Pilotage à distance de la rotation d'une caméra de vidÉo surveillance.

***Thème du programme :*** Informatique embarquée et objets connectés

|  |  |
| --- | --- |
| **Contenus** | **Capacités attendues** |
| Interface homme-machine (IHM) | Réaliser une IHM simple d’un objet connecté |
| Commande d’un actionneur, acquisition des données d’un capteur | Écrire des programmes simples d’acquisition de données ou de commande d’un actionneur. |
| **Exemples d’activités** | |
| * Réaliser une IHM pouvant piloter deux ou trois actionneurs et acquérir les données d’un ou deux capteurs. * Gérer des entrées/sorties à travers les ports utilisés par le système. * Utiliser un tableau de correspondance entre caractères envoyés ou reçus et commandes physiques | |

***Objectifs d'apprentissage :***

Réaliser une IHM sur smartphone. Écrire un programme de commande d'un servomoteur (équipant une caméra). Communiquer les ordres du smartphone vers le servomoteur via Bluetooth afin de le piloter à distance. Utiliser un tableau de correspondance entre caractères envoyés ou reçus et commandes physiques.

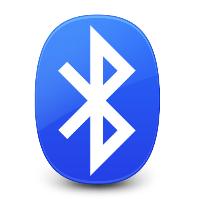
***Matériel mobilisé :*** Carte Arduino, Module Bluetooth HC-05, Smartphone ou tablette, Led, Servomoteur, Caméra (c’est un exemple, mais tout autre système à commander en rotation pourra convenir).

***Durée :*** 1h30

***Période :*** Deuxième ou troisième trimestre

***Remarque :*** Le moteur pas à pas équipant traditionnellement les caméras est remplacé ici par un servomoteur pour sa simplicité de commande. De plus ce servomoteur pourrait être utilisé pour mettre en rotation d’autres systèmes : petit panneau solaire, Pantilt, parabole orientable. La seule contrainte est d’avoir un servomoteur à faire tourner.

***Remarque sur APP inventor hors ligne* :** Une version **portable de APP inventor hors ligne** ne nécessitant **pas la création de compte** en ligne ou d’utilisation de compte Google sera utilisée ici( cf annexes). Cette solution permet de ne pas recourir à un compte Google, et aussi d’éviter de divulguer des données personnelles et ne requiert pas de connexion internet. Cette solution permet également de créer des comptes locaux pour les élèves.

Pilotage à distance de la rotation d'une caméra de vidÉo surveillance.

**Objectif :** Le but est de réaliser une interface graphique pour smartphone permettant de commander la rotation d’une caméra de vidéosurveillance. La caméra est équipée d’un moteur (actionneur). Celui-ci est relié à une carte Arduino équipée d’un module Bluetooth. L’interface graphique est réalisée avec Appinventor. Le pilotage de la caméra est alors réalisé via un curseur sur l’interface graphique ou par accéléromètre en penchant le smartphone.

# Étape : ÉLÉments employÉs

*Indication aux collègues : le montage est déjà prêt et montré aux élèves. Une procédure détaillée de montage est donnée en annexe.*

|  |  |
| --- | --- |
| **Identifier** les éléments ci-dessous sur l’image :   * Carte Arduino * Module Bluetotooth * Servomoteur * Led   **Identifier** les broches de branchement du module Bluetooth, de la LED et du moteur. |  |

# Étape : solutions de communication sans fil existantes

*Indication aux collègues : vidéos en annexe à consulter avant la séance.*

**Compléter le tableau** de comparaison suivant afin d’illustrer les intérêts et limites du Bluetooth.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Technologie | Topologie | Portée | Débit | Avantages | Inconvénients | Applications |
| Bluetooth |  |  |  |  |  |  |
| Wifi |  |  |  |  |  |  |
| Zigbee |  |  |  |  |  |  |

# Étape : COMMUNICATION BLUETOOTH

Le Bluetooth nécessite une communication bidirectionnelle, deux modules peuvent communiquer ensemble en même temps. Le comportement utilisé est de type “maître/esclave”. Un esclave pourra parler avec un seul maître, mais un maître pourra dialoguer avec plusieurs esclaves. L'utilisation se passe en plusieurs étapes :

