

LES OBJETS COMMUNICANTS

Je découvre

Depuis longtemps, les Hommes ont conçu des **objets techniques** dont la fonction d'usage est de leur permettre de communiquer, c'est-à-dire d'échanger des informations entre eux.

Observer et analyser les situations illustrées ci-dessous :

Situation 1 :



Deux personnes discutent par téléphone.

Communication établie entre :

Moyen de communication :

Nature de l'objet :

Situation 2 :



Chaussures permettant de mesurer la distance parcourue, la vitesse de course, les calories brûlées, et d'envoyer ces informations sur l'écran d'un iPod.

Communication établie entre :

Moyen de communication :

Nature de l'objet :

Situation 3 :



Les caméras pour surveiller son domicile avec son smartphone.

Communication établie entre :

Moyen de communication :

Nature de l'objet :

Les besoins des Hommes évoluent : on attend désormais des objets qu'ils nous aident d'eux-mêmes à mieux vivre dans notre environnement et qu'ils interagissent sans intervention humaine avec intelligence et autonomie.

→ De la communication entre humains à la communication entre objets.



Un objet technique **communicant** est un objet capable d'interagir en autonomie avec son environnement et d'échanger de lui-même (sans intervention humaine) des informations avec des utilisateurs ou d'autres objets.

LES OBJETS COMMUNICANTS

J'applique

On distingue 2 familles : les objets communicants non connectés et les objets communicants connectés (ou simplement « objet connecté »).

Exercice 1 : Nommer chacun de ces objets communicants non connectés ci-contre.

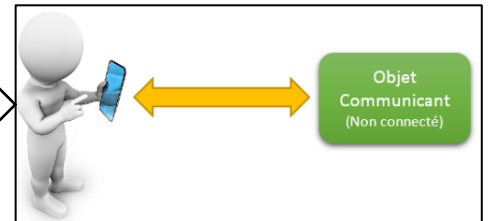


.....

.....

.....

→ Dans ce type d'objet communicant....., l'échange d'informations ne s'effectue qu'avec propriétaire de l'objet via un (tablette, smartphone, ordinateur...) munie d'une application/d'un logiciel.



Exercice 2 : Nommer chacun de ces objets communicants connectés ci-contre.

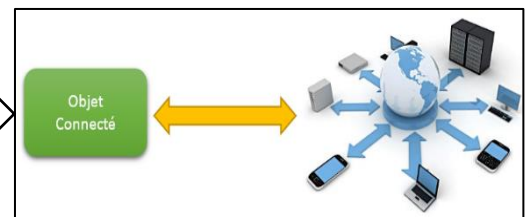


.....

.....

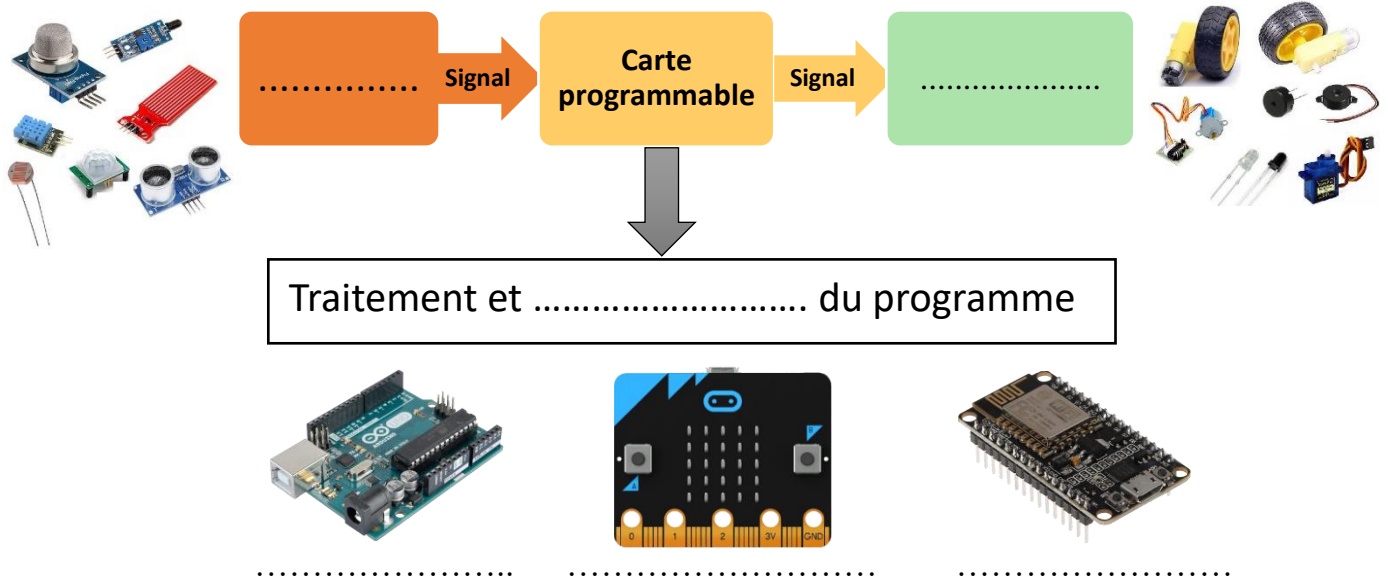
.....

