



Bienvenue sur la plateforme de documentation numérique de Hackable !

(/)

< Article précédent (/Hackable/HK-015/Creez-une-lanterne-qui-reagit-au-toucher)

Article suivant > (/Hackable/HK-015/Bluetooth-pilotez-votre-Arduino-avec-votre-smartphone)

CONNECTEZ VOTRE ARDUINO EN BLUETOOTH : CONFIGURATION DU MODULE

Hackable n° 15 (/Hackable/HK-015) | novembre 2016 | Bodor Denis (/auteur/Bodor-Denis)

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

Mobilité (/search/node?Domaines%5B0%5D=72462)

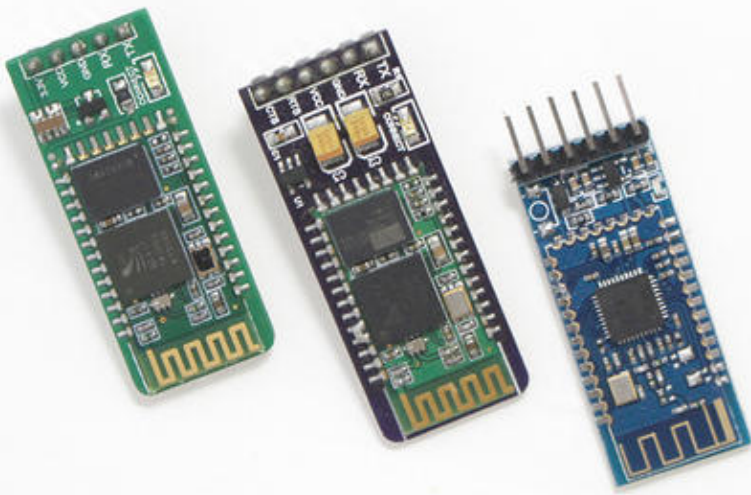
Radio et wireless (/search/node?Domaines%5B0%5D=72465)

Le Bluetooth est partout, du PC portable au smartphone en passant par les montres connectées et même les vieux téléphones mobiles. Mais c'est également une technologie relativement simple à mettre en œuvre dans un projet Arduino et une excellente solution pour faire communiquer différents modules et cartes. Voyons ensemble comment ajouter rapidement une telle connexion Bluetooth pour obtenir une communication série sans fil...

Lorsqu'il s'agit de créer une communication sans fil entre un montage quelconque et un « système plus intelligent » (PC, Mac, smartphone, Raspberry Pi, etc.), les choix possibles sont légion : Wifi, modules radio (APC220, nRF24, etc.), ZigBee... Chaque technologie possède ses avantages et ses inconvénients en termes de facilité de mise en œuvre, de consommation d'énergie, de portée, de vitesse ou encore de coût. Sans oublier un autre aspect en ce qui nous concerne : l'omniprésence de la technologie.

Là, il ne reste finalement plus que deux options : le Wifi et le Bluetooth. La première est séduisante puisque qui dit « Wifi » qui forcément « réseau » et donc, d'une manière ou d'une autre, une connexion au reste du monde, que ce soit le vôtre (LAN) ou le grand monde d'Internet. En revanche, ajouter du Wifi à un projet Arduino, ou même du réseau filaire Ethernet, implique le fait de jongler avec des protocoles, des bibliothèques et des concepts parfois délicats à appréhender. Même en remplaçant la carte Arduino par un ESP8266, dont nous avons parlé dans le numéro 7 et qui se programme directement via l'environnement Arduino, il faut tout de même assimiler quelques concepts (HTTP, MQTT, IP, DNS, SSDP, etc.) avant d'arriver à obtenir un résultat, une communication sous forme de question/réponse.

Dans le cas du Bluetooth en revanche, les choses sont bien plus simples à notre échelle. Certes, il s'agit d'une pile de protocoles comprenant bien des subtilités, mais tout ceci peut être facilement masqué pour arriver à l'essentiel : communiquer avec votre montage exactement comme vous le feriez avec le moniteur série de l'IDE Arduino. En effet, avec les modules très courants dont nous allons parler, il n'y a strictement rien à ajouter côté croquis pour bénéficier du Bluetooth et ajouter cette connectivité dans vos projets.



(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/BT_collec1.jpg)

Voici différents modules Bluetooth tels qu'on peut en trouver sur de nombreux sites de vente. À gauche, un module HC-05 sur un support intégrant un régulateur de tension, au centre un modèle similaire, mais plus ancien comprenant des broches RTS/CTS pour le contrôle de flux et à droite un module Bluetooth Smart HM-10 de piètre qualité.



Note

Bluetooth, LE, BLE, Classic, 4.0, 2.1, EDR...

Afin de vous éviter au maximum une crise de nerfs, il est une chose qu'il est capital de comprendre : le Bluetooth est un standard et il y a des versions à ce standard.

Mais plus important encore « Bluetooth » ne veut pas dire « Bluetooth LE » (alias BLE, « Bluetooth Smart » ou encore « Bluetooth Low Energy ») !

La version 4.0 du standard, rendue publique en 2010, inclut les protocoles *Classic Bluetooth*, *Bluetooth Low Energy* (réduit en « BLE » ou « Bluetooth LE », et commercialement désigné par « Bluetooth Smart ») et *Bluetooth High Speed*. Ceci signifie que le standard 4.0 (et ultérieur) détaille le fonctionnement de ces protocoles, **mais ne les impose pas**. Autrement dit, ce n'est pas parce qu'un périphérique ou un smartphone est Bluetooth 4.0 qu'il est forcément compatible *Bluetooth Low Energy* ou *Classic Bluetooth*. Ce qui est juste, en revanche, c'est de dire qu'un périphérique *Bluetooth Low Energy* est **forcément** compatible Bluetooth 4.0 ou supérieur (puisque c'est à partir de cette version du standard que le protocole a vu le jour).

Ainsi, si votre smartphone est capable de « voir » à la fois les périphériques Bluetooth un peu anciens et le dernier cri en Bluetooth LE, c'est parce qu'il supporte à la fois *Classic Bluetooth* et *Bluetooth Low Energy*. En revanche, un appareil ne supportant que le Bluetooth 3.0 ou le Bluetooth 2.1+EDR sera totalement incapable de voir un périphérique *Bluetooth Low Energy*. Il en va de même pour un appareil effectivement Bluetooth 4.0 ou supérieur, mais non compatible *Bluetooth Low Energy*.

En résumé, nous avons d'une part le *Classic Bluetooth* et toutes les versions du standard d'avant 4.0, et d'autre part le *Bluetooth Low Energy*. Ces deux mondes sont incompatibles au niveau le plus bas et ne peuvent communiquer entre eux.

