

App Inventor ressource Numéro 12



Sources

<http://ai2.appinventor.mit.edu/>

<https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>

<http://appinventor.mit.edu/explore/ai2/setup-emulator.html>

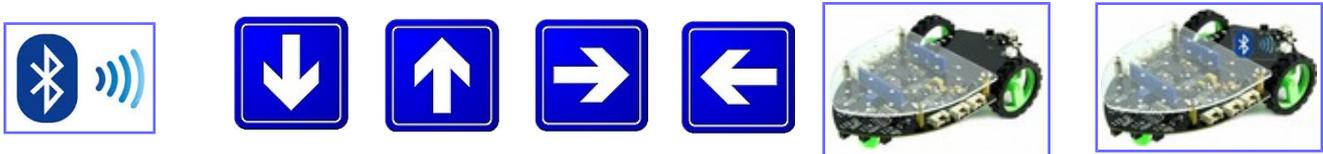
Tutoriels vidéos

<https://www.youtube.com/watch?v=0H9Y2OQohyE>

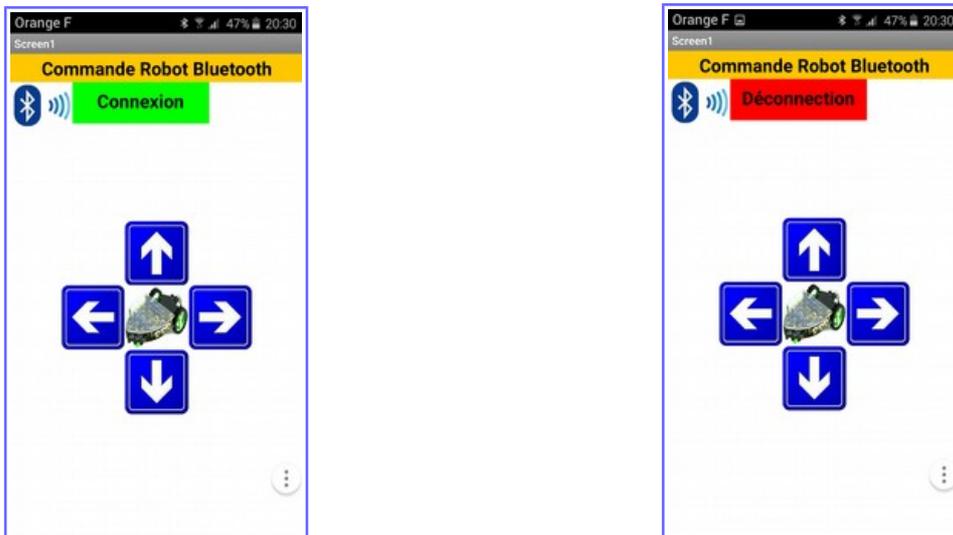
<https://www.youtube.com/watch?v=I2DONpimg1M>

Nouvelle application, commande du robot Shield bot v1.1 via Bluetooth

Images que nous allons utiliser pour cette application



Voici les images de l'application terminée (un exemple de présentation ;)



Afin de ne pas avoir à reprendre toutes les saisies, vous pouvez **reprendre la ressource numéro 8** (De nombreuses informations y sont déjà décrites !) ...

