



## Modules HC-05- Arduino

### 1. Objectif :

- Appairer (associer ensemble) deux modules Bluetooth HC-05 pour leur permettre de communiquer entre eux

### 2. Matériel :

- 2 cartes Arduino UNO
- 2 modules Bluetooth HC-05
- Ordinateur + fil de branchement pour la carte
- Application Arduino IDE
- 2 piles 9V plus support avec Jack ou 2 alimentations Jack qui se branchent au mur.
- Quelques fils et une platine d'essai
- 2 kits : 1 résistance 2k $\Omega$  et 1 résistance 1k $\Omega$  ( ou 3 résistances de 1k $\Omega$ )

### 3. Ce qui sera fait dans ce projet

La première partie du projet consiste à préparer le rôle de chacun des modules HC-05 pour établir une communication entre eux. Ils seront associés l'un à l'autre et n'enverront des informations qu'entre eux.

La deuxième partie du projet sera de vérifier la communication entre les cartes Arduino à l'aide des modules. Les programmes qui sont faits peuvent alors servir de point de départ pour tout projet de type sans fil à l'aide des modules HC-05!

### 4. Petite introduction au Bluetooth

La technologie Bluetooth a énormément évolué depuis ses débuts dans les années 1994. La première norme a été proposée par le fabricant suédois Ericsson<sup>1</sup>. Cette norme de communication devait permettre un échange entre les ordinateurs et les téléphones cellulaires. Son utilisation a depuis été largement généralisée. Le nom Bluetooth (littéralement « dent bleue ») provient de la traduction anglaise du nom du roi viking

---

<sup>1</sup>La plupart des informations de cette section proviennent de : <https://fr.wikipedia.org/wiki/Bluetooth>

Harald Blâtand ou *Harald à la dent bleue*, qui avait unifié les tribus danoises en un seul royaume. La communication Bluetooth, comme le roi qui lui a prêté son nom, permet d'«unifier» la communication entre divers appareils. L'icône représentant le Bluetooth provient des initiales en alphabet runique du fameux roi viking.

Même si la communication Bluetooth peut sembler relativement aisée à utiliser, les choses se compliquent assez rapidement quand on cherche à en comprendre le fonctionnement sous-jacent. La plus grande difficulté réside dans le fait que les nombreuses normes ont apporté des termes différents, des nouveaux rôles pour les appareils et des façons différentes d'établir la connexion. Sans tenter d'offrir des réponses à toutes les questions qui sont soulevées lorsque l'on travaille avec Bluetooth et Arduino, le présent document, et les autres qui traitent de ce sujet, présente des méthodes pour communiquer à l'aide de Bluetooth avec deux appareils qui ont été testées et qui fonctionnent. Le but étant, par exemple, de pouvoir concevoir des robots commandés à distance à l'aide de deux cartes Arduino.

Il est aussi possible de concevoir une application sur votre téléphone cellulaire et de l'utiliser pour contrôler votre robot. Cependant, ce sujet ne sera pas traité ici. Vous pouvez quand même essayer les différentes applications qui existent déjà, mais cela dépasse les objectifs du présent document.

Il existe plusieurs types de Bluetooth, en fonction des possibilités offertes par la technologie et les différentes normes. On retrouve, dans les grandes lignes, le Bluetooth Classique et le Bluetooth *Low Energie* (BLE). Les deux catégories utilisent des façons différentes pour envoyer l'information, des vitesses de transfert différentes, etc. Elles utilisent toutes les deux des ondes radio dont les fréquences sont autour de 2.4 GHz.

Dans tous les cas, un appareil possédant la connectivité Bluetooth possède une antenne qui sert d'émetteur et/ou de récepteur de signal. La portée du signal Bluetooth dépendra de la puissance de cette antenne. De manière générale, on peut s'attendre à une portée d'environ 10 mètres. Le Wi-Fi, les autres appareils Bluetooth et même les fours à micro-ondes (en fonction...) peuvent parfois causer des interférences car ils utilisent des ondes radio à des fréquences similaires.

## **5. Choix et rôles des deux cartes (explications très simplifiées des principes de base)**

Lorsque l'on utilise habituellement la communication Bluetooth entre deux appareils, c'est généralement pour brancher un périphérique (souris, montre intelligente, clavier, ...) à un ordinateur.

