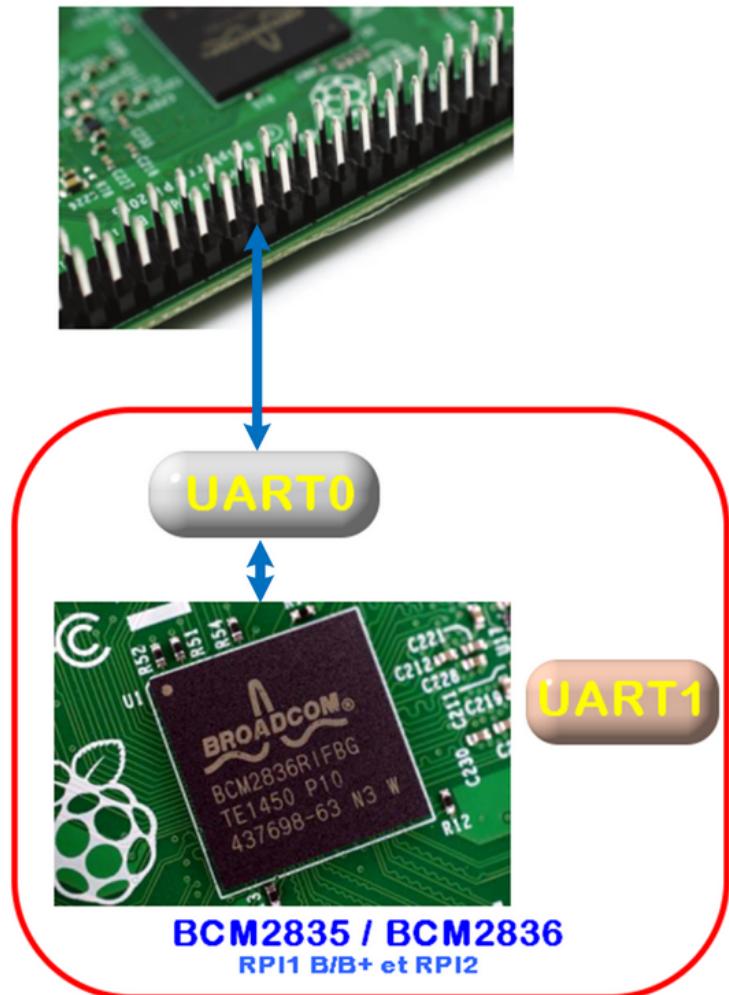
If the system clock is 250 MHz and the baud register is zero the baudrate is 31.25 Mega baud. (25 Mbits/sec or 3.125 Mbytes/sec). The lowest baudrate with a 250 MHz system clock is 476 Baud.

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

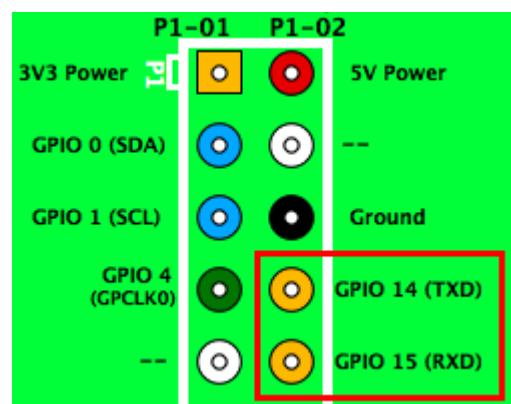


Il ne gère pas non plus la parité.

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



de démarrage sont envoyés par défaut sur ce port. Il suffit de brancher un terminal pour les lire.

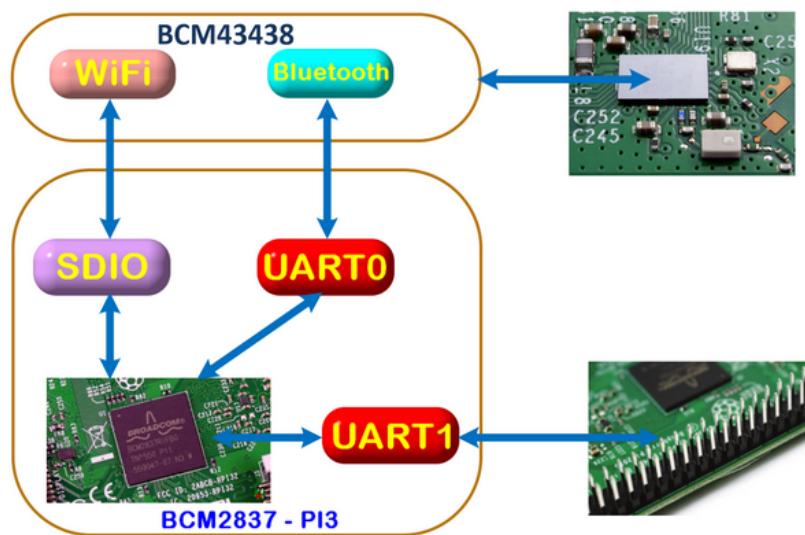


Cette entrée série peut également être connectée à un terminal qui servira alors à se connecter au Raspberry Pi après s'être logué.

