

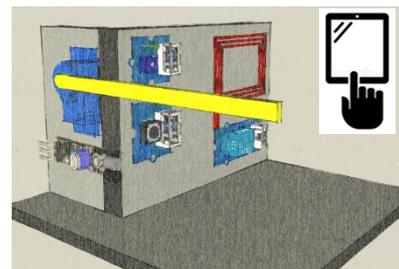
Imaginer concevoir et programmer des applications informatiques nomades

## Situation problème :

On souhaite ajouter la fonction **Piloter à distance une barrière levante avec un appareil mobile**. Vous disposez d'une maquette de barrière munie d'un module Bluetooth, d'une tablette Android et du site App Inventor

Matériel: Pour créer l'application nous avons besoin :

- d'un ordinateur et de se rendre sur le site: <http://code.appinventor.mit.edu/>
- d'une tablette android avec l'application MIT AI2 Companion
- de la maquette de la barrière



## Situation 1 : Créer une application Led ON \_ Led Off pour smartphone ou tablette?

But : passer du mode LED allumée au mode LED éteinte en appuyant sur les boutons allumer et éteindre

### 1.1. CREER LE DESIGN DE L'APPLICATION :

Designer

Suivre le tuto vidéo en ligne

### 1.2. DECRIRE LE COMPORTEMENT DE L'APPLICATION :

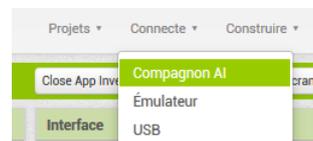
Blocs

### 1.3. TESTER L'APPLICATION CREEE :

Depuis la tablette, lancer l'application MIT AI2 Companion , scanner le QR code ou écrire les caractères obtenu à l'écran de l'ordinateur

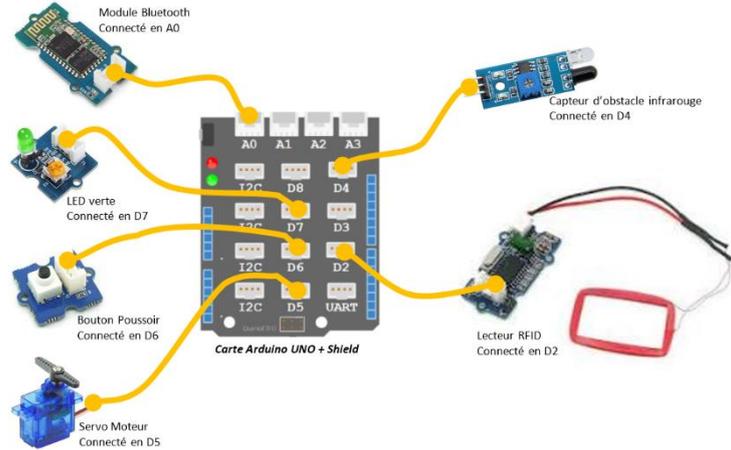
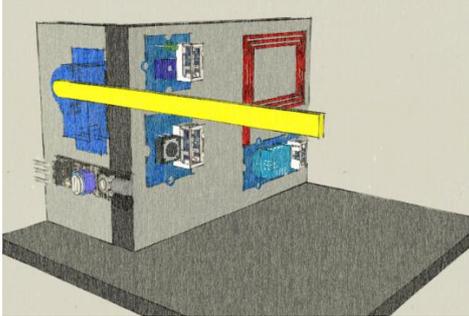
**Attention il faut cocher  Use legacy connection**

tester le résultat sur la tablette



## Situation 2 : Test maquette barrière

### 2. MAQUETTE DE LA BARRIERE



### Programmation de la maquette avec Mblock3



Mode Téléverser  
 Le programme doit être transmis à la carte, il faut cliquer sur Téléverser dans l'Arduino, cette étape peut prendre du temps, soyez patient

```

1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 #include "Seeed_RFID.h"
6
7 double angle_rad = PI/180.0;
8 double angle_deg = 180.0/PI;
9 double code;
10 Seeed_RFID RFID_2(2, 2+1);
11
12 void setup() {
13   pinMode(7, OUTPUT);
14 }
15
16 void loop() {
17   if (RFID_2.isAvailable()) {
18     code = RFID_2.cardNumber();
19     if ((code) == (6696431)) {
20       digitalWrite(7, 1);
21       delay(5);
22     }
23   }
24   digitalWrite(7, 0);
25   _loop();
26 }
27
28 void _delay(float seconds) {
29   long endTime = millis() + seconds * 1000;
30   while (millis() < endTime) _loop();
31 }
32
33 void _loop() {
34 }
    
```

