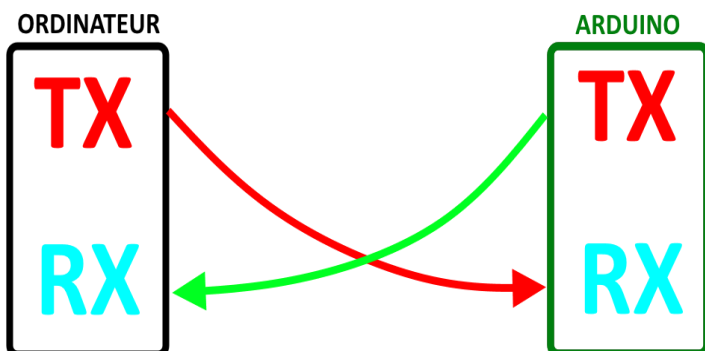


Nous sommes à l'aire des objets connectés et nous allons voir qu'il n'est pas si compliqué de réaliser ses propres prototypes qu'il s'agisse de robotique, de domotique ou tout simplement de systèmes embarqués.

Nous utiliserons des modules HC-06, présentés ci-contre où seuls les quatre pins soudés nous intéresseront.

La communication entre l'ordinateur et l'Arduino se fait en USB qui signifie Universal Serial Bus autrement dit en français Bus Série Universel. Le fonctionnement est simple si je veux parler à quelqu'un je place ma bouche près de l'oreille de mon interlocuteur, à l'inverse quand ce dernier souhaite me répondre, c'est lui qui placera sa bouche à mon oreille, ce n'est pas son oreille qui répond à bouche, on dit qu'il y a deux canaux de discussion.

