

Descriptif de séquence en Technologie

Cycle :4	FICHE ÉLÈVE	
Seq : Les objets connectés		
Séance 3 : Créer une application Android		

Situation problème :



Vous venez de programmer le contrôleur principal Arduino à l'aide de mblock. Notre système est connecté. Vous allez maintenant développer la partie application Android qui permettra de piloter votre système.

Reformulation du problème :



Hypothèses :



1ère PARTIE:

➡ Documents ressources :

➡ Production finale attendue : Créer une application Android pour commander une diode par Bluetooth

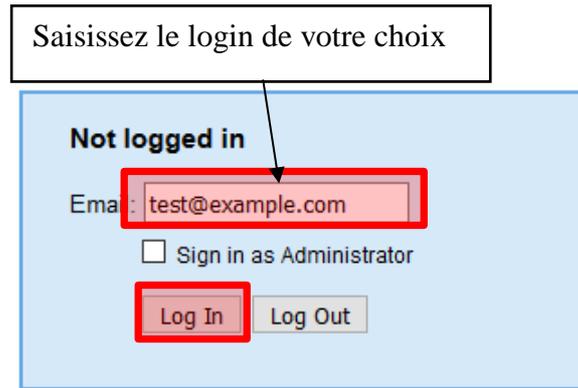


Travail à faire en ilot :

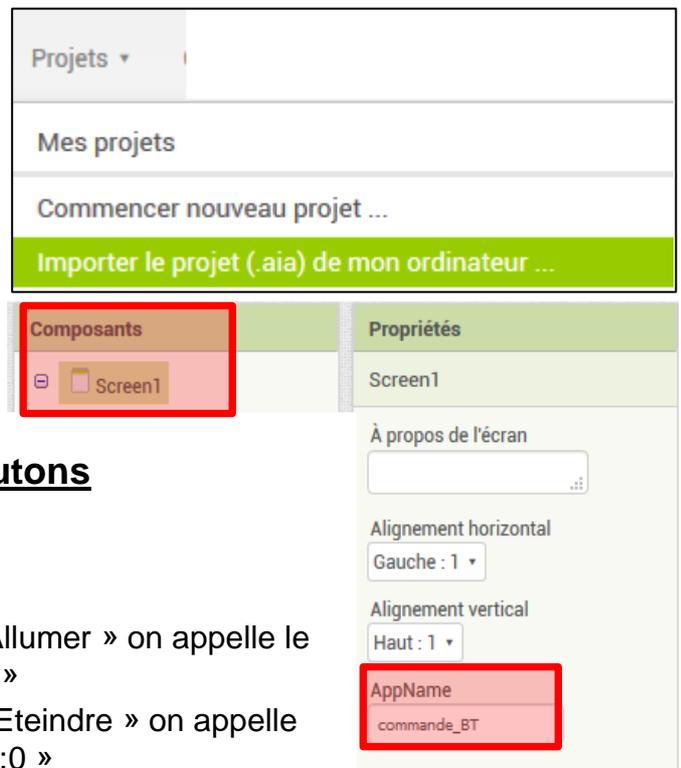
- ➡ Importer un projet
- ➡ Ajouter une interface au projet
- ➡ Programmer l'application
- ➡ Exporter l'application vers la tablette

Partie 1 : Installation

- Se connecter à App-Inventor en créant votre login



- Importer le projet « Commande_BT.aia » dans votre espace. La partie de gestion du Bluetooth est un peu complexe. C'est la raison pour laquelle vous n'avez pas à la programmer.
- Renommer votre application



Partie 2 : Créer une application avec 2 boutons

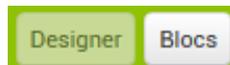
Travail 1 :

Créer une application comportant 2 boutons.

- Allumer** : Lorsqu'on clique sur le bouton « Allumer » on appelle le client_Bluetooth1 pour envoyer le texte « B:1 »
- Eteindre** : Lorsqu'on clique sur le bouton « Eteindre » on appelle le client_Bluetooth1 pour envoyer le texte « B:0 »

Méthode :

- Partie Designer



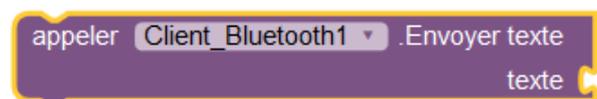
Créer deux boutons (Interface utilisateur), les renommer (Composants), puis modifier leurs Propriétés (largeur, texte, couleur, forme...).

