



Bienvenue sur la plateforme de documentation numérique de Hackable !

(/)

< Article précédent (/Hackable/HK-017/Transformez-un-vieux-materiel-de-30-ans-en-clavier-USB)
Article suivant > (/Hackable/HK-017/Changez-la-configuration-des-leds-de-votre-Raspberry-Pi)

CONFIGUREZ UN CLAVIER BLUETOOTH POUR VOTRE PI

Hackable n° 17 (/Hackable/HK-017) | mars 2017 | Bodor Denis (/auteur/Bodor-Denis)



Système (/search/node?Domaines%5B0%5D=72451)

Embarqué (/search/node?Domaines%5B0%5D=72457)

Radio et wireless (/search/node?Domaines%5B0%5D=72465)

Qu'il s'agisse de la Raspberry Pi 3 ou d'un modèle précédent équipé d'un adaptateur USB Bluetooth, l'un des usages les plus courants pour ce type de connectivité se résume souvent à l'utilisation de périphériques d'entrée comme un clavier ou une souris. Ceci est très facile à configurer via l'interface graphique, mais qu'en est-il lorsqu'on utilise Raspbian Lite et qu'on préfère la ligne de commandes ?

Lorsqu'on joue avec plusieurs Pi pour différents projets qui tantôt nécessitent une intervention, une reconfiguration ou une mise à jour, il n'est pas toujours agréable de s'amuser à brancher/débrancher un clavier, en particulier lorsque l'installation est bien propre et dans un coffret par exemple. La solution consiste donc à utiliser une liaison sans fil et, si l'on souhaite ne pas devoir jongler avec les *dongles*, cette solution est tout naturellement le Bluetooth.

Ceci est pris en charge très facilement via l'interface graphique PIXEL (qui se résume à une adaptation/personnalisation de LXDE) puisqu'il est directement possible de configurer un clavier et une souris depuis une petite icône située en haut de l'écran. Mais voilà, lorsqu'on décide d'utiliser Raspbian Jessie Lite et non la version avec interface graphique, plus lourde, ce n'est pas pour devoir installer tout ce petit monde sous prétexte de devoir configurer un clavier (chose qui peut effectivement être fait en installant le paquet `raspberrypi-ui-mods` et donc toutes ses dépendances). Mieux vaut se creuser un peu les méninges, se documenter et faire cela en ligne de commandes.

Comme souvent, on retrouve sur le Web, quelques guides, billets de blog et *howto* qui détaillent ce genre de configuration et, comme souvent, il ne s'agit que de copier/coller d'autres guides, billets de blog et *howto* qui n'expliquent rien et assument que certaines manipulations font effectivement partie de la procédure, alors que ce n'est pas le cas. Ainsi, pour le sujet qui nous intéresse ici, on retrouve généralement la modification de deux fichiers système, `/etc/default/bluetooth` et `/etc/default/bluetooth`, alors que ceux-ci n'ont rien à voir avec la configuration d'un périphérique comme un clavier ou une souris. Pire encore, cette modification n'est souvent pas nécessaire et en partie totalement obsolète...



(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-017/82938/BTkey_noir1.jpg)

Un clavier Bluetooth de bonne facture sera le compagnon idéal d'une Raspberry Pi 3, puisque celle-ci intègre de base la connectivité adaptée ainsi que le Wifi. Pour les précédents modèles, il vous faudra acquérir un ou des adaptateurs USB Bluetooth.



1. CONNEXION ET APPAIRAGE

Le matériel de démonstration utilisé ici est un sympathique clavier Bluetooth AZERTY de marque Perixx et plus exactement un Periboard-804 KT-1063. Je ne peux guère vous en dire plus puisque je possède ce périphérique depuis bien longtemps et ne sais plus où, quand et comment j'en ai fait l'acquisition. Mais ceci n'est pas important, car tous les périphériques de saisie (clavier, souris, etc.) Bluetooth fonctionnant de la même façon, ce qui va suivre pourra s'appliquer à n'importe quel produit récent ou ancien.



