

## PILOTAGE A DISTANCE DE LA ROTATION D'UNE CAMERA 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.
<b>Exemples d'activités</b>	
<ul style="list-style-type: none"> <li>• Réaliser une IHM pouvant piloter deux ou trois actionneurs et acquérir les données d'un ou deux capteurs.</li> <li>• Gérer des entrées/sorties à travers les ports utilisés par le système.</li> <li>• Utiliser un tableau de correspondance entre caractères envoyés ou reçus et commandes physiques</li> </ul>	

### 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 A DISTANCE DE LA ROTATION D'UNE CAMERA 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.

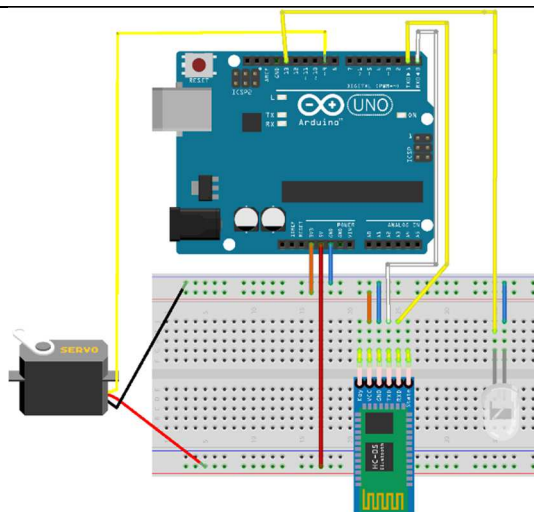
### ÉTAPE : ÉLÉMENTS EMPLOYÉS

*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 Bluetooth
- 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.*

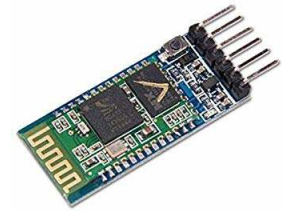
**Compléter le tableau** de comparaison suivant afin d'illustrer les intérêts et limites du Bluetooth.

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.

D	H	Nom	Car	D	H	Car	D	H	Car	D	H	Car	D	H	Car	D	H	Car	D	H	Car	D	H	Car
0	00	NUL		32	20		64	40	0	96	60	*	128	80	ç	160	A0	à	192	C0	ˆ	224	E0	ó
1	01	SOH	0	33	21	!	65	41	A	97	61	a	129	81	ü	161	A1	í	193	C1	±	225	E1	þ
2	02	STX	0	34	22	"	66	42	B	98	62	b	130	82	é	162	A2	ó	194	C2	˜	226	E2	ø
3	03	ETX	0	35	23	#	67	43	C	99	63	c	131	83	ä	163	A3	ü	195	C3	˘	227	E3	ù
4	04	EOT	0	36	24	\$	68	44	D	100	64	d	132	84	å	164	A4	ñ	196	C4	—	228	E4	z
5	05	ENQ	0	37	25	%	69	45	E	101	65	e	133	85	à	165	A5	ñ	197	C5	+	229	E5	ø
6	06	ACK	0	38	26	&	70	46	F	102	66	f	134	86	ä	166	A6	ë	198	C6	ä	230	E6	µ
7	07	BEL	0	39	27	'	71	47	G	103	67	g	135	87	ç	167	A7	ø	199	C7	å	231	E7	þ
8	08	BS	0	40	28	(	72	48	H	104	68	h	136	88	é	168	A8	í	200	C8	ˆ	232	E8	þ
9	09	HT	0	41	29	)	73	49	I	105	69	i	137	89	ë	169	A9	ø	201	C9	˜	233	E9	ü
10	0A	LF	0	42	2A	*	74	4A	J	106	6A	j	138	8A	è	170	AA	˘	202	CA	ˆ	234	EA	ü
11	0B	UT	0	43	2B	+	75	4B	K	107	6B	k	139	8B	ı	171	AB	½	203	CB	˜	235	EB	ü
12	0C	FF	0	44	2C	,	76	4C	L	108	6C	l	140	8C	ı	172	AC	¾	204	CC	˜	236	EC	ý
13	0D	CR	0	45	2D	-	77	4D	M	109	6D	m	141	8D	ı	173	AD	ı	205	CD	=	237	ED	ÿ
14	0E	SO	0	46	2E	.	78	4E	N	110	6E	n	142	8E	ä	174	AE	ˆ	206	CE	˜	238	EE	ˆ
15	0F	SI	0	47	2F	/	79	4F	O	111	6F	o	143	8F	ä	175	AF	˜	207	CF	˜	239	EF	ˆ
16	10	DLE	0	48	30	0	80	50	P	112	70	p	144	90	é	176	B0	˜	208	D0	\$	240	F0	-
17	11	DC1	0	49	31	1	81	51	Q	113	71	q	145	91	ˆ	177	B1	˜	209	D1	0	241	F1	ˆ
18	12	DC2	0	50	32	2	82	52	R	114	72	r	146	92	ˆ	178	B2	˜	210	D2	é	242	F2	=
19	13	DC3	0	51	33	3	83	53	S	115	73	s	147	93	ˆ	179	B3	ı	211	D3	ˆ	243	F3	¾
20	14	DC4	0	52	34	4	84	54	T	116	74	t	148	94	ˆ	180	B4	ı	212	D4	ˆ	244	F4	¾
21	15	NAK	0	53	35	5	85	55	U	117	75	u	149	95	ˆ	181	B5	ä	213	D5	ˆ	245	F5	¾
22	16	SYN	0	54	36	6	86	56	V	118	76	v	150	96	ü	182	B6	ä	214	D6	ı	246	F6	ˆ
23	17	ETB	0	55	37	7	87	57	W	119	77	w	151	97	ü	183	B7	ä	215	D7	ı	247	F7	ˆ
24	18	CAN	0	56	38	8	88	58	X	120	78	x	152	98	ü	184	B8	0	216	D8	ı	248	F8	ˆ
25	19	EM	0	57	39	9	89	59	Y	121	79	y	153	99	ü	185	B9	ı	217	D9	ˆ	249	F9	ˆ
26	1A	SUB	0	58	3A	:	90	5A	Z	122	7A	z	154	9A	ü	186	BA	ı	218	DA	ˆ	250	FA	ˆ
27	1B	ESC	0	59	3B	;	91	5B	[	123	7B	[	155	9B	ˆ	187	BB	ı	219	DB	ˆ	251	FB	ˆ
28	1C	FS	0	60	3C	<	92	5C	\	124	7C	\	156	9C	ˆ	188	BC	ˆ	220	DC	ˆ	252	FC	ˆ
29	1D	GS	0	61	3D	=	93	5D	]	125	7D	]	157	9D	ˆ	189	BD	ˆ	221	DD	ˆ	253	FD	ˆ
30	1E	RS	0	62	3E	>	94	5E	^	126	7E	^	158	9E	ˆ	190	BE	ˆ	222	DE	ı	254	FE	ˆ
31	1F	US	0	63	3F	?	95	5F	_	127	7F	_	159	9F	ˆ	191	BF	ˆ	223	DF	ˆ	255	FF	ˆ

