

App Inventor ressource Numéro 9



Sources

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

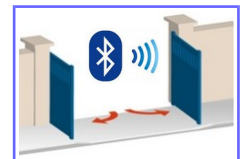
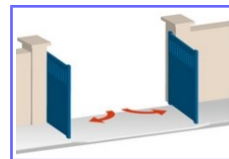
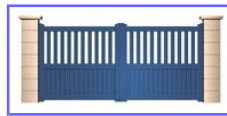
<https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>

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

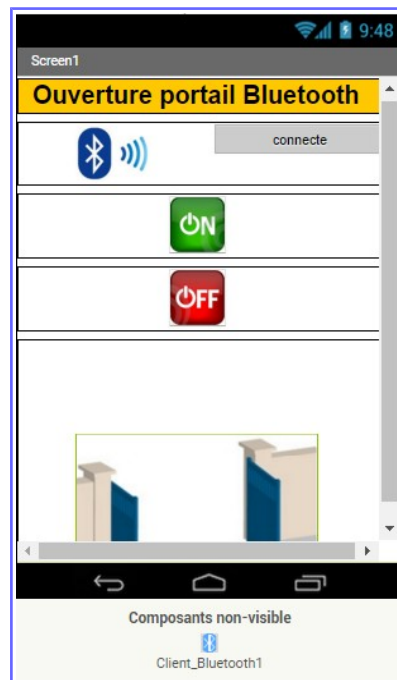
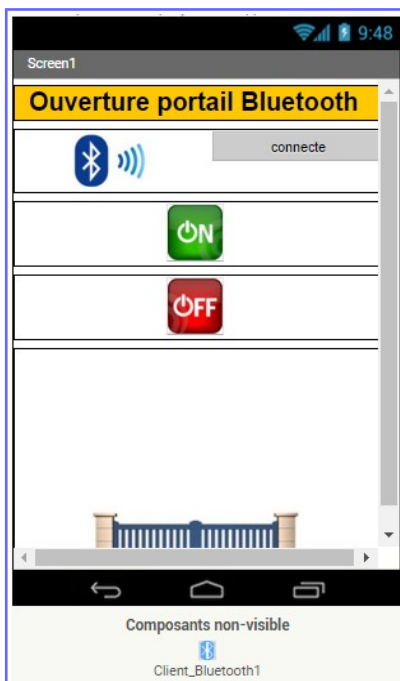
Nouvelle application, ouverture d'un portail via Bluetooth

Nous reprenons l'activité de la ressource numéro 8.....et nous allons utiliser également une carte Arduino et un moteur (pour l'ouverture du portail)....et bien sûr une carte Bluetooth HC-06

