



KUBII

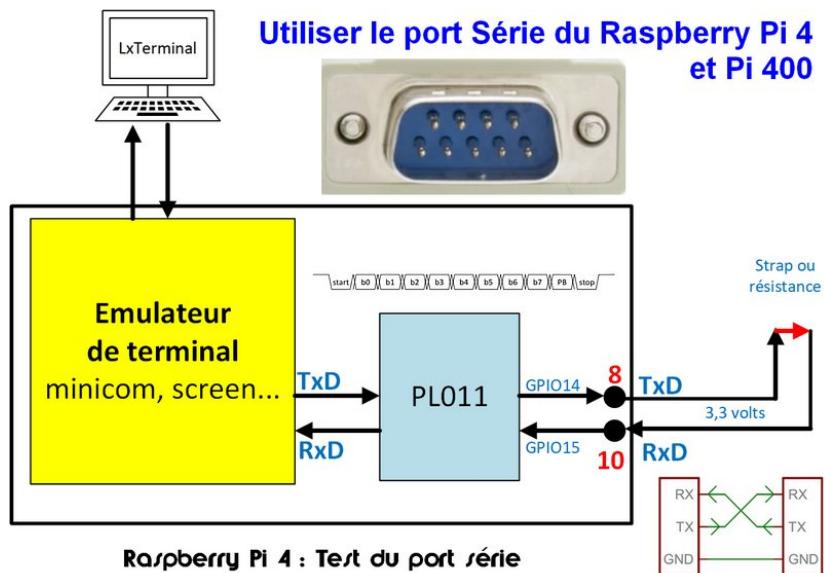
REVENDEUR OFFICIEL EN FRANCE
LIVRAISON GRATUITE DÈS 65€
SATISFAIT OU REMBOURSÉ



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

Raspberry Pi 4

Utiliser l'UART – Port série du

 Rechercher


Utiliser l'UART – Port série du Raspberry Pi 4

[cacher]

- 1 Utiliser l'UART du Raspberry Pi
- 2 Connexion d'origine des UART
- 3 Modification de cmdline.txt

3.1 Manuellement
3.2 Dans raspi-config
3.3 En mode graphique dans la configuration
4 Déconnecter le Bluetooth
4.1 Les overlays
4.2 Désactiver le Bluetooth
4.3 Désactiver le service système
4.4 Activer l'UART physique
5 Minicom
5.1 Installation
5.2 Permettre à l'utilisateur pi d'utiliser minicom
5.3 Configuration
5.4 Test
5.5 Mode Echo
6 Conclusion
7 Sources

Utiliser l'UART du Raspberry Pi

UART = Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel

Attention le Raspberry Pi ne supporte que des niveaux à 3,3 volts maximum. Il vous appartient de prévoir des adaptations de niveau de tension si nécessaire. Si vous envoyez du 5 volts ou du 12 volts sur les GPIO vous allez détruire la carte Raspberry Pi !

Pour de vrai, le Raspberry Pi 4 et le Pi400 (je vais rester sur ceux là car ça varie avec les modèles) comporte plusieurs UART... 6 en fait !

Name	Type
UART0	PL011
UART1	mini UART
UART2	PL011
UART3	PL011
RT4	PL011



Vous voyez qu'il y a deux types d'UART.

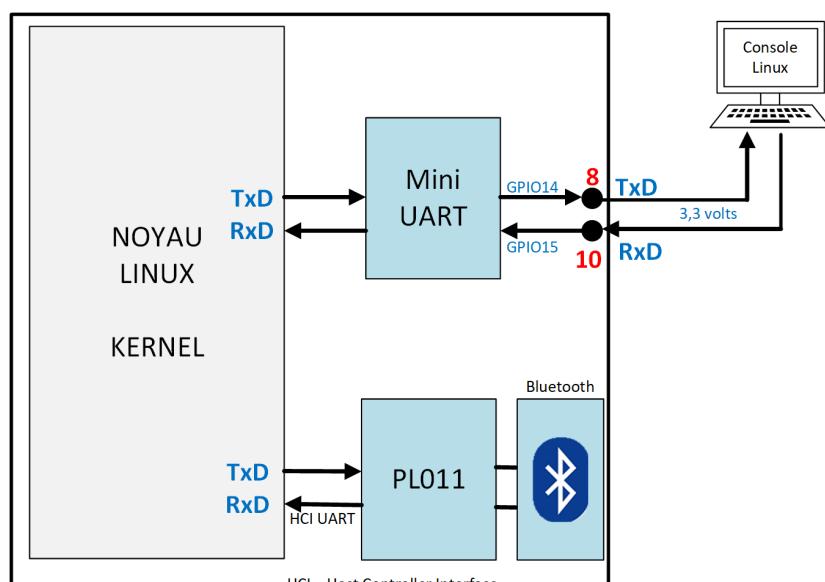
Le premier type est le **PL011** l'autre le **mini UART**

Le **PL011** est un **UART performant**, largement compatible avec le **16550**. Les plus anciens se rappelleront de ce modèle qui équipait l'IBM PC... Le **mini UART** a quant à lui, un ensemble de fonctionnalités réduit.

Tous les UARTs sur le Raspberry Pi sont en 3,3V seulement – *des dommages se produiront s'ils sont connectés à des systèmes 5V ou pire RS232 en +12v et -12v*. Un adaptateur peut être utilisé pour se connecter à des systèmes 5V ou 12v. Des adaptateurs USB à 3,3 V série bon marché sont également disponibles auprès de divers vendeurs en ligne.