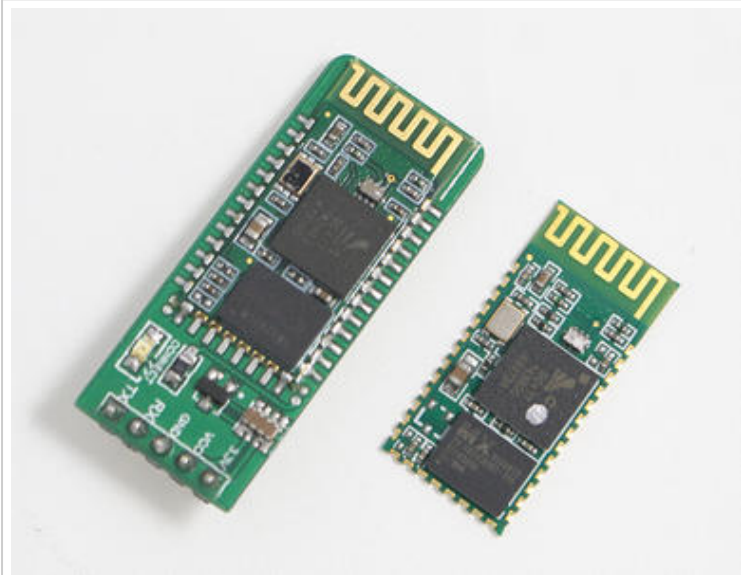
Dans cet article, nous parlons de Bluetooth et donc, pour les périphériques Bluetooth 4.0 ou supérieurs, de *Classic Bluetooth*. Non de *Bluetooth Low Energy*, de BLE, ou de Bluetooth Smart.

1. LES MODULES BLUETOOTH

Une simple recherche de « Bluetooth module » ou « Bluetooth Arduino » sur un site comme eBay, Banggood, Tindie, DealExtreme (DX) ou Alibaba, et vous serez submergé d'offres en tous genres vous proposant des modules à quelques 3€ ou 4€ vous promettant monts et merveilles pour ajouter du Bluetooth à vos projets.

Ceux-ci se présentent généralement sous deux formes : un circuit minuscule possédant de nombreux connecteurs sur trois côtés et une version plus utilisable intégrant ce même circuit, mais soudé sur un support proposant un connecteur au pas de 2,54mm avec entre 4 et 6 broches.

Il va de soi qu'il vaut mieux éviter la première déclinaison, car très délicate à mettre en œuvre et pas vraiment moins chère. En plus de servir de support de connexion, le circuit imprimé de la version avec connecteur intègre généralement un régulateur de tension ainsi qu'une led accompagnée d'une résistance. Au cœur du module, souvent désigné sous le nom de HC-05, se trouve une puce Bluecore 4 de *Cambridge Silicon Radio* (CSR), alimentée en 1,8V. Le circuit intègre, en plus de ce composant une mémoire flash ainsi qu'un autre régulateur de tension. De ce fait, le module de base, sur son support peut être alimenté directement en 5 volts par une carte Arduino, par exemple.



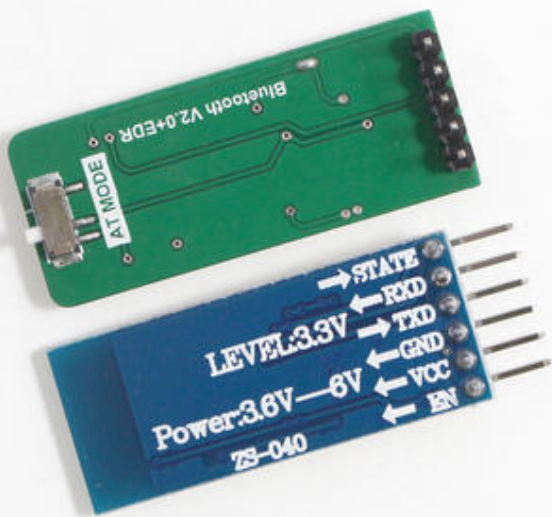
(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/BT_mini_support1.jpg)

Un module Bluetooth au sens strict du terme est le circuit présenté à droite. Il intègre le contrôleur CSR (carré) et de la mémoire flash (rectangle). Afin de rendre son utilisation plus facile, celui-ci est généralement vendu monté/soudé sur un support avec une connectique plus pratique, un régulateur de tension et une led (à gauche).

Attention cependant, en fonction du modèle que vous allez acquérir, il pourra être nécessaire d'utiliser des résistances lors de la connexion à la carte Arduino. En effet, même si le module peut être alimenté en 5V, la communication elle-même se fait avec des tensions 0/3,3V. Parlant de communication, ces modules utilisent tous une communication série. Nous avons donc toujours au minimum 4 connexions :

- Vcc : la tension d'alimentation,
- GND : la masse,
- RX : la broche de réception à connecter au TX d'une carte Arduino,
- TX : la broche d'émission à connecter au RX.

À ce stade, une certaine confusion peut régner concernant votre achat. Soit les résistances permettant l'adaptation de tension sont directement présentes sur le support, soit il faudra les prévoir séparément. Certains modèles de modules comprennent également un circuit intégré chargé de la conversion de tension, mais ils sont relativement rares (et plus chers).



(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/BT_verso1.jpg)

Les modules Bluetooth HC-05 et HC-06 se configurent via des commandes AT, après les avoir basculés dans un mode spécial. Certains supports de module permettent cette opération via une broche, un cavalier, un bouton ou, comme ici, avec un micro-interrupteur (en vert). D'autres supports de modules n'offrent aucune possibilité de configuration (celui en bleu). Il faut alors soit bidouiller, soit simplement utiliser la configuration par défaut.

Ce n'est pas tout, comme nous allons le voir un peu plus loin, ces modules disposent de deux modes de fonctionnement. Nous avons d'une part le mode de configuration (ou mode commande) et de l'autre celui de fonctionnement normal. Le premier mode permet de définir les paramètres de communication du module (vitesse, code, nom, etc.) alors que le second se contente de servir de relais entre la communication série et le Bluetooth : tous les messages qui sont reçus en Bluetooth sont émis sur la broche TX et tout ce qui arrive de la broche RX est envoyé en Bluetooth.

Pour passer d'un mode à l'autre et permettre la configuration, l'une des broches du module (le tout petit) doit être reliée à la tension d'alimentation (3,3V) lors de sa mise en marche. En fonction du module choisi, ceci peut être prévu sur le support sous la forme d'un cavalier, un bouton poussoir, un interrupteur ou... rien du tout. Cela se complique encore davantage lorsqu'il s'agit de savoir précisément quelle broche doit être utilisée (souvent désignée par « KEY »).

Les modules généralement désignés sous le nom HC-05 utilisent la broche 34 (PIO11) pour la sélection du mode alors que les modules désignés par HC-06 utilisent la broche 26 (PIO3). Physiquement, les deux modules sont strictement identiques au composant près. Ce qui change en revanche c'est le code enregistré dans leur mémoire flash (le firmware) : les HC-05 sont capables de prendre le rôle de maître ou d'esclave, alors que les HC-06 ne peuvent être qu'esclave ou en d'autres termes, ne peuvent qu'accepter des connexions et non les initier. Ceci n'est cependant pas très important, car dans la plupart des cas, un projet Arduino sera un périphérique Bluetooth attendant des connexions et ne cherchera de toute façon pas à se connecter de lui-même.