L'ordinateur va rechercher les périphériques disponibles et afficher une liste. Il peut aussi se connecter ou s'associer à plusieurs périphériques différents.

Le périphérique « annonce », quand à lui, son existence aux autres appareils avec connectivité Bluetooth qui sont situés à proximité. Une fois apparié ou associé, il ne se connecte qu'avec un seul appareil.

Dans les réseaux Bluetooth plus simples (*piconets*), l'appareil qui peut s'associer à plusieurs autres se nomment *Master* (maître). Les appareils qui ne se branchent qu'à un seul se nomment *Slave* (esclave). Cependant, des réseaux plus complexes peuvent être formés (*Mesh*) si la technologie le permet, où l'on peut faire l'interconnexion entre plusieurs appareils. Cependant, ce ne sont pas tous les appareils Bluetooth qui permettent ceci. Ici, nous nous contenterons d'associer seulement deux modules HC-05 ensemble, donc le réseau créé sera simple. En général, l'appareil central (ordinateur par exemple) deviendra le *Master* et le périphérique deviendra le *Slave*.

Une fois la connexion effectuée, la communication peut être faite dans les deux directions. Cela signifie que l'ordinateur et l'appareil peuvent envoyer une information (écriture) ou demander une information (lecture) à l'autre. Cependant, il peut arriver que l'écriture ou la lecture soit restreinte pour certaines valeurs. Ce ne sera pas le cas avec les modules HC-05.

Comme on veut appairer les deux modules ensemble, l'un devra jouer le rôle de l'ordinateur (*central* ou *master*) et l'autre celui du périphérique (*peripheral* ou *slave*). Le choix de quel module joue quel rôle est sans importance, puisqu'une fois la connexion établie entre les deux, l'information va pouvoir circuler des deux côtés entre les modules. À noter que les modules HC-05 utilisent la norme Bluetooth 2.0 (Bluetooth classique). Ils pourraient donc ne pas être compatibles avec des modules BLE ou de norme plus récente. Il faut toujours vérifier la compatibilité des appareils Bluetooth, car elle n'est pas garantie, même s'il s'agit de Bluetooth dans tous les cas. Ce ne sont pas tous les appareils Bluetooth qui ne permettent pas la connexion avec des appareils dont la norme est différente. Cependant, cela peut arriver et il faut en être conscient lorsque l'on fait des choix technologiques.

## 6. Circuit

Les modules HC-05 ont en général 6 broches : KEY, VCC, GND, TXD, RXD et STATE.

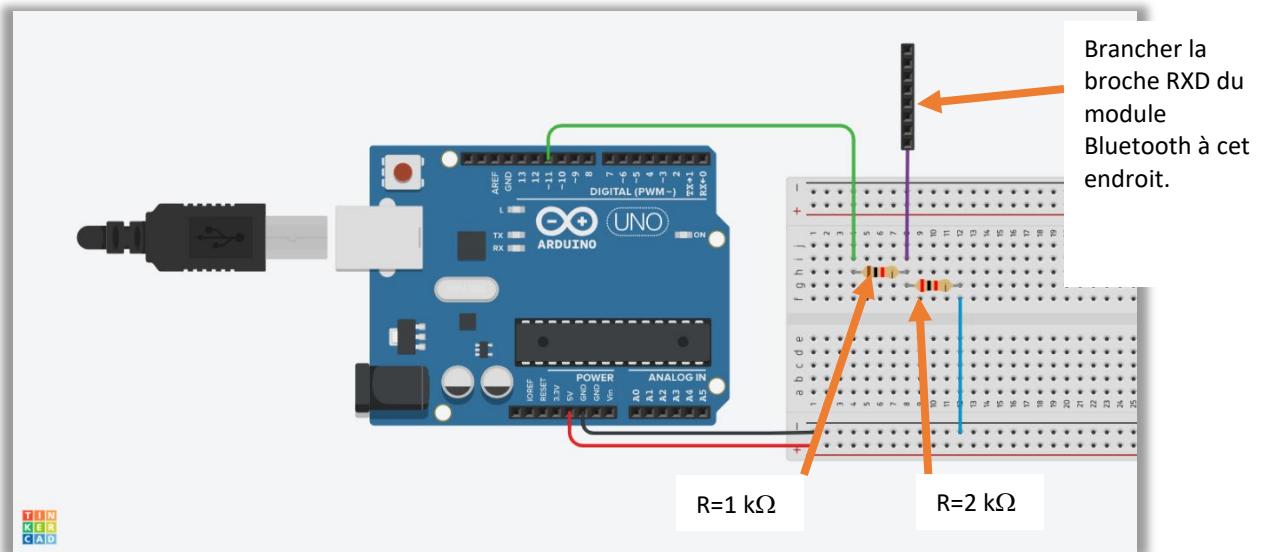
Pour l'instant, nous n'utiliserons pas la broche STATE, donc elle ne sera branchée à rien.