Sur le Raspberry Pi, un UART est connecté aux GPIO 14 (transmission TxD) et 15 (réception RxD). C'est l'UART primaire. Par défaut, ce sera également l'UART sur lequel une console Linux peut être connectée. C'est ce qui est programmé par défaut dans **/boot/cmdline.txt**. Notez que GPIO 14 est relié à la broche 8 sur et GPIO 15 est sur la broche 10 du connecteur GPIO.

Connexion d'origine des UART



Raspberry Pi 4 : Configuration par défaut

Par défaut le noyau Linux est accessible (pour une ouverture de session ou pour l'affichage des messages lors du démarrage du système) sur le port série (GPIO 14 et 15, broches 8 et 10 du Raspberry Pi). L'interface Bluetooth est quant à lui connecté à un UART PL011 qui communique via [HCUART](#).

On peut dire que Linux « confisque » la console à son profit 😊 ... La connexion de la console Linux est paramétrée dans le fichier **/boot/cmdline.txt**. On peut modifier ce fichier manuellement (voir encadré ci-dessous) ou via [raspi-config](#), ou encore par la configuration en mode graphique, ce que je vous conseille si vous êtes débutant(e).

[stextbox id='warning' caption='Attention']Si vous modifiez le fichier **cmdline.txt**, il faut savoir que ce fichier envoie directement des commandes au noyau lors du démarrage. Il ne doit faire qu'une seule ligne et ne pas comporter de saut de ligne, retour à la ligne ou autre caractère exotique. A défaut le système risque de ne pas démarrer. Comme la partition Boot est accessible depuis Windows (ou Linux) je vous conseille de faire une sauvegarde du fichier avant toute modif pour pouvoir revenir en arrière. D'autre part, n'utilisez pas d'utilitaire comme le Bloc Note de Windows qui ajoute des caractères et plante le fichier. Préférez **NotePad++** sous Windows ou **nano** sous Linux.[/stextbox]

Modification de cmdline.txt

En fonction de votre assurance, choisissez l'une ou l'autre de ces trois méthodes.

Manuellement

Commencez par faire une sauvegarde du fichier **cmdline.txt** (on ne sait jamais, ça permettra de revenir en arrière...).

```
sudo cp /boot/cmdline.txt /boot/cmdline.txt.old
```

Le fichier d'origine (installation « fraîche » de Raspberry Pi OS) contient SUR UNE SEULE LIGNE :

```
console=serial0,115200 console=tty1
root=PARTUUID=75afe894-02 rootfstype=ext4 eleva-
ck.repair=yes rootwait quiet splash
```

```
plymouth.ignore-serial-consoles
```

Ouvrez le avec nano (sur le Raspberry Pi) ou Notepad++ sur Windows.

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

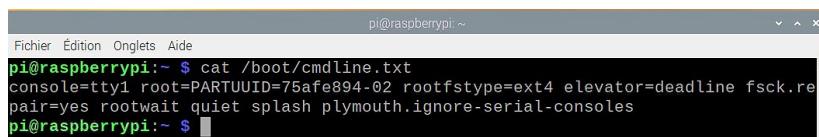
Supprimez ce texte :

```
console=serial0,115200
```

Redémarrez le système pour prendre la modification en compte.

```
sudo reboot
```

Pour vérifier, affichez le contenu du fichier cmdline.txt



```
pi@raspberrypi:~ $ cat /boot/cmdline.txt
console=tty1 root=PARTUUID=75afe894-02 rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait quiet splash plymouth.ignore-serial-consoles
pi@raspberrypi:~ $
```

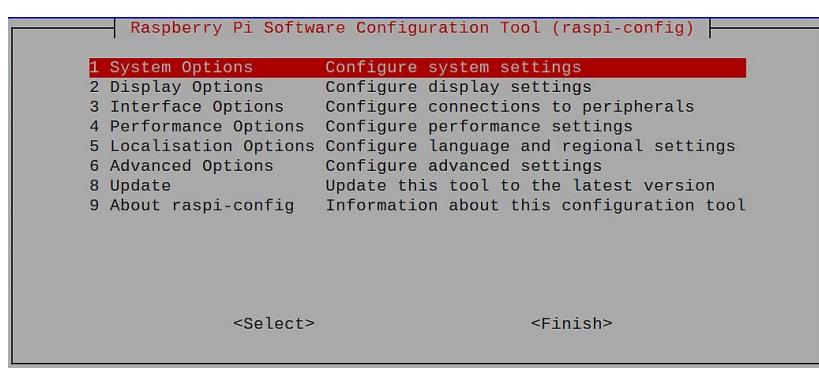
Il ne contient plus la déclaration de la console sur le port série.

Vous pouvez continuer.

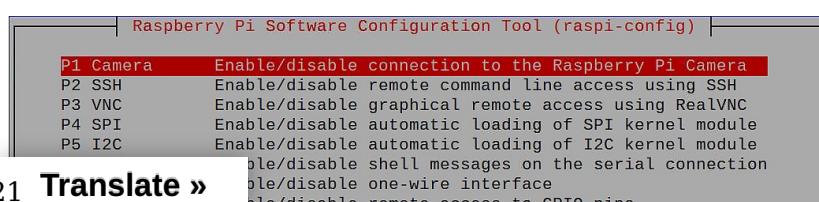
Dans raspi-config

Dans une fenêtre de terminal, lancez **raspi-config** :

```
sudo raspi-config
```

