

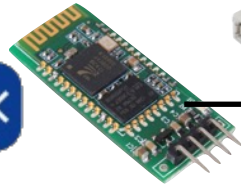
APP INVENTOR



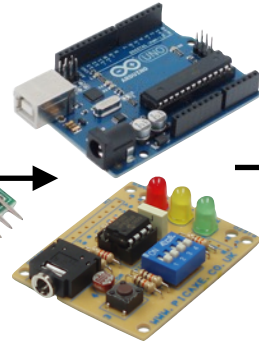
Application Android



Communication sans fil bluetooth



Module bluetooth

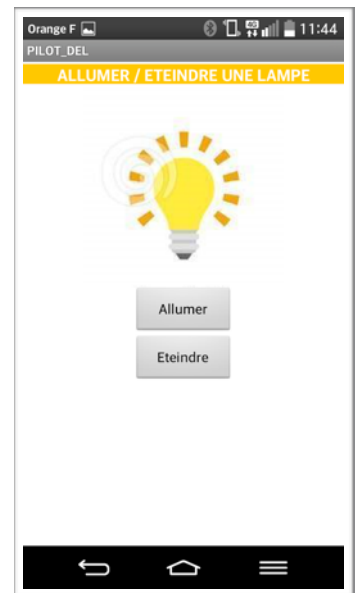
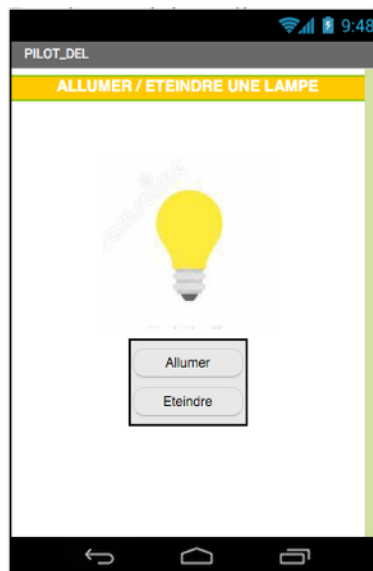
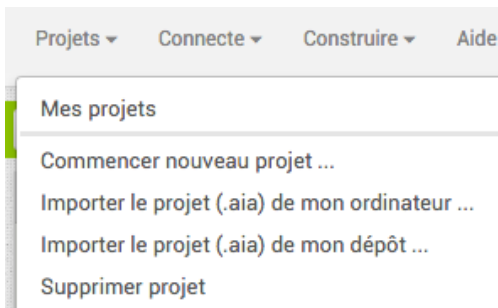


Interface programmable (Arduino ou Picaxe)



Dans cet exemple il s'agit, de piloter une del depuis le smartphone (application Android)

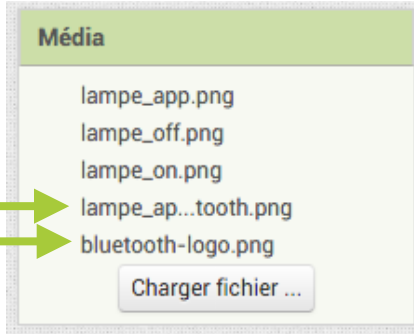
1
Reprendre l'application Appli Lampe vue précédemment et l'enregistrer sous un nouveau nom de projet



Ajouter les images que l'on va utiliser par la suite :
 Icône bluetooth et nouveau logo de l'application

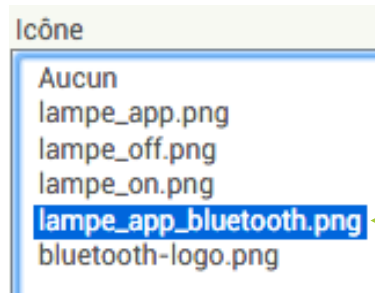


2



Changez les propriétés de l'application : Logo et nomme l'application

3



Ajoutez en dessous du bandeau une disposition horizontale qui intègre une image (pour le logo bluetooth), un sélectionneur de liste (pour le choix de la connexion au réseau bluetooth) et un bouton (pour dé-activer la connexion bluetooth)

4

Propriétés

Screen1

À propos de l'écran

Alignement horizontal: Centrer

Alignement vertical: Haut

AppName: PILOT_DEL_Bluetooth

Couleur de fond: Blanc

Image de fond: Aucun...