Le module idéal sera donc un HC-05 sur un support adapté et disposant d'un interrupteur pouvant permettre la sélection de mode. Il est important de le préciser, car certains modules sont incomplets ou incompatibles :

- un HC-05 sur un support pour HC-06 : le bouton, s'il est présent (généralement pas) est connecté à la mauvaise broche (PIO3 au lieu de PIO11) et n'aura aucun effet. Généralement, les supports pour HC-06 sont bleus avec une sérigraphie au verso indiquant l'usage des broches ;



- un HC-06 sur un support pour HC-05 : même problème, même conséquence ;

- un HC-05 ou HC-06 sur un support ne proposant pas de sélecteur de mode : là vous avez deux solutions, soit l'utiliser avec les paramètres par défaut (vitesse, nom, code), soit bricoler pour pouvoir activer le mode de configuration en reliant la bonne broche à la tension d'alimentation, du module (3,3v) non du support (5V). Ceci peut se faire en soudant délicatement un câble sur la broche et en la reliant au moment opportun à la bonne tension.

Il peut être rapidement difficile de s'y retrouver dans les objets et être plus ou moins sûr de ce qu'on va recevoir. Voici quelques conseils pour vous éviter une déception :

- Évitez comme la peste toutes les annonces ou descriptions qui parlent de RS232. RS232 est une norme pour les connexions séries qui utilisent +/-12v et étaient présentes sur les PC il y a une quinzaine d'années. Le vendeur qui parle de RS232 pour ce type de module ne sait clairement pas de quoi il parle.

- Évitez les modules HC-05 ou HC-06 ne proposant que 4 broches et ne disposant pas de bouton poussoir ou d'interrupteur. La seule solution pour configurer ces modules sera de souder un câble à la minuscule broche permettant l'activation du bon mode.

- À défaut d'interrupteur de sélection de mode, orientez vos préférences vers les modules proposant une broche « KEY ». Il y a de fortes chances que cette broche soit effectivement reliée à celle permettant le changement de mode.

- Un module désigné par HC-05, mais dont la photo présente un support bleu où tous les composants ne sont pas présents a toutes les chances d'être effectivement un HC-05, mais sur un support prévu pour HC-06. Même en soudant un bouton sur le support, vous ne connecterez très probablement pas la bonne broche à la tension d'alimentation.

- Évitez les annonces qui proclament vendre des modules « HC-05/HC-06 ». Cela montre que le vendeur ne fait que copier/coller des termes qu'il ne comprend pas et vous recevrez quelque chose que vous n'attendez certainement pas. Un module est soit un HC-05, soit un HC-06, mais pas les deux.

- Évitez les modules HM-10 ou HM-11, il s'agit de modules Bluetooth LE et vous ne pourrez probablement pas les utiliser avec tous vos équipements Bluetooth. Notez qu'ils ne sont pas foncièrement mauvais, mais il s'agit d'une technologie totalement différente de celle dont nous allons parler.

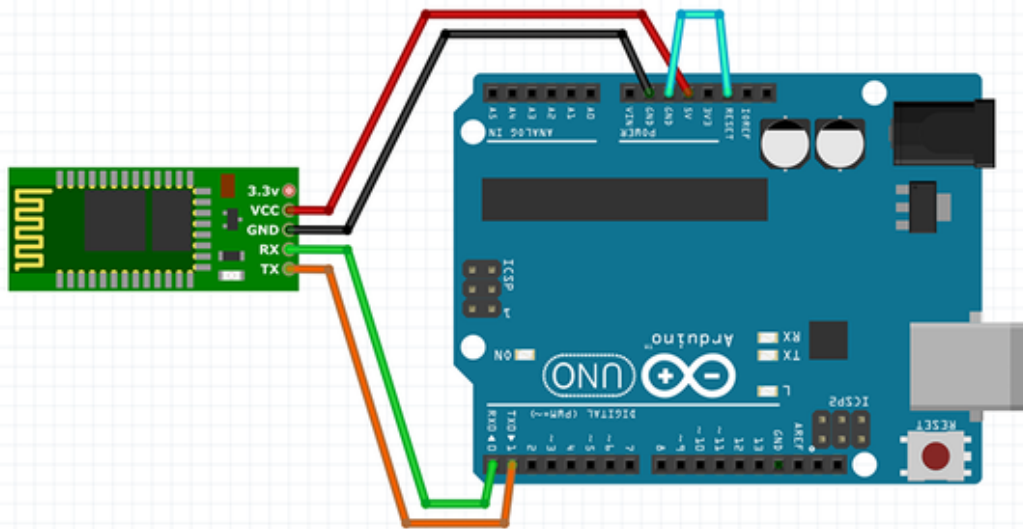
- Si le vendeur précise « Raspberry Pi » dans l'intitulé de son annonce, passez votre chemin, c'est un attrape-nigaud. Ce type de module peut effectivement se connecter à une Pi mais, passez-moi l'expression, c'est totalement stupide. Il existe des *dongles* USB Bluetooth 2.0+EDR pour le tiers du prix d'un module (1€), bien plus faciles à utiliser, directement compatibles avec Linux et ne nécessitant pas de se passer de la console série de la Pi. Le seul intérêt d'une connexion d'un tel module à une Pi est de « déporter » la console série en Bluetooth, mais c'est là quelque chose de totalement annexe.

- Si le vendeur présente dans son annonce un schéma du support, c'est très bon signe. Il aura au moins fait l'effort de fournir un maximum d'informations et il est très probable que le matériel corresponde effectivement au schéma.

- Si vous êtes agile de vos petits doigts, il peut être plus sûr d'acheter séparément le module lui-même et le support. Ceci ne sera pas forcément plus économique et vous prendra un certain temps pour l'assemblage, mais vous saurez avec davantage de précision de quoi il en retourne.

- Préférez, bien entendu, les vendeurs avec une bonne réputation, de nombreuses et bonnes évaluations et des bons commentaires des acheteurs. Préférez également les produits affichant une quantité déjà vendue importante. Généralement, lorsqu'un vendeur se rend compte du mécontentement des acheteurs, il supprime l'annonce et en crée une nouvelle pour le même produit. Les annonces avec beaucoup de ventes à leur actif sont donc généralement un bon signe.