→ Dans ce type d'objet communicant....., l'échange d'informations s'effectue par connexion à un à grande échelle (en général Internet) avec d'autres, des ou des serveurs de stockage et de traitement à distance.



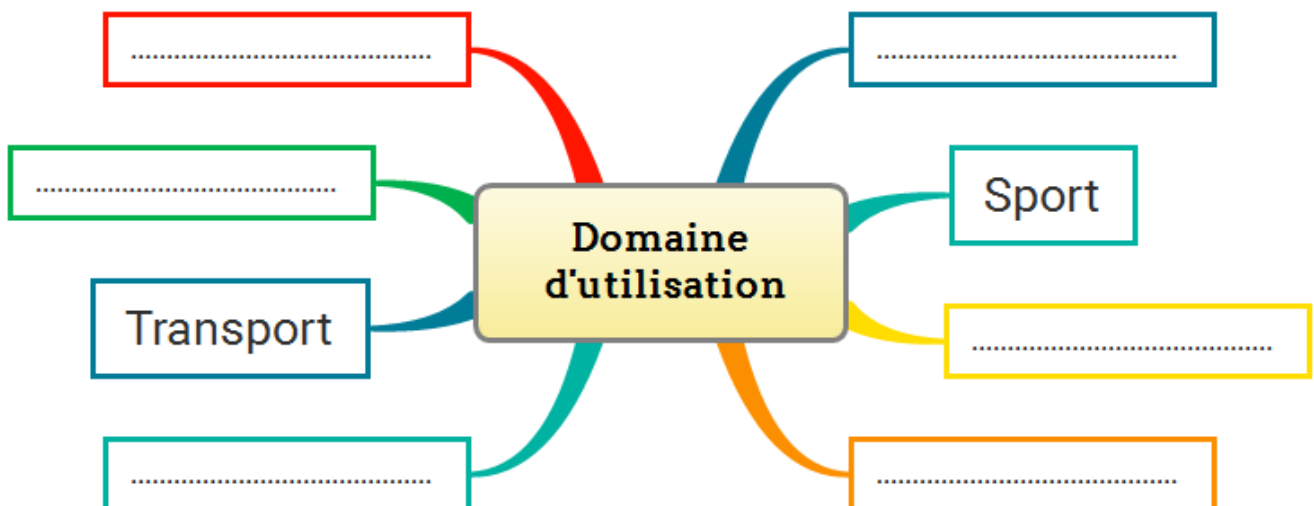
LES OBJETS COMMUNICANTS

Exercice 3 : Observer le fonctionnement des projets créés au sein du club Robotique de l'année dernière, puis compléter la structure technique des objets communicants et des objets connectés.



- **Un capteur :**
-
- **Un actionneur :**
-
- **Technologie de communication :**
-

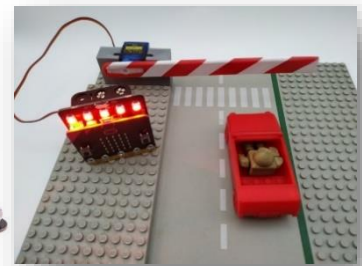
Exercice 4 : Compléter la carte mentale ci-contre par des exemples de domaine d'application des objets connectés.