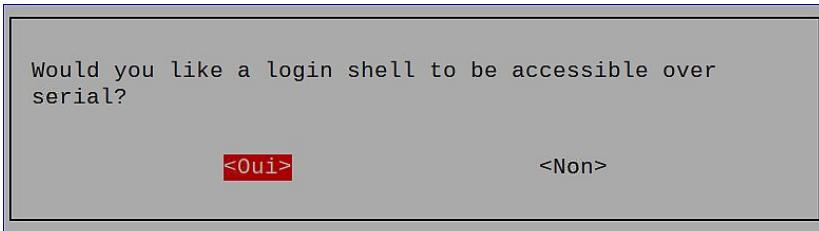


Allez sur 3 => **Interface Options**

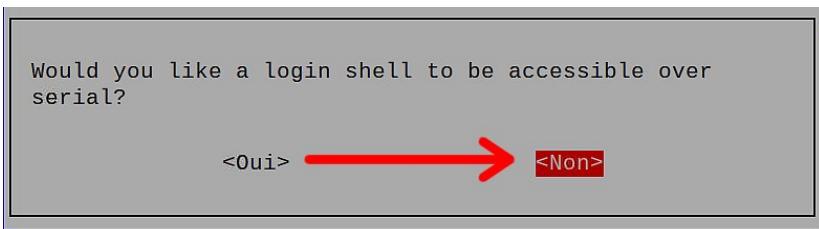




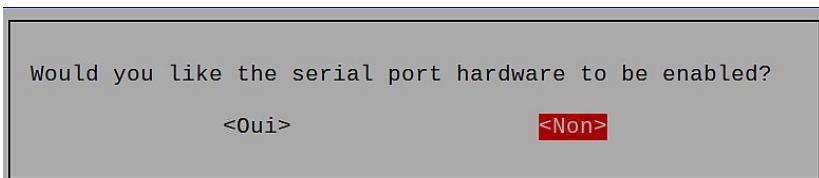
Sélectionnez P6 => **Serial Port**



Vous obtenez cette fenêtre avec **Oui** sélectionné par défaut mais on ne veut plus que le login soit possible sur le port série : répondez **Non** (touche Tab)



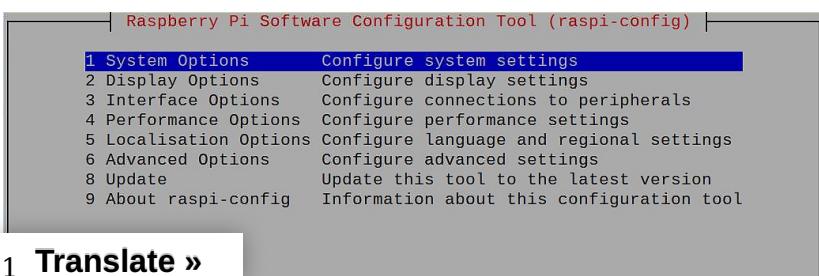
Puis validez (touche Entrée)



raspi-config vous demande maintenant si vous souhaitez activer le port série hardware. Répondez Non et validez.

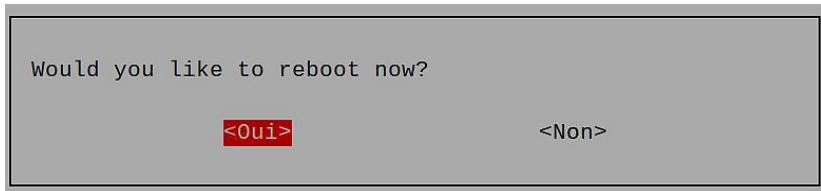


raspi-config vous confirme que le login n'est plus possible sur le port série qui est désactivé. Validez avec la touche Entrée.



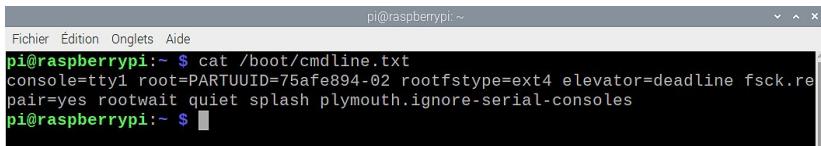


Vous revenez au menu principal. Mettez **Finish** en surbrillance (touche Tab) et validez (touche Entrée)



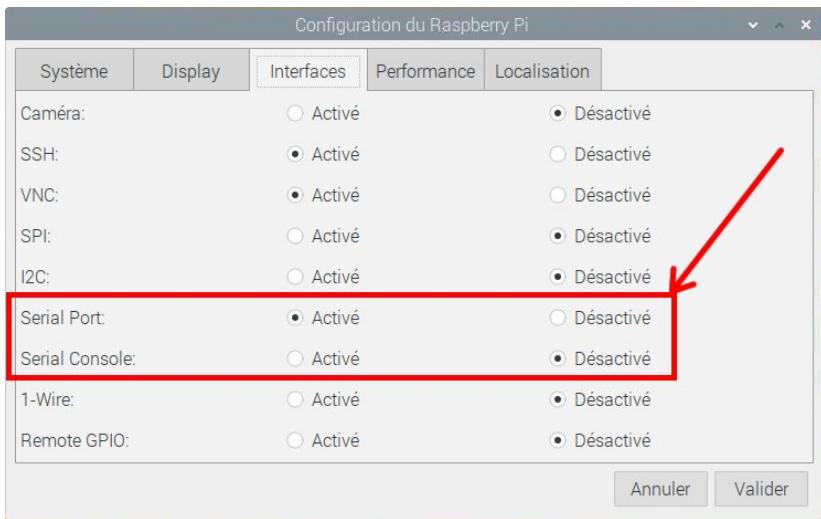
Il reste à rebooter pour prendre ces modifications en compte.