* Le maître se met en mode “reconnaissable” et fait une recherche des périphériques Bluetooth avoisinants
* L’esclave trouve le maître et envoie son nom pour qu'une connexion puisse être établie
* Le maître accepte la connexion en validant par un code à 4 chiffres la connexion avec l'esclave
* Les périphériques sont alors appairés (ou associés)
* La communication peut commencer

*Indication aux collègues : par défaut chaque carte est configurée en tant qu'esclave avec un débit de 38400 et s'appelle HC-05. Le code pour association avec un PC ou autre est 1234. Il est donc intéressant de pouvoir configurer la carte de manière à changer le débit par défaut et son nom. En annexe on trouve la procédure de configuration des modules. Chaque module pourra ainsi comporter une étiquette avec son nom une fois celui-ci paramétré et renommé.*

**Sur votre téléphone, aller dans paramètres / Bluetooth.**

**Sélectionner le module à appairer.**

**Rentrer le code 1234.**

***Remarque :*** *pour mettre en œuvre le Bluetooth avec la carte Arduino, il est nécessaire de disposer d'un module que l'on relie à la carte. Le module Bluetooth HC-05 dispose de 6 broches.*

* *Led : pour brancher une LED et obtenir un signal sur l'état du module.*
* *VCC : alimentation 3,3V*
* *GND : masse*
* *Tx : pour la communication*
* *Rx : pour la communication*
* *Key : pour envoyer des données de configuration au module*

# Étape : RÔLE DE LA CARTE ARDUINO et tÉLÉversement

**Cliquer** sur toolbox\_arduino\_v3.ino puis sur **« téléverser »** dans la carte Arduino.

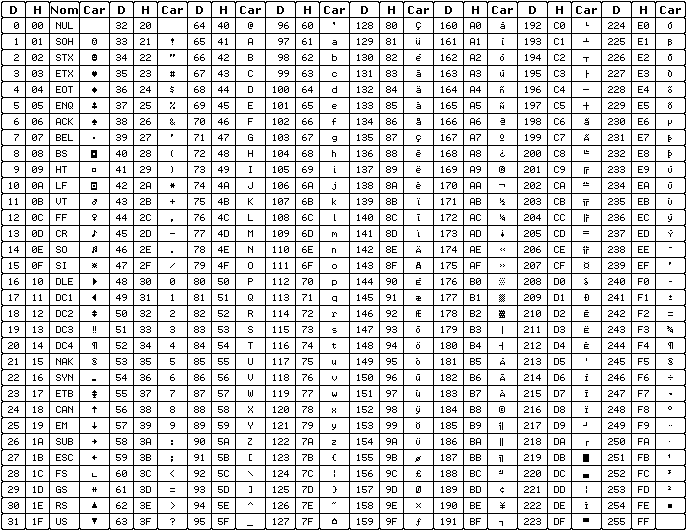
**Débrancher puis rebrancher la carte Bluetooth.** La LED clignote rapidement indiquant ainsi que le mode est celui de la transmission/réception (esclave).

Le câble USB de l'Arduino ne sert alors qu'à alimenter la carte, on ne peut pas transférer de données via USB (ou alors il faut débrancher la carte Bluetooth). On utilise alors le programme chargé dans la carte (toolbox\_arduino\_v3.ino). Chaque commande envoyée à l'Arduino est interprétée et exécutée.

Un tableau partiel de correspondance entre caractères envoyés et les commandes physiques est donné ci-dessous.

|  |  |
| --- | --- |
| **Caractère envoyé** | **Commande physique** |
| Dan0 or Dan1 | Déclarer la broche numérique n comme entrée (0) ou sortie (1) |
| Drn | Lire la valeur numérique (0 ou 1) sur la broche n (ASCII de 2 à b) |
| Dwn0 or Dwn1 | Écrire 1 ou 0 sur la broche n |
| An | Lire la broche analogique n |
| Wnm | Écrire la valeur analogique m (ASCII de 0 à 255) sur la broche n |
| Sa1 or Sa2 | Déclarer le servo 1 (broche numérique 9) ou 2 (broche numérique 10) |
| Sw1n or Sw2n | Déplacer le servo 1 ou le servo 2 en position n (d’ASCII(0) à ASCII(180)) |
| Sd1 or Sd2 | Détacher le servo 1 ou 2 |

Les valeurs n et m sont codées en ASCII. On donne ci-dessous un tableau de correspondance entre les caractères ASCII et le code décimal correspondant. Par exemple pour obtenir le caractère « 0 » la valeur décimale est 48.