Images que nous allons utiliser pour cette application



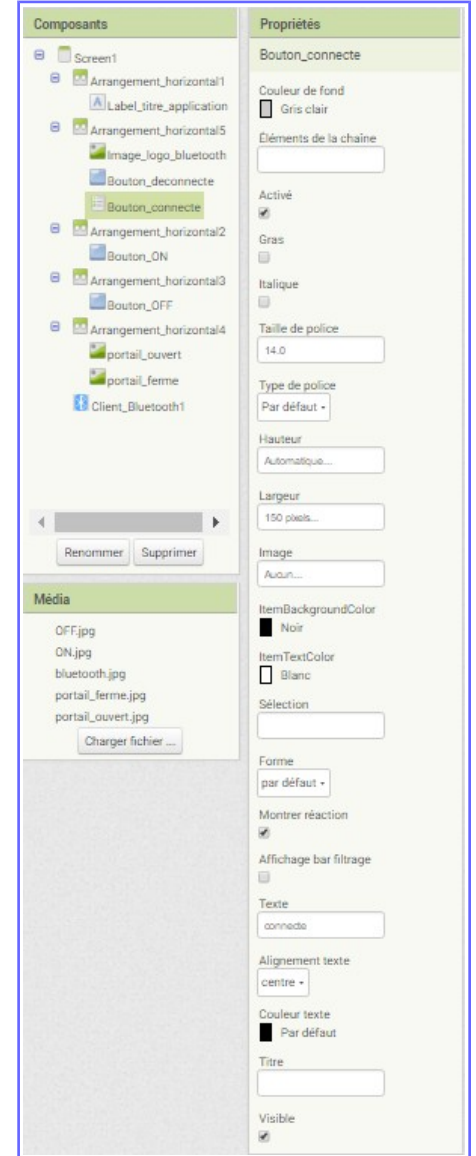
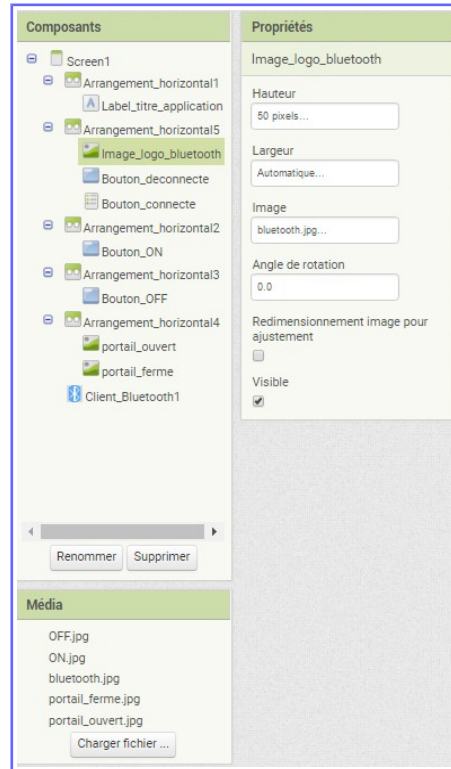
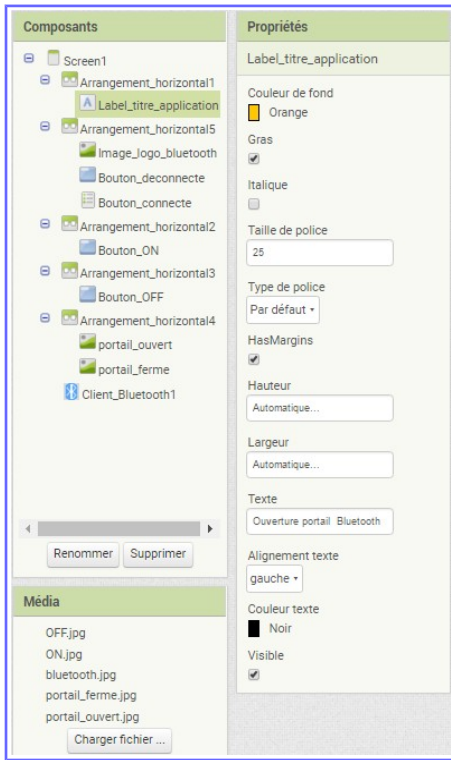
Voici les images de l'application terminée (un exemple;)



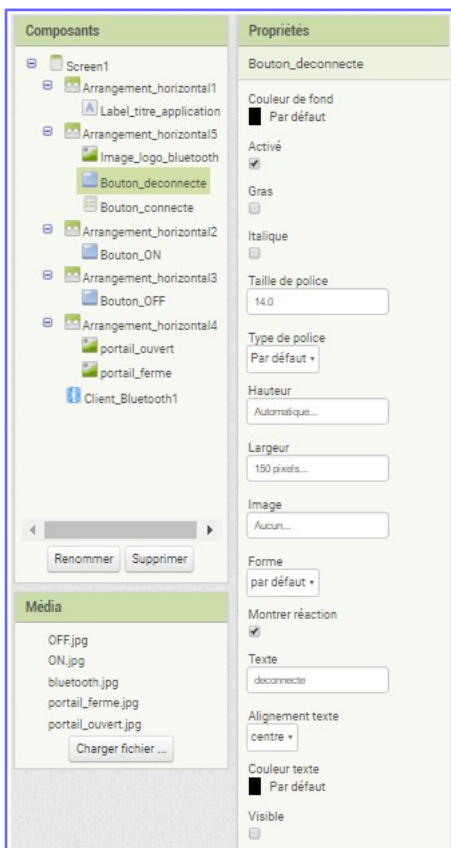
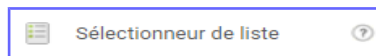
Afin de ne pas avoir à reprendre toutes les saisies de la ressource numéro 8, vous pouvez faire un « enregistrez sous » et renommer l'application « portail_version_manuelle » en « portail_version_Bluetooth »

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

Nous allons décrire toutes les propriétés des composants...