Après le redémarrage, on va quand même vérifier ce qui a été fait dans **cmdline.txt** :



Le résultat est le même que pour la procédure manuelle, la déclaration de la console a disparu (console=serial0,115200) et c'est ce qu'on voulait. C'est juste un peu plus long avec **raspi-config**. On peut passer à la suite.

En mode graphique dans la configuration



En mode graphique, Dans les interfaces, activez le port série, désactivez la console. Après le démarrage Vous disposez du port série sur les GPIO.

Déconnecter le Bluetooth

Les overlays

Pour le moment on a libéré le port UART GPIO (qui utilisait mini UART) de la console. On va maintenant libérer l'UART PL011 qui était relié à l'interface Bluetooth. Cette fois c'est dans **/boot/config.sys** que ça va se passer.

Diverses définitions d'overlay UART se trouvent dans le Device Tree. J'avais écrit [un article sur ce mécanisme](#) en 2015. Les deux overlays les plus utiles sont **disable-bt** et **minuart-bt**.

disable-bt désactive le périphérique Bluetooth et fait du premier PL011 (UART0) l'UART primaire. Vous devez également désactiver le service système qui initialise le modem, afin qu'il ne se connecte pas à l'UART, en utilisant **sudo systemctl disable hcuart**.

minuart-bt permet à l'interface Bluetooth d'utiliser le mini UART, et fait du premier PL011 (UART0) l'UART primaire. Notez que cela peut réduire le débit maximum (en bauds). Vous devez également régler l'horloge centrale du GPU à une fréquence fixe en utilisant soit `force_turbo=1` ou `core_freq=250`.

Les overlays `uart2`, `uart3`, `uart4`, et `uart5` sont utilisés pour activer les quatre UARTs supplémentaires sur le Pi 4. Il y a d'autres overlays spécifiques aux UART dans le dossier. Référez-vous à **/boot/overlays/README** si vous voulez plus de détails sur les overlays, ou exécutez **dtoverlay -h overlay-name** pour plus d'informations sur leur utilisation.

Désactiver le Bluetooth

Vous devez ajouter une ligne au fichier **/boot/config.txt** pour appliquer cet overlay :

Ouvrez le fichier avec nano

```
sudo nano /boot/config.txt
```

Et tout en bas du fichier, ajoutez :

```
dtoverlay=disable-bt
```

prenne la modification en compte.

`sudo reboot`

[stextbox id='info' caption='Info']/dev/serial0 et /dev/serial1 sont des liens symboliques qui pointent vers /dev/ttys0 ou /dev/ttyAMA0.[/stextbox]

Désactiver le service système

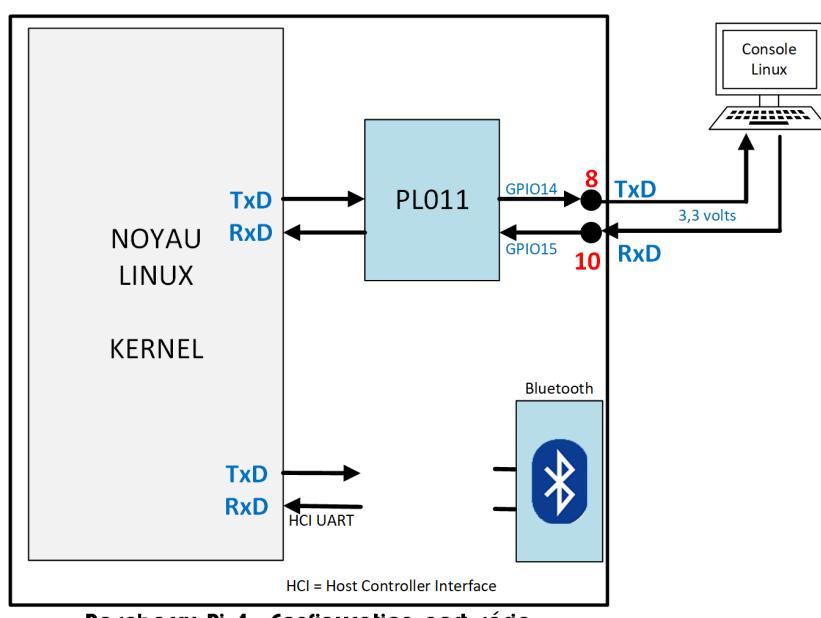
disable-bt que nous venons d'utiliser désactive le périphérique Bluetooth et fait du premier PL011 (UART0) l'UART primaire. Vous devez également désactiver le service système qui initialise le modem, afin qu'il ne se connecte pas à l'UART, en utilisant la commande

`sudo systemctl disable hcuart`

Ce qui se traduit par l'affichage de ce message :

```
pi@raspberrypi:~ $ sudo systemctl disable hcuart
Removed /etc/systemd/system/multi-user.target.wants/hcuart.service.
pi@raspberrypi:~ $
```

Votre système est maintenant prêt pour l'utilisation du port série sur le GPIO :



La configuration est maintenant celle-ci. Pour pouvoir communiquer avec l'extérieur par l'intermédiaire du port série, on va mettre en place un émulateur de terminal sur le Raspberry Pi. Ce

et d'afficher celles qui arrivent. C'est tout ce qu'il faut pour dialoguer avec un Arduino, un Raspberry Pi Pico, une carte de téléphonie... (il y a des articles qui en parlent sur le blog).