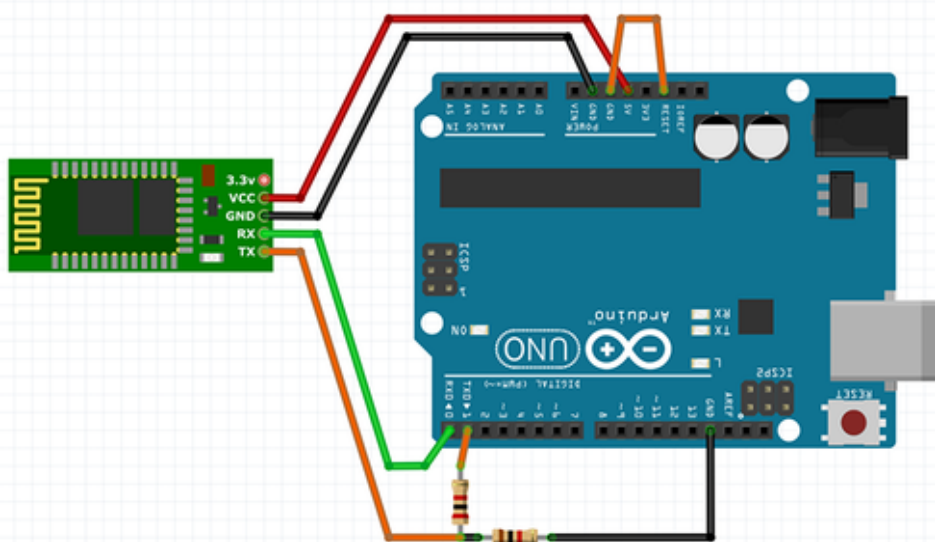


(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/ardu_bt_serial1.png)

Connexion d'un module type HC-05 à une carte Arduino utilisée comme simple convertisseur USB/série. Ici, le module est censé intégrer une conversion de tension et peut donc être branché directement aux ports de la carte Arduino.

À ce stade, vous l'avez très certainement compris, quoi que vous fassiez, il n'est pas certain que vous obteniez le matériel correspondant à vos attentes. En ce qui me concerne, je dispose de ce type de modules depuis fort longtemps et en avait fait l'acquisition auprès d'un vendeur appelé Wide.hk. Le firmware qu'ils contiennent date de 2009 et correspond à un HC-05. Ils présentent cependant l'avantage de posséder un micro-interrupteur permettant la sélection du mode, chose relativement rare dans les produits actuellement vendus.

Le dernier conseil que je vous donnerai concernant l'achat consiste à ne pas acheter en trop grande quantité auprès d'un unique vendeur (à moins d'être à 100% sûr). Comme généralement le port est offert, il est préférable d'acheter un module chez trois vendeurs plutôt que trois modules chez un vendeur. C'est une question de probabilités, tant au niveau du délai de livraison que de la qualité du produit réceptionné. Rien ne vous empêchera, ensuite, une fois un module testé, d'en commander d'autres exemplaires via la même annonce.



(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/ardu_bt_serial_DIV1.png)

Si votre module Bluetooth ne dispose pas des composants nécessaires pour l'adaptation des tensions, mais uniquement un régulateur 3,3V, il n'est pas recommandé de brancher directement la broche TX de l'Arduino sur le module,

car les 5V risquent d'endommager le périphérique. On utilise alors deux résistances, 1K ohms et 2 Kohms, agencées en diviseur de tension pour procéder à la conversion.

2. CONFIGURATION DU MODULE : LE MODE COMMANDE

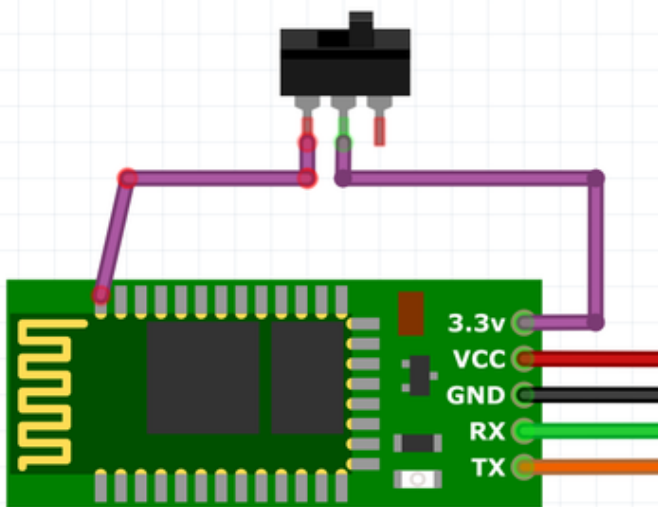
Le module Bluetooth arrive généralement avec une configuration par défaut qu'il est préférable de changer : la vitesse de communication, le nom sous lequel le module apparaît en Bluetooth et le code PIN permettant l'appairage (ou le jumelage).

Pour procéder à cette configuration, il faut passer le module en mode commande et utiliser des instructions spécifiques permettant de changer ses paramètres. La communication se fait directement avec le module via la liaison série. Lorsque le module est démarré de façon classique, les données qui lui sont envoyées sont transmises via Bluetooth, mais lorsqu'il se trouve en mode commande, ces données sont des instructions qu'il accepte et traite.

Il existe plusieurs façons d'établir cette communication avec le module. Celle que je trouve la plus aisée et qui ne nécessite aucun élément/équipement supplémentaire, consiste à tout simplement utiliser la carte Arduino comme convertisseur USB/série. En effet, le microcontrôleur d'une carte comme l'Arduino UNO communique avec le PC/Mac et le moniteur série via une communication série au travers d'un convertisseur USB/série intégré (FTDI ou un ATmega16U2). Mais cette ligne de communication est également connectée aux broches TX et RX de la carte elle-même. De ce fait, en désactivant le microcontrôleur en maintenant la broche reset à la masse, on peut établir une connexion directe entre le convertisseur USB/série et un module connecté aux broches RX et TX.

Un simple câble connectant GND et RESET sur la carte la transforme donc en simple convertisseur USB/série, mais maintenir le bouton reset enfoncé, avec une pince à linge par exemple, fonctionnera tout aussi bien. Il ne vous restera plus qu'à connecter le module Bluetooth sur les broches 5V, GND, RX et TX puis utiliser le moniteur série de l'environnement Arduino pour parler au module, passé pour l'occasion en mode commande.

Notez qu'en fonction du module, il peut être risqué de connecter sa broche de réception de données (notée RX en principe) à la carte Arduino. En effet, si le module ne convertit pas les tensions sur les broches RX/TX, la carte va utiliser 5V alors que le module attend 3,3V. Personnellement, je n'ai jamais eu de problème de ce côté, mais en principe, ce n'est pas correct (pour tout dire, je viens tout juste de me rendre compte que j'ai toujours utilisé ces modules, depuis des années, en 0/5V alors qu'ils ne sont censés supporter que 0/3,3V).



(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/hack_HC051.png)

Si le circuit support du module Bluetooth HC-05 ne permet pas, par une broche, un interrupteur ou un bouton, de passer en mode configuration/commande, un

petit bidouillage fera l'affaire. Il suffit de relier la broche 34 du module (PIO11) à la tension d'alimentation 3,3V lors de sa mise en route.

