Descriptif de séquence en Technologie

Cycle :4	FICHE ÉLÈVE	
Seq : Les objets connectés		
Séance 2 : Piloter une lampe à l'aide du clavier, puis par Bluetooth		

1ère PARTIE:

- Documents ressources :

- Production finale attendue : Commander l'allumage de la diode par Bluetooth

Travail à faire en ilot :

- Commander avec deux touches du clavier la diode
- Ne commander la diode qu'avec une seule touche
- Commander la diode par Bluetooth

Partie 1 : Commander avec deux touches

Vous allez devoir programmer un système constitué :

- d'un module de commande Arduino ;

- d'un shield grove ;

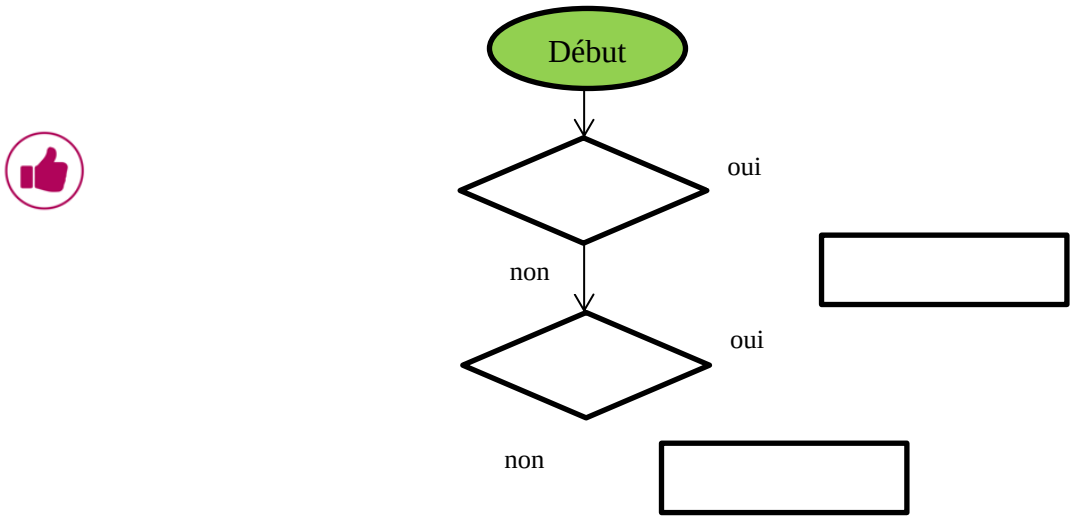
- et d'un module LED Grove.

Identifier sur le croquis ci-dessous le module de commande (en rouge), le capteur (en vert) et l'actionneur (en bleu)

Comportement attendu du système :

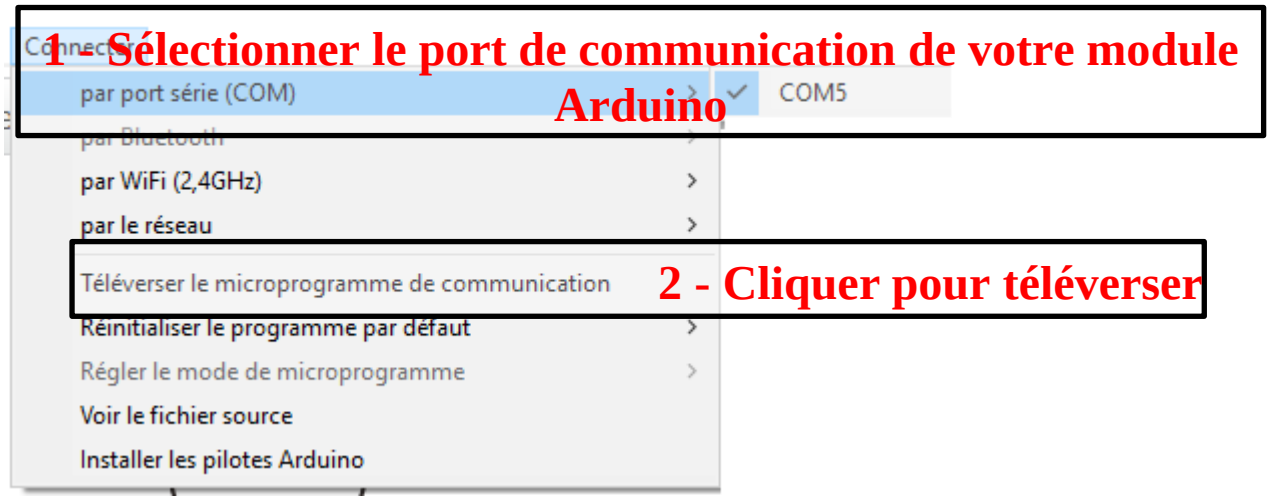
- **Si** j'appuie sur la touche A du clavier **alors** la diode doit s'allumer ;
- **Si** j'appuie sur la touche E du clavier **alors** la diode doit s'éteindre ;