Activer l'UART physique

Dans /boot/config.txt si ce n'est pas fait automatiquement,

modifiez la ligne

```
enable_uart=0
```

pour qu'elle devienne

```
enable_uart=1
```

Pour activer l'UART physique (voyez [sur cette page](#) la partie : *Other DT commands available in config.txt*).

Pour prendre tout ceci en compte, redémarrez votre Raspberry Pi

```
sudo reboot
```

Minicom

L'émulateur de terminal que j'ai choisi s'appelle **minicom**. Ce n'est pas le seul et en fonction de vos habitudes ou de vos préférences, vous pouvez en choisir un autre.

Installation

Dans une fenêtre de terminal, entrez :

```
sudo apt install minicom
```

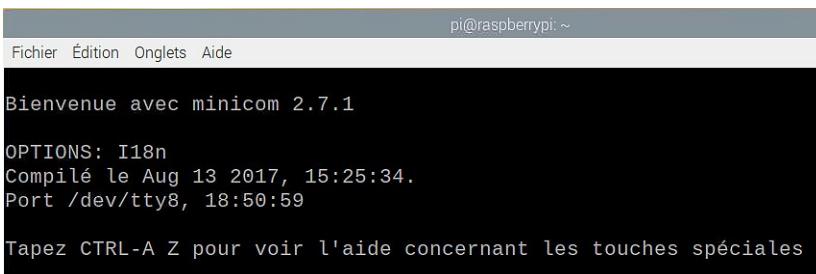


```
pi@raspberrypi: ~
Fichier Édition Onglets Aide
pi@raspberrypi: $ sudo apt install minicom
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  lrzsz
Les NOUVEAUX paquets suivants seront installés :
  lrzsz minicom
0 mis à jour, 2 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 329 ko dans les archives.
Après cette opération, 1 332 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] ■
```

Répondez **O**, l'installation est très rapide.

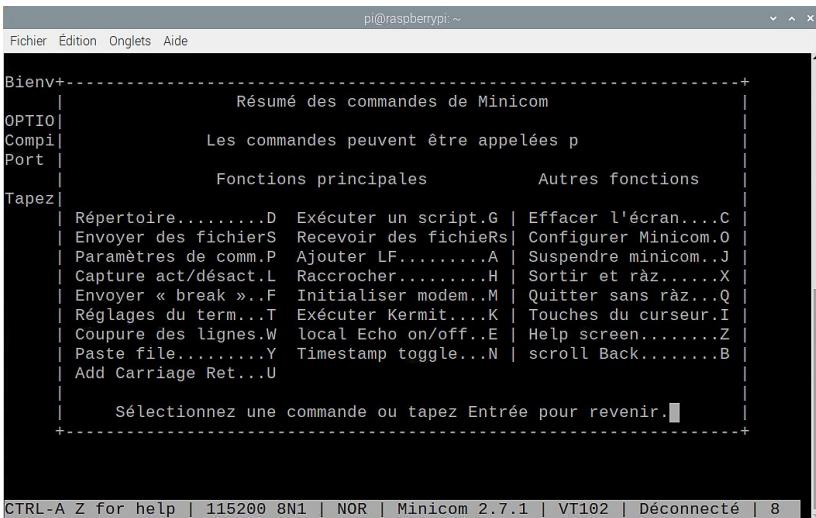
Pour démarrer **minicom** :

`sudo minicom`



```
pi@raspberrypi: ~
Fichier Édition Onglets Aide
Bienvenue avec minicom 2.7.1
OPTIONS: I18n
Compilé le Aug 13 2017, 15:25:34.
Port /dev/tty8, 18:50:59
Tapez CTRL-A Z pour voir l'aide concernant les touches spéciales
```

Minicom se lance et affiche ceci dans la fenêtre de terminal. Pour accéder à l'aide, tapez **CTRL A** puis **Z** :



```
pi@raspberrypi: ~
Fichier Édition Onglets Aide
Bienvenu+-----+ Résumé des commandes de Minicom +-----+
OPTIO+-----+ Les commandes peuvent être appelées p +-----+
Compi+-----+ Fonctions principales Autres fonctions +-----+
Port+-----+
Tapez+-----+
  Répertoire.....D Exécuter un script.G | Effacer l'écran....C
  Envoyer des fichiersS Recevoir des fichiersR | Configurer Minicom.O
  Paramètres de comm.P Ajouter LF.....A | Suspendre minicom..J
  Capture act/désact.L Raccrocher.....H | Sortir et razz....X
  Envoyer « break »..F Initialiser modem..M | Quitter sans razz...Q
  Réglages du term...T Exécuter Kermit....K | Touches du curseur.I
  Coupe des lignes.W local Echo on/off..E | Help screen.....Z
  Paste file.....Y Timestamp toggle...N | scroll Back.....B
  Add Carriage Ret...U

  Sélectionnez une commande ou tapez Entrée pour revenir.■ +-----+
```

CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7.1 | VT102 | Déconnecté | 8

Vous obtenez un résumé des commandes de **minicom**. Pour quitter proprement vous voyez qu'il faut taper **X**, **O** pour configurer **minicom**, **P** pour définir les paramètres de communication etc. La page d'aide de **minicom** (*man page*) [est en ligne et en français](#), merci au traducteur 😊

Permettre à l'utilisateur pi d'utiliser minicom

Pour lancer **minicom** il faut le faire précédé de sudo car **minicom**

accède au port série qui est un port matériel, auquel un utilisateur « normal » n'a pas accès. Pour que **pi** puisse y accéder on va l'ajouter dans le groupe **dialout**.

```
sudo usermod -a -G dialout pi
```

et **redémarrez** pour prendre la modification en compte.