Ici nous utiliserons **Mblock 3** pour programmer la carte Arduino UNO dans le menu **Choix de la carte** sélectionner **Arduino UNO**, avec les extensions Uno Grove, Dans le menu **Gérer les extensions**, sélectionner **UNO et Grove** et **A4 Grove**

#### Mode Téléverser

Une fois la maquette reliée en USB à l'ordinateur il faut la connecter à Mblock 3, dans le menu **Connecter** sélectionner le port **COM\_** (jamais le COM1)

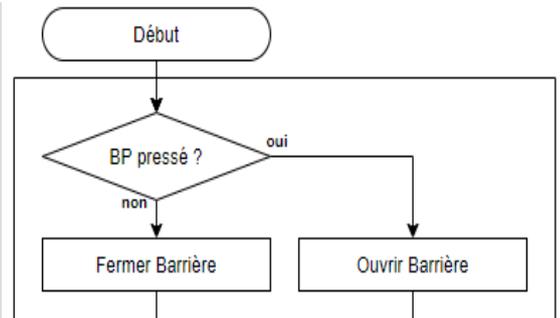
Pour tester le programme il faut cliquer sur **Edition Mode Arduino** puis sur **Téléverser dans l'Arduino**, cette étape peut être longue, soyez patient

#### Piloter l'ouverture avec le Bouton Poussoir

Reproduire et tester le programme suivant en complétant le nom des broches

```

UNO et Grove - générer le code
répéter indéfiniment
si Lire l'état logique du bouton poussoir sur la broche [ ] = [ ] alors
  Définir l'angle du servo-moteur de la broche [ ] à 220 vitesse 80
sinon
  Définir l'angle du servo-moteur de la broche [ ] à 90 vitesse 80
    
```



Ne rien écrire sur ce document

Imaginer concevoir et programmer des applications informatiques nomades

Situation 3 : Créer une application Led ON \_ Led Off pour smartphone ou tablette permettant d'allumer et éteindre une LED en bluetooth?

### 3.1. CREER LE DESIGN DE L'APPLICATION :

Designer

**Interface utilisateur** Sélectionneur de liste  
renommer en Se connecter en Bluetooth

Arrangement Horizontal

Image à renommer LED Off

Image à renommer LED On décocher la case  
Visible   
visible

Bouton à renommer allumer

Bouton à renommer éteindre

**Connectivité** Client Bluetooth

**Composants**

- Screen1
- Se\_connecter
- Arrangement\_horizontal1
- led\_on
- led\_off
- Allumer
- Eteindre
- Client\_Bluetooth1

### 3.2. DECRIRE LE COMPORTEMENT DE L'APPLICATION :

Blocs

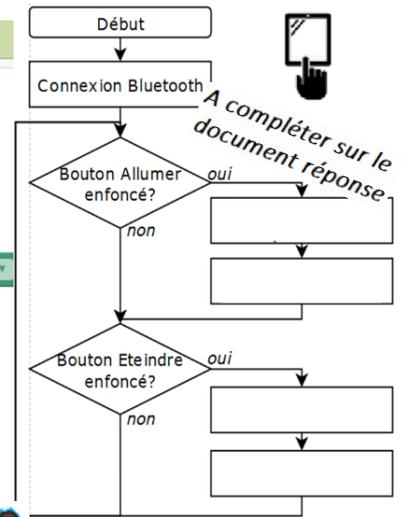
**Interface**

quand Se\_connecter . Avant prise  
faire mettre Se\_connecter . Éléments à Client\_Bluetooth1 . Adresses et noms

quand Se\_connecter . Après prise  
faire mettre Se\_connecter . Activé à appeler Client\_Bluetooth1 . Se connecter adresse Se\_connecter . Sélection

quand Allumer . Enfoncé  
faire mettre led\_on . Visible à vrai  
mettre led\_off . Visible à faux  
appeler Client\_Bluetooth1 . Envoyer1Octet nombre 1

quand Eteindre . Enfoncé  
faire mettre led\_on . Visible à faux  
mettre led\_off . Visible à vrai  
appeler Client\_Bluetooth1 . Envoyer1Octet nombre 0



