Modules Bluetooth HM-10- Arduino

1. Objectifs :

- Appairer (associer ensemble) deux modules Bluetooth HM-10 pour leur permettre de communiquer entre eux

2. Matériel :

- 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 2kΩ et 1 résistance 1kΩ (ou 3 résistances 1kΩ)
- 2 modules HM-10
- 2 cartes Arduino

3. Ce qui sera fait dans ce projet

La première partie du projet consiste à préparer le rôle de chacun des modules HM-10 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 HM-10!

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[1]. 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

[1]La plupart des informations de cette section proviennent de : https://fr.wikipedia.org/wiki/Bluetooth

Harald Blåtand 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 sont 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 », quant 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 certains appareils ou à certaines valeurs. Ce ne sera pas le cas avec les modules HM-10.

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 HM-10 utilisent la norme Bluetooth 4.0 (Bluetooth Low Energy). Ils pourraient donc ne pas être compatibles avec des modules Bluetooth 2.0 (Bluetooth classique) 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. Quoiqu'il en soit, les modules HM-10 peuvent communiquer les uns avec les autres.

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. Avant d'utiliser les modules, il faut les préparer à l'aide des commandes AT.

6. Circuit pour les commandes AT

Les modules HM-10 ont en général 6 broches : STATE, VCC, GND, TXD, RXD et BRK.

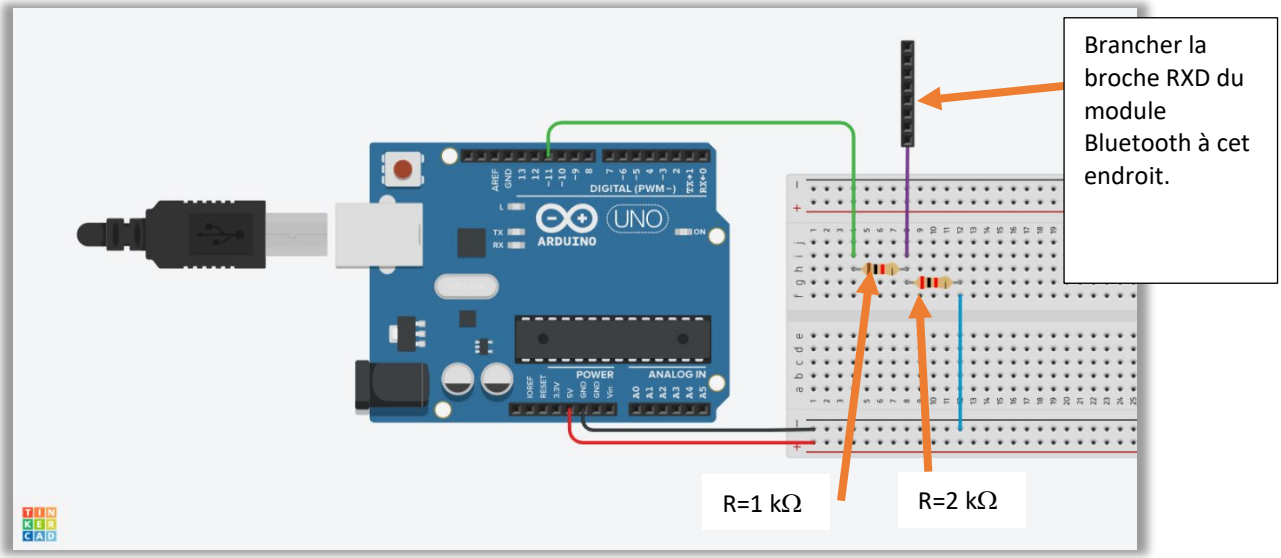
Pour l'instant, nous n'utiliserons pas la broche STATE, donc elle ne sera branchée à rien.

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.

Certains modules HM-10 pourraient demander une alimentation 3.3 V. Toujours vérifier les informations du fabricant pour être certain d'utiliser les bonnes broches.

En fonction des recommandations du fabricant, les modules HM-10 doivent être utilisés avec un diviseur de tension pour la broche RXD. Dans ce cas, faire le circuit suivant :



On peut utiliser deux résistances de 1 kΩ en série au lieu d'une résistance de 2 kΩ.
Il faudra brancher la broche TXD sur la broche digitale 9 de la carte Arduino.

Il faudra brancher la broche RXD sur la broche digitale 8 de la carte Arduino (ou sur le diviseur de tension qui lui est branché dans la broche 8).