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 »



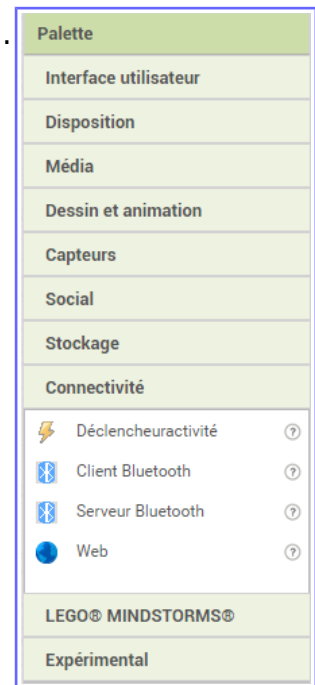
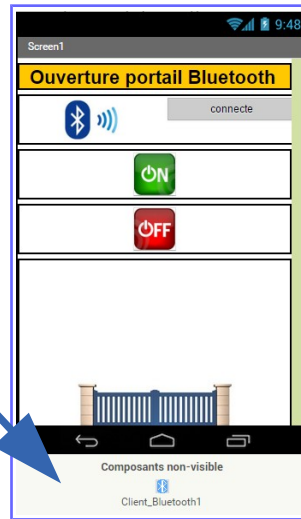
Le « Bouton déconnecte » est un bouton simple...et sera non visible au départ bien sûr ...

Nous ne verrons pas les boutons ON/OFF que vous avez déjà rencontré dans la ressource numéro 8

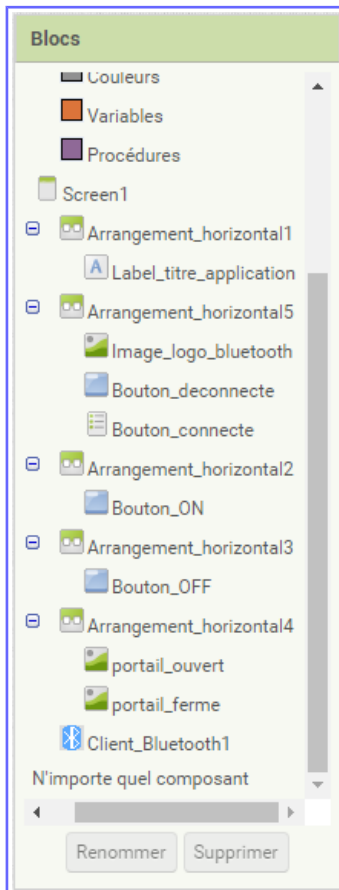
Nous allons maintenant rajouter un « service Bluetooth » à cette application...

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.