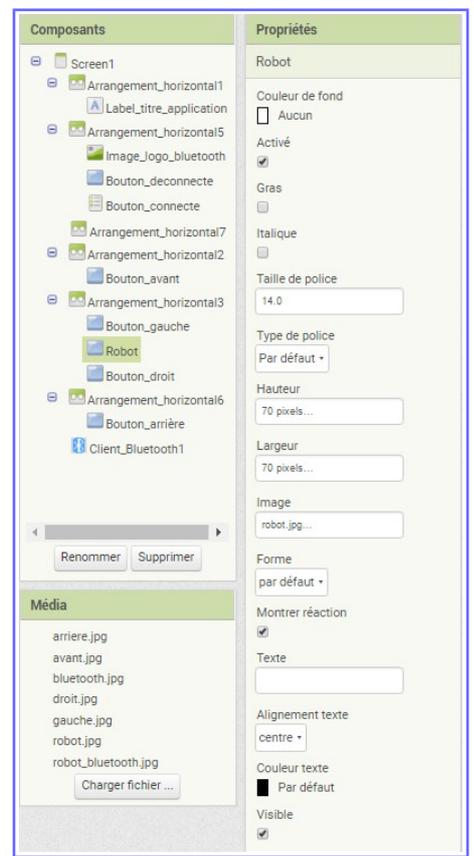
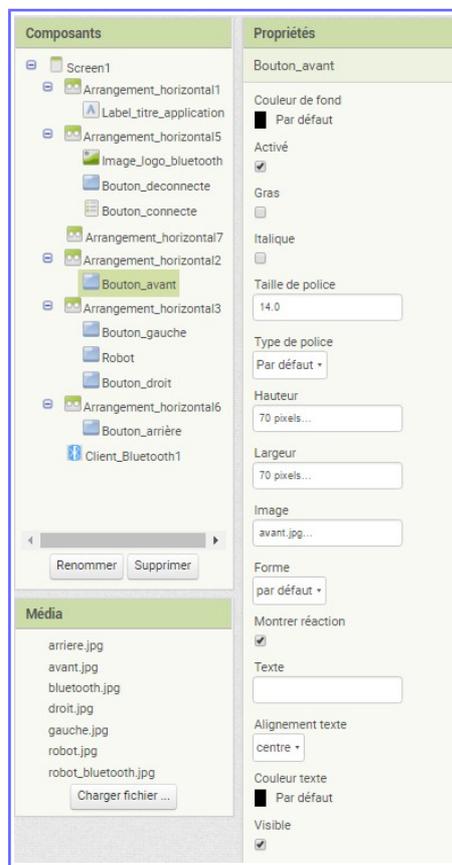
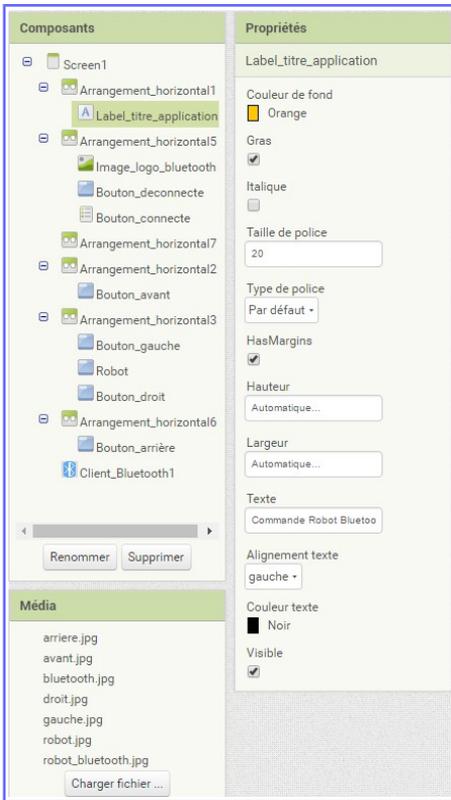
Vous pouvez faire un « enregistrez sous » et renommer l'application en « robot_version_bluetooth_version_1 »

Il ne vous « reste » plus qu'à compléter l'application...

Nous allons décrire toutes les propriétés des composants...certains ne seront pas repris car ils sont déjà décrits dans la ressource numéro 8 (Arrangement horizontal 5). L'arrangement horizontal 7 est là uniquement pour descendre les boutons au milieu du smartphone...Les 4 boutons ont la même structure, nous décrivons seulement le « bouton_avant »



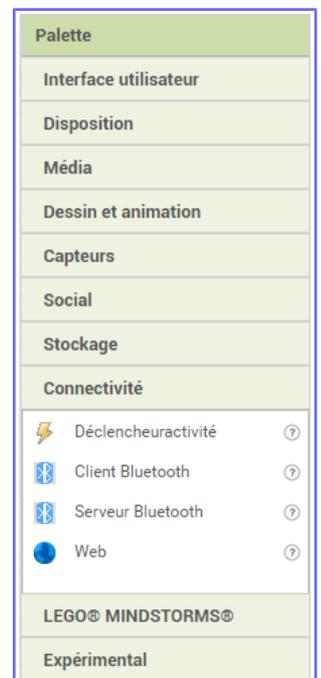
Le bouton « Robot » est ici pour espacer les boutons droite/gauche et pour rappeler l'icône de l'application



Le «

Bouton connecte » va nous permettre de sélectionner la carte Bluetooth à utiliser (par exemple vous pouvez comme moi avoir dans votre liste les connexions Bluetooth de votre voiture...)

Dans la palette « interface utilisateur » il faut choisir « sélectionneur de liste »



Nous allons maintenant rajouter un « service Bluetooth » à cette application...