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

HM-10	Arduino
VCC	+5V ou + 3.3V selon le fabricant
GND	GND
TXD	Broche digitale 8 ou diviseur de tension
RXD	Broche digitale 9
BRK	Rien
STATE	Rien

Une fois que les modules seront configurés, on pourra utiliser une bibliothèque, SoftwareSerial, qui permettra de faire la communication série avec le module Bluetooth à partir de n'importe quelle broche digitale de la carte Arduino, sauf 0 et 1. Il est aussi techniquement possible[2] de brancher les modules dans les broches digitales 0 et 1, qui jouent déjà le rôle de Rx et Tx et d'utiliser les fonctions Serial de Arduino. Cependant, ces broches sont aussi utilisées lors de la communication via le port USB entre la carte et l'ordinateur. Il peut alors arriver des conflits.

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

Pour les modules HM-10, il existe différentes versions du *firmware* (programme interne du module). Cela ne devrait poser de problèmes lors de l'utilisation. Il faut cependant être conscient que les commandes AT peuvent être différentes. On peut se référer aux références du fabricant en cas de problèmes. De plus, la version 7 du *firmware* utilise une vitesse de communication de 115200 bauds par défaut, alors que les versions précédentes utilisent 9600 bauds. On peut aussi changer cette vitesse en la programmant, mais il faut utiliser la bonne vitesse pour accéder aux commandes AT.

[2] Je n'ai personnellement pas testé ce branchement avec les modules HM-10. Il est préférable d'utiliser la bibliothèque SoftwareSerial avec les modules Bluetooth pour éviter les conflits potentiels.

7. Étapes de réalisation

Configuration du module par Commandes AT :

1. Brancher le module Bluetooth à la carte Arduino avec le diviseur de tension, brancher la carte Arduino à l'ordinateur et ouvrir l'application Arduino IDE.
2. 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.
3. Il faut installer la bibliothèque « AltSoftSerial » dans Arduino IDE.
4. Télécharger le programme suivant dans la carte Arduino. Ce programme provient du site : http://www.martyncurrey.com/hm-10-bluetooth-4ble-modules/

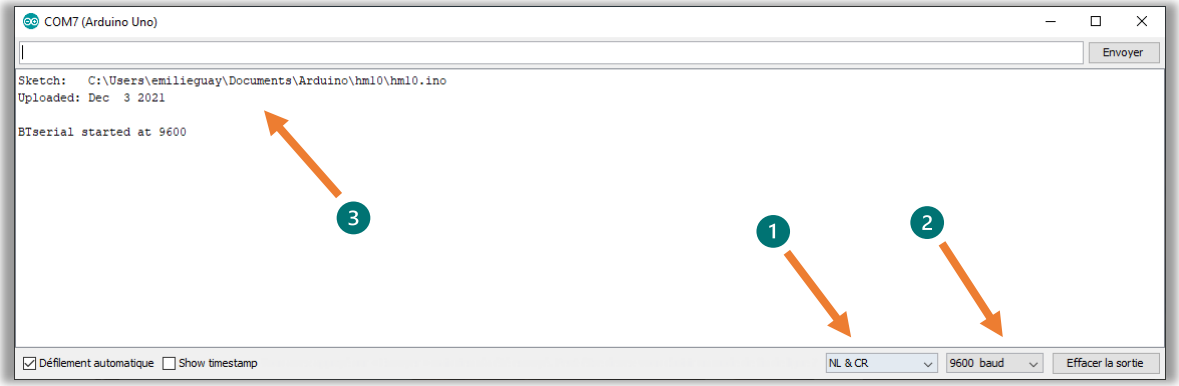
```
// SerialIn_SerialOut_HM-10_01
//
//  Uses hardware serial to talk to the host computer and AltSoftSerial for communication with the bluetooth module
//
// What ever is entered in the serial monitor is sent to the connected device
// Anything received from the connected device is copied to the serial monitor
// Does not send line endings to the HM-10
//
// Pins
// BT VCC to Arduino 5V out.
// BT GND to GND
// Arduino D8 (SS RX) - BT TX no need voltage divider
// Arduino D9 (SS TX) - BT RX through a voltage divider (5v to 3.3v)

#include <AltSoftSerial.h>
AltSoftSerial BTserial;
// https://www.pjrc.com/teensy/td_libs_AltSoftSerial.html

char c=' ';
boolean NL = true;

void setup()
{
  Serial.begin(9600);
  Serial.print("Sketch:   "); Serial.println(__FILE__);
  Serial.print("Uploaded: "); Serial.println(__DATE__);
  Serial.println(" ");
```

```
  BTserial.begin(9600);
  Serial.println("BTserial started at 9600");
}

void loop()
{
  // Read from the Bluetooth module and send to the Arduino Serial Monitor
  if (BTserial.available())
  {
    c = BTserial.read();
    Serial.write(c);
  }


  // Read from the Serial Monitor and send to the Bluetooth module
  if (Serial.available()) {
    c = Serial.read();

    // do not send line end characters to the HM-10
    if (c!=10 & c!=13 )  {
        BTserial.write(c);
    }

    // Echo the user input to the main window.
    // If there is a new line print the ">" character.
    if (NL) { Serial.print("\r\n>");  NL = false; }
    Serial.write(c);
    if (c==10) { NL = true; }

  }
}
```

5. Cliquer sur l'onglet « *Outils* » et ensuite sur « *Moniteur série* ». C'est par le moniteur série que la configuration de la carte sera faite.