La broche VCC devra être branchée à une source de tension. Dans notre cas, la broche VCC sera branchée à la borne +5V de la carte Arduino.

La broche GND sera branchée sur une broche GND de la carte Arduino.

Pour la configuration en commandes AT, la broche KEY sera branchée à une broche digitale (2 à 13) qui sera mise à une valeur HIGH pendant la programmation. On pourrait aussi la brancher sur la broche +5V de la carte Arduino. Certains modules HC-05 doivent être branché plutôt sur la broche +3.3V. Toujours vérifier les informations du fabricant pour être certain d'utiliser les bonnes broches.

En fonction des recommandations du fabricant, certains modules HC-05 doivent être utilisés avec un diviseur de tension pour la broche RXD. Dans ce cas, faire le circuit suivant :



Il faudra brancher la broche TXD sur la broche digitale 10 de la carte Arduino.

Il faudra brancher la broche RXD sur la broche digitale 11 de la carte Arduino (ou sur le diviseur de tension qui lui est branché dans la broche 11).

Sur les cartes Arduino UNO, toutes les broches digitales peuvent être utilisées pour RX et TX, sauf la broche 13. On pourrait aussi utiliser les broches 0 et 1 (qui sont déjà marquées par TX et RX) et utiliser les fonctions `Serial.begin()` et autres que l'on utilise pour communiquer avec le moniteur série dans l'application Arduino IDE. Cependant, si on utilise ces broches pour la communication Bluetooth, on ne pourra pas communiquer avec l'ordinateur par le port USB. On pourra, par contre, utiliser une bibliothèque, `SoftwareSerial`, qui permettra de faire la communication série avec le module Bluetooth. Dans le cas où l'on ne souhaite pas communiquer aussi avec un ordinateur, il serait préférable d'utiliser les broches 0 et 1 de la carte Arduino, puisque ces broches sont conçues pour ce genre de communication.

En résumé, les connexions suivantes doivent être faites pour pouvoir configurer le module en commandes AT :

HC-05	Arduino
VCC	+5V ou + 3.3V selon le fabricant
GND	GND
TXD	Broche digitale 10 ou diviseur de tension
RXD	Broche digitale 11
KEY ou EN	+5V ou Broche digitale 8 (ou autre)
State	Rien

En mode normal, on réalise le même circuit, mais il ne faut pas brancher la broche KEY (ou EN).

À noter qu'il est toujours préférable de vérifier les branchements recommandés par le fabricant du module.

## 7. Étapes de réalisation

### Configuration du module par Commandes AT :

Pour pouvoir accéder aux commandes AT et configurer les modules, il faut alimenter en tension la broche KEY ou EN avant d'alimenter le module.