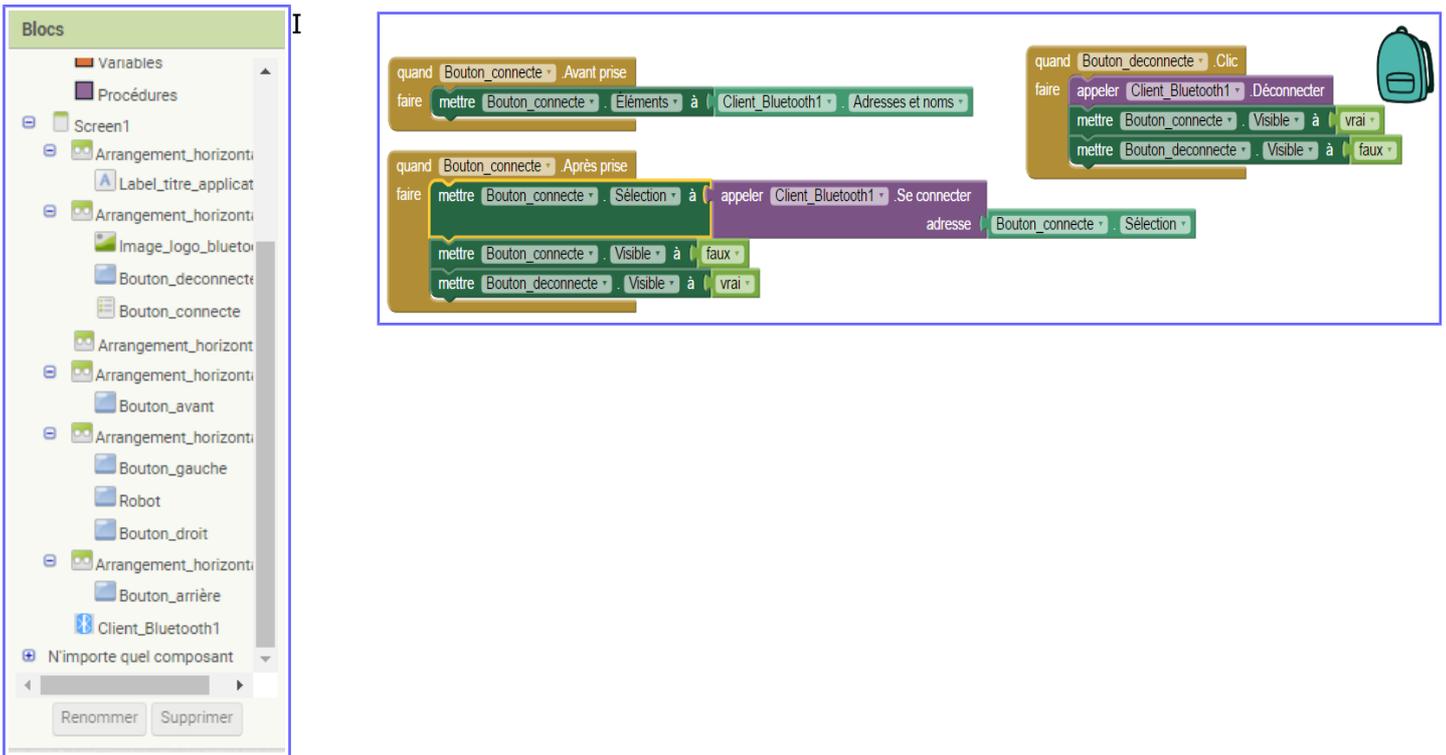
Ce service se place en dessous de votre application (physiquement à l'écran)

Faire un glissé/déposé et le service se place tout seul en bas de l'application.



Pour la partie programmation par Blocs, il va falloir prendre en compte de nombreuses modifications...pas forcément très simples à comprendre...

Voici l'ensemble des blocs que vous avez utilisé...cette partie là ne change pas !!



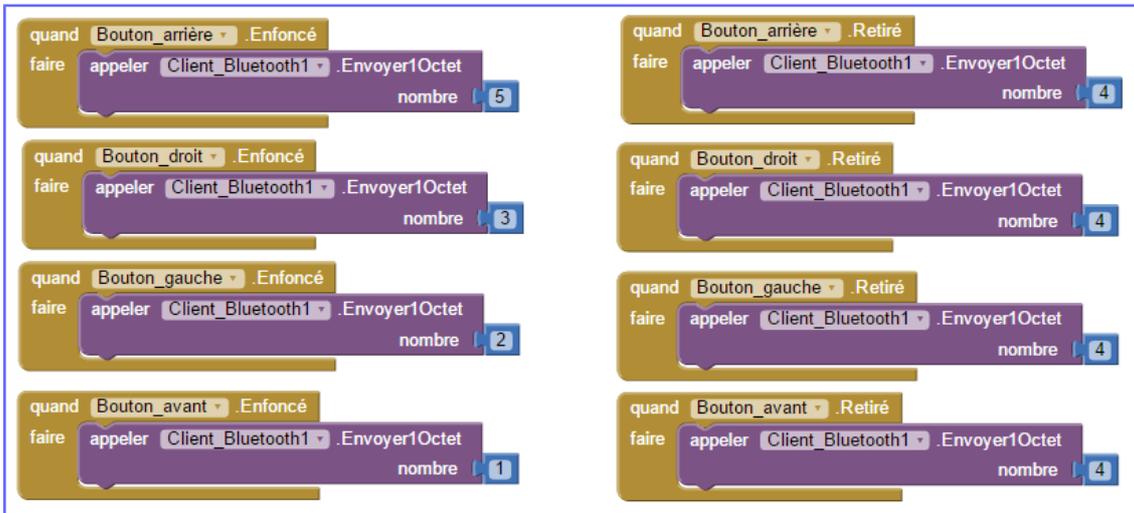
Le « Bouton_connecte », le sélectionneur de liste, va vous permettre de choisir la connexion Bluetooth (ici celle de la carte HC-06)

Une fois le choix de la connexion Bluetooth fait, la liaison va s'établir entre votre terminal (tablette ou smartphone) et le « Bouton_connecte » va disparaître et laisser la place au « Bouton_deconnecte »

Et pour le dernier ensemble, quand en clique sur le « Bouton_deconnecte », la liaison Bluetooth est désactivée, le « Bouton_deconnecte » disparaît et le « Bouton_connecte » revient à l'écran....