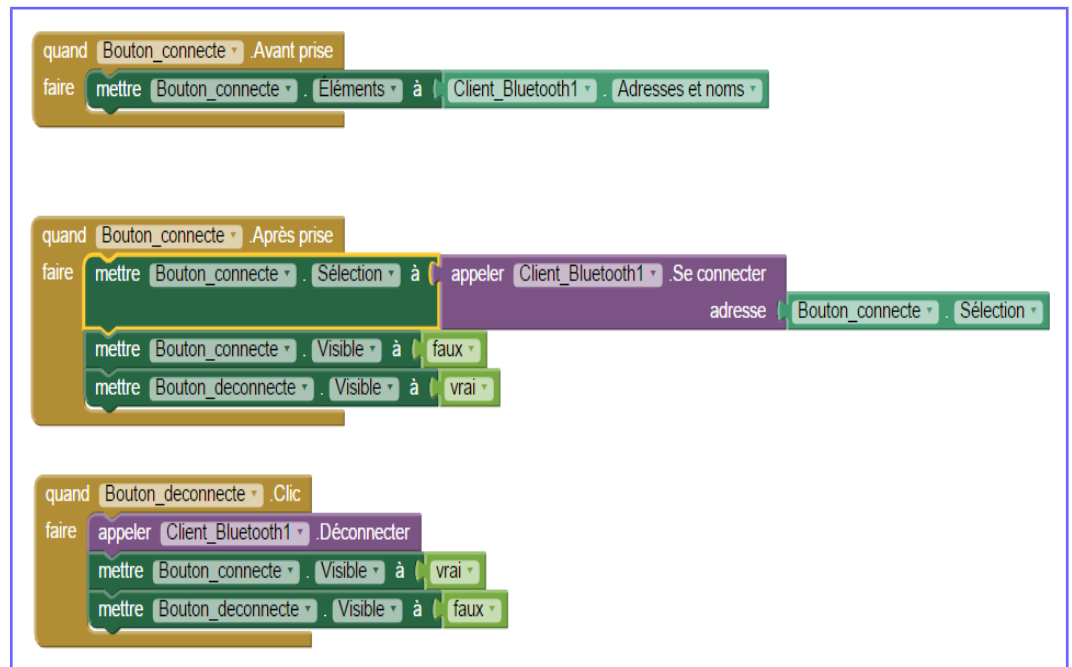


Pour la partie programmation par Blocs, il va falloir prendre en compte de nombreuses modifications...pas forcément très simples à comprendre...



Voici l'ensemble des blocs que vous avez utilisés...

Il va falloir « admettre » quelques fonctionnalités qui sont indispensables.



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_déconnecte »

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

Les boutons ON/OFF vont maintenant pouvoir être utilisés

Quand on clique sur le « Bouton_ON », on envoie un octet via le « Client_Bluetooth1 » (nombre=1) et le portail doit s'ouvrir (enfin, il doit;)

Quand on clique sur le « Bouton_OFF » on n'envoie plus d'octet via le « Client_Bluetooth1 » (nombre=0) et le portail doit se refermer...




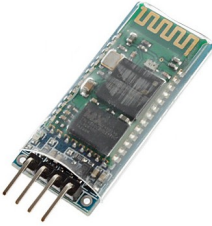
```

quand Bouton_ON .Clic
faire
mettre portail_ouvert .Visible à vrai
mettre portail_ferme .Visible à faux
appeler Client_Bluetooth1 .Envoyer1Octet
nombre 1

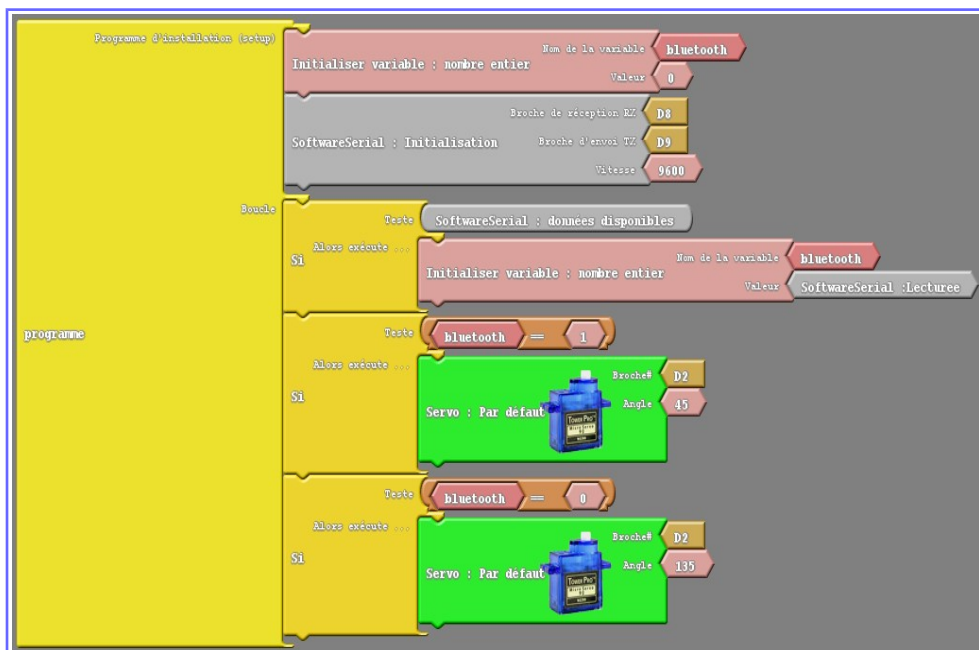
quand Bouton_OFF .Clic
faire
mettre portail_ouvert .Visible à faux
mettre portail_ferme .Visible à vrai
appeler Client_Bluetooth1 .Envoyer1Octet
nombre 0
  