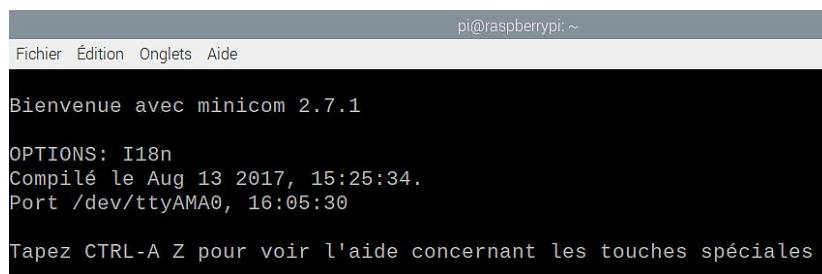
Vous pouvez lancer minicom directement sur le port série avec la commande

```
minicom -D /dev/ttyAMA0
```

Configuration

Lancez minicom, normalement il suffit de faire :

```
minicom -D /dev/ttyAMA0
```



pi@raspberrypi: ~

Fichier Édition Onglets Aide

Bienvenue avec minicom 2.7.1

OPTIONS: I18n
Compilé le Aug 13 2017, 15:25:34.
Port /dev/ttyAMA0, 16:05:30

Tapez CTRL-A Z pour voir l'aide concernant les touches spéciales

Pour voir la configuration de la communication, tapez **CTRL A** puis **P**



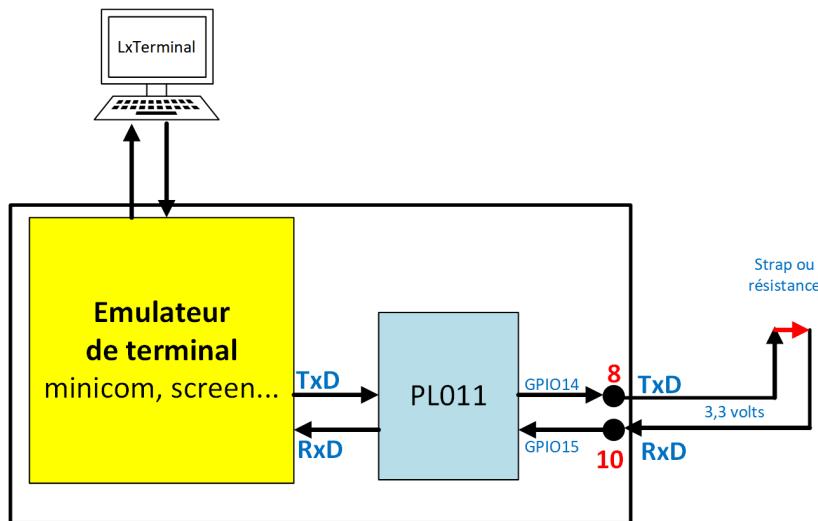
```
+---[Paramètres de communication]---+
| Actuellement : 115200 8N1
| Speed          Parity        Data
| A: <next>      L: None      S: 5
| B: <prev>       M: Even      T: 6
| C: 9600         N: Odd       U: 7
| D: 38400        O: Mark      V: 8
| E: 115200       P: Space
|
| Stopbits
| W: 1            Q: 8-N-1
| X: 2            R: 7-E-1
|
| Quitter      <Entrée> pour sortir ?
```

Par défaut **minicom** est configuré en 115200 8N1

- 115200 bauds
- 8 bits de données
- Parité : sans parité
- 1 stop bit

Pour des liaisons avec des systèmes externes, vous devrez sans doute modifier ces paramètres. Pour nos tests on va laisser ça sans y toucher.

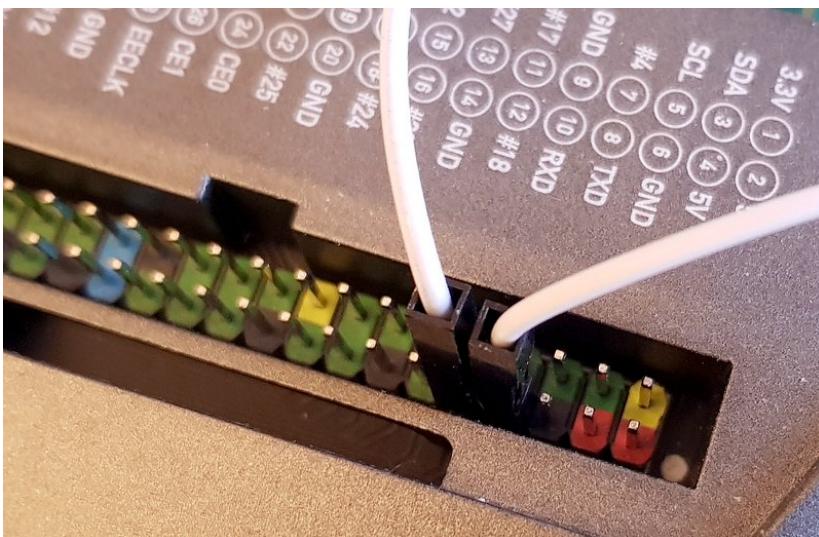
Test



Raspberry Pi 4 : Test du port série

Pour le test on va envoyer des caractères depuis **minicom**. Ils vont être envoyés vers le port GPIO14 TxD (émission de données). Positionnez un **strap** (un court-circuit, un cavalier) entre les broches 8 et 10 du GPIO. Le signal sortant par GPIO14 va rentrer par GPIO15 RxD (réception de données) et revenir vers **minicom** qui l'affichera sur l'écran.