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 (Société Nationale des Objets Connectés)** située à Saint-Sylvain-d'Anjou a sorti fin janvier une carte de prototypage pour SIGFOX. Cette [BRKWS01 distribuée par Ydom](#), 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 permute 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

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. Permettez les UART : ajoutez **dtoverlay = pi3-minuart-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 !* »

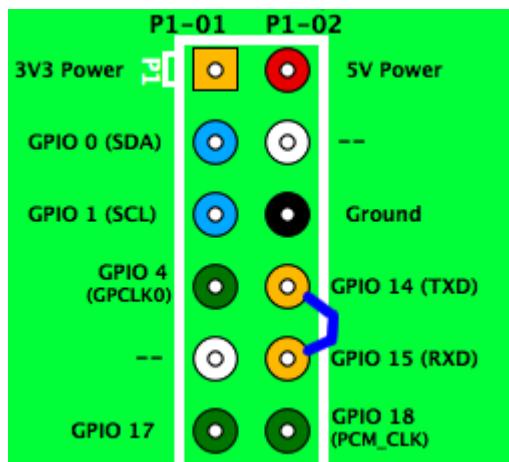
Réponse : « *Est-ce que vous avez testé le port série avant de connecter la carte SIGFOX ?* »

« *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...

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 loop-back) ! 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.

[stextbox id= »warning » caption= »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 ![/stextbox]

Un programme en Python

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
# Test du port série
import serial
test_string = "Je teste le port série 1 2 3 4 5"
port_list = ["/dev/ttyAMA0", "/dev/ttyAMA0", "/dev/ttys0"]
for port in port_list:
    try:
        serialPort = serial.Serial(port, 9600, timeout = 2)
        print "Port Série ", port, " ouvert pour le test :"
        bytes_sent = serialPort.write(test_string)
        print "Envoyé ", bytes_sent, " octets"
        loopback = serialPort.read(bytes_sent)
        if loopback == test_string:
            print "Reçu ", len(loopback), "octets identiques."
        else:
            print "Reçu des données incorrectes : ", loopback,
            serialPort.close()
    except IOError:
        r 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 fonctionne
Port Série /dev/ttyAMA0 ouvert pour le test :
Envoyé 33 octets
Reçu 33 octets identiques. Le port /dev/ttyAMA0 fonctionne
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.

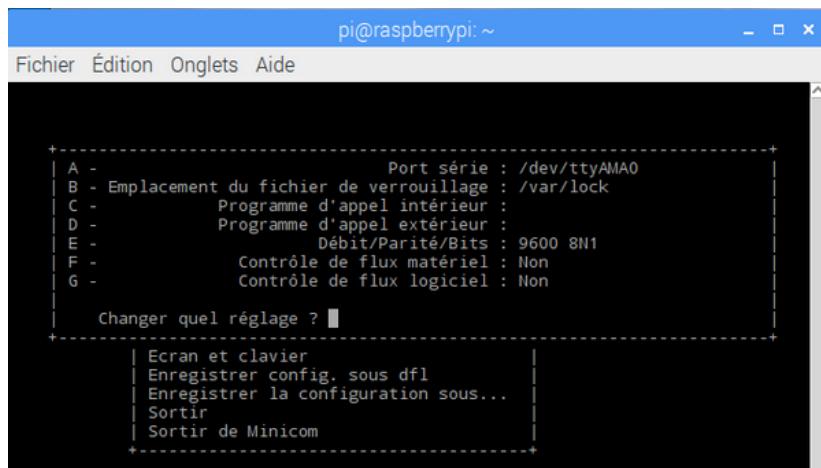
```
pi@raspberrypi:~ $ sudo apt-get install minicom
```

Configurez le :