Pour faire les choses proprement, la solution la plus rapide consiste à créer un diviseur de tension à l'aide de résistances :

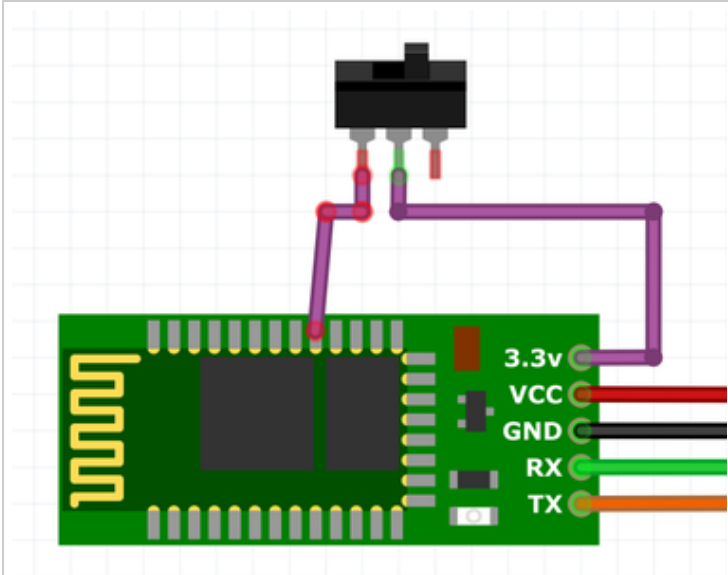
- connectez la sortie de la carte Arduino à une résistance de 1 Kohm (R1) ;
- branchez l'autre borne de la résistance à la broche RX du module ;
- et enfin, connectez cette même borne à la masse via une résistance de 2 Kohms (R2), ou deux fois 1 Kohm en série.

Vous obtenez un diviseur de tension où la différence de potentiel entre la broche du module et la masse sera égale à $U \cdot R2 / (R1 + R2)$, soit $5 \cdot 2000 / (1000 + 2000) = 3,33V$, dès lors que la sortie de l'Arduino sera à 5V. Dans le sens inverse, vous n'avez rien à faire puisque le module utilise 3,3V comme tension à l'état haut, ce qui est parfaitement dans les tolérances de l'Arduino qui y verra donc le même état.

Une fois le module alimenté et démarré en mode commande, ouvrez le moniteur série, réglez la vitesse sur « 38400 bauds » et les fins de lignes en « Les deux, NL et CR ». À ce stade, vous devez être en mesure d'envoyer la première commande, **AT** et la réponse retournée par le module sera **OK**. Si tel n'est pas le cas, vérifiez différents points :

- Le module est-il bien connecté ? La sérigraphie sur le module et les mentions « RX » et « TX » sont parfois inversées. Elles peuvent signifier l'utilité de la broche elle-même au regard du module ou de la connexion. Dans mon cas par exemple avec mes HC-05, « TX » fait référence à la broche permettant d'envoyer des données en Bluetooth (c'est en principe le RX du module lui-même) et est donc connectée au « TX » de la carte Arduino. Ce n'est pas nécessairement votre cas.
- Le module est-il bien en mode commande ? Avec un module HS-05, la led présente sur le support du module clignote de façon continue en mode commande : 1 seconde allumée, 1 seconde éteinte. Lorsque le module est démarré en mode standard, la led clignote rapidement, et si le module est en mode standard et qu'un périphérique Bluetooth y est connecté, la led pulse brièvement une fois toutes les secondes.
- La bonne vitesse est-elle sélectionnée ? La vitesse de communication en mode commande dépend des choix du fabricant et est indépendante de la vitesse en mode standard que vous pouvez configurer. Tous les modules HC-05 que j'ai eus entre les mains utilisaient 38400 bps, mais c'est peut-être différent avec le modèle en votre possession (en fonction de la version du firmware en flash). Essayez de trouver une documentation spécifique à votre module ou testez toutes les vitesses possibles et probables : 9600, 19200, 38400, 57600, 115200, 230400 ou 460800.
- La carte Arduino est-elle effectivement en plein reset grâce au câble ou au bouton poussoir maintenu enfoncé ?

Si vous obtenez la réponse **OK** de la part du module, vous êtes prêt pour la configuration. Les commandes à utiliser débutent toutes par **AT** à l'instar de celles utilisées il y a bien longtemps avec les modems analogiques (commandes Hayes) ou encore les modules ESP8266 utilisés avec leur firmware d'origine. Ces commandes permettent de définir des paramètres gérant son comportement. Nous n'allons ici couvrir qu'un seul cas d'utilisation où le module a un rôle d'esclave en Bluetooth : il est en attente de connexion, mais n'initie pas de communications lui-même.



(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/hack_HC061.png)

Le passage en mode commande sur un module HC-06 fonctionne de la même manière qu'avec un HC-05, seule la broche utilisée est différente. Ici, c'est la broche 26 (PIO3) qu'il faudra mettre à 3,3V pour démarrer le module dans le bon mode.

Les commandes se divisent en deux catégories : celles ne réglant pas un paramètre, mais ordonnant une action, et celles permettant de consulter et de définir des valeurs de configuration. Un exemple de commande déclenchant une action est :

```
AT+VERSION
+VERSION:1.0-20090818
OK
```

La commande envoyée est **AT+VERSION** et la réponse débutant par un + répète l'ordre, suivi de la réponse effective : **1.0-20090818**. La fin d'une réponse est toujours marquée par **OK**. Si un problème survient pour une commande, le module répondra **ERROR: (0)** en lieu et place de **OK** (le 0 correspond à un code d'erreur, mais celui-ci ne semble documenté nulle part).

Une autre commande de ce type permet d'obtenir l'adresse Bluetooth matérielle (MAC) du module :

```
AT+ADDR
+ADDR:19:5d:ee:a4:24
OK
```

Cette adresse ne peut être changée et est unique à chaque périphérique Bluetooth alors que son nom, par exemple, peut être modifié. L'adresse est ici présentée de manière un peu « brouillonne » et se transcrit en **19:5d:ee:a4:24** ou en **19:5D:EE:A4:24**, c'est ainsi qu'elle apparaîtra pour un appareil procédant à une recherche de périphériques Bluetooth.

Les commandes permettant le réglage se terminent soit par un ? pour une consultation de la valeur en cours, soit par = suivi d'une nouvelle valeur pour un réglage. Commençons par consulter le nom actuellement configuré :

```
AT+NAME?
+NAME:WIDE_HK
OK
```

Pour définir un nouveau nom puis vérifier l'enregistrement :

```
AT+NAME=Hackable15
```



```
OK
AT+NAME?
+NAME:Hackable15
OK
```

Nous pouvons également faire de même avec le code PIN permettant l'appairage :

```
AT+PSWD?
+PSWD:1234
OK
AT+PSWD=424242
OK
AT+PSWD?
+PSWD:424242
OK
```

La commande **AT+ROLE?** permet de connaître le rôle que doit prendre le module dans une relation Bluetooth. pour esclave et 1 pour maître. Avec un module HC-06 ne permettant que le rôle d'esclave, cette commande n'existe pas.