Si vous souhaitez mettre ces boutons côte à côte vous devez créer un arrangement horizontal (Disposition) et faire glisser les boutons à l'intérieur.

- Partie Blocs



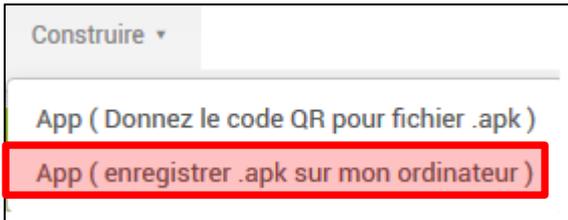
Les boutons que vous avez créés doivent apparaître. Lorsque je clique dessus j'appelle le client Bluetooth pour envoyer le texte B:1 ou B:0



Coup de pouce 1 et 2 : Créer un bouton et le paramétrer

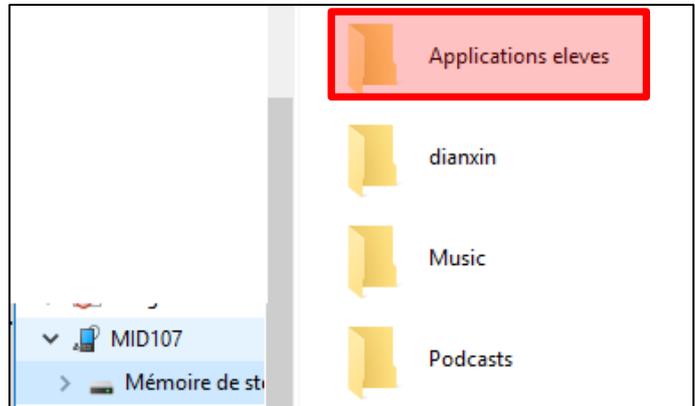
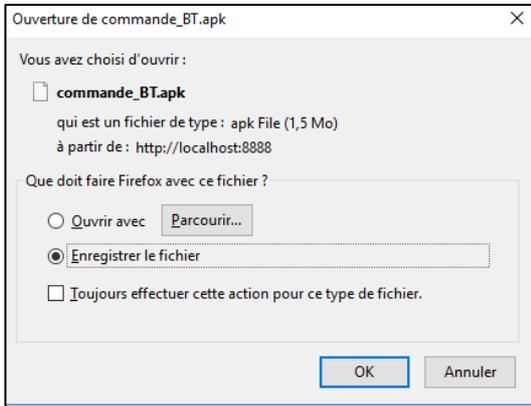
Travail 2 : Installer l'application et tester votre projet

1. Connecter votre tablette à l'ordinateur



2. Cliquer sur Construire et App (enregistrer .apk sur mon ordinateur)

3. Enregistrer le fichier généré sur votre tablette en cliquant sur Enregistrer sous



4. Sur la tablette cliquer sur Mes fichiers

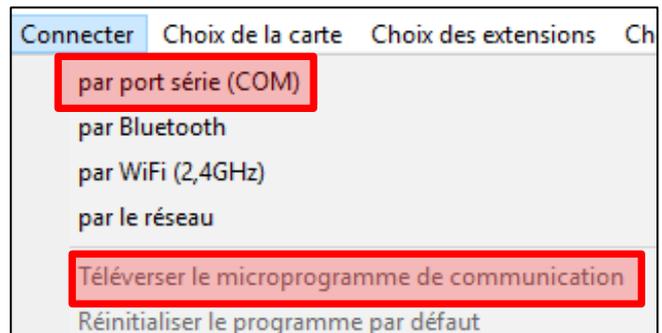


5. Ouvrir le dossier Applications élèves puis votre fichier.apk ;

6. Valider pour lancer l'installation ;

7. Connecter le système Arduino à votre ordinateur ;

8. Ouvrir l'application mBlock et mettez-vous en mode connecté ;



9. Ouvrir le fichier que vous avez fait lors de la séance 2 « BT_2boutons.sb2 »

10. Sur l'application de la tablette cliquer sur le bouton Bluetooth et choisir le module qui correspond à votre îlot ;



11. Piloter le système. Vérifier que cela fonctionne, sinon corriger vos erreurs et recommencer le transfert.

Partie 3 : Système commandé avec un seul bouton (partie difficile)

Travail 1 :

Ajouter au programme précédent une zone de label nommée « fonction ».

