

Ebook Arduino !



Vous avez une question ? Une remarque ? N'hésitez pas à venir en parler dans le chat si je suis en cours de diffusion !

[Arduino](#) / [I - Annexes](#) / Utiliser un module Bluetooth HC-05 avec Arduino

Utiliser un module Bluetooth HC-05 avec Arduino

[Eskimon](#) , le ven. 03 décembre 2010

[arduino](#) [tuto](#)

Un fil USB pour communiquer c'est contraignant ! On est limité en distance et on ne peut pas mettre en boîte notre carte, une vraie laisse ! Heureusement, ce ne sont pas les solutions sans fil qui manquent ! Je vous propose dans cette annexe du tutoriel Arduino de découvrir, utiliser et paramétrer un module bluetooth qui coûte une poignée d'euros : le HC-05.

#_Sommaire

- [Le bluetooth et le module HC-05](#)
 - [Qu'est ce que le bluetooth ?](#)
 - [Présentation du module bluetooth HC-05](#)
 - [Brancher le module à l'Arduino](#)
- [Communication entre HC-05 et Arduino](#)
 - [Une question de voie série](#)
 - [Utiliser la bibliothèque](#)
- [Aller plus loin, la configuration \(AT\)](#)
 - - [Passer en "mode commande"](#)
 - [Envoyer des commandes](#)

#_Le bluetooth et le module HC-05

Afin de bien débiter, voyons d'abord à quoi ressemble le bluetooth et découvrons de quoi est composé le nouveau module que nous allons utiliser.

##_Qu'est ce que le bluetooth ?

Le bluetooth est un protocole de communication sans fil. Il a vu le jour à la fin des années 1990 et n'a vraiment percé que dans les années 2000. Il a subi de nombreuses révisions et évolutions pour atteindre aujourd'hui la version 4.1 depuis la fin 2013. À l'origine, la société Ericsson cherchait à développer un moyen d'utiliser une voie série sans passer par un fil. Petit à petit, ses utilisations se sont étendues pour atteindre ce que l'on connaît aujourd'hui, un moyen de connecter sans fil de nombreux appareils, allant d'une Arduino et sa voie série à un ordinateur, ou pour utiliser un casque audio ou encore une manette de jeu sur une console de salon.

Rentrons un peu dans le détail. Ce protocole est un cousin du Wi-Fi. En effet, ils respectent tous deux une même spécification IEEE et utilisent la même gamme de fréquences : 2.4 GHz (tout comme les téléphones portables et le zigbee par exemple). C'est une communication bidirectionnelle, deux modules peuvent communiquer ensemble en même temps. Le comportement utilisé est "maître/esclave". Un

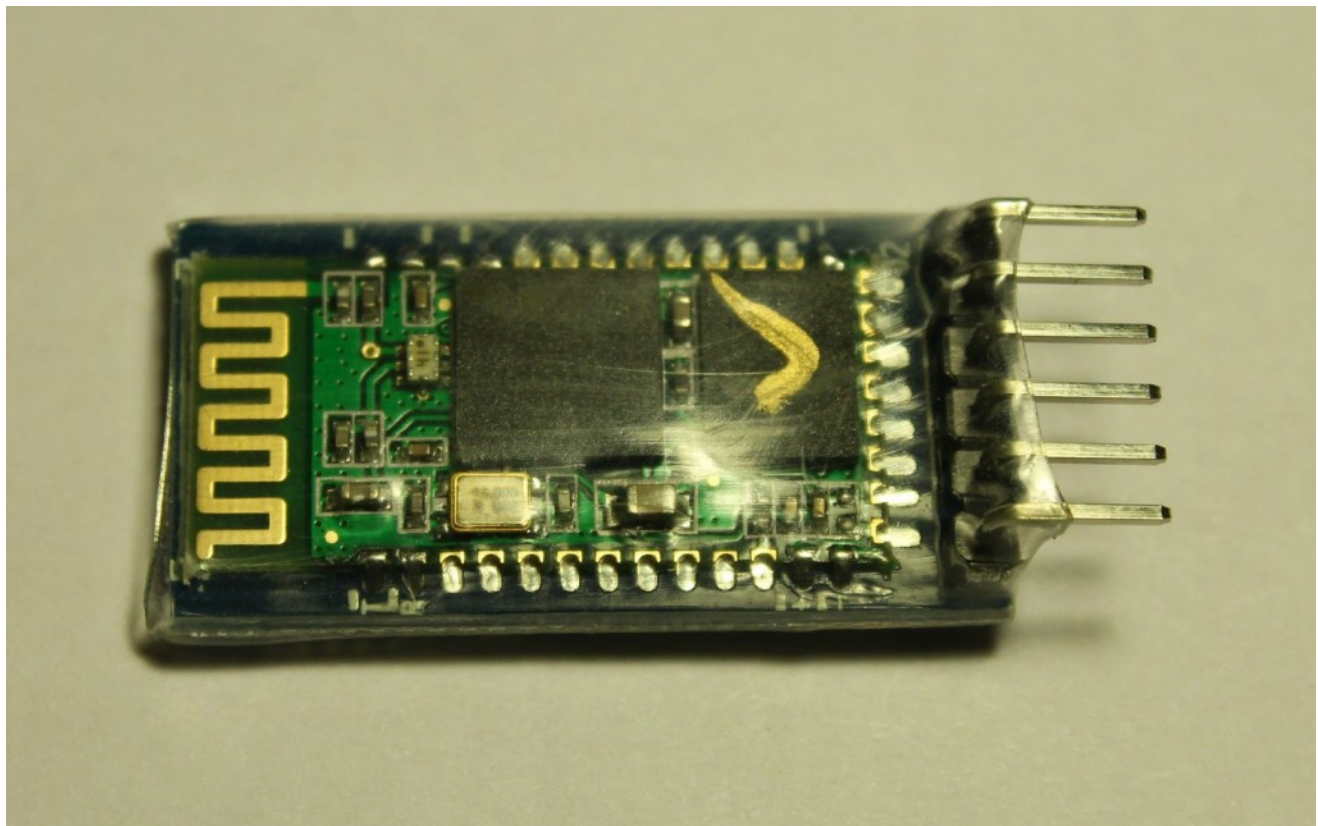
esclave pourra parler avec un seul maître, mais un maître pourra dialoguer avec plusieurs esclaves. Pour son utilisation, elle se passe en plusieurs étapes :