Les boutons de déplacement vont maintenant pouvoir être utilisés



Quand on maintient **le bouton arrière enfoncé**, on envoie via le Bluetooth le **nombre 5**

Quand on maintient **le bouton droit enfoncé**, on envoie via le Bluetooth le **nombre 3**

Quand on maintient **le bouton gauche enfoncé**, on envoie via le Bluetooth le **nombre 2**

Quand on maintient **le bouton avant enfoncé**, on envoie via le Bluetooth le **nombre 1**

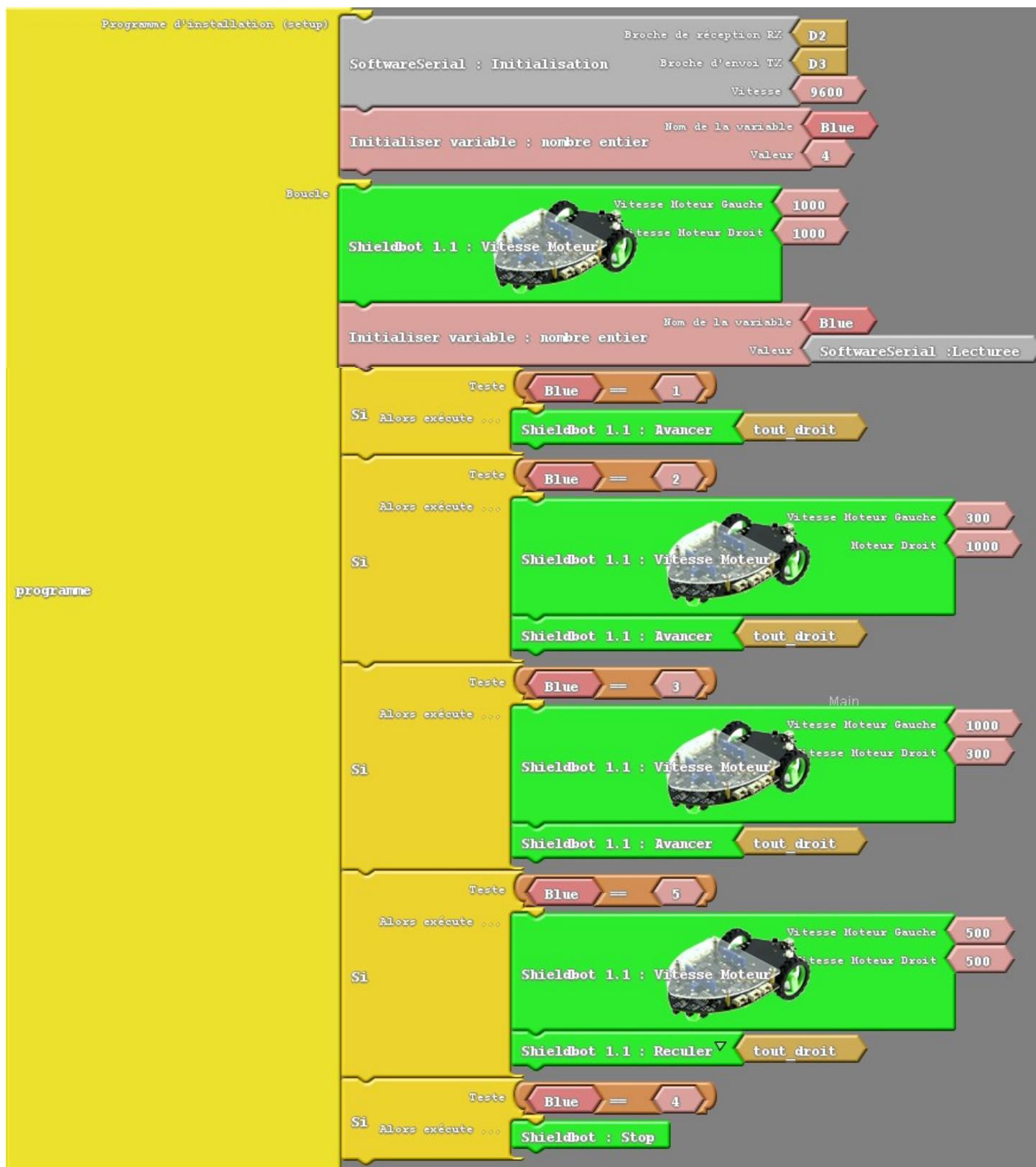
Pour les 4 boutons, lors du relâchement, on envoie via le Bluetooth le **nombre 4** (qui provoquera l'arrêt!)