(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-017/82938/BTkey_bouton1.jpg)

Les claviers Bluetooth disposent généralement d'un petit bouton à l'arrière permettant de rendre le périphérique découvrable pour procéder à l'appairage. Il faudra ensuite taper un code au clavier pour confirmer l'association.

Avec l'implémentation actuelle du support Bluetooth sous GNU/Linux nommée BlueZ (version 5.23 dans Raspbian) tout est directement configurable via une simple commande : **bluetoothctl**. On lancera donc celle-ci accompagnée de l'option **-a** de façon à ce qu'elle intègre directement un agent qui nous permettra de gérer l'authentification lors de l'appairage (on peut également lancer la commande sans l'option puis utiliser **agent on**).

Une fois la commande lancée, on commencera par un petit scan destiné à rechercher des appareils Bluetooth pouvant être découverts. Côté clavier, un petit bouton à l'arrière du matériel permet de rendre le périphérique temporairement visible :

```
[bluetooth]# scan on
Discovery started
[CHG] Controller B8:27:EB:10:1F:A8 Discovering: yes
[NEW] Device 90:7F:61:80:8A:9E Bluetooth Keyboard
```

Le clavier a été détecté par la Pi et nous pouvons nous renseigner sur ce dernier avec **info** suivi de son adresse :

```
[bluetooth]# info 90:7F:61:80:8A:9E
Device 90:7F:61:80:8A:9E
    Name: Bluetooth Keyboard
    Alias: Bluetooth Keyboard
    Class: 0x002540
    Icon: input-keyboard
    Paired: no
    Trusted: no
    Blocked: no
    Connected: no
    LegacyPairing: yes
```



Le périphérique n'est ni appairé, ni connecté, ni de confiance. Nous commençons donc par l'appairage avec :

```
[bluetooth]# pair 90:7F:61:80:8A:9E
Attempting to pair with 90:7F:61:80:8A:9E
[CHG] Device 90:7F:61:80:8A:9E Connected: yes
[agent] PIN code: 358206
```

L'association se fera après authentification. L'idée est ici de prouver qu'on a physiquement accès aux deux matériels. Le code qui nous est alors présenté à l'écran par l'agent doit être utilisé sur le clavier. Cette procédure peut être différente d'un matériel à l'autre. Ici, il suffit de taper les six chiffres sur le clavier (qui est alors en QWERTY et ne demande donc pas d'utiliser la touche Maj pour entrer les chiffres) et de finir par la touche *Entrée*.

Ceci fait, **bluetoothctl** nous confirme le succès de l'opération :

```
[CHG] Device 90:7F:61:80:8A:9E Paired: yes
Pairing successful
```

Pour que la liaison puisse s'établir automatiquement la prochaine fois que les périphériques (Pi et clavier) seront réunis, nous devons faire confiance au clavier. Nous utilisons donc la commande **trust** suivie de l'adresse du périphérique pour entériner la liaison :

```
[bluetooth]# trust 90:7F:61:80:8A:9E
[CHG] Device 90:7F:61:80:8A:9E Trusted: yes
Changing 90:7F:61:80:8A:9E trust succeeded
```

Enfin, on jette un dernier coup d'œil à la configuration :

```
[bluetooth]# info 90:7F:61:80:8A:9E
```

```
Device 90:7F:61:80:8A:9E
  Name: Bluetooth Keyboard
  Alias: Bluetooth Keyboard
  Class: 0x002540
  Icon: input-keyboard
  Paired: yes
  Trusted: yes
  Blocked: no
  Connected: no
  LegacyPairing: yes
```

Les lignes **Paired** et **Trusted** sont à **yes**, le périphérique est donc bien appairé et la liaison est de confiance. Dès cet instant, le clavier doit être utilisable exactement comme un modèle connecté en USB. Il vous sera peut-être nécessaire cependant de configurer son agencement de clavier en lançant `sudo raspi-config` et en faisant un tour dans **Internationalisation Options** et **Change Keyboard Layout** pour choisir un clavier français AZERTY.