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) ?

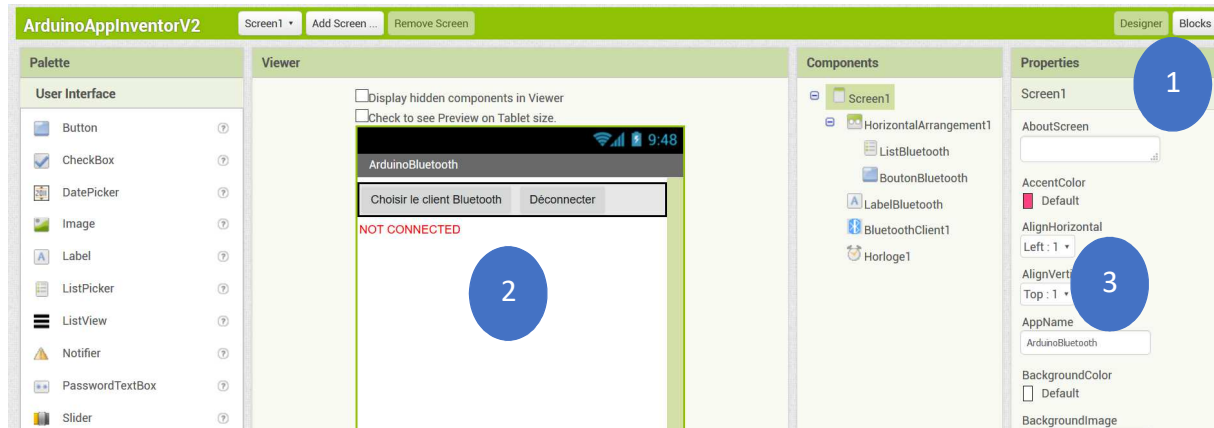
Quels sont les mots clefs à envoyer pour déclarer le servo moteur (sur la broche 9) et pour le faire tourner de 90° ?