PROJET

A partir du matériel disponible dans le laboratoire, créer un **objet communicant** qui répond à un besoin de fonction d'usage technique dont l'objectif est de nous faciliter la vie quotidienne (économie d'énergie, portail automatique, caméra de sécurité, ...), pour cela vous devez :

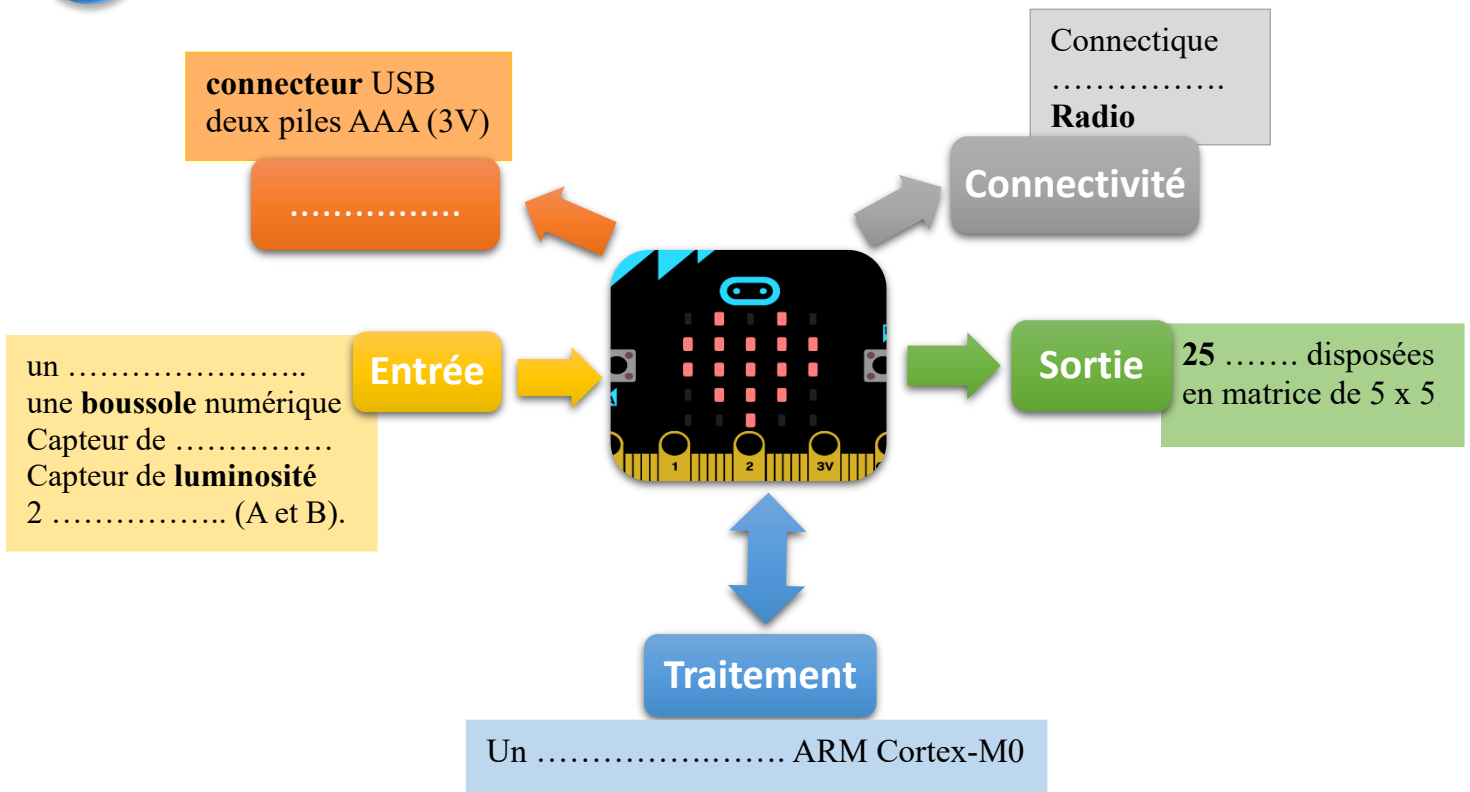
- Programmer la carte de votre objet communicant.
- Développer une application mobile permettant de piloter l'objet à distance.