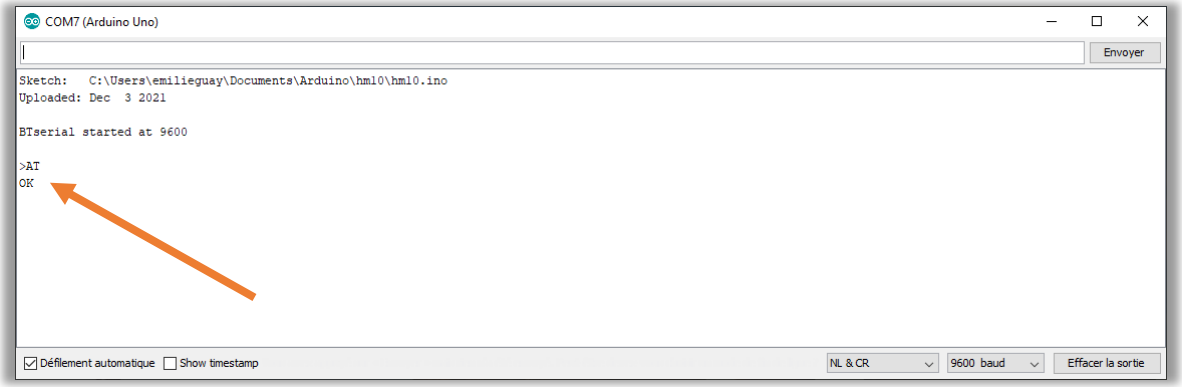
6. Dans la fenêtre du Moniteur série, s'assurer quand même que le moniteur est bien configuré :
    1. Sélectionner « NL&CR » dans la liste déroulante.
    2. Sélectionner « 9600 baud » dans la liste déroulante.

3. Le message de début que l'on peut voir à la figure suivante devrait apparaître. À noter que le nom de fichier devrait être celui du fichier Arduino que vous utilisez et la date devrait être celle d'aujourd'hui.



7. 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. Il peut être possible aussi que la vitesse doive être de 115200 Baud au lieu de 9600 Baud.



Le module est prêt à être configuré.

8. Quelques petits conseils pour les commandes AT, car il peut être légèrement frustrant d'utiliser ces commandes la première fois.
    a. Les majuscules et minuscules sont importantes. « AT+MODE? » par exemple, n'est pas la même chose que « at+mode? ».
    b. Lorsqu'une commande est erronée, il n'y a aucun message affiché, sauf une ligne blanche vide. Il faut entrer la bonne commande encore pour continuer.
    c. 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* ou pour utilisation avec l'application App-IA

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

1. Entrer la commande : « AT »
    a. Si la réponse est : OK passer à l'étape suivante.
    b. S'il n'y a pas de réponse, vérifier les branchements.

2. Entrer la commande : « AT+ROLE? »
    a. Si la réponse est : « OK+Get:0 », passer à l'étape suivante. Cela correspond à une connexion à n'importe quelle adresse.
    b. Si la réponse est : « 1 », entrer la commande « AT+ROLE0 ». Si la réponse est « OK+Set:0 », tout va bien. Sinon, réessayer en vérifiant la frappe.

    Cela permet de configurer la carte pour qu'elle se branche sur n'importe qu'elle carte qui le lui demande.
3. Entrer la commande : « AT+BAUD? »
    a. Si la réponse est : « OK+Get:0», passer à l'étape suivante.
    b. Si la réponse est autre chose, entrer la commande « AT+BAUD0». Si la réponse « OK+Set:0 » 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 peut donner lieu à des comportements qui ne sont pas ceux attendus. Il est possible de configurer d'autres vitesses de communication. Il est important, cependant, que les vitesses des deux modules ou des modules et de l'ordinateur soient les mêmes. Dans le cas de l'utilisation avec l'application développée utilisant les modèles de Teachable Machine,
4. Entrer la commande : « AT+ADDR? »

Cela permet d'obtenir l'adresse (*MAC address*) de la carte. Noter la valeur obtenue, c'est-à-dire les chiffres suivants le « : » dans « OK+ADDR: 0035FF209316 ». Ces chiffres seront utiles pour connecter deux modules HM-10 ensemble.