Nous allons maintenant passer à la partie « Arduino »

Matériel nécessaire pour réaliser cet exercice

Matériel	Matériel
Carte UNO + cordon USB 	Cordon Grove 
Robot Shield bot v1.1 	Une carte Bluetooth HC-06 

Voici les blocs assemblés



Code

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

SoftwareSerial mySerial(2,3);
int _ABVAR_1_Blue = 0 ;
//libraries at http://duinoedu.com/dl/lib/grove/EDU\_Shieldbot\_Grove/
Shieldbot monBot;

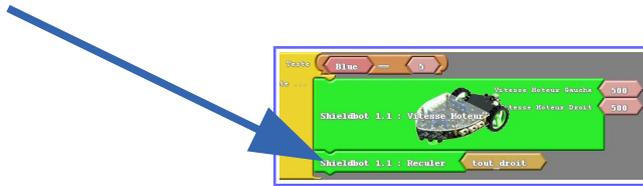
void setup()
{
  mySerial.begin(9600);

  monBot.brancher();

  _ABVAR_1_Blue = 4 ;
}

void loop()
{
  monBot.ecrireVitesse(1000,1000);
  _ABVAR_1_Blue = mySerial.read() ;
  if (( ( _ABVAR_1_Blue ) == ( 1 ) ))
  {
    monBot.avancer("tout_droit");
  }
  if (( ( _ABVAR_1_Blue ) == ( 2 ) ))
  {
    monBot.ecrireVitesse(300,1000);
    monBot.avancer("tout_droit");
  }
  if (( ( _ABVAR_1_Blue ) == ( 3 ) ))
  {
    monBot.ecrireVitesse(1000,300);
    monBot.avancer("tout_droit");
  }
  if (( ( _ABVAR_1_Blue ) == ( 5 ) ))
  {
    monBot.ecrireVitesse(500,500);
    monBot.reculer("tout_droit");
  }
  if (( ( _ABVAR_1_Blue ) == ( 4 ) ))
  {
    monBot.stopper();
  }
}
```

Pour l'avance moteur « tout droit » on choisit une vitesse de 1000/1000
 Pour tourner à droite et à gauche on choisit une vitesse 300/1000, rotation plus fluide
 Pour reculer on choisit une vitesse de 500/500 et on n'oublie pas de changer le sens du déplacement dans le bloc « shieldbot 1.1 reculer »