*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 remplit 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.*

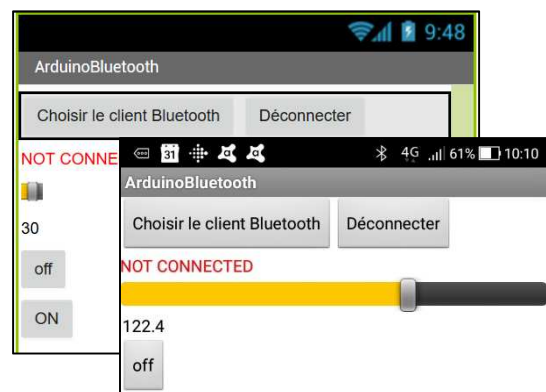
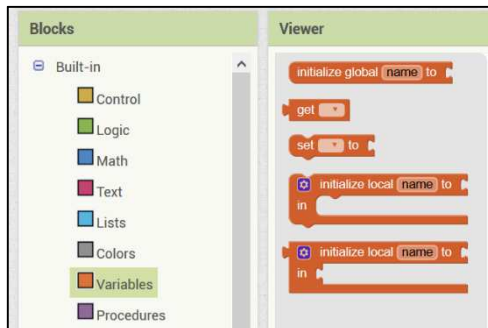


Figure 1 : Interface finalisée sur Appinventor et sur le téléphone

## ÉTAPE : PROGRAMMER LES DIFFÉRENTES INTERACTIONS

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



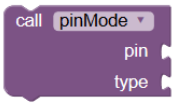
On retrouve à nouveau une zone de visualisation du code graphique et une zone Blocks pour choisir les méthodes, variables... pour programmer. Chaque bloc est exécuté « parallèlement » (programmation par événement). Les instructions s'emboîtent entre elles comme un puzzle.

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

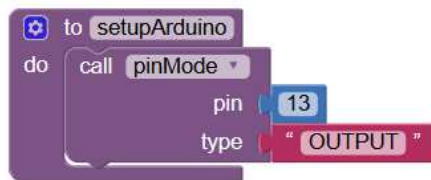
- si on clique sur le bouton ButtonLedON, alors la LED s'allume ;
- si on clique le bouton ButtonLedOFF, alors la LED s'éteint.

### 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

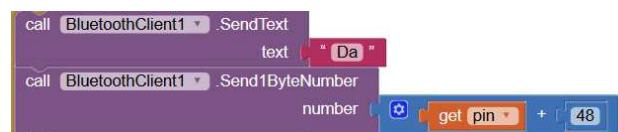
math et ajouter  en modifiant le numéro de la broche. Aller dans Text et ajouter  en renseignant



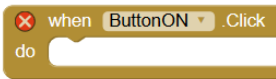
OUTPUT dans la case. On obtient :

**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.

to pinMode pin type do ...



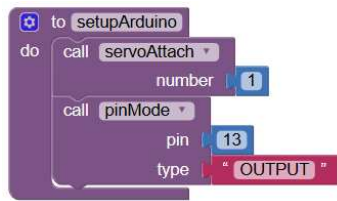
### 2) Programmation des actions à effectuer

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 programmer 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.

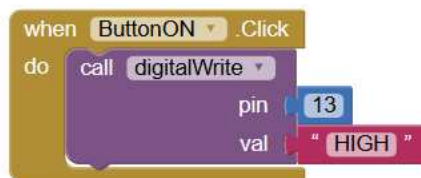
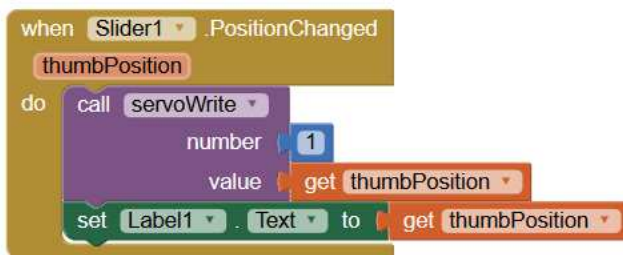
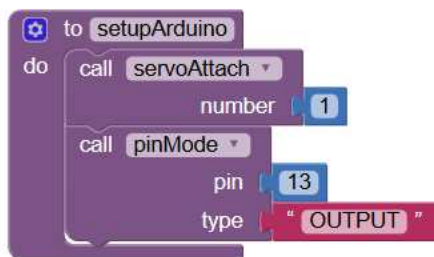
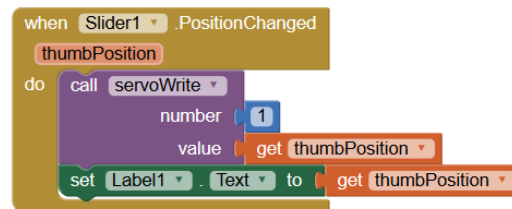


Figure 2 : Programmation par bloc obtenue



## 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.

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



MIT AI2 Companion  
MIT Center for Mobile Computing

## 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.

*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.*



Figure 3 : Autre support possible