5. Pour l'utilisation avec l'application, on doit s'assurer que le nom du module commence avec « HMSoft ». On peut toutefois ajouter des chiffres à la fin ou des lettres pour différencier les différents modules.
    a. Entrer la commande : AT+NAME?
    Si la réponse est « OK+NAME:HMSoft », on peut utiliser le module tel quel.
    b. Pour modifier légèrement le nom, utiliser la commande « AT+NAMEHMSoft01 ». Le nom du module sera changé pour « HMSoft01 ». Il sera plus facile de le repérer dans la liste de modules si plusieurs sont utilisés en même temps.

6. On peut aussi vérifier la version du *firmware* qui est installé dans le module.
    a. Entrer la commande « AT+VERR? ». La réponse devrait être quelque chose comme « HMSoft V610 ». Ici, la version 6.10 du *firmware* est utilisée.

Pour ce qui est de l'utilisation avec l'application, le module est maintenant prêt.

Pour configurer un 2e module si on souhaite en utiliser une paire, ne pas fermer le Moniteur Série. Débrancher la carte Arduino de l'ordinateur. Enlever le module HM-10 et le remplacer par l'autre si l'on souhaite en configurer une paire.

7. S'assurer que le bon port est sélectionné pour la carte dans l'onglet Outils.
8. Télécharger le programme dans la carte Arduino.
9. Refaire les étapes 5 à 8 de la Configuration du module par Commandes AT et continuer avec la *Configuration de la carte en mode Master ou Central :*

Configuration de la carte en mode *Master ou Central*

1. Entrer la commande : « AT »
    c. Si la réponse est : OK passer à l'étape suivante.
    d. S'il n'y a pas de réponse, vérifier les branchements.

2. Entrer la commande : « AT+ROLE? »
    c. Si la réponse est : « OK+Set:1 », passer à l'étape suivante. Cela correspond à une connexion à n'importe quelle adresse.

d. Si la réponse est : « 0 », entrer la commande « AT+ROLE1 ». Si la réponse est « OK+Set:1 », tout va bien. Sinon, réessayer en vérifiant la frappe.
e. Cela permet de configurer la carte comme *Master ou Central*

3. Entrer la commande : « AT+BAUD? »
   c. Si la réponse est : « OK+Get:0», passer à l'étape suivante.
   d. Si la réponse est autre chose, entrer la commande « AT+BAUD0». Si la réponse « OK+Set:0 » apparaît, tout va bien. Sinon, réessayer en vérifiant la frappe.

4. Pour l'utilisation avec l'application, on doit s'assurer que le nom du module commence avec « HMSoft ». On peut toutefois ajouter des chiffres à la fin ou des lettres pour différencier les différents modules.
   a. Entrer la commande : AT+NAME?
      Si la réponse est « OK+NAME:HMSoft », on peut utiliser le module tel quel.
   b. Pour modifier légèrement le nom, utiliser la commande « AT+NAMEHMSoft02 ». Le nom du module sera changé pour « HMSoft02 ». Il sera plus facile de le repérer dans la liste de modules si plusieurs sont utilisés en même temps.

5. On peut aussi vérifier la version du *firmware* qui est installé dans le module.
   a. Entrer la commande « AT+VERR? ». La réponse devrait être quelque chose comme « HMSoft V610 ». Ici, la version 6.10 du *firmware* est utilisée.

6. Entrer la commande AT+CON[adresse]. Remplacer, cependant, la partie [adresse] avec la MAC adress de l'autre module préparé plus haut. Par exemple : AT+CON0035FF209316. La réponse devrait être : AT+CONNE.

Le site http://www.martyncurrey.com/hm-10-bluetooth-4ble-modules/ semble proposer des façons de connecter deux modules. Il reste à vérifier le tout.

7. 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(9600);
      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(9600);
}

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

9. Fonctionnement des programmes

Pour chaque module HM-10, à partir du code proposé, il faut une carte Arduino programmée et un circuit avec la platine d'essai. On peut même alimenter les deux modules séparément avec les piles ou les fils d'alimentation murale.

La communication peut se faire si la DEL rouge qui est sur les deux modules ne clignote plus mais reste allumée. Cela signifie que les deux modules sont connectés ensemble.

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(9600);
```

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 brancher chaque carte dans l'ordinateur et de télécharger le programme destiné à chacune, avec le module HM-10, bien entendu.

2. 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.

3. 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.

4. 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.

À noter : Les modules Bluetooth HM-10 utilisent aussi la norme GATT de communication Bluetooth. Cela ne sera pas utilisé ici, mais cette information pourrait être utilise si l'on souhaite utiliser ces modules avec autre chose.

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.
- Tout autre chose que vous voudriez essayer!