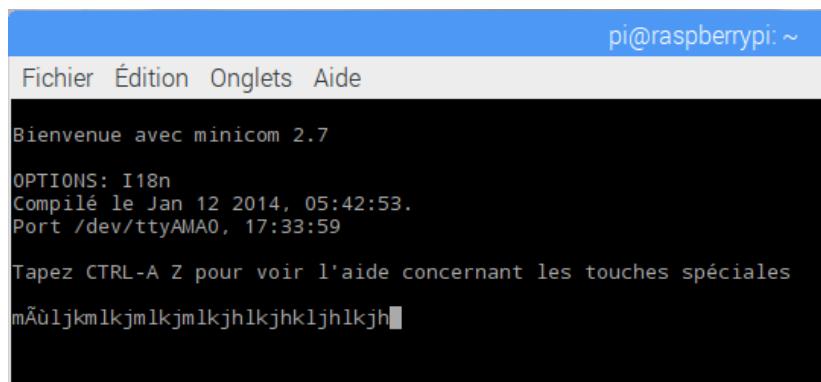


Pour adapter le fonctionnement à la carte SIGFOX qui rejoindra ce Raspberry Pi, il faut régler les paramètres du port série à 9600 bits par seconde, 8 bits de données, pas de parité et un bit de

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'apparaît... C'est que le port série ne fonctionne pas ou que quelque chose est mal configuré.

En Python

=====### Edit du 25 février 2019 ###=====

Étienne m'a envoyé ce message :

*J'ai testé cette excellente modif sur mon Pi3B+ car grand besoin d'un
UART de compétence pour mes projets !*

Attention, si vous travaillez comme moi en python3, il faut faire quelques modifs, rajouter les parenthèses dans les fonctions print, par exemple pour la dernière ligne on écrira :

```
print(« Erreur sur « .port. « \n »)
```

*il se peut aussi que vous ayez des erreurs d'encodage des caractères,
il faudra alors écrire :*

test_string = b »Je teste le port série 1 2 3 4 5”

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 plaît un peu d'infos et de doc, juste un peu 😊

====> Edit du 21 avril 2020 <=====

Merci à Benoît qui m'a transmis ce message pour appliquer au Pi4

Le port série du Raspberry Pi 4

Un grand merci pour ces explications!!! ça m'a sauvé 😊

Je sais bien que cet article est destiné au Pi3 mais je voulais juste faire une remontée d'expérience quant à sa mise en œuvre sur Pi4 (sous Raspbian à jour en date d'avril 2020).

j'ai essayé d'appliquer le méthode 1 n'ayant pas du tout besoin du Bluetooth intégré au Pi4:

Dans **/boot/config.txt** au lieu de **dtoverlay = pi3-disable-bt** j'ai rajouté :

dtoverlay = disable-bt en fin de fichier
et bien entendu j'ai **supprimé** dans **/boot/cmdline.txt** :
console=serial0,115200

PS: j'ai également désactivé le service hciuart en exécutant la commande :

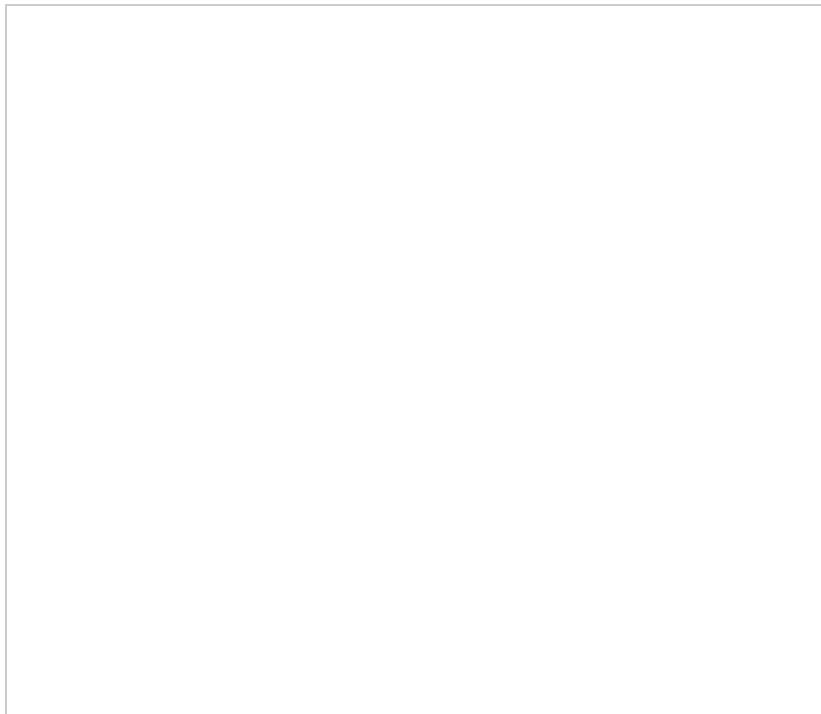
puis reboot...