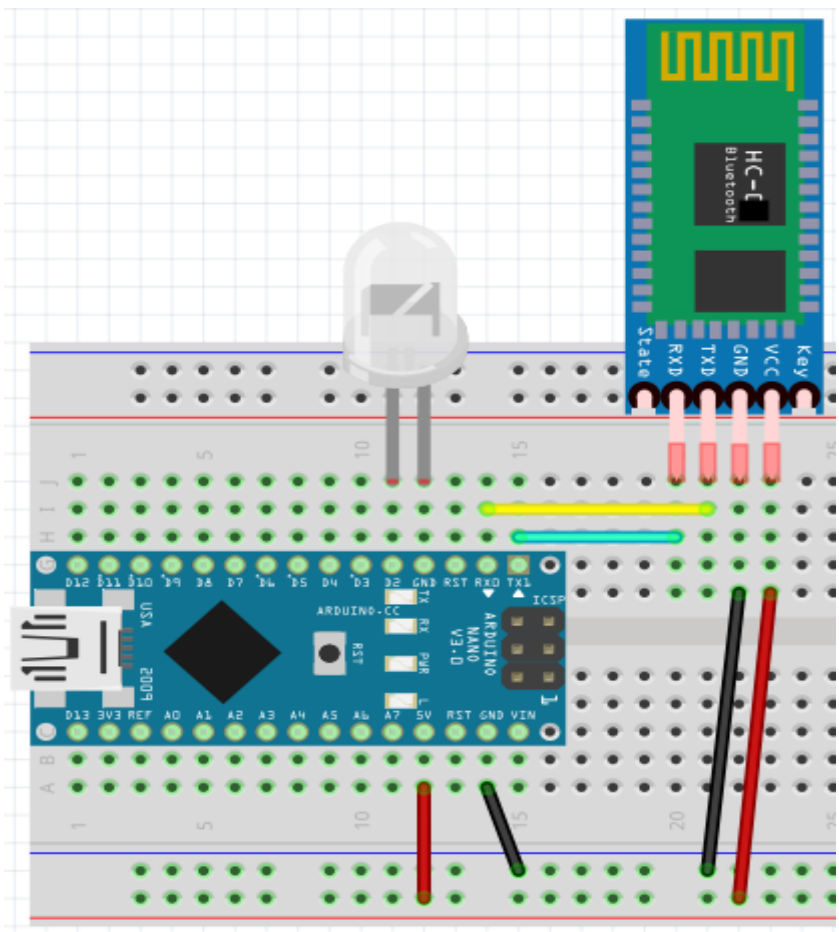


La discussion en série est basée sur ce schéma, l'ordinateur parle sur un fil électrique qui part de sa sortie TX (avec T comme transmission) vers l'entrée de l'Arduino RX (avec R comme réception). Si l'Arduino doit lui répondre, il utilisera alors un autre fil qui part de son RX vers le TX de l'ordinateur.

Il existe d'autres modes de communication, des modes qui n'utilisent qu'une seule liaison comme la CB ou les Talkies Walkies, d'autres qui en utilisent plus comme la téléphonie mobile...

Ne perdons pas de temps et passons au montage, nous utiliserons une Led que nous ferons attention à bien câbler (la plus petite « patte » sur la pin GND (masse) de l'Arduino). Ensuite nous connecterons le module HC-06 en vérifiant bien que sa borne TXD est reliée à la RX de l'Arduino et que sa borne RXD est bien reliée à la TX de l'Arduino.

Une fois que cela est bien réalisé, vous pouvez relier les alimentations (VCC) et les masses (GND) manquantes.



### Réalisation du sketch :

Nous allons voir qu'aucune bibliothèque n'est nécessaire pour communiquer en Bluetooth puisqu'il ne s'agit que d'un échange entre TX et RX du module et de l'Arduino :

```

1  /*  Module bluetooth HC-06  */
2
3  //Assignment des pins de l'Arduino
4  #define Led 2
5
6  //Déclaration de variable
7  char instruction; // le message reçu par bluetooth;
8
9  void setup()
10 {
11     Serial.begin(9600);
12     //On initialise la LED
13     pinMode(Led, OUTPUT);
14     //On la fait clignotter pour vérifier son
15     //bon fonctionnement
16     digitalWrite(Led, HIGH);
17     delay(1000);
18     digitalWrite(Led, LOW);
19     delay(1000);
20     digitalWrite(Led, HIGH);
21     delay(1000);
22     digitalWrite(Led, LOW);
23 }

```

```

26 void loop()
27 {
28     char message;
29     if (Serial.available()) // réception d'un message
30     {
31         message = Serial.read(); // lecture du message reçu
32     }
33     if (message != instruction) { // alors c'est un nouveau message
34
35         instruction = message;
36
37         if (instruction == 'a')
38         {
39             digitalWrite(Led, LOW);
40         }
41         else if (instruction == 'b')
42         {
43             digitalWrite(Led, HIGH);
44         }
45     }
46     delay(100);
47 }

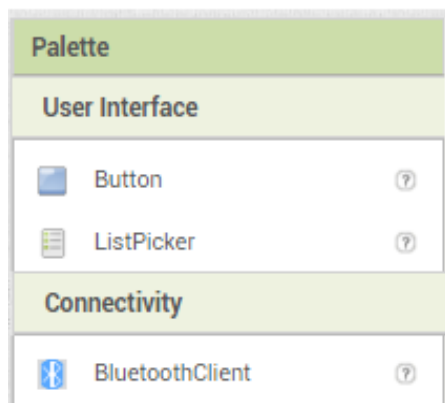
```

Si on envoie la lettre **a** au module Bluetooth, ce dernier la transmettra au port série de l'Arduino qui éteindra la LED. En revanche, si on envoie la lettre **b**, cela permettra d'allumer la LED. Mais encore faudra-t-il pouvoir envoyer ce message au module Bluetooth, pour cela il existe pas mal d'applications sur le play store tels que Bluetooth terminal mais nous allons de ce pas réaliser une application Android personnelle qui sera alors personnalisable.

**Rq :** Lorsque l'on voudra envoyer un sketch dans l'Arduino, il faudra veiller à couper (débrancher) l'alimentation du module Bluetooth sans quoi les bornes RX et TX de l'USB vont discuter avec celles de l'Arduino mais également celles du modules, et votre interface Arduino sur l'ordinateur vous affichera une erreur.

