

## App Inventor

**Modifie tes programmes pour que les actions d'allumer/éteindre la DEL soient confirmées par le programme Arduino avant d'être prises en compte par l'application Android.**

### Première étape : Créer le projet

Pour ce projet, tu peux, soit réutiliser ton application et ton code Ardublock de la séance précédente (fiche Piloter une DEL par Bluetooth), soit utiliser les fichiers suivants :

- '**applicationCtrlDELAvecRetourGabarit.aia**' qui se trouve sur App Inventor 2. Il s'agit d'une application qui envoie des commandes via Bluetooth
- '**ardublockCtrlDELAvecRetourGabarit.abp**' qui te sera donné par l'enseignant. Il contient le code Arduino qui interprète les commandes pour allumer/éteindre la DEL .

Crée une copie de ces fichiers que tu appelleras selon les consignes données par l'enseignant.

### Deuxième étape : Interaction Android/DEL

Pour que l'application Android sache si le programme Arduino a bien pris en compte l'action à réaliser sur la DEL (allumer/éteindre) :

- le programme Arduino doit envoyer des messages à l'application Android (via Bluetooth) lorsqu'il réalise une action sur la DEL
- l'application Android doit traduire les messages reçus comme des confirmations des actions sur la DEL.

Il faut donc que l'application Android et le programme Arduino soient d'accord sur ces messages et leur signification.

Pour ton cas, les messages envoyés par le programme Arduino à l'application Android seront :

- le chiffre '1' pour confirmer que la DEL a été allumée (l'application Android interprète le '1' comme le message « la DEL vient d'être allumée »)
- le chiffre '0' pour confirmer que la DEL a été éteinte (l'application Android interprète le '0' comme le message « la DEL vient d'être éteinte »)

## Troisième étape : Interaction Android/DEL (AI2)

Pour que l'application Android puisse recevoir des messages par Bluetooth, elle doit utiliser un composant **Horloge** (que tu trouveras dans **Capteurs**). Ce composant est en fait une alarme qui se déclenche toutes les secondes pour exécuter les blocs qui sont dans le bloc **quand Horloge1.Chronomètre**. Ci dessous le code de l'application.

Teste le et écris sur papier l'algorithme qu'il implémente.

```
initialise global etatLampe à " OFF "
```

```
quand Horloge1.Chronomètre  
faire  
si Client Bluetooth1.Est connecté  
alors  
si  
appeler Client Bluetooth1.Octets disponibles pour le réception >> 0  
alors  
initialise local etatRecu à 0  
dans  
mettre etatRecu à appeler Client Bluetooth1.Recevoir texte  
nombre d'octets appeler Client Bluetooth1.Octets disponibles pour le réception  
si  
obtenir global etatLampe == " OFF " et obtenir etatRecu == 1  
alors  
mettre global etatLampe à " ON "  
mettre imgAmpoule.Image à " ampouleAllumee.jpeg "  
mettre Horloge1.ChronomètreActivé à faux  
sinon  
si  
obtenir global etatLampe == " ON " et obtenir etatRecu == 0  
alors  
mettre global etatLampe à " OFF "  
mettre imgAmpoule.Image à " ampouleEteinte.jpeg "  
mettre Horloge1.ChronomètreActivé à faux
```

```
quand btnAllumer.Clic  
faire  
mettre Horloge1.ChronomètreActivé à vrai
```

```
quand btnEteindre.Clic  
faire  
mettre Horloge1.ChronomètreActivé à vrai
```

## Quatrième étape : Interaction Android/DEL (Ardublock)

Et voici le programme pour la carte Arduino :

```
Initialisation  
SoftwareSerial : Initialisation  
Broche de réception RX: D2  
Broche d'envoi TX: D3 ~ Uno  
Vitesse: 9600
```

```
Boucle  
Teste SoftwareSerial : données disponibles  
Alors exécute ...  
Teste SoftwareSerial :Lecturee == 1  
Alors exécute ...  
Si SoftwareSerial :Lecturee == 1  
Si - Sinon  
Sinon exécute ...  
Fixe la sortie numérique au niveau # D8 HAUT  
SoftwareSerial : écrire message Collier true nouvelle ligne  
Fixe la sortie numérique au niveau # D8 BAS  
SoftwareSerial : écrire message Collier 0 true nouvelle ligne
```

Écriture des données  
sur le module  
Bluetooth

# Objets Connectés

Piloter une  
DEL par  
Bluetooth.  
Exercices.

## App Inventor

### Exercice 1

Modifie ton application Android pour que tant qu'elle n'a pas reçu de confirmation du programme Arduino, elle ne permette pas des actions sur les boutons allumer/éteindre.

*Piste* : utilise la propriété **Activé** des boutons allumer/éteindre.

### Exercice 2

Simplifie le code de ton application Android en utilisant des procédures.