Quant au code pour tester les ports série j'ai été obligé de le modifier un petit peu pour qu'il soit compatible avec mon environnement python:

```
#!/usr/bin/env python
# Test du port série
import serial
test_string = "Je teste le port série 1 2 3 4 5".encode()
port_list = ["/dev/ttyAMA0", "/dev/ttyAMA0", "/dev/ttyS0",
for port in port_list:
    try:
        serialPort = serial.Serial(port, 9600, timeout = 2)
        print ("Port Série", port, "ouvert pour le test:")
        bytes_sent = serialPort.write(test_string)
        print ("Envoyé", bytes_sent, "octets")
        loopback = serialPort.read(bytes_sent)
        if loopback == test_string:
            print ("Reçu ",len(loopback), "octets identiques.")
        else:
            print ("Reçu des données incorrectes:", loopback,
        serialPort.close()
    except IOError:
        print ("Erreur sur", port, "\n")
```

Sources

- [https://www.element14.com/community/thread/55627/I/how-to-use-serial-port-in-raspberry-pi-3?
displayFullThread=true](https://www.element14.com/community/thread/55627/I/how-to-use-serial-port-in-raspberry-pi-3?displayFullThread=true)
- <https://www.element14.com/community/message/195438/I/re-raspberry-pi-3-und-enocean-pi-kompatibilit%C3%A4tsproblem#195438>
- <https://www.raspberrypi.org/forums/viewtopic.php?f=28&t=153514>



Raspberry Pi 3 + UART/Bluetooth issues from yeokm1

Partager      

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 →](#)

59 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.

13 sur 39 [Translate »](#)

Confidentialité
12/03/2024 07:32

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 hcuart
```

```
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, malgré tout, lorsque je lance le script python j'ai ça 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

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'à 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 écrire 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



RogerT

26 août 2018 à 12 h 44 min

bonjour,

j'ai eu le même problème (Rasbian Jessie – voir mon message du 25/08/2018) :

J'ai Rajouter à /boot/config.txt

enable_uart=1

après reboot :

\$ ls -l /dev | grep serial

lrwxrwxrwx 1 root root 5 Aug 25 21:59 serial0 -> ttyS0

lrwxrwxrwx 1 root root 7 Aug 25 21:59 serial1 -> ttyAMA0



Pointillé

9 mars 2020 à 16 h 48 min

Bonjour,

Nous avions le même problème que boubou, nous avons suivi vos conseils. Nous avons bien serial0 pointant vers ttyS0, et se-

Cependant, après l'exécution du programme python, le terminal affiche " Port Série /dev/ttyAMA0 ouvert pour le test : ", et pas de suite.

Merci d'avance.



François MOCQ Auteur de l'article

10 mars 2020 à 17 h 41 min

bonjour
vous avez bien strappé les 2 pinoches GPIO ?
cdt
françois



Pointillé

1 avril 2020 à 17 h 52 min

Oui les deux broches GPIO étaient bien strappées, après re-commencement depuis le début, cela a fonctionné, donc plus de problèmes.

Cordialement



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 delà 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



Michel

27 novembre 2017 à 6 h 47 min

Tu utilises la commande sudo chmod. (Sudo c'est pour avoir les droits de root, c'est à dire de tout faire). chmod c'est la commande que tu change les droits en lecture, écriture et exécution.



David.B

6 octobre 2017 à 22 h 15 min

Salut à tous,

Article très instructif et qui fonctionne parfaitement, mais je souhaitais partager avec vous un problème annexe, résolu mais inexpliqué, alors si ça peut éviter une galère à d'autres et qu'un spécialiste peut nous éclairer sur ce phénomène...

J'ai utilisé la procédure ci-dessus sur un Raspberry Pi 3 sous Raspbian 9.1 et Python 2.7 dont l'objectif était d'utiliser le port série depuis PHP 7.0.19-1. En suivant les instructions ci-dessus, tout fonctionne parfaitement. Alors vous allez me dire où est le problème ? Eh bien, si on omet le test du script python, d'autant qu'on pourra procéder aux mêmes vérifications sous PHP, rien ne fonctionne avec cette technologie. La lecture de Rx ne fournit cette fois absolument rien ! Pourtant minicom démontre que le shunt est opérationnel.

Comment résoudre ce problème ? Sachant qu'il faut initier une communication série via un script python (même plus élémentaire que celui décrit ci-dessus) pour que PHP s'en sorte jusqu'au prochain démarrage, il faut lancer ce script automatiquement au lancement de l'OS et le problème n'existe plus.

Maintenant je suis preneur de toutes explications sur ce phénomène pour éviter ce genre de bidouille. En tout cas vous voilà averti, si vous pensiez maîtriser le port série via PHP, redémarrez avant de crier victoire...