1. Le maître se met en mode "reconnaissable"
2. L'esclave trouve le maître et demande à s'y connecter
3. Le maître accepte la connexion
4. Les périphériques sont alors appairés (ou associés)
5. La communication peut commencer

Ensuite, selon le type de composant que vous utilisez (une oreillette bluetooth, une manette de jeu-vidéo etc) la communication pourra se faire selon un protocole ou un autre. Dans notre cas cela consistera simplement en une liaison série. Au final, nous aurons donc le même fonctionnement qu'une liaison série habituelle (partie 3 du tuto Arduino) mais toute la partie "filaire" de la communication sera englobée dans des trames bluetooth gérées par le module. C'est totalement transparent pour nous (ou presque). Faisons donc un peu plus connaissance avec "HC-05", le petit module bluetooth si pratique.

_Présentation du module bluetooth HC-05

Comme expliqué plus tôt, le module utilisé se nomme HC-05 et est trouvable assez facilement pour quelques euros (via des sites d'import de Chine). Il est aussi gros que le pouce et est en fait un montage d'un module bluetooth sur un petit PCB. Cela permet de s'affranchir de certaines contraintes comme la soudure du module (qui est très délicate), la conversion 5V -> 3.3V, la régulation de l'alimentation (3.3V de nouveau) ou encore l'ajout de LEDs de signal. Tout cela est déjà intégré ! Alors que trouvons nous sur ce petit module ? Tout d'abord, un ensemble de broches. VCC et GND pour l'alimentation (5V), Rx/Tx pour la communication. On y voit aussi une broche "Key" qui servira à envoyer des commandes de configuration au module (nous y reviendrons plus tard). La dernière broche nommée "Led" permet de brancher une LED pour obtenir un signal sur l'état du module.



Le module bluetooth HC-05

_Brancher le module à l'Arduino

Voyons comment brancher tout ça. Tout d'abord, l'alimentation. C'est assez habituel, le VCC sera relié au 5V de l'Arduino et le GND à la masse. Ensuite, viennent les broches de transmission de données, Rx et Tx. Vous pourrez au choix les connecter aux broches de liaison série de l'Arduino (0 et 1) pour utiliser la voie série native, ou alors les brancher sur n'importe quelle autre pin afin de garder la voie série disponible pour du debug avec l'ordinateur (ce que je conseille). **Nous reviendrons sur ce point** . Il ne reste alors plus que les broches Key et Led. Led sert à connecter une LED de statut, vous pouvez la laisser déconnectée cela n'influencera pas le comportement du module. Key sert à utiliser le mode "commande" du module. Avec cette dernière vous pourrez reconfigurer la voie série (vitesse, parité etc...) et d'autres options liées au bluetooth (nom du module, mot de passe d'appairage, mode esclave/maître...). Cette broche est à connecter à n'importe quelle sortie numérique de l'Arduino.