 En passant la souris sur cet élément, une flèche de déroulement apparaît et nous avons « avancer, reculer et pivoter »



Il faudra dans un premier temps appairer votre carte HC-06 avec votre terminal et remarque très importante pour le câblage, il faudra croiser RXD et TXD entre la carte HC-06 et la carte Arduino !!

Pour appairer votre carte Bluetooth à votre périphérique Android, le code est 1234



Ne pas oublier de mettre à jour votre logiciel Ardublock
<http://duinoedu.com/telecharger.html>

Rappel de la ressource numéro 4

Activer le débogage USB sous Android,

Étape 1 : Passer en mode administrateur

Pour commencer, rendez-vous dans les « Paramètres », puis cherchez l'onglet « À propos » ou « À propos du téléphone ». Il est généralement situé tout en bas de la liste.

Cliquez dessus. Ici, on peut voir la catégorie « Options pour les développeurs ». Il n'apparaîtra pas tant que vous ne serez pas en mode administrateur.

Cliquez ensuite sur « Informations Logicielles ». Le nom peut varier d'un modèle à un autre, mais le dossier est forcément présent. Il recense toutes les informations liées aux logiciels de votre mobile. Il vous faut trouver le « Numéro de build ». Si vous ne savez pas ce que c'est, pas de panique ça n'a aucune importance pour arriver à bout de la manœuvre.

Une fois le numéro de build visible, cliquez dessus plusieurs fois de suite. Six, sept ou plus si nécessaire. Au bout d'un moment, une notification apparaîtra à l'écran.