David.B



M. BLIN

2 mars 2018 à 23 h 21 min

Bonjour,

Merci pour cet article très bien fait qui explique clairement les possibilités pour faire fonctionner le UART du Raspberry pi 3. J'ai suivit les instructions, ai fait les mise à jour qui s'impose, ...

des périphériques i2c tel que des afficheurs oLed, LCD, pilote de moteur pas à pas, le GrovePi+, ... mais certain périphériques fonctionnent avec le port UART. Lorsque je l'utilise en lecture seul, tout fonctionne bien (exemple d'un module GPS). mais lorsque je dois transmettre des messages, il se met à déblatérer des informations dont je ne suis pas l'auteur. Comment savoir d'où cela vient ? Quelqu'un aurai-t-il déjà rencontré puis résolu ce problème ?

Je vous remercie par avance,
Jean BLIN

**M. BLIN**

5 mars 2018 à 19 h 51 min

Bonjour,
Je me répond à moi même, mais j'ai fini par trouver la solution,
...
La carte GrovePi+ qui est vendu avec un grand nombre d'équipement prévu pour communiquer avec le port série... perturbe et ne permet pas d'utiliser le port série.
Il y a des concepteurs qui devraient réfléchir au moins une fois avant d'agir...
Donc désolé d'avoir posé une question avant de tester la chose la plus simple... Retirer tous les équipement et faire un test à vide.
Encore désolé et que ma déconvenue serve au minimum à guider les futures devs qui oublierons, sans doute par rêverie, de faire les testes les plus simple.
Cordialement,
M BLIN

**didi**

29 mars 2018 à 10 h 34 min

Bonjour

J'ai essayé votre option numéro 1 et j'ai pas été dessus : « kernel panic » et impossible de démarrer la Raspberry , j'ai essayer de re-modifier les fichiers directement sur la carte SD mais rien n'y fait , obliger de reformater la carte et de refaire la config de la Rasp. et c'est un peu près le même problème pour tout vos tutos il ne sont ni à jour et ne fonctionnent pas pour stretch ... merci de donner dans le futur qu'elles sont les versions des logiciels et la date de création du post

bonne journée



tnatf

12 janvier 2019 à 16 h 21 min

UP !

De mon côté, j'arrive à éditer le fichier depuis un autre système linux.

c'est le rajout de enable_uart=1 à /boot/config.txt qui fait planter



Fleron

30 mai 2018 à 15 h 20 min

Bonjour,

Je viens régulièrement sur votre site ... Merci pour le boulot!
J'ai un soucis de parasites sur la sortie de la liaison série, suffisamment important pour ne pas pouvoir récupérer qq chose propre pour être traduit en DMX.

Ces parasites semblent constants et répondre à une fréquence précise

Je suis sur un rasp 3 B+

Avez-vous des pistes sur la cause de ces parasites?

Encore merci.

**Don**

2 août 2018 à 3 h 22 min

Étrange... le UART fonctionne bien Rx & Tx lorsque le programme est démarrer en ligne de commande « python3 main.py », mais le UART ne fonctionne pas une fois la distribution compléter avec pyinstaller...

```
« /home/pi/.local/bin/pyinstaller --onefile --hidden-import=tkinter -y --hidden-import='PIL._tkinter_finder' main.py » « cd dist/ » « ./main »
```

Le programme sombre dans boucle infini avec l'appel de: « device.readline() »....

J'ai l'impression que RaspberryPi3 contient des permissions et que la documentation est nul part.... :s

**ZAABOUBI**

9 août 2018 à 15 h 27 min

Bonjour,

A ce que il y a une possibilité de lire les messages envoyée par la raspberry via l'UART, j'utilisé le port 8 pour TX pour l'envoi des donnée. lire les données envoyées sur la broche 8

**MARTIN**

24 août 2018 à 18 h 48 min

Bonjour François,

Le Rapsberry me paraît intéressant pour résoudre mon problème :

Je cherche un matériel qui peut remplacer un modem en protocole minitel par un modem 4G.

J'ai actuellement une connexion à distance entre un minitel d'une part et un modem analogique bas débit de protocole Minitel d'autre part, connecté à un système de chauffage

Clipsol PSDAI, me permettant de piloter ce chauffage à distance.

La connexion s'établit en composant sur le Minitel le n° de téléphone à 10 chiffres de la ligne où est branché le modem.

C'est super pratique et ça marche très bien.

Malheureusement, Orange va bientôt supprimer la liaison filaire analogique et du coup, plus rien ne fonctionnera avec le minitel.

Je voudrais donc faire évoluer le système sur internet : une solution serait de remplacer le modem Minitel par un modem internet, 4G si possible.

D'un côté, à partir de l'entrée du modem à remplacer (Light Modem MV 123 V23 de COM1), installer un modem 4G et envoyer les pages de dialogue sur internet.

De l'autre côté, recevoir ces pages sur un PC ou un smartphone, et échanger des informations pour contrôler et programmer le chauffage.