Ainsi pour déclarer la LED, on envoie la commande "Dan1" avec n= 48+numéro de la broche Arduino sur laquelle est reliée la LED sous forme d’un caractère ASCII.

Quelle est la commande à envoyer pour déclarer la LED sur la broche 9 ?

Quelle est la commande pour allumer la LED sur la broche 9 ?

Quelle est la commande à envoyer pour déclarer la LED sur la broche 13 ?

Quelle est la commande à envoyer pour allumer la LED (sur la broche 13) ?

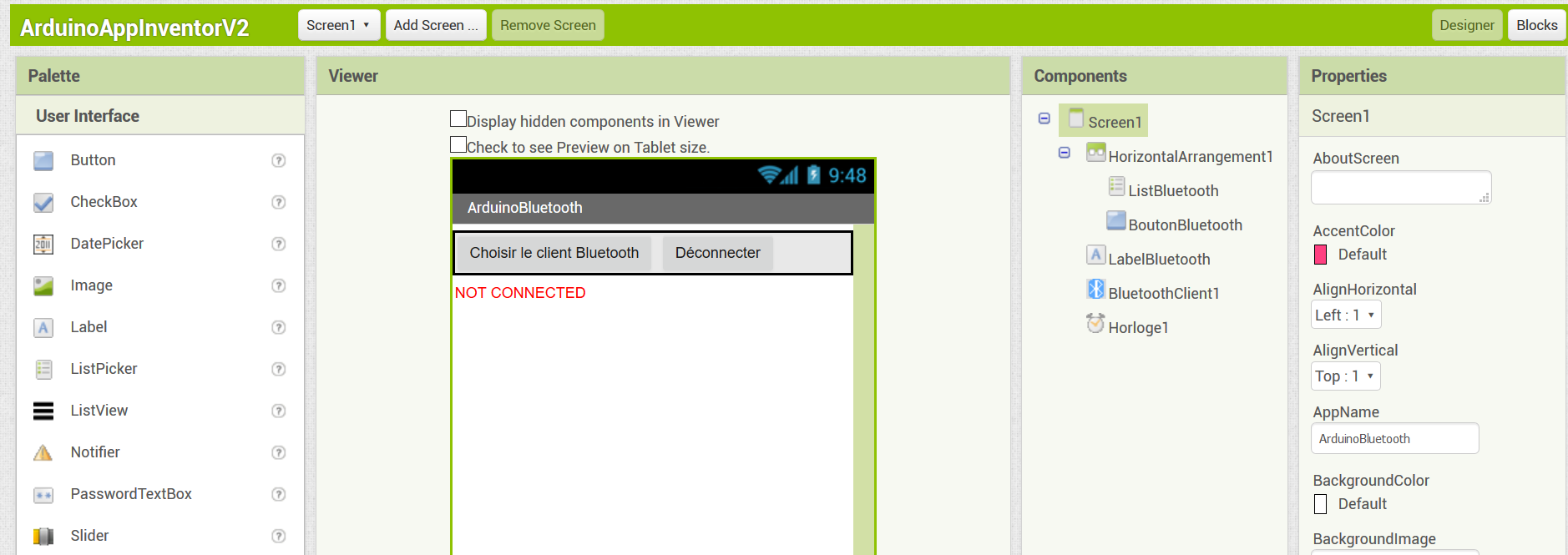
Quels sont les mots clefs à envoyer pour déclarer le servo moteur (sur la broche 9) et pour le faire tourner de 90° ?

*Indication aux collègues : la distinction entre les broches 9 et 13 est que pour la broche 9 le caractère sera facilement identifiable « 9 » tandis que pour la broche 13 le caractère sera « = ».*

# Étape : RÉaliser une interface graphique sous Android avec AppInventor

Se connecter au site : <http://ai2.appinventor.mit.edu/>. Il est alors possible d’ouvrir un projet existant ou en créer un nouveau.