1. Brancher le module Bluetooth à la carte Arduino, brancher la carte Arduino à l'ordinateur et ouvrir l'application Arduino IDE.
2. Débrancher la broche Vcc du module HC-05.
3. On doit ensuite s'assurer que le bon type de carte et le bon port de communication sont utilisés. Dans l'onglet *Outils*, modifier le *Type de carte* dans la liste déroulante si « Arduino/Genuino UNO » n'est pas sélectionné. Sélectionner aussi le bon port de communication.
4. Télécharger le programme suivant dans la carte Arduino :

```
#include <SoftwareSerial.h>

//RXD dans la broche 11 et TXD dans la broche 10 (ou diviseur de tension)
SoftwareSerial moduleSerial(10, 11);

void setup() {

    //mettre la broche 8 à +5V pour la broche KEY pour entrer en commandes AT
    pinMode(8, OUTPUT);
    digitalWrite(8, HIGH);
}
```

```

//commencer la communication avec le moniteur Série
Serial.begin(9600);
Serial.println("Entrer les commandes AT : ");

//commencer la communication avec le module HC-05
moduleSerial.begin(38400);
}

void loop(){

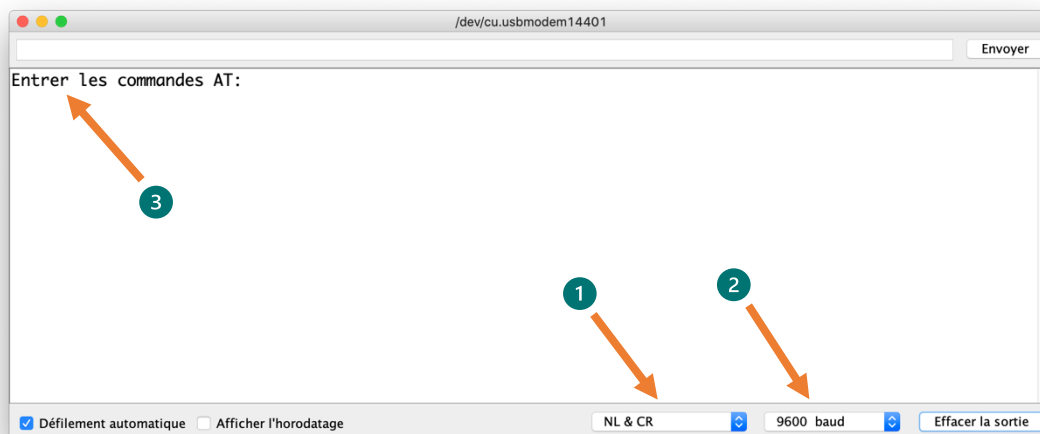
// Lire les données du module et les envoyer dans le moniteur série
if (moduleSerial.available())
    Serial.write(moduleSerial.read());

// Lire les données du moniteur série et les envoyer au module
if (Serial.available())
    moduleSerial.write(Serial.read());
}

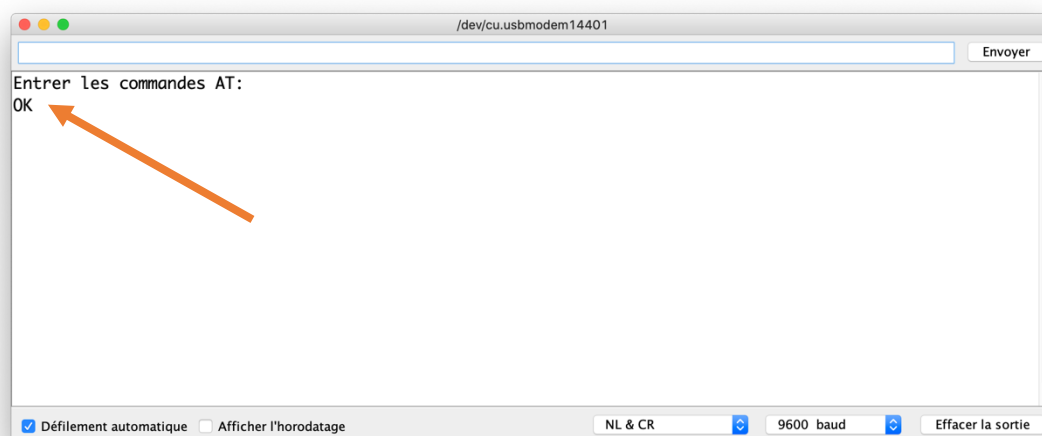
```

5. Brancher la broche VCC à la borne +5V de la carte Arduino. La DEL sur le module HC-05 devrait maintenant clignoter à un intervalle de 4 secondes (allumée 2 secondes, éteinte 2 secondes). Cela signifie que le module peut être configuré avec les commandes AT. Si cela n'a pas fonctionné, débrancher la broche VCC du module et la rebrancher dans la borne +5V.
6. Cliquer sur l'onglet « *Outils* » et ensuite sur « *Moniteur série* ». C'est par le moniteur série que la configuration de la carte sera faite.
7. Dans la fenêtre du Moniteur série, s'assurer quand même que le moniteur est bien configuré :
  1. Sélectionner « NL&CR » dans la liste déroulante.
  2. Sélectionner « 9600 baud » dans la liste déroulante.