- ✓ Vous pouvez utiliser d'autres ressources pour la conception de votre objet (carton, plexiglass, colle, ...).
- ✓ Trouver le logiciel adéquat pour programmer la carte Micro bit.
- ✓ Développer une application mobile qui permet de piloter l'objet.
 - Créer l'interface de l'application contenant des titres et des boutons pour déclencher **l'action** avec un design bien structuré.
 - Programmer l'application
- ✓ Les programmes doivent s'écrire d'une manière précise : (Entrée, traitement, sortie)
 - Entrée → Evènement : clic sur des boutons, saisie d'un texte, parler...
 - Traitement → Ensemble d'instructions ordonnées
 - Sortie → Affichage du résultat
- ✓ Soyer logique et méthodique pour résoudre les problèmes rencontrés.
- ✓ Enregistrer votre projet sous le nom « **projet_ votre prénom** ».
- ✓ Enregistrer votre application sous le nom « **app_ votre prénom** ».

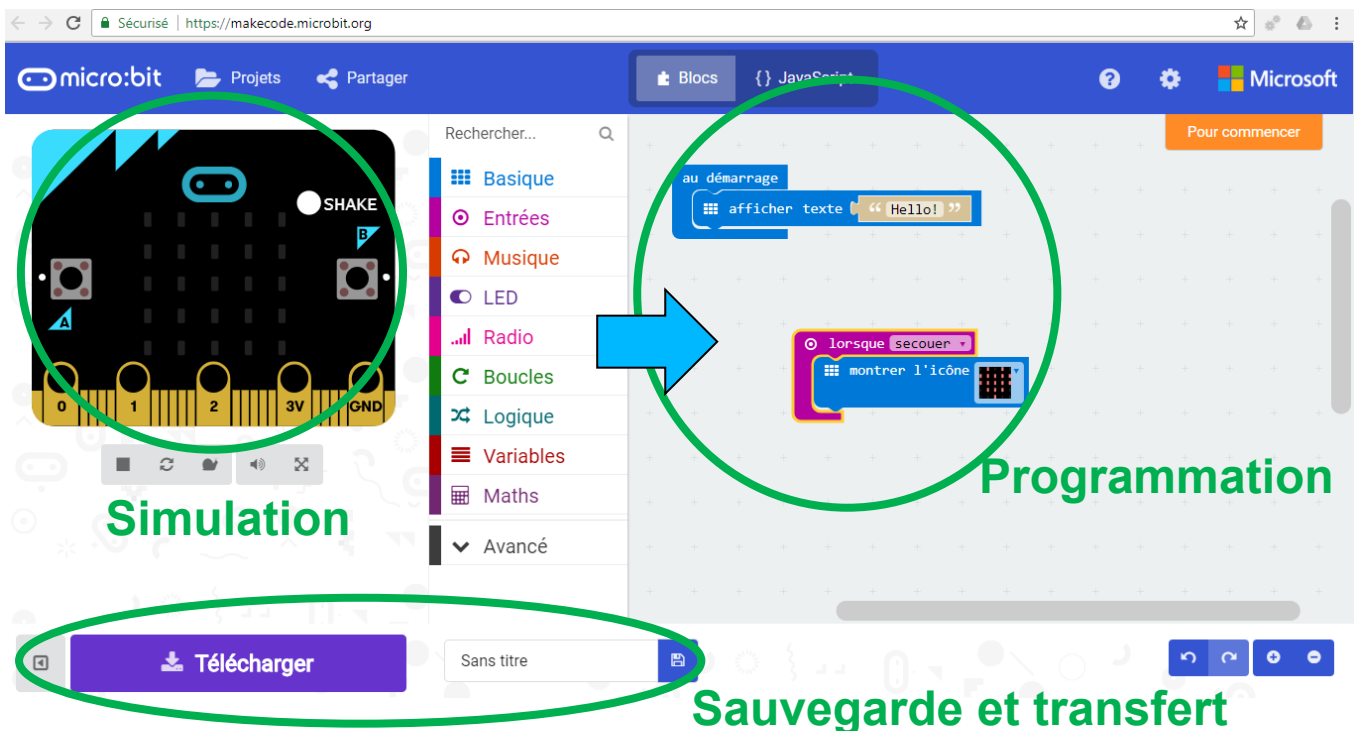
PROGRAMMATION D'OBJET

Je découvre



Rq : Branchement à l'aide de cinq anneaux et 20 broches de connexion.