Animation fermeture écran: Par défaut

Icône: lampe_app_bluetooth.1

Animation ouverture écran: Par défaut

Orientation écran: Indéterminé

Défilant:

ShowStatusBar:

Sizing: Fixed

Titre: PILOT_DEL_Bluetooth

PaLETTE

Interface utilisateur

- Bouton
- Zone de texte
- Vue liste
- Sélecteur de date
- Sélecteur temps
- Case à cocher
- Label
- Sélecteur de liste
- Ascenseur
- Zone texte mot de passe
- Notificateur
- Image
- Afficheur Web
- Curseur animé

Disposition

Média

Dessin et animation

Interface

Afficher les composants cachés dans l'interface

Cochez pour voir un aperçu sur un appareil de taille tablette.

PILOT_DEL_Bluetooth

ALLUMER / ETEINDRE UNE LAMPE

Texte pour Sélectionneur_de_liste1

Image

Allumer

Eteindre

Arrangement_horizontal1

- Image1
- Sélectionneur_de_liste
- Bouton1

Components

- Screen1
 - Titre
 - Arrangement_horizontal1
 - Bluetooth
 - BP_Connecte
 - BP_Deconnecte
 - LAMPE_ON
 - LAMPE_OFF
 - Arrangement_vertical1
 - BP_ON
 - BP_OFF

Renommer Supprimer

Pensez à renommer les composants pour une lecture plus facile

Image

- Aucun
- lampe_app.png
- lampe_off.png
- lampe_on.png
- lampe_app_bluetooth.png
- bluetooth-logo.png

Charger fichier ...

Annuler OK

Largeur: 150 pixels...

Texte: Connecté

Texte: Déconnecté

Largeur: 150 pixels...

Visible:

5

Modifiez les propriétés des 3 composants ajoutés en suivant les images ci-contre

Pourquoi ne pas mettre un simple bouton pour la connexion ? Car le « sélectionneur_de_liste » permet d'afficher toutes les connexions bluetooth disponibles

Il est normal que le bouton « Déconnecté » ne soit pas visible. Il sera visible uniquement quand la connexion en bluetooth sera activée

Palette

- Interface utilisateur
- Disposition
- Média
- Dessin et animation
- Capteurs
- Social
- Stockage
- Connectivité
 - Client Bluetooth
 - Serveur Bluetooth
 - Web
 - Déclencheuractivité

LEGO® MINDSTORMS®

screen1

ALLUMER / ETEINDRE UNE LAMPE

Connecté

Allumer

Eteindre

Composants non-visible

Client_Bluetooth1

6

Il ne reste plus qu'à ajouter un service bluetooth (Client Bluetooth) à notre application

En glisser/déposer sur l'écran de simulation, il s'affiche automatique en bas

Côté programmation ... Quelques ajouts et modifications sont à réaliser pour communiquer via le bluetooth

Designer

Blocs

```

quand BP_Connecte .Avant prise
faire
mettre BP_Connecte . Eléments à Client_Bluetooth1 . Adresses et noms
  
```



Quand BP_Connecte (le sélectionneur_de_liste) cliqué :
Afficher le nom des connexions disponibles en bluetooth

```

quand BP_Connecte .Après prise
faire
mettre BP_Connecte . Activé à appeler Client_Bluetooth1 .Se connecter
adresse BP_Connecte . Sélection
mettre BP_Connecte . Visible à faux
mettre BP_Deconnecte . Visible à vrai
  
```



Quand BP_Connecte (le sélectionneur_de_liste) a été sélectionné :
Se connecter au bluetooth sélectionné
Ne plus afficher le bouton qui permet d'afficher cette liste de bluetooth : BP_Connecte
Afficher le bouton « se déconnecter » : BP_Déconnecte

```

quand BP_Deconnecte .Clic
faire
appeler Client_Bluetooth1 .Déconnecter
mettre BP_Connecte . Visible à vrai
mettre BP_Deconnecte . Visible à faux
  
```



Quand BP_Deconnecte cliqué :
Se déconnecter du bluetooth
Afficher le bouton qui permet d'afficher cette liste des bluetooth disponibles : BP_Connecte
Ne plus afficher le bouton « se déconnecter » : BP_Déconnecte

```

quand BP_ON .Clic
faire
mettre LAMPE_ON . Visible à vrai
mettre LAMPE_OFF . Visible à faux
appeler Client_Bluetooth1 .Envoyer1Octet
nombre 1
  