Je cherche donc quelqu'un pour réaliser cette transformation.

Merci pour votre proposition.

Cordialement.



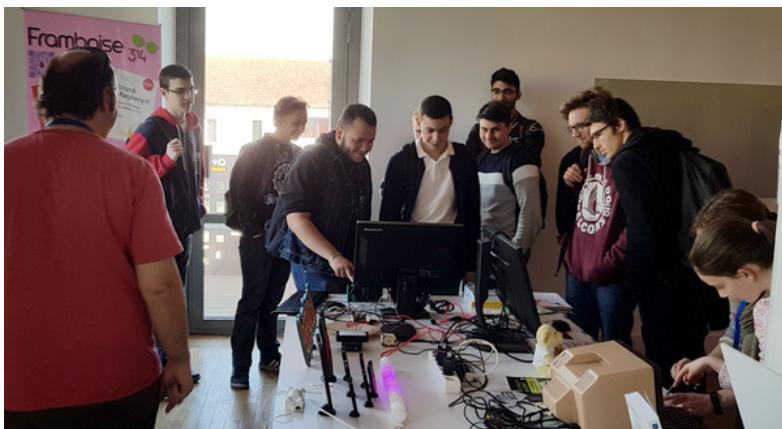
François MOCQ

Auteur de l'article

25 août 2018 à 9 h 59 min

Bonjour

je sais que des solutions existent pour interfaçer un minitel et un Raspi mais je n'ai pas pratiqué
si le Raspi sert à envoyer/recevoir les données, vous devriez pouvoir passer par la prise série à l'arrière du minitel (moyennant une adaptation de niveau)



ici sur le stand de SLMEDIATION un Minitel piloté par un
Raspberry Pi lors de #RNRPI3 à Nevers
essayez de poser la question dans le forum vous aurez peut
etre des solutions ?
cordialement
François

**RogerT**

25 août 2018 à 16 h 47 min

Bonjour,

comme de nombreuses personnes, j'ai perdu _beaucoup_ de temps à configurer ce fichu port série du Rpi 3 ArgX!(@@)%\$ V...

Dans mon cas (RPi3 model B), c'est le commentaire de G DE LA RUE qui a finalisé une solution opérationnelle, cad l'option 2 de François + désactivation du BT :

```
& cat /boot/config.txt
```

```
#...
```

```
dtoverlay = pi3-disable-bt
```

```
enable_uart=1
```

(et en modifiant aussi /boot/cmdline.txt)

Je m'occupe maintenant de communiquer enfin avec mon modem.

Je testerai plus tard toutes les autres options avec BT et je vous mettrai au parfum.

Merci à vous deux

**François MOCQ**

Auteur de l'article

25 août 2018 à 17 h 00 min

merci

éventuellement je mettrai un complément en fin d'article car beaucoup de monde galère avec ça !

**RogerT**

26 août 2018 à 13 h 59 min

Pour information à ceux qui galèrent dans leur communication avec leur modem (après avoir enfin réglé le problème du port série du RPi3) :

J'ai eu et j'ai réglé un souci bloquant (avec minicom, non lié à

minicom) : impossible de causer avec le modem via ce port série.

En fait, ça ne marchait pas car la communication était réglée à 9600 bauds. A 115200 bauds, débit du modem, ça marche impeccable.

AT

OK

Question : quel composant de communication série le plus léger recommandez-vous d'utiliser pour intégrer à un programme _quelconque_ (éventuellement un script bash) ? A ce propos, quels composants utilisent Python et sa bibliothèque (directive « import serial ») ?

Savez-vous comment utiliser minicom en CLI (ligne de commande) plutôt qu'en tant que terminal ?

Merci

**etiennes33**

25 février 2019 à 11 h 50 min

Hello la team 😊

J'ai testé cette excellente modif sur mon Pi3B+ car grand besoin d'un UART de compète pour mes projets !

Attention, si vous travaillez comme moi en python3, il faut faire quelques modifs :

rajouter les parenthèses dans les fonctions print, par exemple pour la dernière ligne on écrira :
print (« Erreur sur « , port, « \n »)

il se peut aussi que vous ayez des erreurs d'encodage des caractères, il faudra alors écrire :
test_string = b »Je teste le port série 1 2 3 4 5”

Et avec ça, c'est nickel ...

Merci framboise314 pour tout ce que tu fais !!