Pour programmer la carte, on peut utiliser l'éditeur : Microsoft MakeCode



J'applique

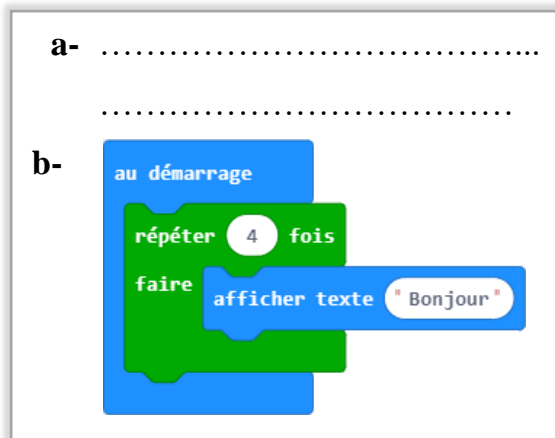
Exercice 1 : Afficher un texte

- 1) La 1^{ère} fonction tout simple de la carte *Micro:bit* consiste à afficher un texte qui va défiler lettre par lettre sur l'écran. Afficher le texte "**votre prénom**".
- 2) Le texte de l'activité précédente ne s'est affiché qu'une seule fois, mais il est possible de reproduire cet affichage avec une instruction qui permet de répéter **plusieurs fois** une action.

a- Afficher le texte **bonjour** indéfiniment

b- Quel est le rôle du 2^{ème} exemple

.....



Exercice 2 : Afficher un dessin

Il est possible d'afficher des **icônes** sur l'écran constitué d'une matrice 5 x 5 LEDs grâce à l'instruction du menu de Base. Contrairement au texte l'icône reste affichée à l'écran.

Créer le programme qui permet de :

Au démarrage de l'écran afficher 3 fois :

- a. Le nombre 9
- b. L'icône " tortue"
- c. Effacer l'écran

.....

Exercice 3 : Utiliser les boutons

La carte Micro:bit dispose de deux boutons A et B permettant de déclencher des actions. Les instructions concernant ces deux boutons se trouvent dans le menu **Entrées**

- a. Afficher votre prénom lorsqu'on appuie sur A.
- b. Afficher votre nom lorsqu'on appuie sur B.
- c. Afficher nom & prénom si on appuie sur A et B.

c.

Exercice 4 : Secouer la carte

La carte *Micro:bit* dispose d'un accéléromètre qui offre plusieurs possibilités. Décrire ce programme :

.....

.....

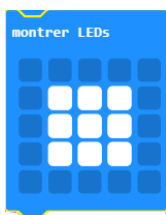
.....



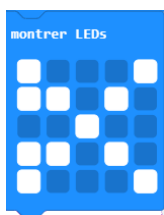
Exercice 5 : Créer le jeu Pierre-ciseaux-papier

Lorsque la carte est secouée, l'un des trois choix (pierre, ciseaux, papier) est sélectionné au hasard. Il est donc nécessaire de définir une variable qui sera un nombre compris entre 0 et 2, ce qui donnera 3 choix possibles. Exemple :

- Le 0 pour la pierre
- Le 1 pour les ciseaux
- Le 2 pour le papier



La pierre



Les ciseaux



Le papier

.....

.....

.....

.....

.....

.....

.....

.....

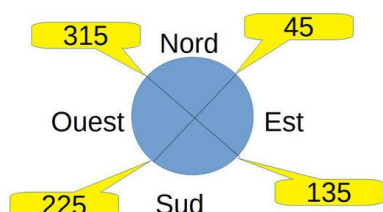
.....

Exercice 6 : Créer une boussole

La carte *Micro:bit* possède un capteur magnétomètre détectant le champ magnétique terrestre, elle peut afficher un nombre compris entre 0 et 359 pour représenter tous les degrés d'une boussole.

Créer un programme pour afficher la direction de la carte.

- Commencer par calibrer la boussole
- Toujours, détecter la direction de la boussole
 - ✓ Si le degré < 45 alors afficher le texte "N"
 - ✓ Sinon Si le degré < 135 alors afficher le texte "E"
 - ✓ Sinon Si le degré < 225 alors afficher le texte "S"
 - ✓ Sinon Si le degré < 315 alors afficher le texte "O"
 - ✓ Sinon afficher le texte "N"



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



Je m'évalue

Lancer l'éditeur MakeCode et créer un nouveau projet "eval_nom" dans D:\9^{ème}\9B...

- 1) Afficher au démarrage de l'écran le texte "INFORMATIQUE".
- 2) Afficher (4 fois) :
 - a. Le nombre 8 pendant 1s
 - b. L'icône " diamant
 - c. Effacer l'écran

.....

.....

.....