*Indication aux collègues : il faut utiliser un compte Google existant ou créer un nouveau compte : à anticiper en créant les comptes nécessaires.*



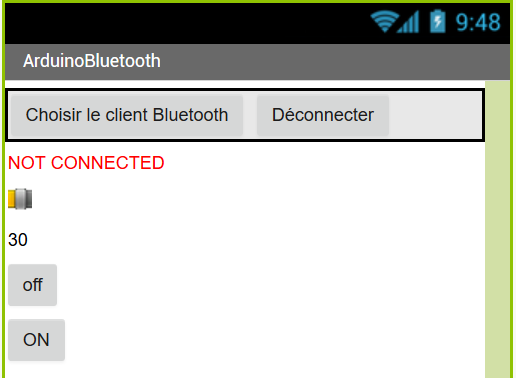
Les boutons (1) permettent de changer de mode :

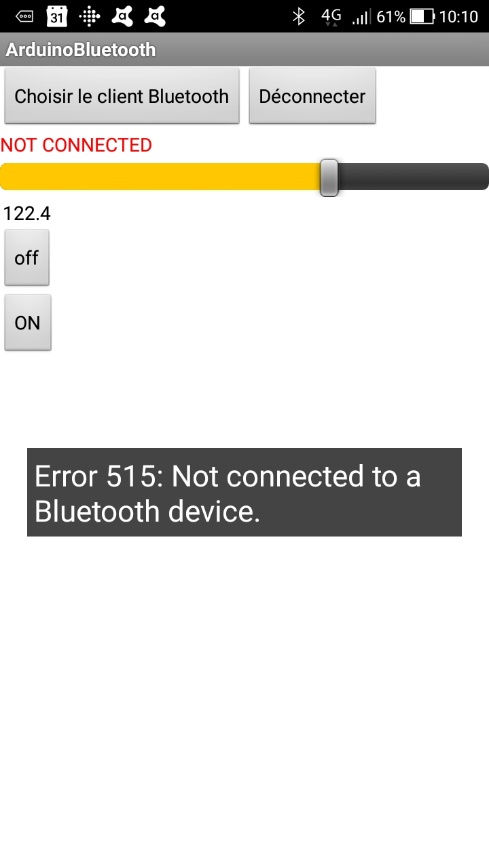
* le mode **Designer** pour réaliser ce que les utilisateurs verront sur leur smartphone ou tablette ;
* le mode **Block** pour réaliser la programmation graphique de l'application.

Pour réaliser la partie Design, il faut déplacer des éléments de la palette dans la zone Viewer (2) puis paramétrer cet élément dans la zone Properties (3).

**Importer** la base déjà préparée ArduinoAppInventor.aia en cliquant sur Projects / Import Project from my computer.

Notez que 5 éléments ont été ajoutés et configurés.

**Glisser** à partir de Layout un élément HorizontalArrangement et choisir dans Properties Width / Fill Parent pour que la zone remplisse toute la fenêtre horizontalement. Il est important d'utiliser ces éléments d'arrangement graphique pour que l'application ait la même allure, quel que soit le périphérique utilisé (tablette ou smartphone).

**Glisser** un Bouton de User Interface et dans Text de Properties taper LED OFF et mettre le Background dans une couleur donnée (verte).

**Renommer** aussi le bouton par ButtonLedOff.

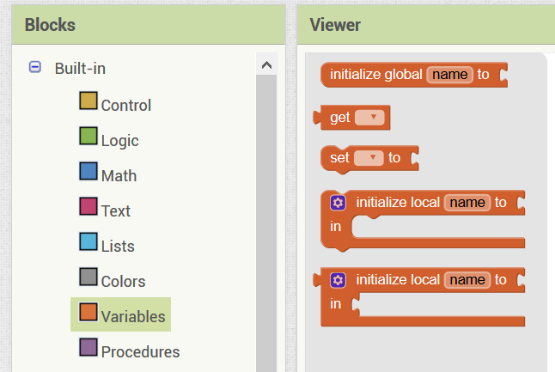
**Ajouter** un bouton ButtonLedOn et mettre une couleur blanche par défaut.

*Indication aux collègues : pour aller plus loin, demander d’ajouter le Slider permettant de piloter le moteur ou donner l’interface complétée.*

Figure 1 : Interface finalisée sur Appinventor et sur le téléphone

# Étape : Programmer les diffÉrentes interactions

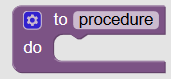
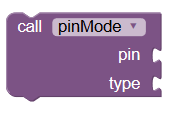
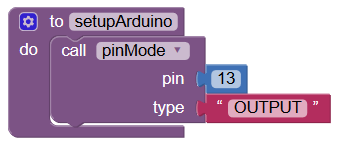
**Cliquer** maintenant sur Blocks pour accéder à la partie programmation graphique.

