

# TRAVAUX DIRIGÉS IOT

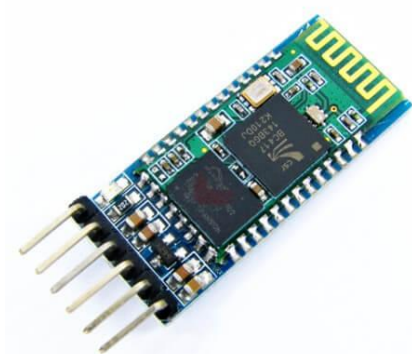
## TD1 :

La carte Arduino Uno peut communiquer avec d'autres appareils (Smartphone, ordinateur ou un autre microcontrôleur) par liaison Bluetooth en utilisant un module HC-05 (maître/esclave). Cela permet de connecter la carte Arduino à un autre système afin d'envoyer et de recevoir des données. La communication par Bluetooth permet, par exemple, de piloter un robot via une application Smartphone, récupérer les mesures d'un capteur sur votre ordinateur ou encore permettre à deux robots de communiquer entre eux.

### Matériels

- Ordinateur
- Arduino UNO
- câble USB pour connecter l'Arduino Uno
- Module Bluetooth HC-05
- Câbles Dupont M/F

### Présentation du module Bluetooth HC-05



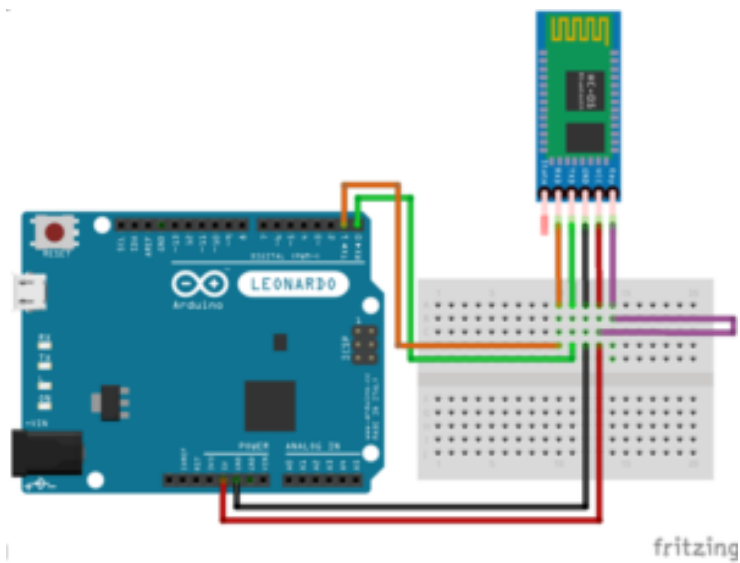
Le module Bluetooth HC-05 présente 6 broches pour permettre d'établir la connexion.

- VCC broche d'alimentation connectée à la broche 5V de l'Arduino.
- GND masse connectée à la broche GND de l'Arduino
- RX broche de réception connectée à la broche de transmission (TX) de l'Arduino
- TX broche de transmission connectée à la broche de réception (RX) de l'Arduino
- State retourne 1 lorsque le module est connecté
- Key ou EN doit être alimentée pour entrer dans le mode de configuration et ne doit pas être connectée pour être en mode communication.

**Note :**

La particularité du module Bluetooth HC-05 est qu'il peut être utilisé en mode esclave (dans ce cas il est équivalent à un module HC-06 et s'utilise de la même manière) ou en mode maître, ce qui veut dire qu'il peut de manière autonome se connecter à un autre module Bluetooth (HC-06, HC-05 en mode esclave ou autre) sans action extérieure.

## Schéma de câblage pour la configuration



Nous utilisons les broches Rx et Tx pour la communication série avec Arduino UNO.

## Code de communication

```
/*  
 * Simple application pour allumer une led  
 * Depuis son smartphone en utilisant un module Bluetooth HC-05  
 */  
  
#define PIN 2  
  
void setup()  
{  
  pinMode(PIN, OUTPUT);  
  Serial.begin(9600);  
  Serial.println("Initializing com");  
  delay(1000);  
}
```

```
void loop()
{
  // Lecture du Port Rx pour savoir si des données sont envoyées
  if (Serial.available())
  {
    // On lit le buffer jusqu'a rencontrer le saut de retour de chaine
    String sig = Serial.readStringUntil("\n");

    // Si on lit 1 on allume la led
    if (sig == "1") { digitalWrite(PIN, HIGH);}

    // Si on lit 0 on eteint la led
    else if (sig == "0") {digitalWrite(PIN, LOW);}
  }
}
```

---

## Code Application Android et Iphone

App Inventor est un outil en ligne, ce qui signifie que vous pouvez créer des applications pour téléphones ou tablettes directement dans votre navigateur Web. Nous allons créer une simple application Android (si vous êtes sur iphone suivez les instructions [ici](#) pour mettre en place votre environnement de travail)

Voici l'application que nous allons développer



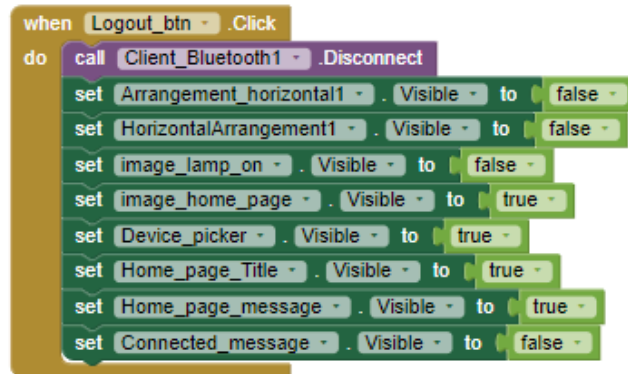
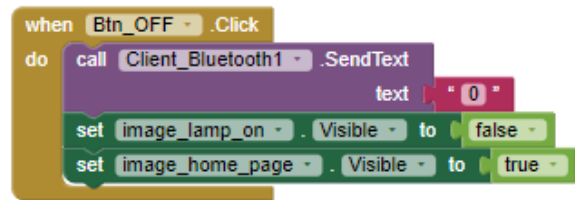
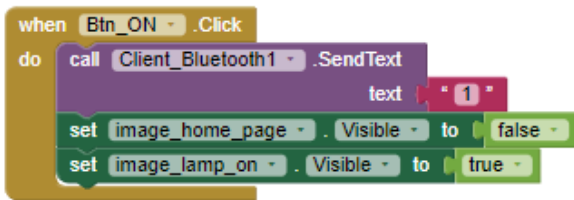
Une fois le design de notre application est fait, il ne reste que le développement des fonctionnalités.

```

when Screen1.Initialize
do
  set ActivityStarter1.Action to " android.bluetooth.adapter.action.REQUEST_ENABLE "
  call ActivityStarter1.StartActivity

when Device_picker.BeforePicking
do
  set Device_picker.Elements to Client_Bluetooth1.AddressesAndNames

when Device_picker.AfterPicking
do
  if
    call Client_Bluetooth1.Connect
    address Device_picker.Selection
  then
    set Device_picker.Elements to Client_Bluetooth1.AddressesAndNames
    set Home_page_Title.Visible to false
    set Home_page_message.Visible to false
    set Connected_message.Visible to true
    set Device_picker.Visible to false
    set Arrangement_horizontal1.Visible to true
    set HorizontalArrangement1.Visible to true
  
```



Le fichier Apk (pour android) est disponible [ici](#)