Elle vous signale que vous êtes désormais en mode administrateur / développeur. C'est parfait, c'est exactement ce que l'on veut. Retournez dans les paramètres généraux.

Étape 2 : activer / désactiver le débogage USB

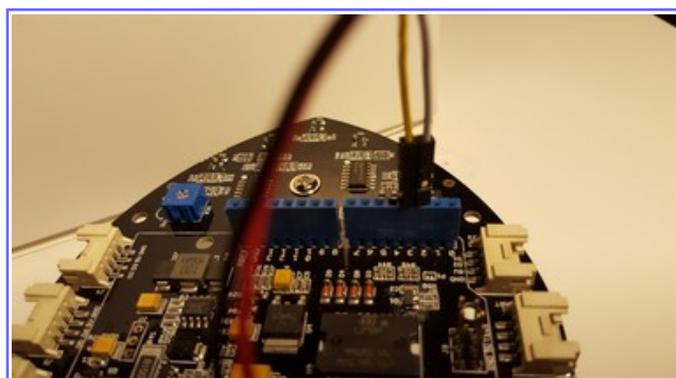
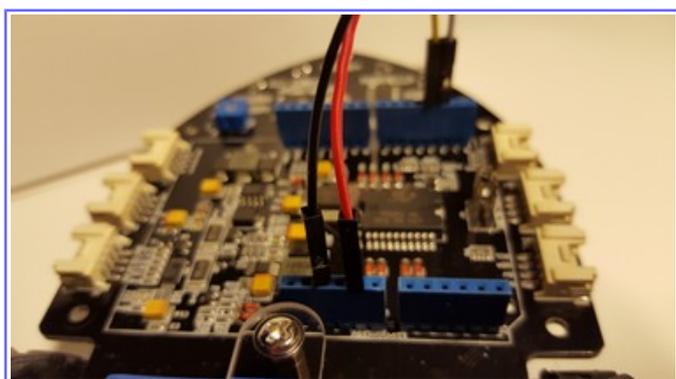
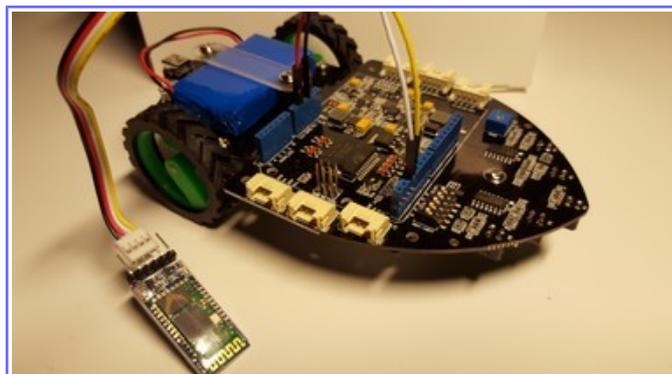
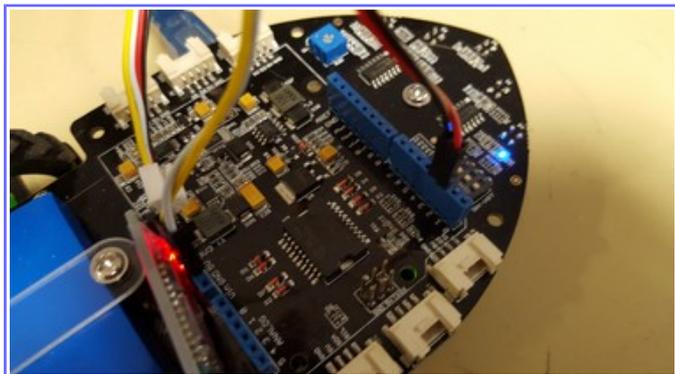
L'onglet « Options pour les développeurs » est maintenant disponible dans les « Paramètres ». Cliquez dessus.

Un message d'alerte apparaît. En effet, modifier des paramètres à tort et à travers en mode administrateur peut avoir un impact néfaste sur votre téléphone.

Cliquez sur « OK », puis tenez-vous en à cocher la case « Débogage USB ».

Voilà, votre mobile peut maintenant être connecté à votre PC et il sera considéré comme un périphérique de stockage. À l'image d'une clé USB classique ou d'un disque dur externe.

Les raccordements sur la carte Arduino



Pour information sur le câblage réalisé ici, j'ai utilisé un câble Grove éclaté sur 4 picots (TX et RX croisés! Avec la partie blocs ci-dessus)

Carte HC-06	Câble Grove	Carte Arduino
RXD	Fil jaune	Pin 3
TXD	Fil blanc	Pin 2
GND	Fil rouge	GND
VCC	Fil noir	+5V

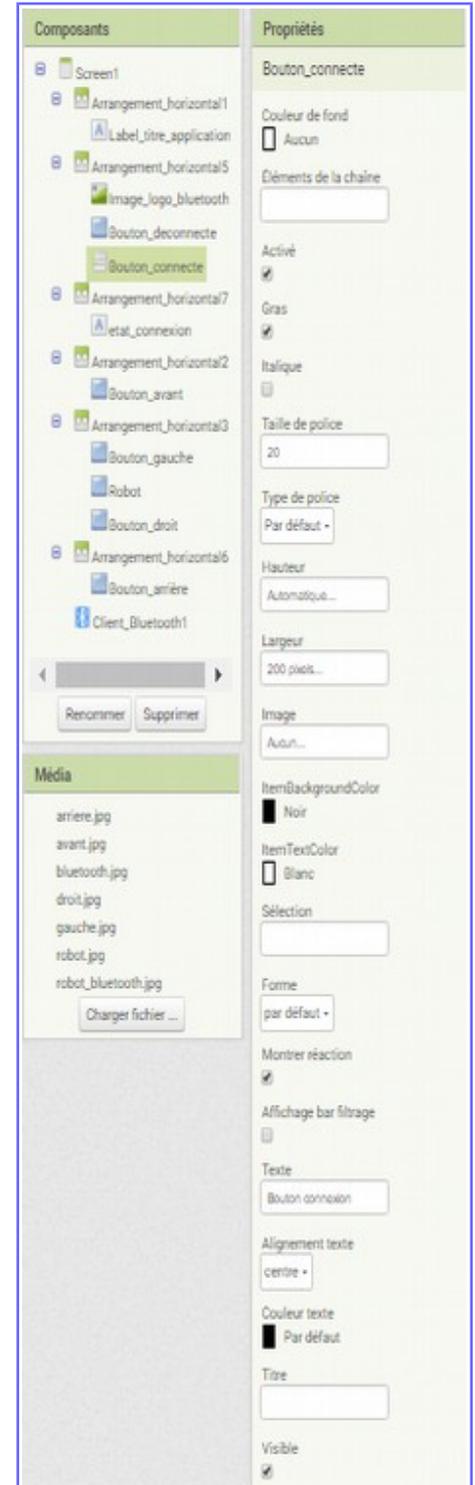
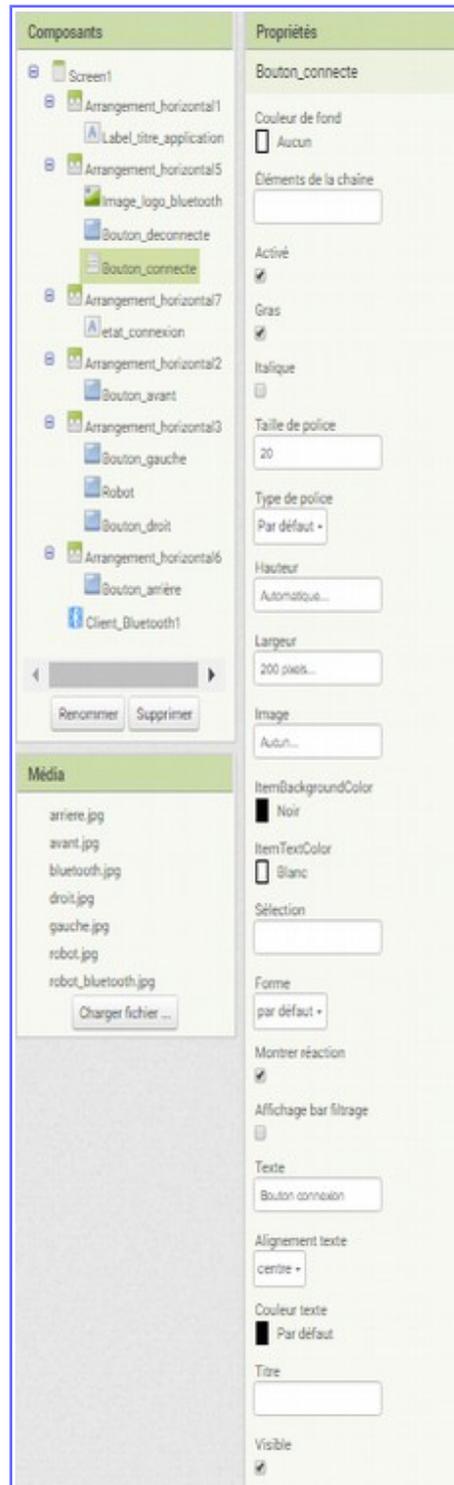
Remarque

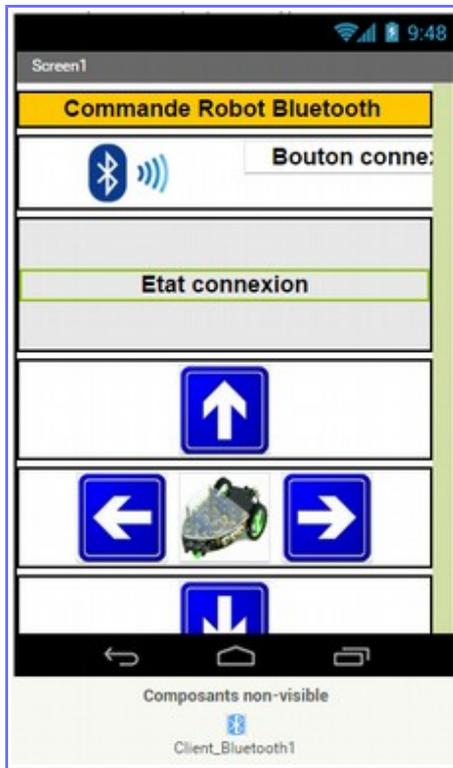