3. Le message « Entrer les commandes AT : » devrait apparaître. Il se peut qu'il apparaisse deux fois, mais ça n'est pas un problème.



8. Entrer « AT » sans les guillemets à l'endroit indiqué. Appuyer sur la touche « Entrée » ou cliquer sur *Envoyer*.  
Le message « OK » devrait apparaître. Sinon, vérifier le programme et les connexions du module.



Le module est prêt à être configuré.

9. Quelques petits conseils pour les commandes AT, car il peut être légèrement frustrant d'utiliser ces commandes la première fois.
- Les majuscules et minuscules sont importantes. « AT+CMODE=? » par exemple, n'est pas la même chose que « at+cmode=? ».
  - Lorsqu'une commande est erronée, il y aura un message « ERROR:(0) » affiché. Si la commande est effectuée correctement, le message sera plutôt « OK ». Il n'y aura pas plus de détails. Le chiffre entre parenthèses permet d'identifier le genre d'erreur. On peut retrouver plus de détails dans la documentation du module.
  - Il peut arriver qu'une commande ne fonctionne pas uniquement pour une faute de frappe ou encore parce qu'on n'a pas changé la sélection « Pas de fin de ligne » pour « NL & CR ». Réessayer la commande à nouveau malgré tout en vérifiant bien la frappe.
  - Ne jamais entrer les guillemets dans les commandes. Ils sont présents uniquement pour permettre une lecture plus facile.

### **Configuration de la carte en mode *Peripheral* ou *slave***

Nous devons d'abord vérifier la configuration de la carte et faire les modifications qui sont nécessaires seulement.

- Entrer la commande : « AT+ROLE? »
  - Si la réponse est : « 0 », passer à l'étape suivante.
  - Si la réponse est : « 1 », entrer la commande « AT+ROLE= 0 ». Si la réponse « OK » apparaît, tout va bien. Sinon, réessayer en vérifiant la frappe.

Cela permet de configurer la carte comme *slave* ou *peripheral*

- Entrer la commande : « AT+CMODE? »
  - Si la réponse est : « 1 », passer à l'étape suivante. Cela correspond à une connexion à n'importe quelle adresse.
  - Si la réponse est : « 0 », entrer la commande « AT+CMODE=1 ». Si la réponse « OK » apparaît, tout va bien. Sinon, réessayer en vérifiant la frappe.

Cela permet de configurer la carte pour qu'elle se branche sur n'importe quelle carte qui le lui demande.

- Entrer la commande : « AT+UART? »
  - Si la réponse est : « 38400,0,0 », passer à l'étape suivante.



- b. Si la réponse est autre chose, entrer la commande « AT+UART=38400,0,0 ». Si la réponse « OK » apparaît, tout va bien. Sinon, réessayer en vérifiant la frappe.

Cela permet de configurer la carte pour qu'elle communique au bon rythme. Si l'on choisit d'autres configurations (38400, 1, 2, par exemple), peut donner lieu à des comportements qui ne sont pas ceux attendus. Par contre, on peut très bien choisir d'autres vitesses de transfert que 38400 bauds. L'important est que les deux modules soient configurés à la même vitesse.

#### 4. Entrer la commande : « AT+ADDR? »

Cela permet d'obtenir l'adresse (*MAC address*) de la carte. Noter la valeur obtenue (qui devrait être sous un format semblable à +ADDR:98d3:32:711103).

Ne pas fermer le Moniteur Série. Débrancher la carte Arduino de l'ordinateur. Enlever le module HC-05 et le remplacer par l'autre.

5. S'assurer que le bon port est sélectionné pour la carte dans l'onglet Outils.
6. Ouvrir le moniteur série.
7. Télécharger le programme dans la carte Arduino.
8. Refaire les étapes 5 à 8 de la Configuration du module par Commandes AT.