```

Nous allons maintenant passer à la partie « Arduino »

Matériel nécessaire pour réaliser cet exercice

Matériel	Matériel
Carte UNO + cordon USB 	Cordon Grove 
Un servomoteur 	Une carte Bluetooth HC-06 

Voici les blocs assemblés



```

sketch_apr24a | Arduino 1.0.5-r2
Fichier Édition Croquis Outils Aide

sketch_apr24a $
#include <SoftwareSerial.h>
#include <Servo.h>

int _ABVAR_1_bluetooth = 0 ;
SoftwareSerial mySerial(8,9);
Servo servo_pin_2;

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

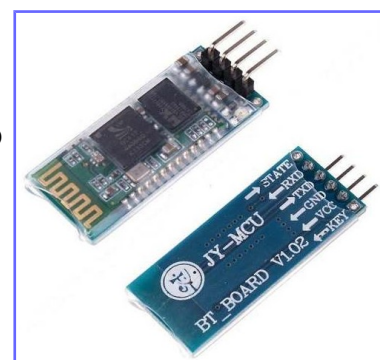
  servo_pin_2.attach(2);
  _ABVAR_1_bluetooth = 0 ;
}

void loop()
{
  if (mySerial.available())
  {
    _ABVAR_1_bluetooth = mySerial.read() ;
  }
  if ( ( ( _ABVAR_1_bluetooth ) == ( 1 ) ) )
  {
    servo_pin_2.write( 45 );
  }
  if ( ( ( _ABVAR_1_bluetooth ) == ( 0 ) ) )
  {
    servo_pin_2.write( 135 );
  }
}

```

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 !!

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



Les images de cette application, captures d'écrans du smartphone

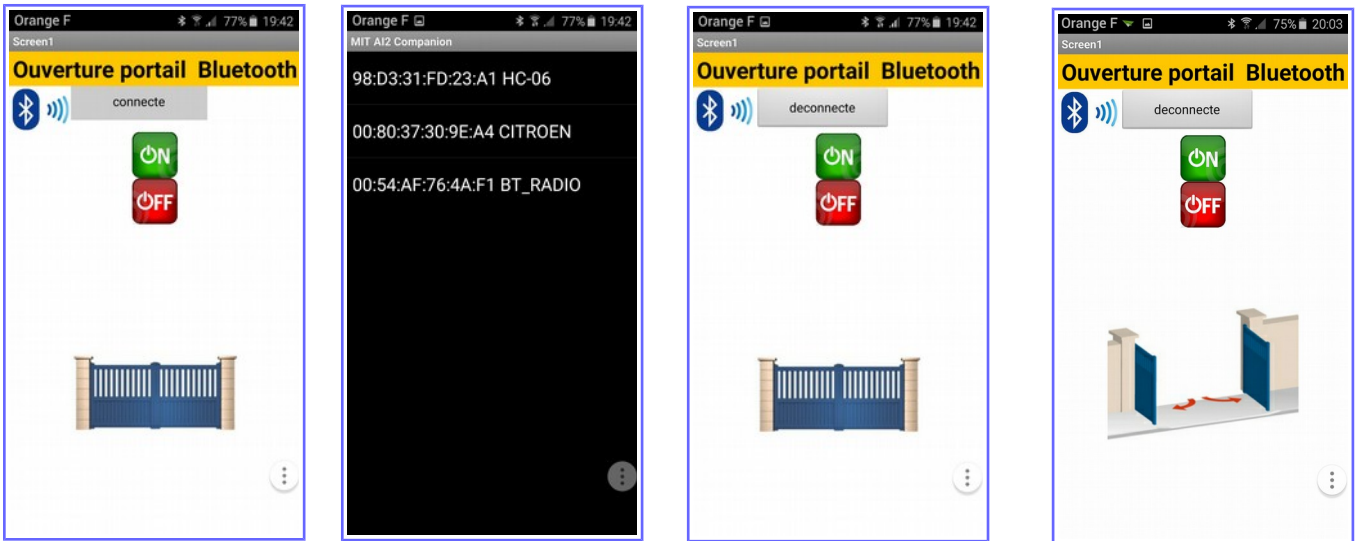
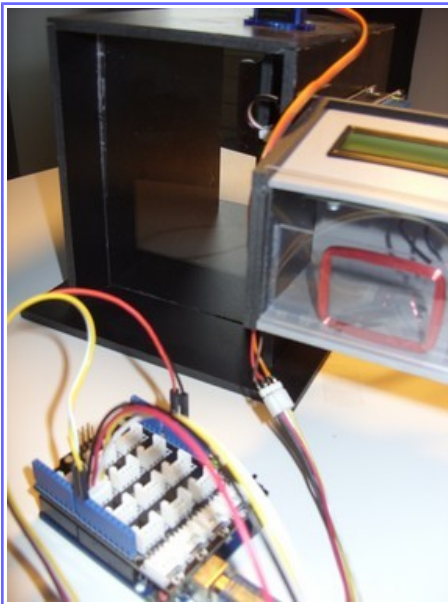