Notez la ligne **connected** dans la sortie. Ceci pourra vous permettre de vous assurer que le clavier Bluetooth est bel et bien connecté, en cas de problème d'utilisation. N'oubliez pas, non plus, que les périphériques Bluetooth passent généralement rapidement en veille et qu'une frappe sur un clavier qui est endormi mettra parfois quelques secondes pour le réveiller et rétablir la connexion.



(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-017/82938/BTkey_deux1.jpg)

En termes de qualité de fabrication et de solidité, ce qu'on trouve sur le Web est assez variable. Ici au premier plan un clavier très peu cher (~15€), peu solide, avec une frappe très moyenne et une alimentation via une paire de piles AAA. En arrière-plan, un modèle de meilleure qualité, rechargeable via une connexion micro-USB.

2. CONFIGURATION DU CLAVIER

En dehors de l'agencement des touches, il n'y a strictement rien à configurer pour faire fonctionner votre clavier. D'où sortent donc ces informations concernant la modification de certains fichiers ?

Le premier, `/etc/default/bluetooth`, permet de régler un problème rencontré avec certains adaptateurs Bluetooth. Le matériel moderne s'utilise généralement avec un protocole appelé HCI pour *Host Controller Interface*. Celui-ci décrit la façon dont l'hôte (votre Pi) et le contrôleur (l'adaptateur Bluetooth) communiquent. Cependant, certains modèles, lorsque vous les connectez à la machine, apparaissent comme des périphériques HID (*Human Interface Device*), une classe de périphériques USB correspondant généralement aux périphériques de saisie.

Pour contrôler l'adaptateur nous devons lui parler HCI et non HID, ces adaptateurs doivent donc changer de mode de communication pour être utilisés. C'est là qu'intervient un outil appelé `hid2hci` (placé dans `/lib/udev`). L'idée est, lors de la détection de l'adaptateur USB, de demander au système d'automatiquement lancer `hid2hci` pour le faire changer de mode (via `udev`). Ceci se configure en changeant une ligne dans le fichier `/etc/default/bluetooth` :

```
HID2HCI_ENABLED=1
```

Dans le cas d'une Raspberry Pi 3 et son adaptateur Bluetooth intégré, le changement de mode n'est pas nécessaire puisqu'il communique, de base, en HCI. Je pense que la confusion concernant l'utilisation de ce fichier vient du fait qu'il est question de HID, terme qui rappelle les périphériques USB HID et donc les claviers/souris USB. Pourtant ceci n'a rien à voir avec le périphérique et la prise en charge de claviers Bluetooth, cela ne concerne que l'adaptateur et sa connexion.

L'autre fichier généralement cité est `/etc/bluetooth/hcid.conf`. Là, c'est encore pire, car ce fichier n'est tout simplement pas utilisé. Il s'agissait, avec BlueZ 3.x, du fichier de configuration de `hcid`, le démon de gestion HCI. Celui-ci, avec l'arrivée de la version 4.0 de BlueZ (en août 2008 !), a laissé place à `bluetoothd`, encore utilisé aujourd'hui. `hcid.conf` n'a aucune utilité puisque `hcid` n'est plus là depuis plus de 8 ans...

Moralité, mieux vaut regarder les dates des pages (biller, wiki, etc.) avant de suivre une procédure et s'assurer que ce qui est décrit correspond bien à la documentation **officielle**. Car, bien entendu, en ajoutant un `hcid.conf` et en modifiant son `/etc/default/bluetooth`, cela fonctionne, mais en ajoutant un `pouet.conf` dans `/etc/bluetooth`, cela fonctionnera tout aussi bien...

Étiquettes :

Raspberry Pi (/search/node?field_ct_article_tags_target_id_selective%5B0%5D=72484),
Bluetooth (/search/node?field_ct_article_tags_target_id_selective%5B0%5D=72511),
clavier (/search/node?field_ct_article_tags_target_id_selective%5B0%5D=72514)

< Article précédent (/Hackable/HK-017/Transformez-un-vieux-materiel-de-30-ans-en-clavier-USB)
Article suivant > (/Hackable/HK-017/Changez-la-configuration-des-leds-de-votre-Raspberry-Pi)

RECHERCHER

UN ARTICLE HACKABLE

parmi plus de 320 articles !