### **Configuration de la carte en mode *Master ou Central***

1. Entrer la commande : « AT+ROLE? »
  - c. Si la réponse est : « 1 », passer à l'étape suivante.
  - d. Si la réponse est : « 0 », entrer la commande « AT+ROLE= 1 ». Si la réponse « OK » apparaît, tout va bien. Sinon, réessayer en vérifiant la frappe.

Cela permet de configurer la carte comme *Master ou Central*

2. Entrer la commande : « AT+CMODE? »
  - c. Si la réponse est : « 1 », passer à l'étape suivante. Cela correspond à une connexion à une adresse fixe.
  - d. Si la réponse est : « 0 », entrer la commande « AT+CMODE=1 ». Si la réponse « OK » apparaît, tout va bien. Sinon, réessayer en vérifiant la frappe.

Cela permet de configurer la carte pour qu'elle se branche sur le module dont l'adresse est fixée.

3. Entrer la commande : « AT+UART? »
  - c. Si la réponse est : « 38400,0,0 », passer à l'étape suivante.

- d. Si la réponse est autre chose, entrer la commande « AT+UART=38400,0,0 ». Si la réponse « OK » apparaît, tout va bien. Sinon, réessayer en vérifiant la frappe.

Cela permet de configurer la carte pour qu'elle communique au bon rythme avec l'autre carte.

4. Entrer la commande : « AT+BIND=98d3,32,711103 » (voir plus haut pour correspondance avec l'adresse de l'autre module ).

Bien entendu, vous devez remplacer l'adresse indiquée ici par l'adresse de votre autre carte. Attention à la casse. La réponse devrait être « OK ». Sinon, il vous manque peut-être un chiffre.

Pour vérifier que l'adresse a bien été entrée, entrer la commande « AT+BIND? ». La réponse devrait être l'adresse entrée. Sinon, recommencer l'étape au complet.

5. Fermer le Moniteur Série et débrancher la carte Arduino.

Les deux modules sont maintenant prêts à être utilisés.

## 8. [Code](#)

### Carte Peripheral

```
#include <SoftwareSerial.h>
SoftwareSerial moduleSerial(10, 11);
byte i=0;
void setup() {
    moduleSerial.begin(38400);
    pinMode(13, OUTPUT);
}
void loop() {
    if(moduleSerial.available()>0) {
        i=moduleSerial.read();
        if(int(i)%2==0){
            digitalWrite(13, LOW);
        }
        else{
            digitalWrite(13, HIGH);
        }
    }
}
```

## Carte Central

```
#include <SoftwareSerial.h>

SoftwareSerial moduleSerial(10, 11);
void setup() {
  moduleSerial.begin(38400);
}

void loop() {
  for(byte i=0; i<=10; i++){
    moduleSerial.write(i);
    delay(1000);
  }
}
```

## 9. Fonctionnement des programmes

Si l'on veut vérifier que les valeurs reçues par la carte Arduino sont celles qui ont été envoyées, on peut utiliser le moniteur Série et ajouter un `Serial.begin(9600)` à la carte qui sert à recevoir les données. On peut alors afficher ce qui a été envoyé à la carte dans le moniteur série. Comme les données sont communiquées en *byte* ici, il est préférable d'utiliser les fonctions `Serial.write()` et `Serial.read()` pour communiquer avec le moniteur série.

Pour vérifier que la communication est établie, il a été choisi de faire clignoter une DEL à un rythme dépendant du temps écoulé entre deux valeurs reçues. Il serait possible de vérifier autrement, faire varier l'intensité, changer le rythme en fonction des valeurs reçues, ... ce qui vous sera proposé de faire dans les défis! Au lieu de faire un circuit et d'utiliser une DEL extérieure, pour des raisons de simplicité, la DEL embarquée sur la carte est utilisée. Celle-ci est verte et se trouve en-dessous de la DEL « ON ». Elle est contrôlée par la broche digitale 13.