.....

.....

.....

.....

- 3) Que fait ce programme ?

.....

.....

.....

.....

.....

```

lorsque le bouton A + B est pressé
  répéter 2 fois
    faire
      afficher texte " Petit "
      montrer l'icône [diamant]
  efface l'écran
    
```

- 4) Ecrire le programme qui permet de : Lorsqu'on secoue la carte on affiche un nombre au hasard de 1 à 6.

.....

.....

.....

- 5) Compléter ce programme qui permet de simuler un chronomètre
 - Lorsque le bouton A est pressé, afficher les chiffres de 0 à 9
 - Chaque chiffre reste affiché pendant 1s

```

lorsque le bouton A est pressé
  définir x à ...
  répéter ... fois
    faire
      ...
  changer x par ...
    
```


DEVELOPPEMENT D'APPLICATION MOBILE



C'est un logiciel de création d'application Android. Il fonctionne par l'assemblage de

En ligne

- Lancer un navigateur (On ne peut pas utiliser le navigateur Internet explorer)
- Saisir l'adresse suivante: "ai2.appinventor.mit.edu"
- Se connecter avec votre adresse

Hors ligne

- Cliquer sur "App Inventor 2 Ultimate"
- Cliquer sur "Ai2 Starter"
- Réduire les commandes d'invites qui s'affichent
- lancer le navigateur et saisir l'adresse suivante : "localhost:8888"

Il y a trois possibilités de transmission de programme du PC au smartphone, par :

Wifi	Emulateur	Câble USB
<p>MIT AI2 Companion</p> <p>! Connecter le smartphone et le PC sur le <u>même réseau Wifi</u>.</p>	<p>ai2 Starter</p> <p>Lancer et laisser la fenêtre qui apparaît ouverte en barre des tâches.</p>	<p>Installer sur le PC le programme que vous trouverez sur cette page https://appinventor.mit.edu/explore/ai2/set-up-device-usb.html</p> <p>Sur votre smartphone, aller dans les propriétés puis Options pour les développeurs puis cocher Débogage USB.</p>

Je découvre

Exemple : Ouverture d'une application existante

- 1) Importer le projet dessin.aia se trouvant dans l'emplacement D:\9^{ème}
- 2) Tester l'application avec l'émulateur.
- 3) Décrire l'objectif de cette application :
- 4) Identifier les différents composants de cette application :
.....
.....
- 5) Ouvrir le code de cette application

```

quand Cadre1 .Glissé
  X début Y début X précédent Y précédent Xactuel Yactuel draggedAnySprite
  faire
    appeler Cadre1 .Dessiner ligne
      x1 obtenir X précédent
      y1 obtenir Y précédent
      x2 obtenir Xactuel
      y2 obtenir Yactuel
quand Rouge .Clic
  faire mettre Cadre1 . Couleur de dessin à
quand Vert .Clic
  faire mettre Cadre1 . Couleur de dessin à
quand Effacer .Clic
  faire appeler Cadre1 .Supprimer
    
```

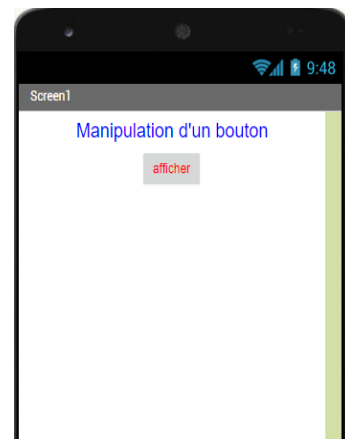
- 6) Ajouter un autre bouton pour dessiner avec une autre couleur.
- 7) Tester l'application avec l'émulateur.

J'applique

Exercice 1 : clic de bouton

Faire afficher une image en cliquant sur le bouton afficher

1. Créer l'interface utilisateur content un titre et un bouton



2. Coder le bouton « afficher » de telle sorte lorsqu'on clique dessus, l'image apparait à l'écran

.....

.....

Exercice 2 : Jouer le son

Faire une application qui fait des bruits d'animaux en appuyant sur des boutons.

- 1) Créer l'interface utilisateur contenant :
 - a. Un titre
 - b. Un arrangement dans lequel vous allez insérer quatre boutons
 - c. Quatre sons (non visibles)
- 2) Récupérer les images et les fichiers sons depuis D:\9^{ème}\app2.
- 3) Changer le fond de chaque bouton pour que ce soit les images des animaux.
- 4) Coder chaque bouton de telle sorte lorsqu'on clique dessus, on entend le son adéquat.