Travail 1 : Compléter l'organigramme ci-dessous en complétant le texte des cases et en ajoutant les flèches manquantes :



Travail 2 : Programmer et piloter le système avec mBlock

- Brancher le module LED sur la broche D4 du shield Grove
- Brancher le module de commande Arduino sur un port USB de l'ordinateur
- Lancer l'application mBlock
- Mettez-vous en mode connecté



- Programmer le système :
- Enregistrer votre travail.

Méthode :

- Le programme doit commencer par 

-
- Pensez à utiliser le bloc ci-dessous pour piloter le module de LED :

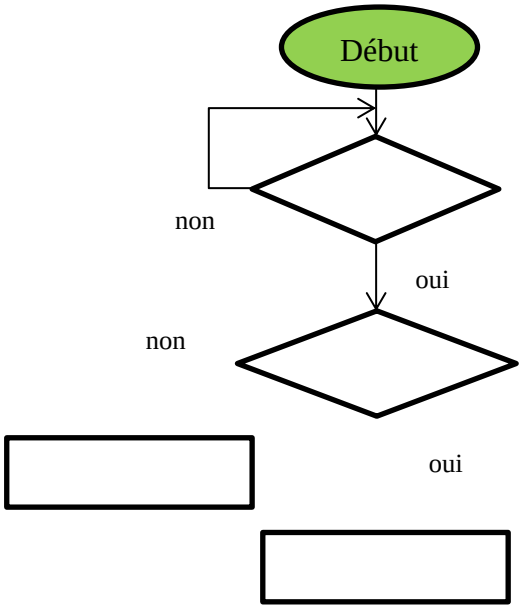
Coup de pouce 1

Partie 2 : Commander avec une seule touche

Comportement attendu du système :

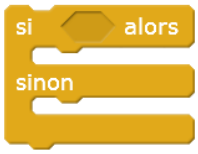
- **Si** j'appuie sur la touche « Espace » du clavier **alors** la LED doit s'allumer **si** elle était éteinte **sinon** elle doit s'éteindre (si elle était allumée).

Travail 1 : Compléter l'organigramme ci-dessous en complétant le texte des cases et en ajoutant les flèches manquantes :



Travail 2 : Programmer et piloter le système avec mBlock

Coup de pouce 1
Pensez à utiliser le bloc suivant :



Utiliser une variable « état » et mettre état=1 lorsque la diode est allumée et état=0 lorsque la diode est éteinte

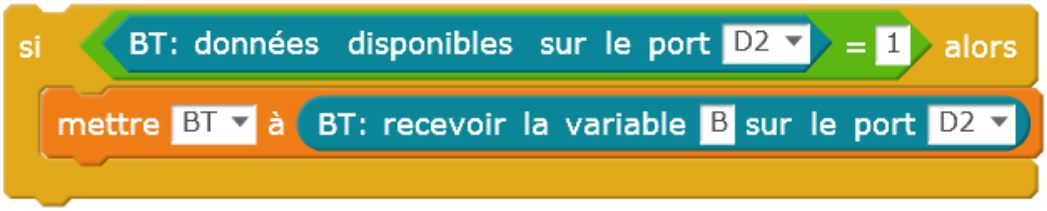
Important : Une temporisation vous permettra de résoudre l'hypersensibilité du système.