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

On utilise la bibliothèque « `SoftwareSerial` » pour générer un port série permettant la communication à l'aide de broches différentes de 0 et 1 de la carte Arduino.

```
SoftwareSerial moduleSerial(10, 11);
```

On initialise le port série moduleSerial (ce nom peut être modifié). Les broches 10 et 11 sont utilisées pour Rx et Tx. Sur une carte Arduino Uno, on peut utiliser n'importe quelle broche digitale de 2 à 12, inclusivement.

```
byte i=0;
```

Par le port série, les données sont transmises sous forme de *Byte*, ce qui signifie octet. Les valeurs de ce type de données sont entières et situées entre 0 et 255. Pour commencer, il est plus simple et rapide d'envoyer les données sous cette forme.

```
moduleSerial.begin(38400);
```

Pour les deux cartes, la communication Bluetooth est effectuée à une vitesse de 38400 baud. Le *baud* est unité de rapidité de modulation du signal envoyé. À toutes fins pratiques, plus ce chiffre est grand, plus il y aura d'information transmise par seconde. Toutefois, le lien entre cette rapidité et le débit de transfert d'information est plus complexe. L'important est que la valeur en baud du signal soit la même pour les deux cartes.

```
if(moduleSerial.available()>0) { }
```

On vérifie s'il y a des données à lire à partir du port Série. Cette condition est remplie seulement si des informations sont envoyées à la carte qui est programmée avec cette condition. Elle est présente pour la carte qui reçoit un signal de l'autre carte, mais pas pour celle qui l'envoie. En effet, la fonction « `moduleSerial.available()` » renvoie le nombre de octets (bytes) disponibles à la lecture dans le port Série et la carte *Central* ne reçoit rien pour l'instant.

```
int(i)%2==0
```

Ici, cette condition vérifie si la valeur de *i* qui est lue au port Série est paire (un multiple entier de 2). L'opération « `a%b` » retourne le reste de la division entière entre *a* et *b*. Noter que *a* et *b* doivent être des nombres entiers. Pour que *i*, qui est un *Byte*, soit du bon format, on le transforme à l'aide de la fonction `int(i)`.

Par exemple, `3%2` vaut 1  
`8%2` vaut 0.

### Réalisation :

1. La première étape est de préparer les modules avec les commandes AT.
2. La deuxième étape est de brancher chaque carte dans l'ordinateur et de télécharger le programme destiné à chacune, avec le module HC-05, bien entendu.
3. Ensuite, alimenter chaque carte avec sa propre pile. Si une des cartes est toujours branchée dans l'ordinateur, ça fonctionnera aussi. Si l'on souhaite envoyer la valeur lue au panda avec l'extension Diffusion en mode Téléchargement, bien s'assurer de brancher la borne carte Arduino à l'ordinateur et de télécharger son programme en dernier.
4. Si la DEL des modules clignote rapidement et sans arrêt ou pas au même rythme sur les deux modules, vérifier les connexions et ensuite vérifier les programmes. Les deux modules ne sont pas bien associés. Il faut parfois un peu de temps (quelques secondes) avant que les deux modules soient bien appariés. Avant de refaire les étapes de préparation, vérifier que tout est conforme, s'assurer que la broche KEY n'est plus branchée et ensuite, mettre hors tension et sous tension les modules. Si cela ne fonctionne toujours pas, vérifier à partir du début.
5. Après quelques secondes, les DEL sur les modules devraient se mettre à clignoter à un rythme régulier. La DEL embarquée sur la carte qui a reçu le programme « Carte Peripheral » devrait se mettre à clignoter en s'allumant et s'éteignant pendant 1 seconde à répétition.

### IMPORTANT

Une fois que la communication a pu être établie entre les deux modules, il n'est pas nécessaire de refaire leur configuration en commandes AT. Cette configuration restera tant qu'elle n'est pas refaite (en commandes AT), même si les modules n'ont plus d'alimentation pendant longtemps, que l'on appuie sur le bouton « RESET » ou que l'on télécharge un nouveau programme. À noter que le module configuré en *Peripheral* ou *Slave* peut se brancher à n'importe quel module *Master* qui est configuré avec son adresse.

### Défis :

- Modifier le rythme d'envoi des valeurs de la carte Central
- Modifier le délai de clignotement en fonction de la valeur lue plutôt que sa parité ou le délai de réception des valeurs.
- Modifier le programme pour faire allumer plus d'une DEL à partir des valeurs lues (implique un petit circuit!)
- Utiliser le moniteur série pour afficher les valeurs reçues.
- Toute autre chose que vous voudriez essayer!