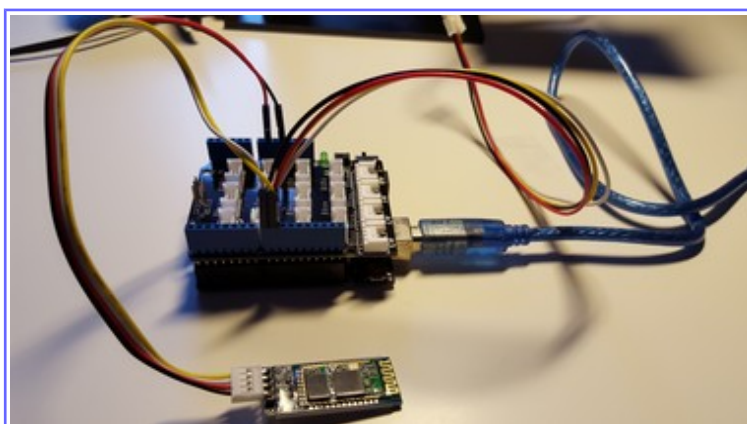
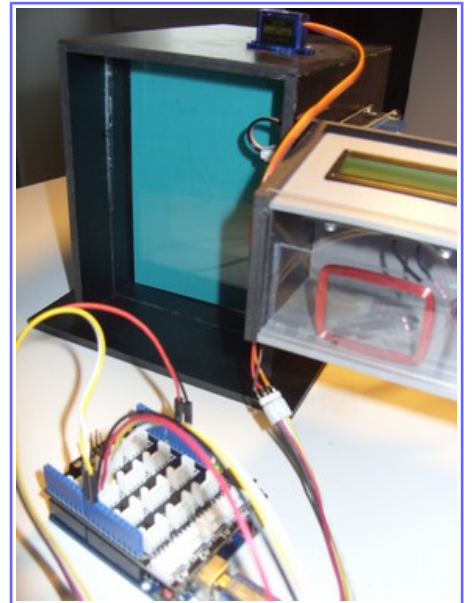
Image 1, je clique que « connecte », j'arrive sur l'image 2, là, je sélectionne le réseau Bluetooth de mon choix (HC-06) et j'arrive sur l'image 3....l'application est en fonctionnement (la LED de la carte HC-06 ne clignote plus, elle est allumée en fixe)

Image 4, je clique sur « ON » le portail s'ouvre en virtuel sur mon smartphone et en réel sur la maquette !

Les raccordements sur la carte Arduino



A gauche, porte ouverte, à droite porte fermée !



App Inventor ressource numérol

Pour information sur le câblage réalisé ici, j'ai utilisé un câble Grove éclaté sur 4 picots
(TX et RX croisés!)

Carte HC-06	Câble Grove	Carte Arduino
RXD	Fil jaune	Pin 9
TXD	Fil blanc	Pin 8
GND	Fil rouge	GND
VCC	Fil noir	+5V

Le câble du servomoteur et lui aussi éclaté sur un Grove,

Coté servomoteur	Coté Grove
Fil marron	Fil noir
Fil rouge	Fil rouge
Fil orange	Fil jaune
NC	Fil blanc