Partie 3 : Commander par Bluetooth

Brancher le module de communication Bluetooth sur la broche D2 du shield Grove.

Travail 1 : La LED va être commandée à distance en Bluetooth par une tablette.

- Si on appuie sur la touche « allumer » de l'application installée sur la tablette on envoie alors par Bluetooth la chaîne de caractères « B:1 » (la variable B a pour valeur 1)
- Si on appuie sur la touche « éteindre » de l'application installée sur la tablette en envoie alors par Bluetooth la chaîne de caractères « B:0 » (la variable B a pour valeur 0)



Dans le bloc ci-dessus : Si une donnée Bluetooth est reçue, alors on affecte à la variable BT (comme bluetooth) de notre programme la valeur de la variable B qui nous a été transmise.

Autrement dit :

- **Si BT=1 alors la LED doit s'allumer ;**
- **Si BT=0 alors la LED doit s'éteindre.**

Ajouter les blocs ci-dessus au début de votre premier programme (Partie 1 : Commander avec 2 touches) juste en dessous de répéter indéfiniment, et adapter la suite du programme en remplaçant les conditions

Coup de pouce 2 : Variable et Bluetooth

Enregistrer votre travail dans le fichier « BT_2boutons.sb2 »

Travail 2 :

Il peut être utile de renvoyer au smartphone l'état de la lampe.

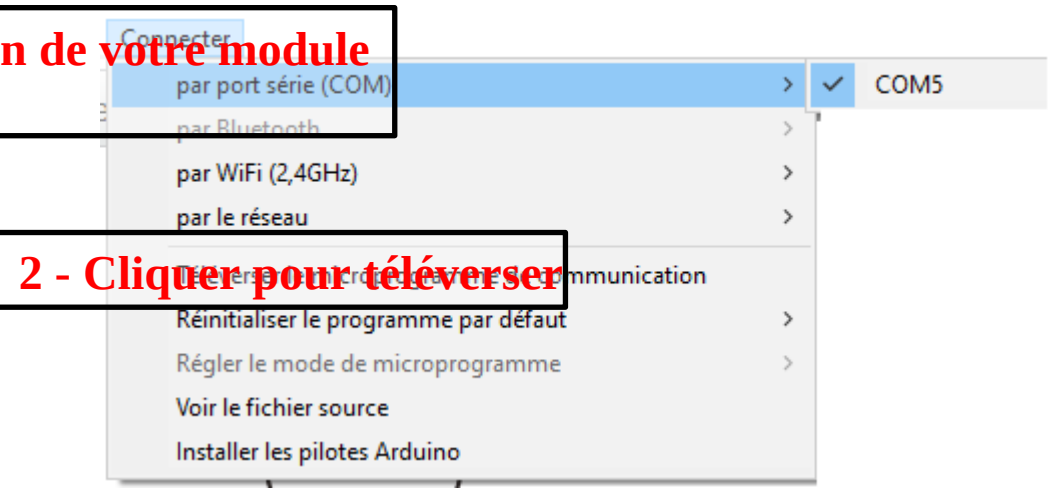
- **Lorsque la lampe s'allume on envoie une seule fois le texte B:3 ;**
- **Lorsqu' elle s'éteint on envoie une seule fois le texte B:4.**

Compléter le programme du travail précédent avec les blocs BT adéquat:

Enregistrer votre travail dans le fichier « BT_retourinfo.sb2 »

Coup de pouce 1 : Programmer avec mBlock

1- Se mettre en mode connecté



Votre programme doit alors commencer par pour piloter le système

2- en 

Votre programme doit comporter le bloc Répéter indéfiniment ce qui permet de le faire tourner indéfiniment boucle

3- 

Si une condition est vraie alors on exécute l'action

4- Test : Est-ce que la touche espace est pressée. Retourne l'état logique 1 si c'est vrai