Si on reçoit par BT du module Arduino la chaîne de caractères B:3, alors on écrit dans cette zone de label « Diode allumée »

Si on reçoit par BT du module Arduino la chaîne de caractères B:4, alors on écrit dans cette zone de label « Diode éteinte »

Méthode :

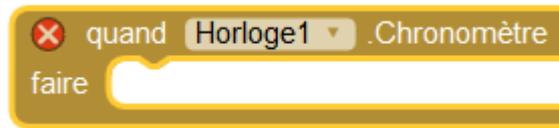
1 - Partie designer

Ajouter une horloge (Capteurs) et régler son intervalle chronomètre à 200.

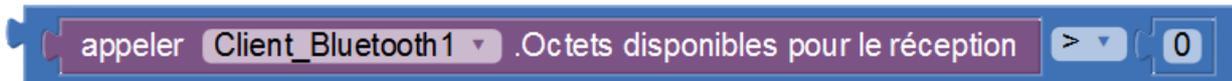
Ajouter une zone de label nommée « fonction » (interface utilisateur)

2 - Partie blocs

Quand horloge1.chronomètre .



SI le Bluetooth est connecté **ET** que des données sont disponibles (ci-dessous)



FAIRE :

- Placer les données dans une variable nommée « BT » (bloc ci-dessous)



- **SI** BT=B:3 **FAIRE** : Mettre fonction.texte à « Diode allumée »
- **SI** BT=B:4 **FAIRE** : Mettre fonction.texte à « Diode éteinte »
-

Installer l'application et tester votre programme. La carte Arduino doit être branchée en mode connecté (voir mblock) avec le programme « **BT_retourinfo.sb2** » actif.



Coup de pouce 3 : Recevoir un message par BT

Travail 2 :

Adapter le programme précédent de manière à ce qu'il n'y ait plus qu'un seul bouton pour commander l'éclairage.

Si BT=B:3 il faut mettre le texte du bouton à « Eteindre ». Si on clique sur le bouton on envoie B:0 (pour éteindre la lampe)

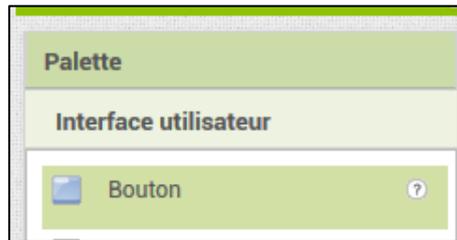
Si BT=B :4 ; il faut mettre le texte du bouton à « Allumer ». Si on clique sur le bouton on envoie B:1 (pour allumer la lampe).

Installer l'application et tester votre programme. La carte Arduino doit être branchée en mode connecté (voir mblock) avec le programme « **BT_retourinfo.sb2** » actif.



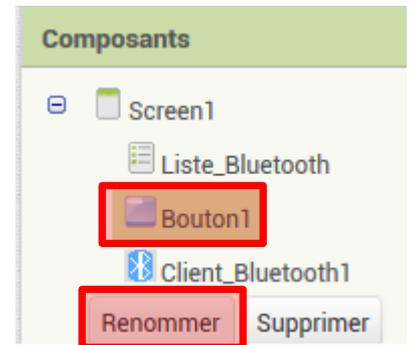
Coup de pouce 1 : Créer un bouton

1 – Interface utilisateur



2 - Cliquer sur bouton, faire glisser le bouton sur l'interface en maintenant le doigt appuyé sur le bouton gauche de la souris. Relâché lorsque le bouton est au bon endroit

3 – Renommer le bouton dans la partie composants



4- Modifier les propriétés du composant dans la fenêtre de droite.