### 3.3. PROGRAMMATION DE LA CARTE ARDUINO DE LA BARRIERE avec MBlock3

UNO et Grove - générer le code

mettre BT à 0

répéter indéfiniment

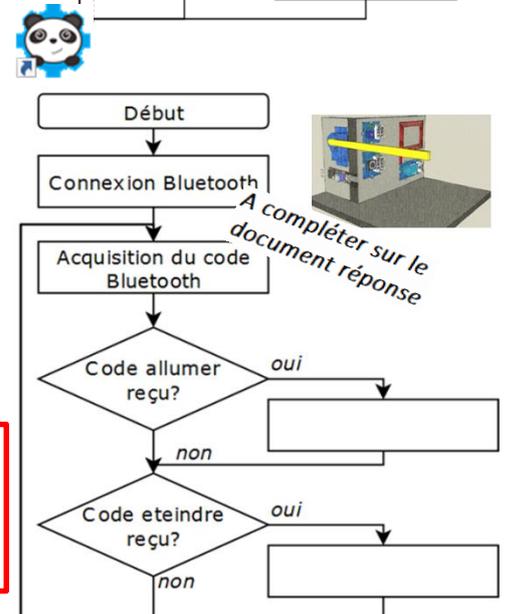
mettre BT à Module Bluetooth sur A0 recevoir donnée

si BT = 1 alors  
LED sur D7 Activer

si BT = 0 alors  
LED sur D7 Désactiver

La variable BT devra être créée et permettra de stocker la

**!** Lors du téléversement du programme dans la carte il est impératif de déconnecter le module Bluetooth ou le transfert échouera



Imaginer concevoir et programmer des applications informatiques nomades

## Situation 4 : Créer une application pour piloter le portail avec un smartphone ou tablette en Bluetooth?

But : piloter la barrière à distance

### 4.1. CREER LE DESIGN DE L'APPLICATION :

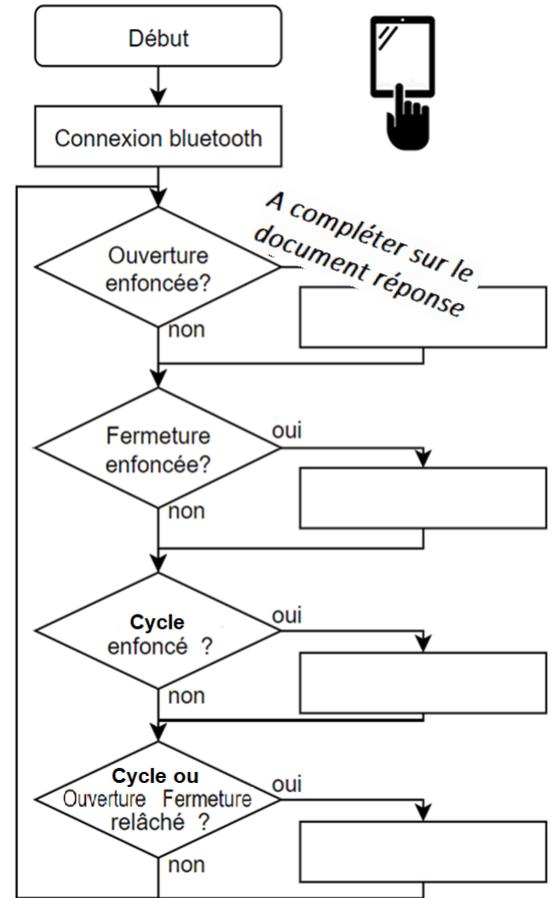
Designer

**Interface**

**Composants**

- Screen1
  - Se\_connecter
- Arrangement\_horizontal1
  - Image1
  - Ouverture
  - Fermeture
  - Cycle\_Complet
  - Client\_Bluetooth1

Cycle Complet = Ouvrir, Attendre 5 secondes, Fermer si obstacle non détecté



### 4.3. PROGRAMMATION DE LA CARTE ARDUINO DE LA BARRIERE avec MBlock3