**François MOCQ**

Auteur de l'article

25 février 2019 à 12 h 06 min

Bonjour Etienne
un grand merci
je rajoute ça en fin d'article pour ceux qui ne lisent pas les commentaires 😊

**ebac1**

13 avril 2019 à 17 h 08 min

Bonjour,
etant novice en la matière comment faisons nous un retour en arrière remise en service du bluetooth ?

merci d'avance pour votre aide

Ping : [Le port série du Raspberry Pi 3 – very Usefull tricks, or not](#)

Ping : [Un GPS USB pour le Raspberry Pi - Framboise 314, le Raspberry Pi à la sauce française....](#)



Valentin COLLIEZ

27 avril 2020 à 18 h 02 min

Bonjour,

Est-ce que couper le Bluetooth interne au Rpi comme indiqué ici, pour faire fonctionner la Pizigate, et utiliser un dongle Bluetooth usb me permettrait d'avoir Zigbee et Bluetooth sur mon Rpi 3 ?



Phillou53

11 mai 2020 à 11 h 04 min

Bonjour François,

Un IMMENSE merci pour ce site et pour cet article très intéressant en particulier. Il m'a sauvé de la crise de nerf avec une interface z-wave.me Razberry particulièrement récalcitrante sur une Jeedom RPi ! D'autres détails ici : <https://community.jeedom.com/t/interface-z-wave-razberry-qui-ne-fonctionnait-pas-et-mesure-niveau-du-signal-z-wave/27953>

Avec un brin de nostalgie en pensant au chip USART 8251 d'un autre temps à x bits/s, 1 stop bit, part paire ou impaire ... 200 bits/s au début avec les « fameux » Teletype et autres Dumb Terminal vt100 ... Un autre monde 😊

**François MOCQ**

Auteur de l'article

11 mai 2020 à 11 h 26 min

merci 😊

et merci pour le lien qui pourra sans doute servir à d'autres domoticiens

ah oui j'ai bien connu aussi notre ami 8251 puisque j'ai été élevé avec du Z80 dans mon biberon 😊

la prise uFL du raspi est là pour les tests lors de la certification de la partie radio (mesures de puissance et de « propreté » du signal), on n'est pas censé ajouter une prise et une antenne

car on perd la certif 😊

cdt

francois

Ping : Synchronisez la date et l'heure de votre Raspberry PI avec un GPS – Société Havraise de Télégraphie Sans Fil -- 25 rue des Iris

**fred**

22 décembre 2020 à 21 h 21 min

Bonsoir,

Merci pour cet article.

Je cherchais pourquoi je n'arrivais pas à communiquer avec mon compteur LINKY avec un RPI3B+.

Après avoir parcouru tous les commentaires j'ai enfin réussi à faire fonctionner le programme de test en Python 3.5 sur l'option 1 et je suis sous stretch. Mais le résultat n'est pas satisfaisant et je ne comprends pas pourquoi!

Je vous donne donc toutes mes étapes en espérant que quelqu'un comprenne le problème.

Dans config.txt, j'ai ajouté: dtoverlay=pi3-disable-bt et mis en commentaire #enable_uart=1

Dans cmdline.txt j'ai supprimer console=serial0,115200 et en plus également désactivé le service hcuart : sudo systemctl disable hcuart

J'ai corrigé toutes les « » et () du fichier original pour que le prog fonctionne et bouclé avec une résistance de 560 ohm pour le test.

et le test me renvoie:

Port Série /dev/ttAMA0 ouvert pour le test:

Envoyé b'Je teste le port serie 1 2 3 4 5' soit 32 octets

Reçu des données incorrectes: b » sur le port série /dev/
ttyAMA0 bouclé

j'ai descendu la vitesse du port à 1200bauds mais rien n'y fait
je ne reçois pas ce que j'envoie sur les pin 8 et 10.
merci pour vos propositions.



Matthieu

13 février 2022 à 17 h 55 min

ça fait deux fois que je tombe sur cette page et qu'elle me sauve la mise. Merci pour cet article complet et bien écrit.



François MOCQ

Auteur de l'article

13 février 2022 à 17 h 58 min

merci 😊



Matthieu

13 février 2022 à 18 h 06 min

De rien. Et bravo pour le truc du loopback. C'est une astuce que peu de bricoleurs connaissent étonnamment, et qui rend des grands services pour débuguer ces satanées UARTs 😊



Mickael

6 mai 2022 à 21 h 39 min

Bonjour,

des ports TX/RX + GND et VCC sur un composant, je ne vois
nulle part sa référence ?

Est-il possible de se passer de ce composant en reliant les pins
TXD/RXD à un DB9 sur les pins 2-3 ?