5- 

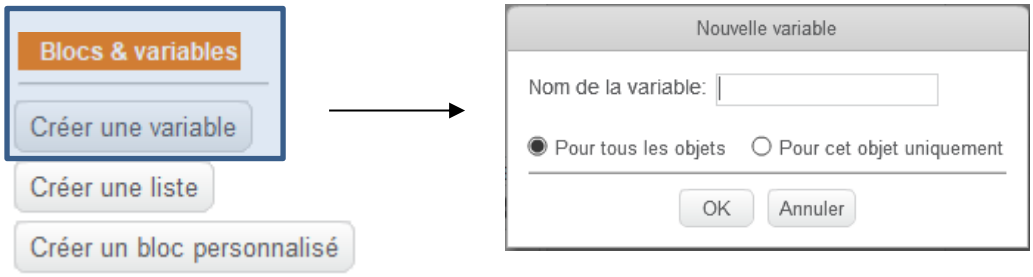
Permet d'activer ou de désactiver la diode branchée sur la broche D4

6- 

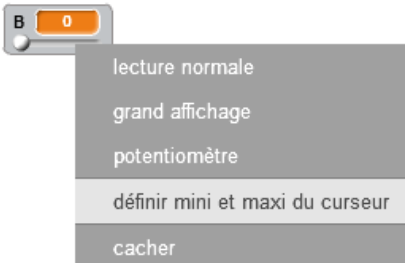
Cette condition est vraie si la broche D4 est à l'état logique 1. Dans notre cas cette condition sera vraie si la diode branchée sur la broche D4 est allumée

Coupsde pouce 2 : Variables et Bluetooth

Variable dynamique : Une variable dynamique est une lettre ou une chaine de caractères à laquelle on affecter une valeur qui peut varier au cours du déroulement du programme.



Permet d'affecter une valeur à la variable dynamique B.



Lorsqu'on utilise le mode potentiomètre, on peut modifier manuellement la valeur de la variable. Il est également possible de définir la valeur maxi et la mini de la variable.

Bluetooth

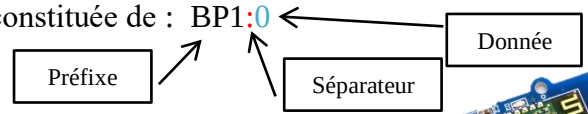
Bloc « BT : données disponible…»

Ce bloc permet de savoir si des données sont disponibles sur le port choisi.
La valeur retournée est de type numérique, « 0 » lorsqu'il n'y a pas de données disponibles et « 1 » lorsque des données sont disponibles sur le port série sélectionné.

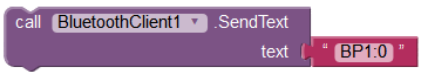
Bloc « BT : recevoir la variable »

Ce bloc permet de recevoir une valeur transmise sur le port série choisi.
Le bloc reçoit une chaine de caractère constituée de la « valeur » et utilise un préfixe pour identifier la donnée.
Par exemple pour recevoir la valeur d'un bouton poussoir n°1 issu d'une application de type APPInventor mon bloc doit être paramétré comme ci-dessous :

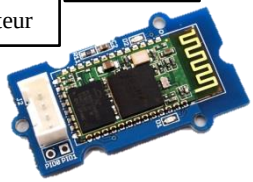
Le bloc s'attend à recevoir cette chaine de caractères constituée de : BP1:0



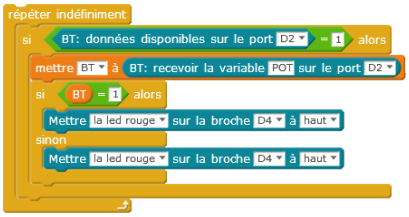
Dans APPInventor, il faudra générer cette chaine comme ci-dessous :



ou



Il est impératif que le nom de ce préfixe soit identique dans APPInventor et mBlock.



Exemple de code avec un module Bluetooth

Sous AppInventor :