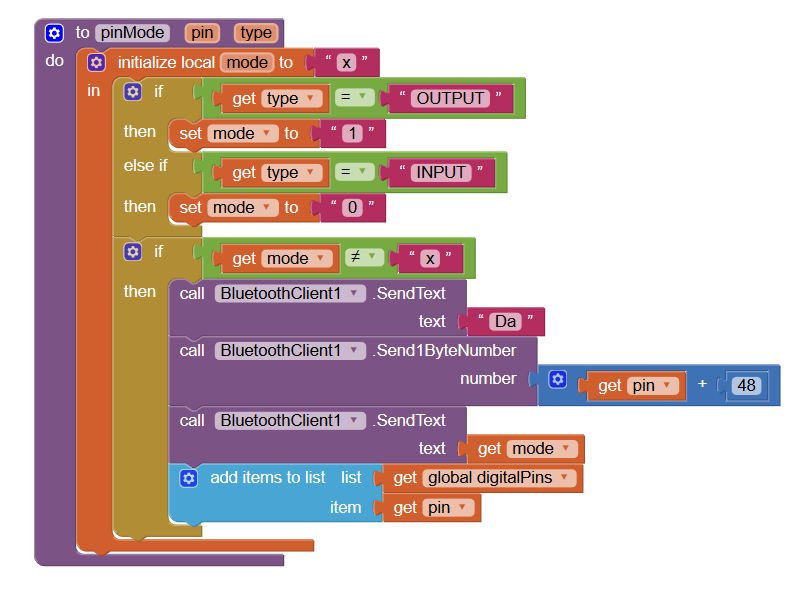
On retrouve à nouveau une zone de visualisation du code graphique et une zone Blocks pour choisir les méthodes, variables… pour programmer. Chaque bloc est exécuté « parallèlement » (programmation par événement). Les instructions s’emboîtent entre elles comme un puzzle.

On veut réaliser l'opération suivante pour allumer la LED :

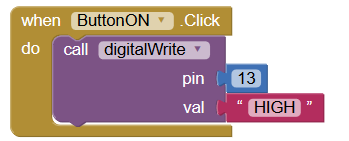
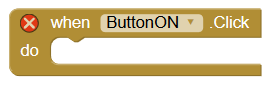
* si on clique sur le bouton ButtonLedON, alors la LED s'allume ;
* si on clique le bouton ButtonLedOFF, alors la LED s'éteint.

**1) Déclaration de la LED 13 en tant que sortie digitale**

Cliquer sur Procédures, faire glisser  dans le Viewer, rentrer « setupArduino2 » dans « procédure ». . Insérer un bloc  à l’intérieur. Choisir pinMode. Aller dans math et ajouter en modifiant le numéro de la broche. Aller dans Text et ajouter en renseignant OUTPUT dans la case. On obtient : 

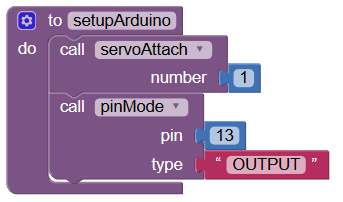
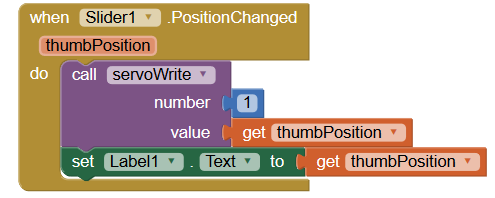
**Remarque :** cliquer droit sur expand de pinMode  : cela ouvre le détail de la fonction correspondante. On retrouve alors dans la partie du bas la création du mot clef vue précédemment DaX avec X numéro de la broche +48. Par la suite on utilisera directement les fonctions prédéfinies.

**2) Programmation des actions à effectuer**

**Cliquer** sur ButtonON et choisir . On insère ensuite les commandes ou méthodes appelées quand le bouton On sera cliqué. **Ajouter** alors les éléments permettant d’allumer la LED sur la broche 13.