AU SOMMAIRE DU MÊME NUMÉRO

Utilisez un programmeur d'EPROM avec votre Raspberry Pi (/Hackable/HK-017/Utilisez-un-programmeur-d-EPROM-avec-votre-Raspberry-Pi)

Créez un effet « feu » avec une matrice de leds (/Hackable/HK-017/Creez-un-effet-feu-avec-une-matrice-de-leds)

Créez des boutons à copier/coller (/Hackable/HK-017/Creez-des-boutons-a-copier-coller)

Créez un contrôleur de volume pour votre ordinateur (/Hackable/HK-017/Creez-un-controleur-de-volume-pour-votre-ordinateur)

Transformez un vieux matériel de 30 ans en clavier USB (/Hackable/HK-017/Transformez-un-vieux-materiel-de-30-ans-en-clavier-USB)

► **Configurez un clavier Bluetooth pour votre Pi** (/Hackable/HK-017/Configurez-un-clavier-Bluetooth-pour-votre-Pi)

Changez la configuration des leds de votre Raspberry Pi (/Hackable/HK-017/Changez-la-configuration-des-leds-de-votre-Raspberry-Pi)

Utilisez votre Arduino UNO comme périphérique USB (/Hackable/HK-017/Utilisez-votre-Arduino-UNO-comme-peripherique-USB)

Les codes tournants ou comment ne pas envoyer le même message deux fois (/Hackable/HK-017/Les-codes-tournants-ou-comment-ne-pas-envoyer-le-meme-message-deux-fois)

SUR LE MÊME SUJET

Reprenez le contrôle ! Faites tourner Linux nativement sur vos téléphones et tablettes (/GNU-Linux-Magazine/GLMF-237/Reprenez-le-contrôle-Faites-tourner-Linux-nativement-sur-vos-telephones-et-tablettes)

GNU/Linux Magazine n° 237 (/GNU-Linux-Magazine/GLMF-237) | mai 2020 | [Meauzoone Arnaud](#) (/auteur/Meauzoone-Arnaud)

Embarqué (/search/node?Domaines%5B0%5D=72457)

J'ai toujours été admiratif des personnes qui arrivent à faire tourner le noyau Linux sur de nouvelles plateformes. J'ai passé un nombre de soirées incalculables à étudier différents portages de Linux (Nintendo Switch, Nintendo 3DS, PlayStation...) et par la suite, j'ai moi-même passé beaucoup de temps à essayer de porter Linux sur tout ce qui me tombait sous la main. C'est une passion pour ...

Google Skaffold (/GNU-Linux-Magazine/GLMF-237/Google-Skaffold)

GNU/Linux Magazine n° 237 (/GNU-Linux-Magazine/GLMF-237) | mai 2020 | [Blaskow David](#) (/auteur/Blaskow-David)

Système (/search/node?Domaines%5B0%5D=72451)

J'aime par-dessus tout les choses simples et, selon moi, si je dois faire la même chose deux fois, c'est qu'il est temps d'automatiser. Quand je développe une application, j'ai régulièrement besoin de la déployer et de la tester ; aussi, j'ai mes astuces. Mais depuis quelque temps, je dois aussi conteneuriser cette même application, et voilà maintenant que je dois également la ...

Apprenez à utiliser kubeadm (/GNU-Linux-Magazine/GLMF-236/Apprenez-a-utiliser-kubeadm)

GNU/Linux Magazine n° 236 (/GNU-Linux-Magazine/GLMF-236) | avril 2020 | [Blaskow David](#) (/auteur/Blaskow-David)

Système (/search/node?Domaines%5B0%5D=72451)

Combien de fois m'avez-vous entendu dire « nous allons utiliser kubeadm pour faire ceci ou faire cela », et puis boum, un kubeadm init plus tard, tout est prêt ? Souvent ? Très souvent ? Trop souvent ? Alors pour une fois, pourquoi ne pas consacrer un article entier à ce merveilleux projet ? ...