```



Quand BP_ON cliqué (allumer lampe) :
Le Client_Bluetooth envoie le nombre 1

```

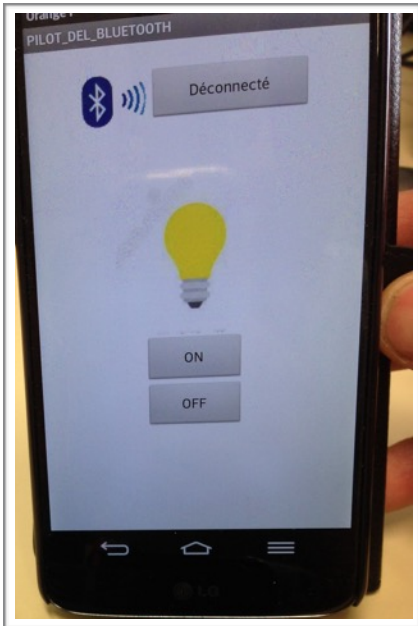
quand BP_OFF .Clic
faire
mettre LAMPE_ON . Visible à faux
mettre LAMPE_OFF . Visible à vrai
appeler Client_Bluetooth1 .Envoyer1Octet
nombre 0
  
```



Quand BP_OFF cliqué (éteindre lampe) :
Le Client_Bluetooth envoie le nombre 0

L'application est terminée, vous pouvez la tester et l'installer sur la tablette ou smartphone Android

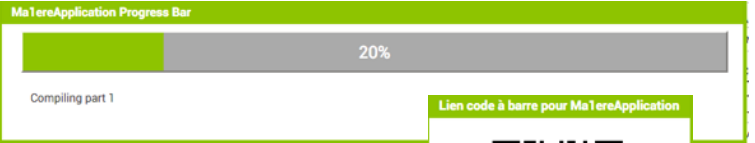
8



Construire ▾ Aide ▾ Mes Projets

App (Donnez le code QR pour fichier .apk)

App (enregistrer .apk sur mon ordinateur)



Lien code à barre pour MaTereApplication



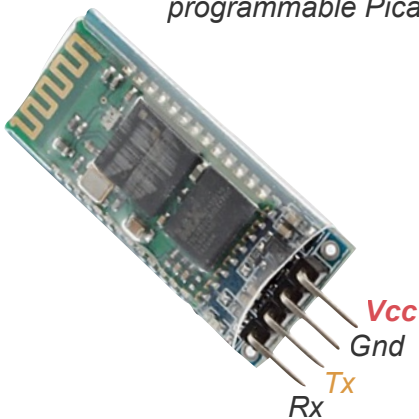
OK

Note: ce code à barre n'est valide que 2 heures. Regardez la FAQ pour voir des informations sur comment partager votre application avec les autres.



Il reste maintenant à réaliser un montage électronique qui permet de recevoir en bluetooth le 1 et 0 générés par l'application.

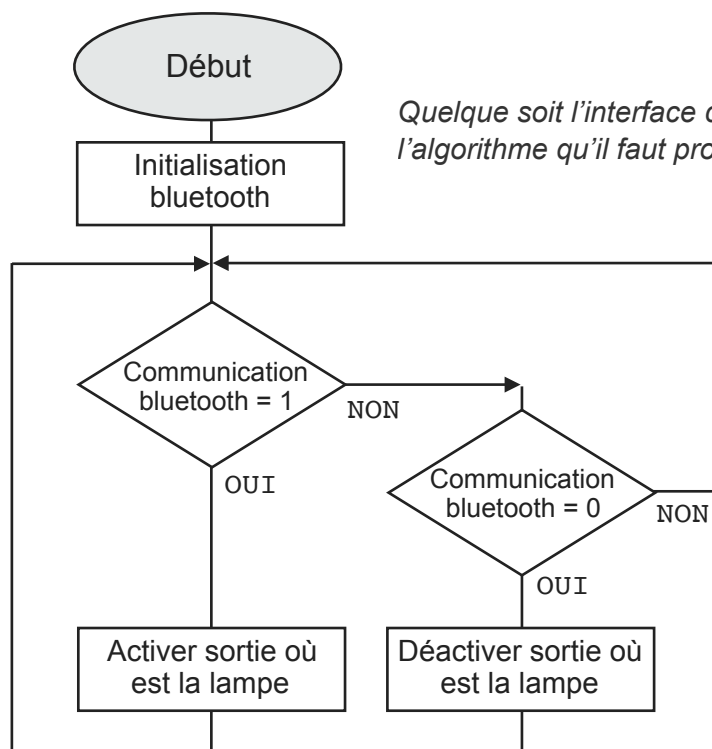
La solution la plus simple étant d'utiliser une interface programmable Picaxe ou Arduino ...