**Ajouter** maintenant l’ensemble les éléments permettant de programme l’extinction de la LED.

**Ajouter** la déclaration du Servomoteur dans le setup.



**Ajouter** les actions devant se produire (rotation du moteur) lorsque l’utilisateur déplace le curseur.

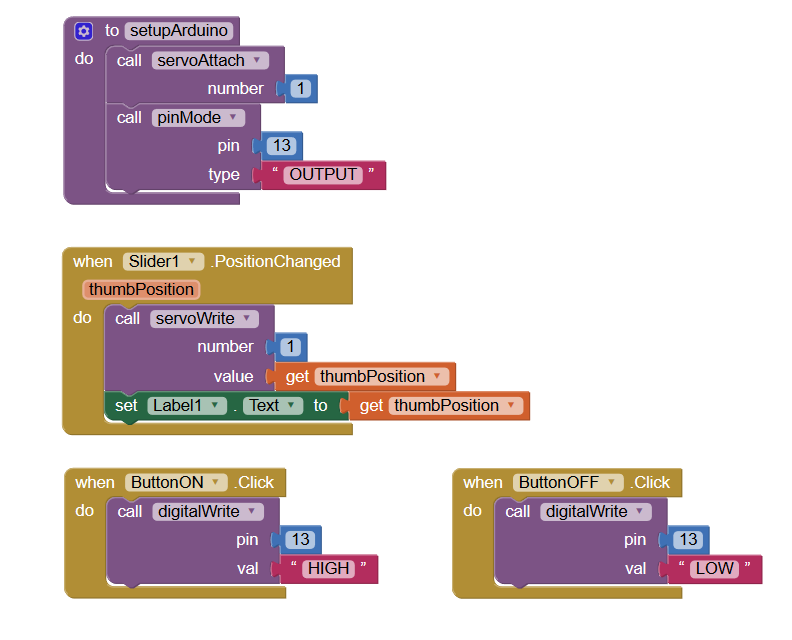
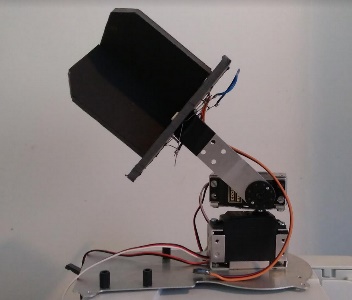


Figure 2 : Programmation par bloc obtenue

# Étape : Charger l'application dans la tablette ou smartphone.

**Remarque à destination des collègues :** Il y a plusieurs manières possibles de tester l'application créée. Le plus rapide est d'utiliser l' « AI companion ». Cependant cette méthode nécessite d’autoriser l’application à accéder aux données du téléphone, elle ne peut donc pas être utilisée sur les téléphones des élèves. Ci-dessous deux méthodes seront données, une sans utiliser l’application AI2 companion pour éviter ce problème et une avec l’application qui pourrait être utilisée par exemple dans une classe disposant de tablettes.

|  |  |
| --- | --- |
| ***Sans AI2 companion***  **Sur l’ordinateur, cliquer sur Construire / App (Enregistrer .apk sur mon ordinateur). Après génération de l'application un QR code apparaît à l’écran.**  **Il faut ensuite récupérer le fichier . apk sur le smartphone (envoi par mail, usb, solution de partage de fichier).**  **Cliquer sur le fichier. Apk**  **L’application n’exige aucun accès particulier, il faut alors cliquer sur installer.**  **Cliquer sur l’application**  **Appuyer sur Scanner**  **Sélectionner la carte**  **Cliquer sur "connecter"**  **Tester l’application** | ***Avec AI2 companion***  **Installer AI2 companion (cf. playstore – AI2 companion)**  **Sur l’ordinateur, cliquer sur Construire / App (Donner le code QR pour fichier .apk). Après génération de l'application un QR code apparaît à l’écran.**  **Sur le téléphone démarrer l’AI companion, cliquer sur « scan QR code », installer.**  **Cliquer sur l’application**  **Appuyer sur Scanner**  **Sélectionner la carte**  **Cliquer sur "connecter"**  **Tester l’application** |



# Pour aller plus loin

Remplacer le curseur par l’utilisation de l’information issue de l’accéléromètre du téléphone : lorsque l’on penche le téléphone, le moteur doit tourner.

Programmer le deuxième axe afin que le système puisse s’orienter suivant deux directions.

Figure 3 : Autre support possible

*Indication aux collègues : d’autres supports d’utilisation sont possibles pour remplacer la caméra, par exemple une parabole orientable ou un système 2 axes comme sur la figure ci-contre.*