Analyse d'un brouilleur GPS (/Hackable/HK-033/Analyse-d-un-brouilleur-GPS)

Hackable n° 33 (/Hackable/HK-033) | avril 2020 | [Friedt Jean-Michel](#) (/auteur/Friedt-Jean-Michel)

Radio et wireless (/search/node?Domaines%5B0%5D=72465)

KiCAD a récemment été illustré sur le NE555, le composant réputé comme le plus commercialisé dans l'histoire de l'électronique [1]. Nous pouvons nous interroger sur l'utilisation actuelle de ce composant analogique aux fonctionnalités réduites. Dans le cadre d'une étude sur la robustesse des systèmes de navigation par satellite, nous avons acquis un « bloqueur » GPS. Le budget que nous ...

Préparer un système GNU/Linux temps réel pour vos applications audio (/Linux-Pratique/LP-118/Preparer-un-systeme-GNU-Linux-temps-reel-pour-vos-applications-audio)

Linux Pratique n° 118 (/Linux-Pratique/LP-118) | mars 2020 | [Laichaoui Malik](#) (/auteur/Laichaoui-Malik)

Système (/search/node?Domaines%5B0%5D=72451)

Audio/Vidéo (/search/node?Domaines%5B0%5D=72453)

Pendant longtemps, je me suis amusé à compiler des noyaux Linux afin de paramétrer mes ordinateurs avec les options correspondantes au matériel les composant. Par la suite, j'ai découvert la possibilité d'optimiser encore plus le cœur de mes systèmes GNU/Linux avec les fonctionnalités dites « temps réel », notamment pour faire tourner des logiciels audio avec le serveur de son Jack. ...

Plus de 30 cas concrets pour être plus efficace avec Git (/GNU-Linux-Magazine/LMHS-107/Plus-de-30-cas-concrets-pour-etre-plus-efficace-avec-Git)

GNU/Linux Magazine HS n° 107 (/GNU-Linux-Magazine/LMHS-107) | mars 2020 | [Colombo Tristan](#) (/auteur/Colombo-Tristan)

Système (/search/node?Domaines%5B0%5D=72451)

Code (/search/node?Domaines%5B0%5D=72464)

Avec un outil aussi complet que Git, il est normal de se poser des questions sur l'utilisation des commandes, que ce soit en tant que débutant ou en tant qu'utilisateur expérimenté. Quoi de mieux alors que des recettes répondant à des questions précises, pour gagner du temps ? ...

PAR LE MÊME AUTEUR

UNE CAMÉRA WEB ESP32 POUR SURVEILLER SON ENVIRONNEMENT (/HACKABLE/HK-033/UNE-CAMERA-WEB-ESP32-POUR-SURVEILLER-SON-ENVIRONNEMENT)

Hackable n° 33 (/Hackable/HK-033) | avril 2020 | [Bodor Denis \(/auteur/Bodor-Denis\)](/auteur/Bodor-Denis)

Domotique (/search/node?Domaines%5B0%5D=72455)

Électronique (/search/node?Domaines%5B0%5D=72452)

L'ESP32-CAM permet de très simplement et très économiquement créer une simple webcam Wi-Fi. Ceci est très pratique, mais ne présente pas réellement d'intérêt lorsqu'on compare cela à la myriade de produits clé en main similaires et disponibles à bas prix. Ce qui est plus intéressant en revanche, c'est l'opportunité d'avoir totalement la main sur les fonctionnalités embarquées et donc ...

LE MODULE DU MOMENT : BOUTON CAPACITIF (/HACKABLE/HK-033/LE-MODULE-DU-MOMENT-BOUTON-CAPACITIF)

Hackable n° 33 (/Hackable/HK-033) | avril 2020 | [Bodor Denis \(/auteur/Bodor-Denis\)](/auteur/Bodor-Denis)

Électronique (/search/node?Domaines%5B0%5D=72452)

Le bouton-poussoir est la forme la plus basique d'interface humaine qu'un montage puisse avoir. Ceci suppose cependant de prendre en compte bon nombre de caractéristiques mécaniques et électroniques, comme par exemple le fait de gérer les rebonds, de manière matérielle ou logicielle. Pouvoir disposer d'une telle interface sous une forme plus simple et plus « moderne » est l'objet ...