Je viens d'essayer mais ça ne fonctionne pas. (en loopback ma
config ça fonctionne)



François MOCQ

Auteur de l'article

7 mai 2022 à 10 h 17 min

Bonjour

Attention, les E/S GPIO sont limitées à 3,3 volts
au delà vous allez les détruire
si vous connectez directement TxD et RxD du GPIO à une DB9
vous risquez de griller le RasPi !!
le RS232 « normal » est en +/- 12v
il y a un circuit convertisseur de niveau sur la carte comme ici
https://s.click.aliexpress.com/e/_A1co1R
cdt
francois



Framont

22 juillet 2023 à 6 h 34 min

Bonjour,

Merci pour votre billet. J'essaye de connecter une imprimante
thermique (9600 baud) sur le port série d'un Raspberry Pi Zero
W (raspbian bullseye), sans succès.

J'ai connecté GPIO14 et GPIO15 avec une résistance tampon de
680 Ohm ; testé, c'est ok.

J'ai suivi votre option 1 (je n'ai pas besoin de bluetooth) :
- ajouté dtoverlay=pi3-disable-bt dans /boot/config.txt
- effacé console=serial0,115200 dans /boot/cmdline.txt

ls -l /dev donne

lrwxrwxrwx 1 root root 7 Jul 21 16:28 serial0 -> ttyAMA0

lrwxrwxrwx 1 root root 5 Jul 21 16:28 serial1 -> ttyS0

Après reboot, le script python de test ne fonctionne pas.

J'ai aussi essayé en désactivant le service hciuart

- sudo systemctl disable hciuart

Puis modifié la ligne du script python en test_string = b »test-test»

Une fois lancé le test, j'obtiens toujours

Port Série /dev/ttyAMA0 ouvert pour le test:

Envoyé 8 octets

Reçu des données incorrectes: b » sur le port série /dev/ttyAMA0 bouclé

Erreur sur /dev/ttyS0

Erreur sur /dev/ttyS

J'ai aussi essayé avec minicom, qui reste muet et remplacé le Pi Zero avec un autre... même résultat.

stty -F /dev/serial0 donne :

speed 9600 baud; line = 0;

min = 0; time = 0;

-brkint -icrnl -imaxbel

-opost -onlcr

-isig -icanon -iexten -echo -echoe -echok -echoctl -echoke

Y a-t-il un erreur ou oubli de ma part ?



François MOCQ

Auteur de l'article

22 juillet 2023 à 8 h 39 min

Bonjour

l'article date un peu (2017) je ne l'ai plus en tête 😊

regardez aussi celui ci

<https://www.framboise314.fr/utiliser-le-port-serie-du-pi-zero/>

en tout cas il faut arriver à faire fonctionner le port série en boucle fermé sinon on ne peut pas l'utiliser

tenez moi au courant il faudrait que je refasse les essais sur un pi zero récent

cdt

francois



Framont

22 juillet 2023 à 10 h 16 min

Merci pour votre réponse. En effet, mon projet est bloqué à cause de ce problème de port série. J'ai lu aussi votre autre article au sujet... sans malheureusement trouver de quoi résoudre mon problème.



vincent

5 septembre 2023 à 21 h 46 min

Bonjour, idem pour moi, impossible d'accéder à /dev/serial.

Merci



Nkuindjeu Frank

9 janvier 2024 à 14 h 54 min

Salut !

Vraiment merci pour ce tuto complet cela m'a beaucoup aide. Cependant lors d'un essaie de communication entre mon pi 3 et le terminal sur mon pc (RealTerm) je rencontre un problème de décodage sur le terminal en question.

Voici le code sur mon pi :

« » » »import serial

```
ser = serial.Serial('/dev/ttyAMA0')

print(ser.name)
ser.write(b »Hello world\r\n »)

ser.close() « » » »
```

Et voici le résultat que j'obtiens :

» »' 5 »!¿!7åë » »

Quel peut en être la cause svp ?



François MOCQ Auteur de l'article

9 janvier 2024 à 15 h 07 min

Bonjour

la vitesse de transmission est la même des 2 côtés ?



Nkuindjeu Frank

11 janvier 2024 à 10 h 04 min

Merci monsieur pour votre réponse aussi rapidement et désole de vous répondre aussi tardivement.

Oui la vitesse de transmission est bien similaire des 2 cotés



François MOCQ Auteur de l'article

11 janvier 2024 à 10 h 12 min

Si la vitesse est bonne et que le test en bouclant le Tx et le Rx est bon, c'est qu'il y a une autre différence il faut recontrôler

9600 N 8 1 Vitesse N = Pas de parité 8 = 8 bits de data 1 = 1

stop bit

Vitesse (c'est bon)

mais il faut vérifier le reste aussi y compris l'adaptateur

(<https://forum.arduino.cc/t/resolu-port-serie-et-mauvais-caracteres/559187/12>)

je ne vois pas d'autre chose

cdt

francois

Ce site utilise Akismet pour réduire les indésirables. En savoir plus sur [comment les données de vos commentaires sont utilisées](#).

[Haut de la page](#)