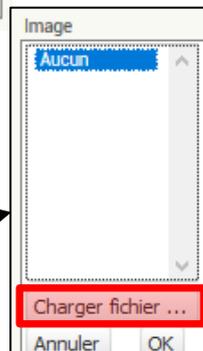
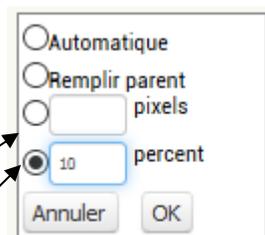
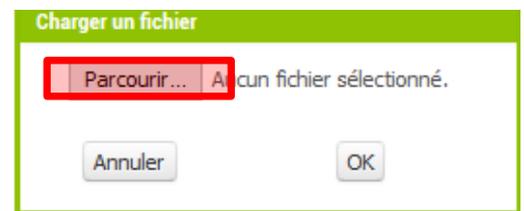


Image qui apparaîtra sur le bouton

Texte qui apparaîtra sur le bouton





Coup de pouce 2 : Partie Blocs

Allumer

quand Allumer .Clic
faire

Permet de définir ce que va faire le programme lorsqu'on clique sur le bouton « Allumer »

Liste_Bluetooth

appeler Client_Bluetooth1 .Envoyer texte
texte

Permet d'envoyer un texte au client Bluetooth

Texte

joint " B: " " 1 "

Permet de joindre plusieurs textes. Ici on constitue le texte « B:1 »

quand Allumer .Clic
faire appeler Client_Bluetooth1 .Envoyer texte
texte joint " B: " " 1 "

Quand on clique sur le bouton allumer on envoie au client Bluetooth le texte B:1

Explication de la partie Bluetooth

quand Liste_Bluetooth .Avant prise
faire mettre Liste_Bluetooth . Éléments à Client_Bluetooth1 . Adresses et noms

Lorsque l'application s'initialise on crée la liste des éléments de « Liste_Bluetooth ». Cette liste est constituée de l'adresse et du nom de tous les appareils appairés par Bluetooth à notre tablette.

quand Liste_Bluetooth .Après prise
faire évaluer mais ignorer résultat appeler Client_Bluetooth1 .Se connecter
adresse Liste_Bluetooth . Sélection

Lorsqu'un client bluetooth est sélectionné dans la liste on vérifie que la communication est fonctionnelle.

Important : Si on connaît déjà le nom du client auquel on veut se connecter on peut remplacer les blocs précédents par

quand Screen1 .Initialise
faire évaluer mais ignorer résultat appeler Client_Bluetooth1 .Se connecter
adresse " robot groupe1 "



Coup de pouce 3 : Recevoir un message

Dans mBlock

- Si la diode est allumée alors on envoie à l'application le texte E:1
- Si la diode est éteinte on envoie à l'application le texte E:0

Bloc « BT envoyer la valeur... »

BT: envoyer <Saisie libre> de valeur 0 sur le port D2

Ce bloc permet d'envoyer une valeur ou état d'un capteur sur le port série choisi.

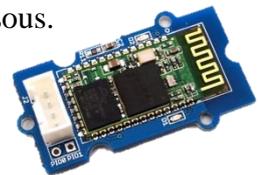
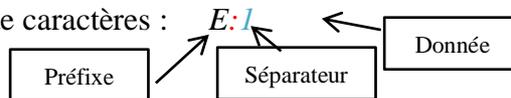
Le bloc envoie une chaîne de caractères constituée de la « valeur » et utilise un préfixe servant à transmettre la donnée.

Ce préfixe permet d'envoyer plusieurs données dans un seul programme comme celle d'un potentiomètre, capteur de température, PIR en choisissant un préfixe différent pour chaque valeur.

Par exemple pour transmettre la valeur d'un potentiomètre à l'application distante, choisir un préfixe exemple « POT » et placer la donnée à transmettre après « valeur » comme ci-dessous.

BT: envoyer POT de valeur Lire la valeur du capteur potentiomètre sur la broche A0 sur le port D2

Le bloc va générer cette chaîne de caractères :



Dans App Inventor

Votre application APPInventor par exemple devra pouvoir retrouver la valeur transmise en utilisant la fonction « split at fist » afin de récupérer les données après « POT : »

Il est impératif que le préfixe soit identique dans APPIncentor et mBlock.