Si vous n'êtes pas confiant(e), remplacez le strap par une résistance de 1KΩ qui protègera vos GPIO en cas de bêtise.



Mise en place du court-circuit entre les broches 8 et 10 du port GPIO pour les tests du port série. (Raspberry Pi 4 en boîtier Argon One)

En résumé : quand le strap est présent, les caractères que je tape au clavier sont affichés sur l'écran. Enlevez le strap : plus rien ne s'affiche sur l'écran quand je tape du texte.

Si ceci fonctionne, votre port série est opérationnel et vous pouvez dialoguer avec des cartes extérieures (Arduino, Raspberry Pi Pico...) via le port série avec ***minicom***.

Mode Echo

En mode normal, chaque caractère frappé au clavier sort par le port TxD et revient sur l'écran. C'est ce qui se passe généralement quand vous connectez le port série à une carte extérieure, c'est elle qui assure le retour des caractères pour confirmer qu'ils ont été bien reçus.

```
pi@raspberrypi: ~
Fichier Édition Onglets Aide

Bienvenue avec minicom 2.7.1

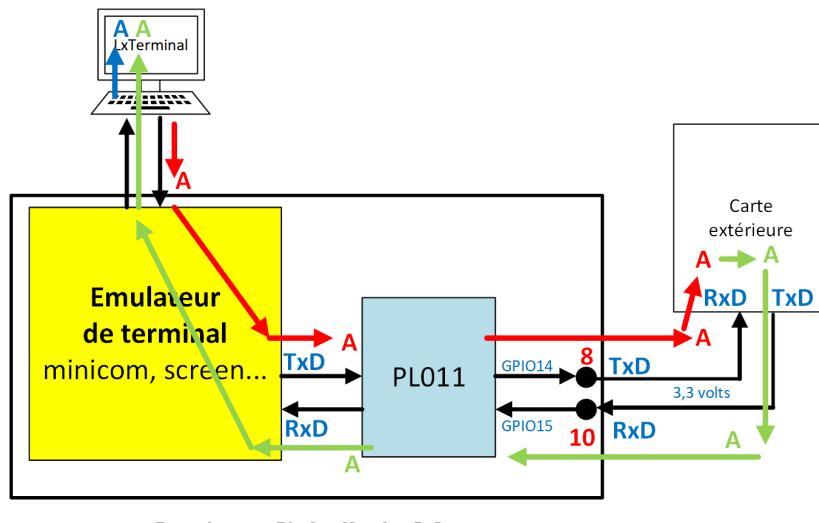
OPTIONS: I18n
Compilé le Aug 13 2017, 15:25:34.
Port /dev/ttyAMA0, 16:05:30

Tapez CTRL-A Z pour voir l'aide concernant les touches spéciales

Essai du port série du Raspberry Pi ... █
```

Ici lors des tests j'ai saisi la phrase « **Essai du port série du Raspberry Pi ...** » notez la transformation du é en 2 caractères ... je vous laisse trouver le pourquoi de la chose 😊 Chaque caractère a été écrit une seule fois à l'écran.

Pour activer le mode Echo, dans minicom tapez **CTRL A** puis **E**. Le mode Echo est activé :



Lorsque vous tapez un caractère au clavier (un A) il est envoyé sur l'écran du terminal (A bleu). Minicom l'envoie également vers le port série TxD (A rouge). Le A est transmis à la carte extérieure. Il arrive sur son port RxD. La carte exploite le caractère reçu mais le renvoie également sur son port TxD pour confirmer qu'il a bien été reçu (A vert). Le A revient donc sur le Raspberry Pi via le port RxD (GPIO15) puis vers minicom qui l'affiche sur l'écran.

En mode Echo, à chaque fois que vous tapez un caractère, si la transmission se passe bien, il est affiché deux fois de suite sur l'écran.



La commande du mode Echo est une bascule. Pour l'enlever faites de nouveau **CTRL A** puis **E**. Les caractères saisis au clavier ne sont affichés qu'une seule fois.

Pour quitter **minicom**, tapez **CTRL A** puis **X** et validez avec la touche **Entrée**.

Conclusion

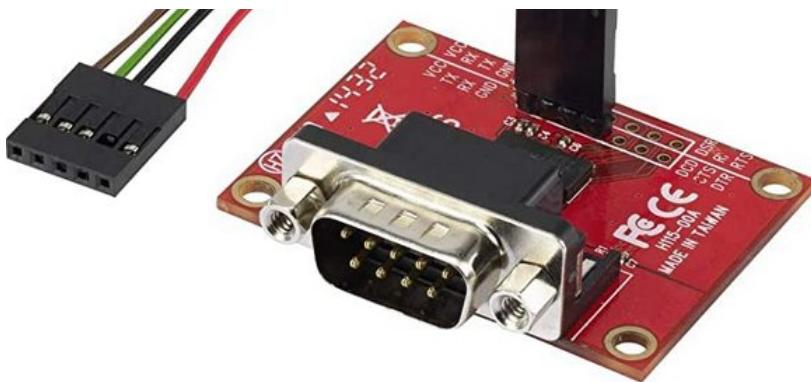
Nous voilà arrivés au terme de cet article. Pour un ancien comme moi qui ai utilisé des lignes RS232, RS422 et RS485 pour des liaisons en industrie et dans les systèmes de pointage horaire, la notion de liaison série a un sens. C'est vrai qu'aujourd'hui on parle plutôt d'USB ou d'I2C. Mais je pense que la bonne vieille liaison série a encore de l'avenir car elle est facile à mettre en oeuvre et à « déverminer ». A une époque (lointaine) j'ai utilisé régulièrement une valise d'analyse de liaison série ainsi que des boîtes de coupe pour dépanner ces liaisons qui ont toujours leur place dans les systèmes actuels.