Et enfin, nous avons la configuration de la vitesse de communication en mode standard :



```
AT+UART?
+UART:9600,0,0
OK
```

Cette commande permet de configurer non seulement la vitesse entre la carte Arduino et le module, mais également le format de données. Nous avons là trois paramètres séparés par des virgules : vitesse, bit de stop, parité. Un croquis de carte Arduino utilise généralement quelque chose comme `Serial.begin(9600)` en ne précisant que la vitesse, mais le format de données peut également être spécifié. Par défaut, il sera de 8 bits de données, pas de parité et un bit de stop (alias 8N1). Dans le cas du module nous n'avons pas le choix du nombre de bits de données, mais pouvons régler le reste. Ainsi pour 19200, pas de parité et un bit de stop nous ferons :

```
AT+UART=19200,1,0
OK
AT+UART?
+UART:19200,1,0
OK
```

Une dernière chose devra sans doute être configurée, en particulier si vous rencontrez des problèmes de connexion avec certains appareils. Chaque périphérique Bluetooth annonce un certain type lorsqu'il est scanné, c'est ce qui permet aux smartphones, par exemple, d'afficher une petite icône lors d'une recherche Bluetooth (casque, PC, etc.). Ce type est défini par une classe de périphérique et de service ou *Class of Device/Service* (CoD) en anglais. Il s'agit d'une valeur qu'il est possible de configurer par une commande **AT** et qui est souvent à `0x000000` par défaut. Certains appareils n'aiment pas que la classe soit à zéro et refusent de lister le périphérique.

Pour définir cette valeur, il vous suffit d'aller sur un site comme <http://www.ampedrftech.com/cod.htm> (<http://www.ampedrftech.com/cod.htm>), et de choisir la classe la plus proche de ce que vous comptez faire avec le module. Cochez les cases de votre choix sur la page, affichez la valeur, puis passez-la avec la commande

AT+CLASS= :

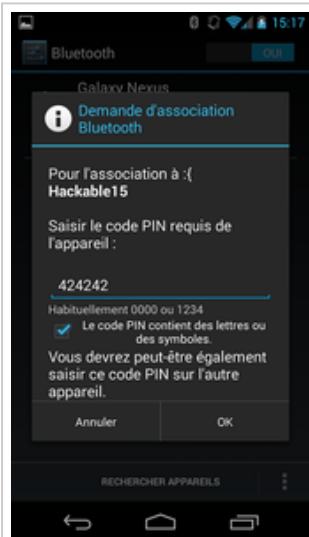
```
AT+CLASS?
+CLASS:0
OK
AT+CLASS=0C0800
OK
AT+CLASS?
+CLASS:c0800
OK
```

00800 définit un périphérique proposant des services de rendu (*Rendering*) et de capture (*Capturing*) et étant un jouet (*Toy*) de type non défini (*Undefined*). Généralement, cette valeur satisfait la plupart des appareils et smartphones.

Notre module est maintenant configuré :

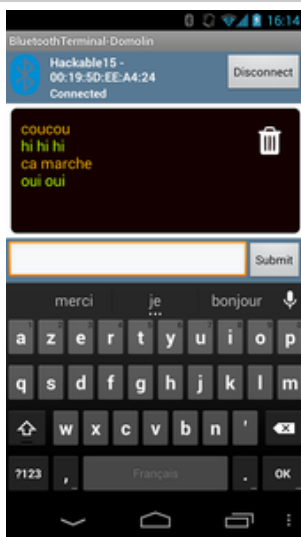
- nom : « Hackable15 »,
- code PIN : « 424242 »,
- vitesse : 19200 8N1.

Il suffit alors de déconnecter le module de son alimentation, repasser en mode standard en le reconnectant sans forcer le mode commande (soit en basculant l'interrupteur s'il est présent, soit en ne connectant plus la broche adéquate à 3,3V). Nous sommes prêts pour le premier essai.



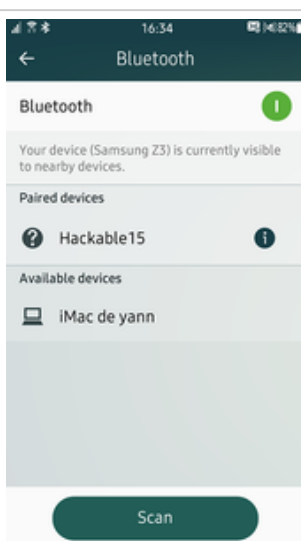
(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/bt_android_paire1.png)

Une fois configuré, le module Bluetooth pourra être appairé (ou jumelé, ou associé) comme n'importe quel périphérique. Il vous suffira de saisir le code PIN défini lors de la configuration et le tour sera joué.



(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/bt_android_term1.png)

Une application
comme Bluetooth
Terminal de Domolin
Bolivia, disponible sur
Google Play, vous
permettra de
rapidement valider le
fonctionnement de la
liaison. Tout ce que
vous taperez côté
Android apparaîtra
côté Arduino dans le
moniteur série et
inversement.



(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/BT_tizen1.png)

Cette communication
Bluetooth marchera
sur la plupart des

périphériques supportant Classic Bluetooth ou Bluetooth inférieur à 4.0, et même sous Tizen (mais il n'y a pas encore d'application pour s'en servir). Seuls les iPod/iPad/iPhone seront incapables de prendre en charge ces modules parce que la firme à la grande pomme a décidé de ne pas supporter le profil Bluetooth associé (SPP). En même temps, ce n'est pas étonnant, ce sont les mêmes qui ont décidé dernièrement que la prise audio jack était obsolète et n'avait plus rien à faire sur un smartphone...



3. UN PREMIER ESSAI AVEC ANDROID ET LE PROBLÈME D'IOS

Pour nous assurer que la configuration est appliquée et que nous pouvons communiquer avec le module en Bluetooth, nous n'allons pas utiliser de croquis. La configuration matérielle actuelle, avec le module directement connecté au moniteur série, sera largement suffisante (n'oubliez pas cependant de changer la vitesse en celle configurée précédemment). Il nous suffit alors de prendre un smartphone ou une tablette Android compatible Classic Bluetooth ou Bluetooth inférieur à 4.0 et de trouver la bonne application.

Pourquoi forcément un périphérique Android ? C'est simple, les iPhone et iPad ne peuvent pas communiquer avec le module. Ceci découle d'un choix d'Apple concernant les profils Bluetooth supportés par leurs périphériques. Nous avons parlé précédemment des normes Bluetooth, mais nous n'avons couvert qu'une partie de ce qui existe : les *Bluetooth Core Specifications*. En plus de cela, nous trouvons la notion de profils déterminant le type de communication supporté par un périphérique Bluetooth. En l'occurrence, un iPhone ou un iPad supporte les profils suivants (dixit la page du support Apple, <https://support.apple.com/fr-fr/HT204387> (<https://support.apple.com/fr-fr/HT204387>)) :

- HFP (*Hands-Free Profile*) : le profil pour les kit mains libres ;
- PBAP (*Phone Book Access Profile*) : accès aux annuaires, carnet d'adresses et contacts ;
- A2DP (*Advanced Audio Distribution Profile*) : diffusion audio ;
- AVRCP (*Audio/Video Remote Control Profile*) : contrôle de diffusion audio/vidéo (télécommande) ;
- PAN (*Personal Area Networking*) : réseau personnel ;
- HID (*Human Interface Device*) : interface homme/machine (clavier, souris, etc.) ;
- MAP (*Message Access Profile*) : messagerie.