## **Réalisation de l'appli de commande.**

Rendez-vous sur App Inventor 2, crée un nouveau projet puis insérez les composants indiqués ci-dessous :



On insère en premier, deux boutons, un pour allumer et un pour éteindre. Ensuite on place un ListPicker, qui ressemble à un bouton mais ouvre une fenêtre qui permettra de choisir parmi les périphériques Bluetooth enregistrés dans notre téléphone ou notre tablette.

Et bien entendu le module « invisible » BluetoothClient qui communiquera avec notre module HC-06.

Dans un premier bloc nous allons indiquer au ListPicker qu'il doit afficher les éléments listés dans le module Bluetooth de notre appareil Android. L'application risque de vous demander un code pin se connecter, tapez alors 1234.

```

when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

```

Ensuite quand le choix du périphérique sera fait, il faut dire à la puce Bluetooth de notre téléphone ou tablette de se connecter à l'appareil sélectionné. On pourra alors changer la couleur du bouton ListPicker pour indiquer à l'utilisateur que la connexion est réussie :

```

when ListPicker1 .AfterPicking
do set ListPicker1 . Selection to call BluetoothClient1 .Connect
address ListPicker1 . Selection
set ListPicker1 . TextColor to

```

Il est temps à présent de paramétrer nos boutons pour envoyer le texte à notre Arduino pour allumer la Led, mais avant cela on vérifie que la connexion Bluetooth est toujours active.

```

when Button1 .Click
do if BluetoothClient1 . IsConnected
then call BluetoothClient1 .SendText
text " b "

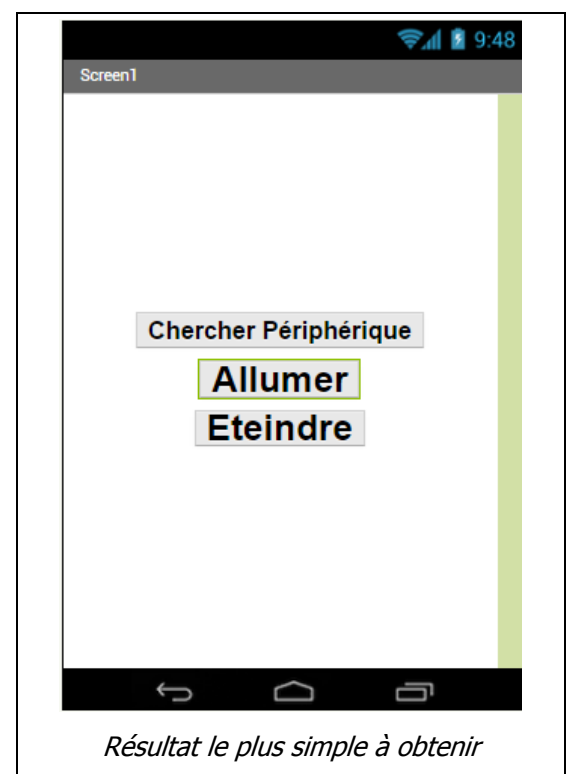
```

On fait de même pour le second bouton qui permettra d'éteindre :

```

when Button2 .Click
do if BluetoothClient1 . IsConnected
then call BluetoothClient1 .SendText
text " a "

```



Bien entendu vous pouvez personnaliser l'application avec des images dans les boutons pour rendre tout cela plus convivial.

**DEFIS** Rajouter au montage un thermomètre 18B20 et un capteur de luminosité. Modifier alors le sketch et l'application pour recevoir les données de ces capteurs et les afficher sur votre téléphone.

Indice :

```

when Clock1 .Timer
do if Client_Bluetooth1 . IsConnected
then set Message . Text to call Client_Bluetooth1 .ReceiveText
numberOfBytes call Client_Bluetooth1 .BytesAvailableToReceive

```