_Communication entre HC-05 et Arduino

Maintenant que le module est connecté, il est temps de le mettre en œuvre avec notre Arduino ! Pour garder les choses simples, nous allons simplement faire une connexion qui permet de renvoyer tout sur la voie série de l'ordinateur. Tout le long de ce tutoriel, seul le mode "esclave" sera utilisé mais le mode "maître" n'est pas beaucoup plus compliqué à atteindre 😊 .

_ Une question de voie série

Comme il a été dit et répété tout au long de la présentation, le module fonctionne via une voie série. Sur une Arduino Uno nous pouvons en trouver seulement une que nous allons garder pour faire du *debugging* avec l'ordinateur le temps de mettre en œuvre le module. Il y a donc un problème, comment allons nous faire pour communiquer avec le module si la voie série est déjà prise ? La réponse est assez simple, nous allons *émuler* une voie série ! Ainsi, nous allons pouvoir utiliser deux broches numériques classiques pour créer une voie série par dessus. C'est une méthode assez habituelle de contournement, c'est équivalent à trouver une solution logicielle à un problème matériel. Pour mettre en œuvre cela il y a deux écoles : Coder la nouvelle voie série soi-même ou utiliser une librairie existante. Si vous avez du temps, la première solution est réellement intéressante et formatrice. Pour ma part, le temps est une denrée ce faisant rare, je vais donc utiliser une bibliothèque existante qui a été codée par des professionnels et testée par des milliers de personnes à travers le monde : SoftwareSerial.

_ Utiliser la bibliothèque

Pour mettre en œuvre cette bibliothèque c'est assez simple. Il va tout d'abord falloir l'inclure dans vos projet. Au choix, soit en cliquant sur "Library/Import/SoftwareSerial" dans l'IDE Arduino (inclus de base) ou alors en ajoutant la ligne suivante en haut de votre programme :

```
1 #include <SoftwareSerial.h>
```

Une fois que la bibliothèque est là, il nous faudra l'utiliser. Cette bibliothèque nous donne l'accès à un objet nommé...SoftwareSerial (quelle coïncidence 🤪). On peut donc créer une instance de cette objet en écrivant :

```
1 SoftwareSerial mavoieserie(Rx, Tx);
```

Rx et Tx seront les numéros des broches sur lesquelles sont connectées les broches d'émission/réception de votre module bluetooth. Par exemple j'ai pour ma part la broche Rx du module branchée sur la pin 10 de l'Arduino et la broche Tx reliée a 11. J'ai donc dans mon programme :

```
1 SoftwareSerial mavoieserie(10, 11);
```

Une fois cela fait, tout est prêt ! En effet, cette bibliothèque offre les mêmes fonctions que la voie série habituelle, la seule différence sera qu'au lieu d'utiliser *Serial* on utilisera *mavoieserie* , comme on peut le voir dans le listing ci-dessous :

```
1 mavoieserie.begin(speed); // démarre la voie série à la vitesse speed
2 mavoieserie.available(); // retourne le nombre de caractère à lire
3 mavoieserie.read();      // retourne le prochain caractère reçu
4 mavoieserie.write(val);  // envoie le char "val" sur la voie série
```

Notre module fonctionnant par défaut en 9600 bauds, nous allons pouvoir le préparer en écrivant :

```
1 #include <SoftwareSerial.h>
2
3 SoftwareSerial mavoieserie(10, 11);
4
5 void setup() {
6     mavoieserie.begin(9600);
7 }
```

Puis afin de tester tout cela, nous allons faire un petit code qui se contentera de faire un "echo" entre la voie série liée à l'ordinateur et celle liée au module bluetooth. Ainsi, chaque fois qu'un caractère/texte sera reçu sur la voie série de l'un, il sera renvoyé sur la voie série de l'autre. Notre Arduino servira alors de relais/adaptateur entre l'USB de l'ordinateur et le module bluetooth 🤪 . Ce petit programme vous donne alors tout ce qu'il faut savoir pour faire vos premières applications avec 🤪 . Juste une chose : branchez la broche "Key" à la masse pour que le module soit en mode "communication" et non pas en mode commande.

```
1 #include <SoftwareSerial.h>
2
3 SoftwareSerial mavoieserie(11, 10); // (RX, TX) (pin Rx BT, pin Tx BT)
4
5 void setup()
6 {
7     // Ouvre la voie série avec l'ordinateur
8     Serial.begin(9600);
9     // Ouvre la voie série avec le module BT
10    mavoieserie.begin(9600);
11 }
12
13 void loop() // run over and over
14 {
15     if (mavoieserie.available()) {
16         Serial.write(mavoieserie.read());
17     }
18     if (Serial.available()) {
19         mavoieserie.write(Serial.read());
20     }
21 }
```

An error occurred.