Le problème est que nos sympathiques petits modules utilisent le profil pour lequel ils sont faits : *Serial Port Profile* ou SPP, un profil permettant d'émuler une communication série via Bluetooth. Il semblerait que le choix de ne pas supporter SPP soit en rapport avec le contrôle des périphériques autorisés (au sens industriel du terme) et le programme de licences *Made For iPod* (MFi) permettant à Apple de choisir qui a le droit de fabriquer des accessoires pour leurs périphériques. Le fait que les produits Apple ne soient pas très bidouille-compatibles n'est pas nouveau, mais dans ce cas précis, c'est tout simplement éliminatoire. Si vous avez un iPod, un iPhone ou un iPad, vous devrez vous rabattre sur la solution du Bluetooth LE avec des modules comme le HM-10 par exemple, au fonctionnement assez différent de ceux traités ici.



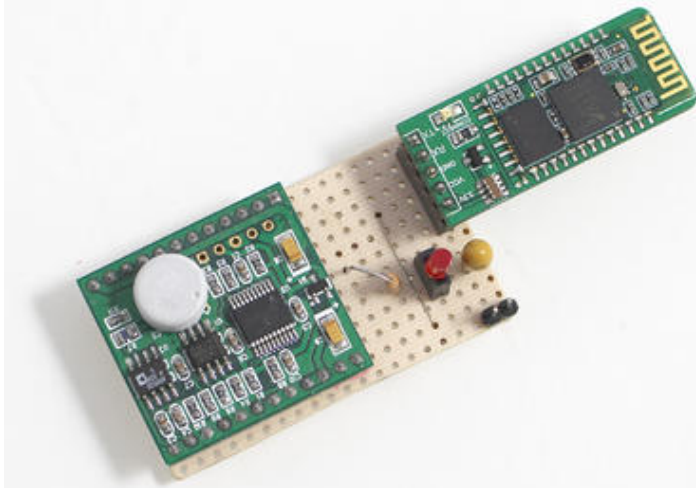
(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/BT_hack_antenne1.jpg)

L'antenne intégrée aux modules sous la forme d'une piste de cuivre est souvent suffisante pour des besoins courants. Pour augmenter la portée cependant, rien n'empêche de bricoler un peu et d'ajouter un connecteur RP-SMA et une « vraie » antenne...

Sur un smartphone Android, dès le module démarré en mode standard, celui-ci devrait être visible dans les préférences Bluetooth lors d'une recherche (scan). Celui-ci devra apparaître soit avec son nom, soit avec son adresse matérielle. Vous pourrez alors le sélectionner pour procéder à l'appairage et entrer le code PIN que vous aurez défini précédemment. Ceci fait et validé, votre smartphone pourra se connecter au module sans autre forme de procès.

Pour procéder au test, vous devrez utiliser une application faisant office de moniteur série exactement comme celui de l'IDE Arduino. Une simple recherche de « Bluetooth terminal » sur Google Play devrait vous afficher une sélection relativement conséquente. Après plusieurs tests d'applications, ma préférence s'est arrêtée sur « Bluetooth Terminal » de Domolin Bolivia. L'application est modeste, mais fait parfaitement son travail. Il vous suffira alors de choisir le périphérique Bluetooth dans la liste, vous connecter et envoyer des messages. Tout ce que vous validerez côté Android devra apparaître dans le moniteur série Arduino et inversement. Ceci validera l'état de la communication et le fonctionnement de l'ensemble.





(https://connect.ed-diamond.com/sites/default/files/articles/hackable/hk-015/82835/BT_no_ardu1.jpg)

Le fait que ces modules Bluetooth soient de simples convertisseurs Bluetooth/série présente un avantage intéressant : il n'est pas toujours utile de faire intervenir une carte Arduino. Ici, le module sur la gauche est un capteur de température et d'humidité relative communiquant via une liaison série. Pour créer un capteur Bluetooth, il suffit de configurer le module Bluetooth avec la bonne vitesse et de relier les deux en ajoutant une alimentation.

CONCLUSION TEMPORAIRE

Notez qu'il est déjà possible de créer un projet sur cette base. Tout ce que vous aurez à faire sera de développer un croquis exactement de la même manière que s'il s'agissait d'interpréter des ordres provenant du moniteur série. Cela ne fait absolument aucune différence qu'il s'agisse d'une liaison filaire (USB) ou Bluetooth. Pour la carte Arduino, c'est exactement la même chose.

Notez toutefois que le fait d'avoir le module Bluetooth connecté au seul port série d'une carte comme l'Arduino UNO peut poser problème lors de l'enregistrement du croquis. Mieux vaut débrancher le module, programmer la carte, puis le rebrancher. Une autre solution sera d'utiliser la bibliothèque *SoftwareSerial* pour connecter le module sur d'autres broches de la carte et réserver le vrai port série pour la programmation et la mise au point du croquis.

Vous conviendrez avec moi que le fait d'envoyer des commandes ainsi à votre croquis peut être très amusant, mais pas vraiment ergonomique, et encore moins impressionnant si vous comptez faire la démonstration de vos projets. Nous allons régler ce petit problème dans l'article qui suit et, par la même occasion, user justement de *SoftwareSerial* pour nos communications...

Étiquettes :

Arduino (/search/node?field_ct_article_tags_target_id_selective%5B0%5D=72477),
Bluetooth (/search/node?field_ct_article_tags_target_id_selective%5B0%5D=72511),
Android (/search/node?field_ct_article_tags_target_id_selective%5B0%5D=72536),
iOS (/search/node?field_ct_article_tags_target_id_selective%5B0%5D=72537)

< Article précédent (/Hackable/HK-015/Creez-une-lanterne-qui-reagit-au-toucher)

Article suivant > (/Hackable/HK-015/Bluetooth-pilotez-votre-Arduino-avec-votre-smartphone)



RECHERCHER

UN ARTICLE HACKABLE

parmi plus de 309 articles !



AU SOMMAIRE DU MÊME NUMÉRO

Contrôlez votre alimentation de laboratoire avec votre Raspberry Pi (/Hackable/HK-015/Controlez-votre-alimentation-de-laboratoire-avec-votre-Raspberry-Pi)

Créez un enregistreur autonome de température, d'humidité et de pression (/Hackable/HK-015/Creez-un-enregistreur-autonome-de-temperature-d-humidite-et-de-pression)

Créez une lanterne qui réagit au toucher (/Hackable/HK-015/Creez-une-lanterne-qui-reagit-au-toucher)

► **Connectez votre Arduino en Bluetooth : configuration du module** (/Hackable/HK-015/Connectez-votre-Arduino-en-Bluetooth-configuration-du-module)

Bluetooth : pilotez votre Arduino avec votre smartphone (/Hackable/HK-015/Bluetooth-pilotez-votre-Arduino-avec-votre-smartphone)

Connectez un module LCD à votre Raspberry Pi pour afficher son adresse réseau (/Hackable/HK-015/Connectez-un-module-LCD-a-votre-Raspberry-Pi-pour-afficher-son-adresse-reseau)

Partagez vos projets et vos créations sur GitHub (/Hackable/HK-015/Partagez-vos-projets-et-vos-creations-sur-GitHub)

Vos projets Arduino et Raspberry Pi sur une batterie externe pour smartphone. Pas si simple... (/Hackable/HK-015/Vos-projets-Arduino-et-Raspberry-Pi-sur-une-batterie-externe-pour-smartphone.-Pas-si-simple...)