ÉDITO (/HACKABLE/HK-033/EDITO)

Hackable n° 33 (/Hackable/HK-033) | avril 2020 | [Bodor Denis \(/auteur/Bodor-Denis\)](/auteur/Bodor-Denis)

Comment ne pas en parler ? À l'heure où je couche ces lignes au clavier, les mots « épidémies », « contagion », « coronavirus », « stade 3 » et bien d'autres sont sur toutes les lèvres, sur tous les smartphones et sur tous les écrans. C'est un véritable climat de psychose qui s'installe sinon, pour certains, littéralement d'hystérie. Dans ce monde hyper-connecté où l'information se propage ...

PRISE EN MAIN DE L'ESP32-CAM (/HACKABLE/HK-033/PRISE-EN-MAIN-DE-L-ESP32-CAM)

Hackable n° 33 (/Hackable/HK-033) | avril 2020 | [Bodor Denis \(/auteur/Bodor-Denis\)](/auteur/Bodor-Denis)

Électronique (/search/node?Domaines%5B0%5D=72452)

Domotique (/search/node?Domaines%5B0%5D=72455)

En septembre dernier, Espressif annonçait l'arrivée prochaine d'un nouveau microcontrôleur ESP32-S2, venant compléter la famille des biens connus ESP8266 et ESP32. C'est en cherchant à me renseigner sur cette nouveauté et sur la disponibilité des premiers devkits que je suis tombé, par hasard, sur un module à base d'ESP32, à très bas coût, mais proposant une caméra intégrée avec ...

ÉDITO (/HACKABLE/HK-032/EDITO)

Hackable n° 32 (/Hackable/HK-032) | janvier 2020 | [Bodor Denis](#) (/auteur/Bodor-Denis)

De Noël à mars... Avez-vous remarqué qu'il y a un climat de crise énergétique ambiant en ce moment ? Tout devient économe en énergie, les ampoules à filament ont laissé place aux leds dans les rayons des magasins et petit à petit dans les rues, de manière générale les choses énergivores sont devenues le mal incarné car, comme dirait la pub, « ce n'est pas Versailles ici ». ...

MODULE INTERFACE I2C POUR ÉCRAN LCD (/HACKABLE/HK-032/MODULE-INTERFACE-I2C-POUR-ECRAN-LCD)

Hackable n° 32 (/Hackable/HK-032) | janvier 2020 | [Bodor Denis](#) (/auteur/Bodor-Denis)

Électronique (/search/node?Domaines%5B0%5D=72452)

Les afficheurs LCD alphanumériques disposant d'une interface compatible HD44780 (composant Hitachi à l'origine) se pilotent tous de la même façon et peuvent avoir différentes caractéristiques et tailles : une ligne de 8 caractères, quatre lignes de 20 caractères, deux lignes de 16 caractères, etc., tantôt avec rétroéclairage, tantôt sans. ...

1 (?page=0%2C0) 2 (?page=0%2C1) 3 (?page=0%2C2) 4 (?page=0%2C3) 5 (?page=0%2C4)
6 (?page=0%2C5) 7 (?page=0%2C6) 8 (?page=0%2C7) 9 (?page=0%2C8)

Suivant > (?page=0%2C1)

[GNU/LINUX MAGAZINE \(/GNU-LINUX-MAGAZINE\)](#)

[LINUX PRATIQUE \(/LINUX-PRATIQUE\)](#)

[MISC \(/MISC\)](#)

[HACKABLE \(/HACKABLE\)](#)

[A PROPOS \(/A-PROPOS\)](#)

[ABONNEZ-VOUS \(/ABONNEZ-VOUS\)](#)

[INFOS LÉGALES \(/MENTIONS-LEGALES\)](#)

[CONTACTEZ-NOUS \(/CONTACTEZ-NOUS\)](#)