Vcc pour l'alimentation 3,3V ou 5V

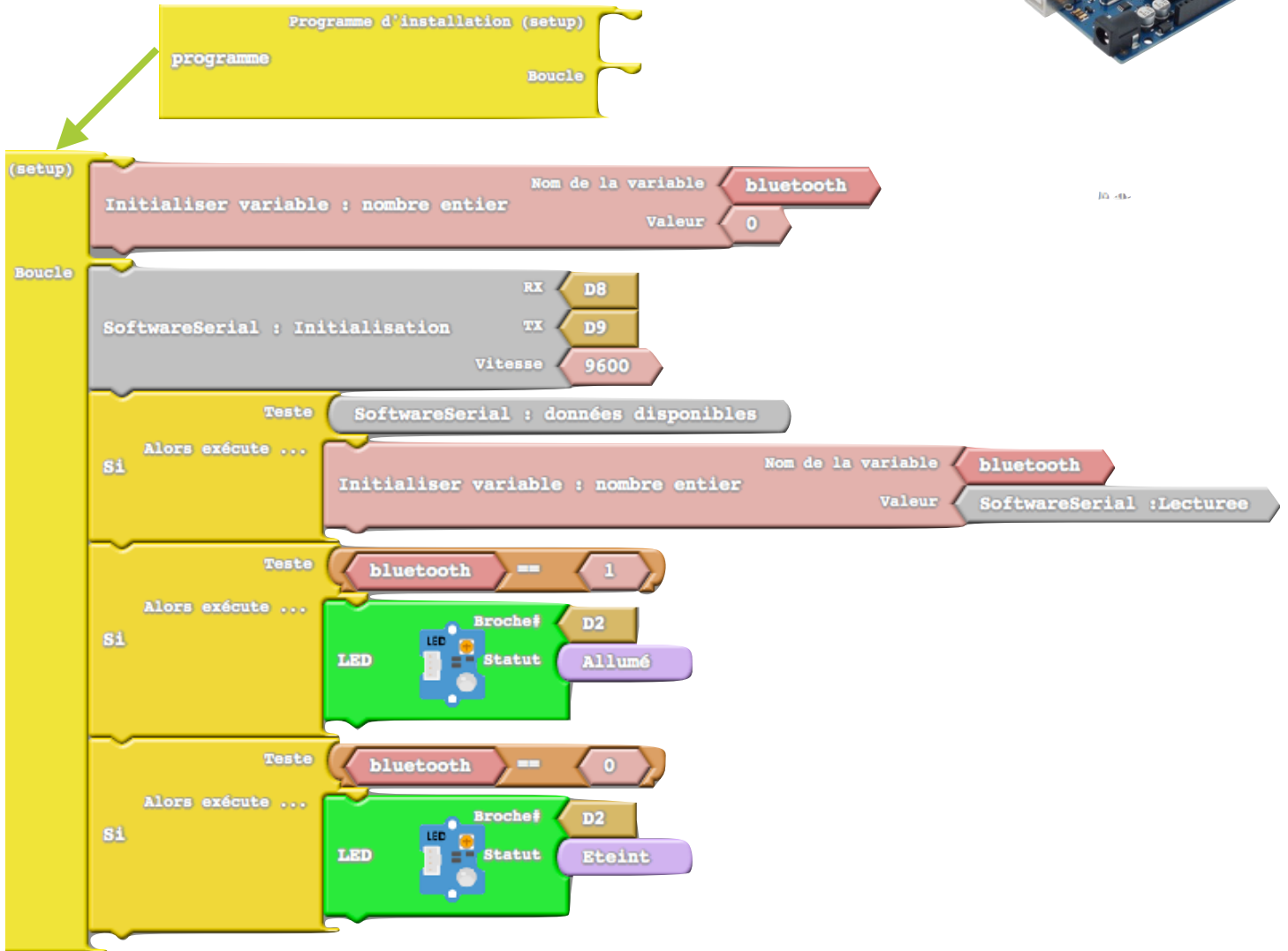
Gnd pour l'alimentation : 0V

Tx et Rx pour la communication

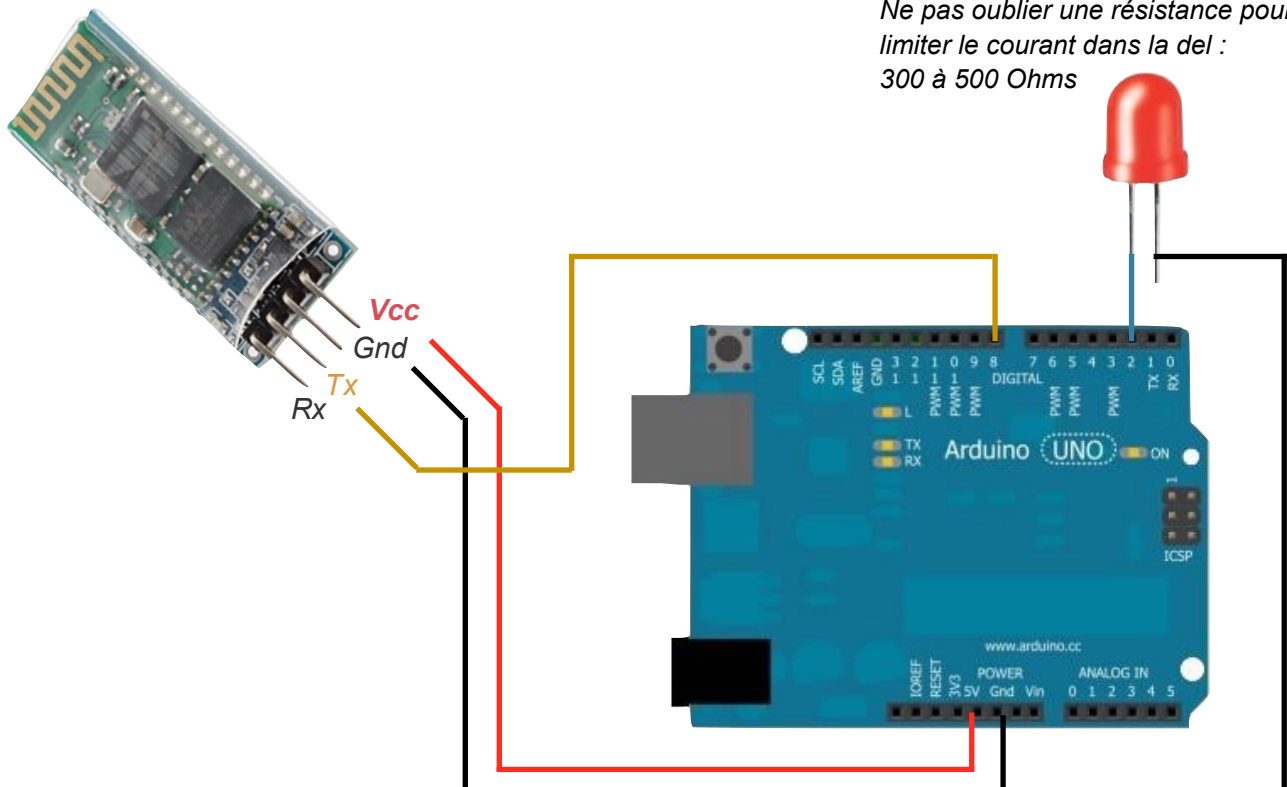


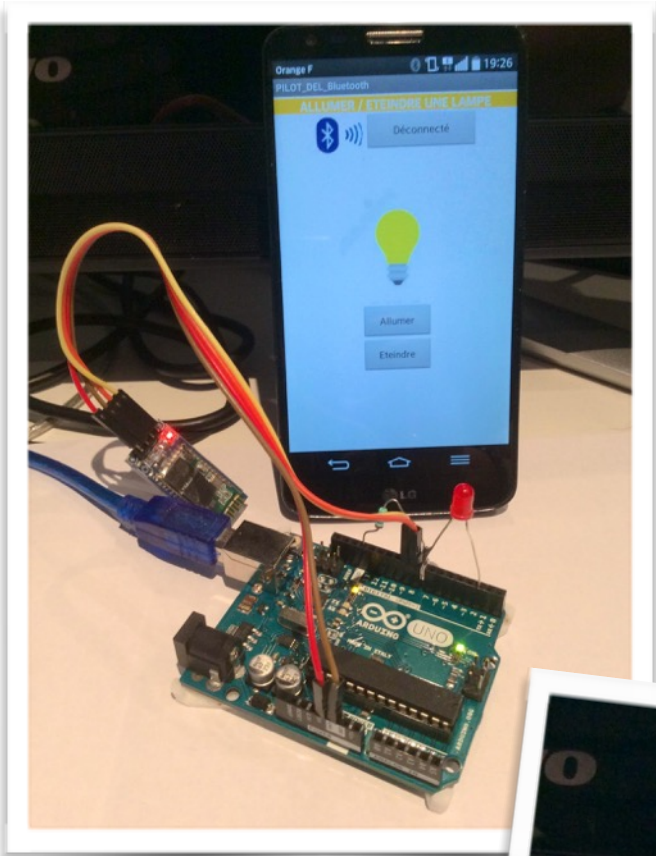
Quelque soit l'interface choisie, voici l'algorithme qu'il faut programmer

Exemple avec une interface Arduino sous Ardublock



Ne pas oublier une résistance pour limiter le courant dans la del : 300 à 500 Ohms





App Inventor - Appli Lampe via Bluetooth