Lors de votre connexion Bluetooth, il s'avère que nous ne savons pas si les appareils sont bien connectés entre-eux.

Pour cela, nous allons mettre en place un test de connexion et nous allons l'afficher sur l'écran du smartphone (ou tablette)

Quelques modifications dans la partie design de votre application...Nous allons créer un label « état de connexion » qui sera modifié lors des connexions Bluetooth

Par la même occasion, on renomme les boutons « connecté et déconnecté » par « Bouton de connexion » et « Bouton de déconnexion »



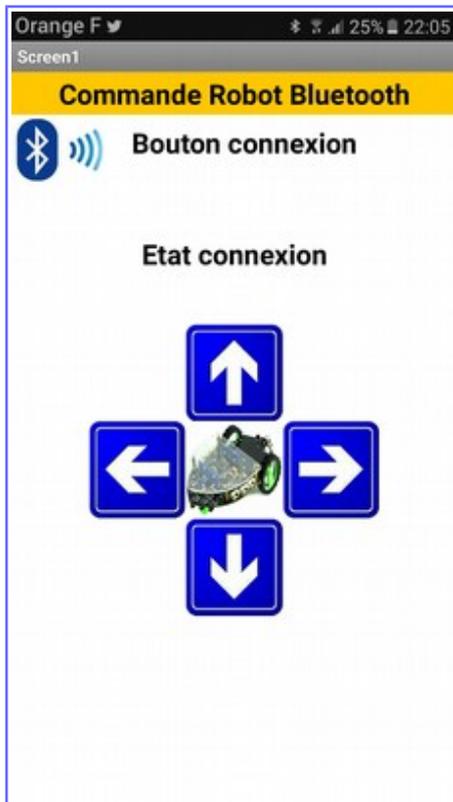


Voici une capture d'écran sur App Inventor et ci-dessous les 3 états successifs de l'application,

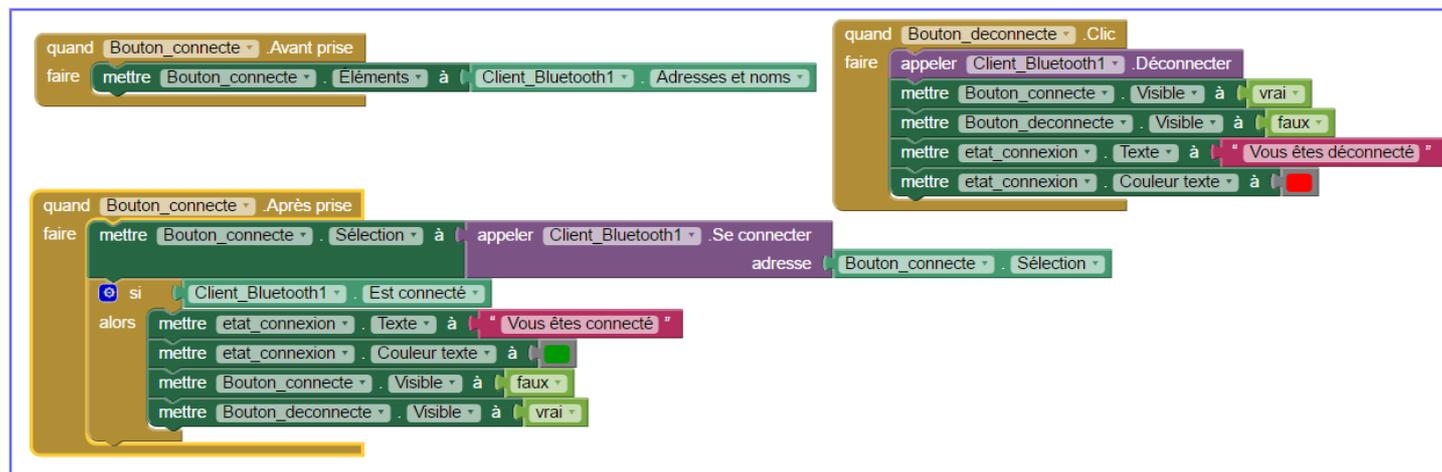
avant la connexion, « État connexion »

puis connecté en Bluetooth « Vous êtes connecté »

et pour finir, après la déconnexion « Vous êtes déconnecté »



Passons maintenant à la partie « Blocs »



On remarque sur ces captures d'écrans

Une fois la sélection réalisée, on vérifie l'état de la connexion Bluetooth (« Client_Bluetooth1 » « Est connecté »)

On effectue un test de « Si / Alors »

Si la connexion Bluetooth est effective alors je mets « Etat_connexion » à « Vous êtes connecté » et je le mets en vert

Pour la partie déconnexion, même principe, si la connexion Bluetooth n'est plus active, je mets « Etat_connexion » à « Vous êtes déconnecté » et le mets en rouge

Cette modification sera nommée **robot_version_bluetooth_version_2**