[Try watching this video on www.youtube.com](#), or enable JavaScript if it is disabled in your browser.

_Aller plus loin, la configuration (AT)

Si vous voulez pouvoir utiliser plus de fonctionnalités avec votre module, il va falloir modifier ses paramètres internes. Pour pouvoir être configuré, le module utilise un jeu de commandes plus ou moins standards que l'on appelle "commande AT" (**AT** tention) ou *Commande Hayes* du nom de leur inventeur. Pour les utiliser, il faudra que le module soit placé dans un mode particulier, le mode "commande". Dans ce mode, toutes les informations envoyées au module seront interprétées pour faire de la configuration, rien ne sera envoyé en bluetooth à un autre appareil.

Les commandes AT portent plutôt bien leur nom. En effet, tous les ordres de configuration commencent par les caractères "AT+" suivis de l'ordre en question puis un retour à la ligne via les caractères "\r\n". Voyons cela plus en détail.

_Passer en "mode commande"

Pour que le module décide que ce qu'il reçoit doit être traité comme un ordre de configuration et non comme un bout de texte à transmettre, il faudra le placer en mode "commande". Dans ce mode, plus aucun texte ne sera transmis. Afin d'établir cette communication, il existe deux méthodes. La première consiste à démarrer le module avec sa broche "Key" à l'état HAUT (reliée au 5V). La LED présente sur le module devrait alors clignoter rapidement, signalant que le module est prêt à recevoir des ordres AT. Dans cette configuration, tous les messages envoyés seront interprétés comme des commandes et le seul moyen d'en sortir sera de redémarrer le module (en supprimant son alimentation). Attention, dans ce mode la connexion s'effectue à une vitesse de 38400 bauds ! N'oubliez pas de le prendre en compte dans votre programme. Je déconseille cette première méthode qui est finalement assez contraignante.

Une deuxième méthode, plus souple, sera d'utiliser la broche "Key" dans notre programme. Il faut donc dans un premiers temps déclarer cette dernière en sortie et la mettre à l'état bas par défaut. Elle sera connectée à la pin 12 pour ma part. Dans cette situation, la vitesse restera à 9600 bauds.

```
1  const int key = 12;
2
3  void setup() {
4      pinMode(key, OUTPUT);
5      pinMode(key, LOW);
6  }
```

Maintenant, chaque fois que vous souhaitez passer dans le mode "commande" vous devrez mettre cette broche à l'état haut et la maintenir dans cet état tant que vous envoyez des commandes.

Envoyer des commandes

Comme je l'écrivais plus tôt, chaque commande envoyée doit commencer par "AT+..." et finir par un retour à la ligne "\r\n". Il existe plusieurs commandes que l'on trouve en général dans les documents fournis par le constructeur. Vous pourrez trouver ce jeu de commande en faisant une recherche du type "HC-05 AT commandes" dans votre moteur de recherche préféré ou en téléchargeant [ce fichier](#) (en anglais of course). Que voyons nous ici ? À partir de la page 3, on peut voir une série de commandes et leur retour. Essayons tout cela. Pour cela, passez tout d'abord votre module en mode commande en mettant sa broche Key à 1. Maintenant, en reprenant le programme vu plus haut faisant un echo dans la voie série, envoyons la chaîne suivante "AT+NAME". Cette commande passée sans paramètre donne le nom visible du module. Par défaut vous devriez obtenir "HC-05" en retour. Si maintenant nous voulions changer ce nom, nous ferions "AT+NAME=TEST" pour l'appeler "TEST" par exemple. Si tout se passe bien, le module doit vous répondre "OK". C'est aussi simple que ça ! D'une manière générale, pour lire un paramètre on fera "AT+[paramètre]" et pour le changer on fera **AT+[paramètre]=[valeur]** . Lorsque vous utiliserez le module sans aucune liaison série externe pour vérifier, il peut être bon de s'assurer que tout se passe bien en vérifiant que le module renvoie bien "OK" lorsque vous changez un paramètre. Si au début du mode commande vous voulez vérifier que le module est bien présent et prêt à répondre, vous pouvez envoyer simplement "AT" et il vous répondra "OK" !

Une erreur s'est produite.

[Essayez de regarder cette vidéo sur \[www.youtube.com\]\(http://www.youtube.com\)](#), ou activez JavaScript dans votre navigateur si ce n'est pas déjà le cas.

Vous devriez maintenant être en mesure d'utiliser votre module bluetooth HC-05. Il ne reste plus qu'à coder une petite application sur votre PC ou un téléphone et vous pourrez faire du pilotage à distance complètement autonome, sans fil relié à un ordinateur pour recevoir les ordres !



Vous avez une question ? Vous souhaitez partager votre experience ? Venez nous rejoindre sur Discord !