N'hésitez pas à faire part de vos remarques sur cet article, positives et négatives, ça permettra de l'améliorer pour les utilisateurs intéressés.



A noter qu'on trouve pour une vingtaine d'euros (le lot de 5 cartes) [des cartes d'extension RS232 pour le Raspberry Pi sur Aliexpress](#). Elles sont équipées d'une prise DB9 et d'un MAX232.





Il y a également des équivalents chez Amazon avec [cette carte Raspberry PI GPIO TX/RX ZU RS232](#) à 7,50€ = 6€ de frais de ports.

Sources

- <https://www.raspberrypi.com/documentation/computers/configuration.html#configuring-uarts>
- https://di-marco.net/blog/it/2020-06-06-raspberry_pi_3_4_and_0_w_serial_port_usage/
- <https://doc.ycharbi.fr/index.php/Minicom>

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 →](#)

6 réflexions au sujet de « Utiliser l'UART – Port série du Raspberry Pi 4 »

**Matthieu**

5 novembre 2021 à 6 h 53 min

On a toujours besoin d'un uart

J'avais utilisé les versions précédentes de cet article pour les Rpi3, je vois que ça n'a pas beaucoup changé pour le 4.

Merci pour la maj !

**François MOCQ**

Auteur de l'article

5 novembre 2021 à 10 h 22 min

merci Matthieu

**lolio**

7 novembre 2021 à 20 h 12 min

Merci, article très intéressant comme toujours !

Juste une remarque sur les convertisseurs MAX232 cités en fin d'article: attention à la tension d'alimentation utilisée en entrée car ces cartes doubles seulement cette dernière en sortie: si on l'alimente en 3.3V, la tension de sortie côté DB9 sera comprise entre 6.6 et -6.6V... cela peut poser problème lorsque l'on souhaite s'en servir pour faire du RS232 avec un port physique d'un ancien PC (gérant ses niveaux entre -12 et +12V) => cela m'a posé plusieurs fois des soucis sur certains montage car les fronts de niveaux (0 ou 1) étaient mal reconnus... La solution que j'ai trouvé: alimenter le module en +5V et utiliser un pont diviseur de tension sur la patte RX du raspberry pour ramener la tension en 3.3V max. Il y avait peut-être moyen de faire mieux mais je reste encore novice en électronique (même

si j'ai beaucoup progressé grâce à ce blog)... 😊

Sinon autre question: comment utiliser les UART2 à 5 ? Y a-t-il déjà des gpio définis pour les utiliser ou doit-on les configurer ? Si oui, serait-il possible d'avoir un exemple ?

J'aurai un projet pour utiliser le raspberry comme concentrateur d'équipements série vers IP... 😊

En tout cas, bravo François ! 😊



François MOCQ

Auteur de l'article

7 novembre 2021 à 21 h 00 min

Bonjour et... merci

Les autres uart s utilisent en les declarant dans config.txt

[voir ce forum pour plus d'infos](#)

Plus d infos sur les uarts dont les broches tx et rx

Dans <https://datasheets.raspberrypi.org/bcm2711/bcm2711-peripherals.pdf>

Page 142

Cdt

Francois

Ping : [Carte SeeedStudio Lora Grove E5 - Framboise 314, le Raspberry Pi à la sauce française....](#)



Yann

22 novembre 2023 à 17 h 36 min

Bonsoir François

Merci pour l'acceptation

Merci pour cet article, qui peut-être va m'aider à sortir de l'im-passe dans laquelle je suis, actuellement j'ai une carte HAT

RS485, or impossible de lui sortir du signal depuis l'application

OLADEMIX Du coup, avec votre article j'ai déjà dans un

premier temps vérifié que mes UARTs fonctionnent et étape par étape vérifié quel UART se retrouve sur le GPIO.

Petit doute dans ce que vous écrivez, via **raspi-config**, vous dites

disable serial login
disable serial interface

Or en mode **graphique** vous indiquez

Enable serial port
Disable serial console

Du coup sauf erreur de ma part via raspi-config on devrait répondre **no** et ensuite **yes** ?!

Autre point, vous parlez du fichier config.sys, ne l'ayant pas trouvé, n'est pas un *.txt dont vous parlé ? config.txt ?

Encore merci pour cet article, qui m'aide bien et doit certainement aider d'autres personnes comme moi =)

Je précise, même s'il s'agit d'un article pour RPi4, je m'adapte avec mon Pi3, c'est peut-être aussi ça les différence *.sys/*.txt ?



François MOCQ

Auteur de l'article

22 novembre 2023 à 18 h 25 min

Bonjour Yann

j avoue que ca date un peu...

vous avez raison c est config.txt

je ne sais pas d ou sort ce config sys (enfin si... des vieux restes de MS DOS sans doute !)

c'est bien /boot/config.txt comme indiqué dans les lignes de commande 😊

je n'ai pas trop le temps de refaire les manips en ce moment
mais vous pouvez tester les différentes options
Après quand vous êtes certain que l'UART fonctionne vous de-
vriez pouvoir causer avec la carte HAT
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](#)