SUR LE MÊME SUJET

Exodus Privacy – Vos applications n’auront plus de secrets pour vous (/Linux-Pratique/LP-117/Exodus-Privacy-Vos-applications-n-auront-plus-de-secrets-pour-vous)

Linux Pratique n° 117 (/Linux-Pratique/LP-117) | janvier 2020 | [Hof Aline](#) (/auteur/Hof-Aline)

Mobilité (/search/node?Domaines%5B0%5D=72462)

Sécurité (/search/node?Domaines%5B0%5D=72467)

Si vous êtes soucieux de votre vie privée, Exodus Privacy pourra sans doute vous plaire. Cette application va en effet vous lister les permissions accordées aux applications installées sur votre appareil, mais aussi vous indiquer si celles-ci utilisent des pisteurs qui collecteront des données vous concernant.

...

Reverse-engineering d'une alimentation numérique et contrôle avec bash (/Hackable/HK-032/Reverse-engineering-d-une-alimentation-numerique-et-contrôle-avec-bash)

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

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

Le marché propose de nos jours des alimentations de laboratoire aux caractéristiques très intéressantes : compactes, puissantes, programmables... Conçues et fabriquées en Chine, évidemment, elles sont économiques et très peu concernées par les standards ou les protocoles, contrairement aux équipements professionnels des grandes marques, chers, mais relativement ouverts. Cet article va examiner un modèle ...

Désinstaller toutes vos applications Android indésirables sans « rooter » votre appareil (/Linux-Pratique/LP-117/Desinstaller-toutes-vos-applications-Android-indesirables-sans-rooter-votre-appareil)

Linux Pratique n° 117 (/Linux-Pratique/LP-117) | janvier 2020 | [Delmas Laurent](#) (/auteur/Delmas-Laurent)

Mobilité (/search/node?Domaines%5B0%5D=72462)

Lorsque vous faites l'acquisition d'un appareil, smartphone ou tablette, fonctionnant sous Android, ce dernier est généralement livré avec de nombreuses applications par défaut dont vous n'avez pas nécessairement l'utilité. Bien souvent il vous est impossible de les désinstaller de façon traditionnelle. Afin d'y parvenir, nombre de sites Internet vous préconisent de « rooter » votre appareil. En réalité, il n'en est rien et vous allez découvrir ...



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. ...

Simulez vos circuits électroniques avant de réaliser vos cartes électroniques avec KiCAD (/Hackable/HK-032/Simulez-vos-circuits-electroniques-avant-de-realiser-vos-cartes-electroniques-avec-KiCAD)

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

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

Tout comme la conception et la modélisation de systèmes mécaniques via des logiciels libres tels que FreeCAD, SaloméMeca ou bien des logiciels propriétaires comme Catia/Solidworks, la conception, simulation et réalisation de circuits électroniques n'ont jamais été aussi simples depuis l'arrivée de la version 5 de KiCAD [1, 2, 3]. Bien qu'existant depuis de nombreuses années, la dernière mouture de KiCAD intègre la librairie partagée ...

Des kits de développement FPGA à moins de 30 € (/Hackable/HK-032/Des-kits-de-developpement-FPGA-a-moins-de-30-EU)

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

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

Même s'il est possible de faire 90 % du développement en simulation et sans matériel, quand on «fait du FPGA», on souhaite pouvoir toucher du concret. Il est donc nécessaire d'avoir une carte électronique permettant de faire fonctionner son projet en réel. Les outils de développement sur FPGA ont la réputation d'être chers et réservés aux universités et bureaux d'études. Ce n'est pourtant plus le cas, il existe de plus en plus de kits de ...

[1 \(?page=0\)](#) [2 \(?page=1\)](#) [3 \(?page=2\)](#) [4 \(?page=3\)](#) [5 \(?page=4\)](#) [6 \(?page=5\)](#) [7 \(?page=6\)](#)
[8 \(?page=7\)](#) [9 \(?page=8\)](#) [» \(?page=1\)](#)

PAR LE MÊME AUTEUR

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.

VITE FAIT : CRÉER UN THERMOSTAT D'AMBIANCE PROGRAMMABLE (/HACKABLE/HK-032/VITE-FAIT-CREER-UN-THERMOSTAT-D-AMBIANCE-PROGRAMMABLE)

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

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

Dans ma nouvelle maison, j'ai découvert les joies du chauffage au fioul et les limitations d'un système de régulation de la température intérieure le plus simpliste qui soit. La simplicité a ses avantages, et le fioul aussi, mais lorsqu'on regarde sa facture, on se rend rapidement compte que cette simplicité a un coût, qui peut être important. Pour régler le problème, j'ai décidé de faire rapidement évoluer mon installation, avec l'aide d'une ...

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 ».

ARDUINO MKR VIDOR 4000 : UN ARDUINO PAS COMME LES AUTRES (/HACKABLE/HK-031/ARDUINO-MKR-VIDOR-4000-UN-ARDUINO-PAS-COMME-LES-AUTRES)

Hackable n° 31 (/Hackable/HK-031) | octobre 2019 | [Bodor Denis](#) (/auteur/Bodor-Denis)

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

Les microcontrôleurs existaient bien avant l'arrivée des cartes Arduino. La révolution provoquée par l'initiative italienne n'a pas été technologique, mais pédagogique. C'est le fait de totalement démocratiser et de rendre accessible le développement sur microcontrôleurs qui a mis le mot « Arduino » sur toutes les lèvres. Aujourd'hui, Arduino récidive avec un autre domaine de l'électronique numérique et s'attaque à une ...

ÉDITO (/HACKABLE/HK-031/EDITO)

Hackable n° 31 (/Hackable/HK-031) | octobre 2019 | [Bodor Denis](#) (/auteur/Bodor-Denis)

Dans la vie, il y a ceux qui savent être « open » et ceux qui ne savent pas. ...

CONVERTISSEUR HDMI VERS VGA (/HACKABLE/HK-031/CONVERTISSEUR-HDMI-VERS-VGA)

Hackable n° 31 (/Hackable/HK-031) | octobre 2019 | [Bodor Denis](#) (/auteur/Bodor-Denis)

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

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

L'intérêt premier d'un convertisseur HDMI vers VGA est de permettre l'utilisation de matériels plus anciens avec des cartes et équipements modernes. Le cas typique ici est, bien entendu, de recycler un vieil écran VGA comme moniteur pour une carte Raspberry Pi. Ceci ne sera pas nécessairement très intéressant pour une utilisation « desktop », mais sera parfaitement viable pour le jeu, l'émulation, la création d'un media center ou ...



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