Bloc du bouton Canard :

.....

Exercice 3 : Jeu de lampe

1. Faire allumer une lampe en cliquant sur le bouton ON et la faire éteindre en cliquant sur OFF.

Interface	Code	Aide
	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Fermée</p> <div style="display: flex; justify-content: center; gap: 10px;"> OFF ON </div> <p>Quand le bouton off est cliqué :</p> <ul style="list-style-type: none"> - La lampe s'éteint - un message affiche « Fermée » </div> <div style="text-align: center;"> <p>Allumée</p> <div style="display: flex; justify-content: center; gap: 10px;"> OFF ON </div> <p>Quand le bouton on est cliqué :</p> <ul style="list-style-type: none"> -La lampe s'allume - un message affiche « Allumée » </div> </div>	<pre> quand ON .Clic faux vrai faire "Allumée" mettre lampe_ouverte . Visible à mettre message . Texte à </pre>

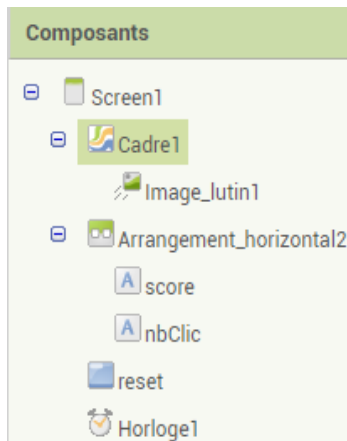
2. Tester l'application avec l'émulateur.

Exercice 4 : Catch Me

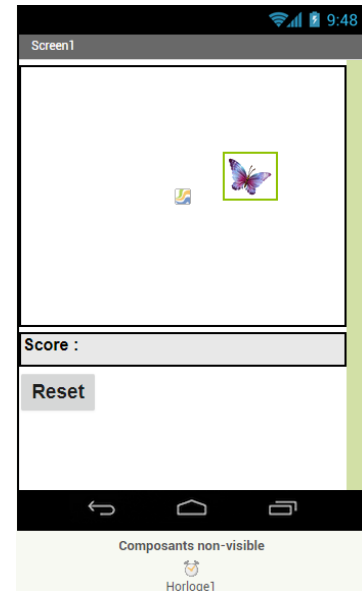
Réaliser une application dont le principe est de gagner un maximum de point en touchant une **image_lutin** qui bouge aléatoirement sur l'écran.

Créer l'interface utilisateur contenant :

- 1) Créer l'interface utilisateur contenant les composants :

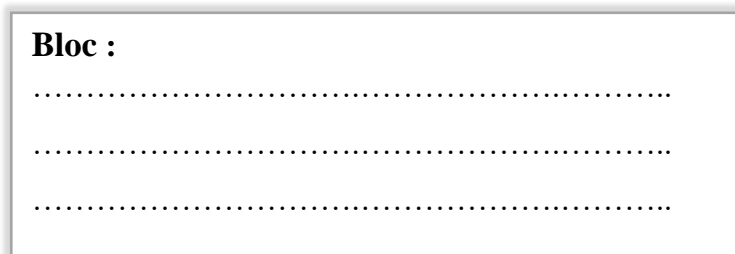


Votre application devrait ressembler à cela :

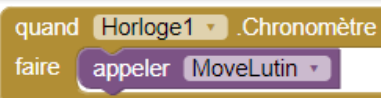
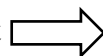


- 2) Créer la procédure "**MoveLutin**" qui permet de bouger le lutin aléatoirement sur votre écran.

Rq : On fixe (met) les positions X et Y de l'image_lutin à des nombres aléatoires compris entre 0 et la taille du cadre.

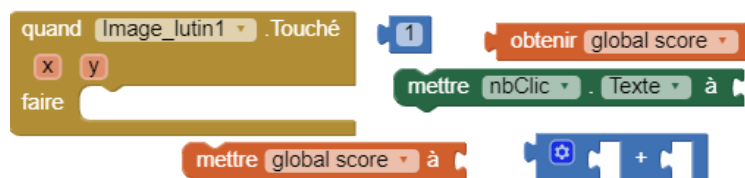


- 3) Mettre le lutin en mouvement indéfiniment



- 4) Créer une variable nommée "**score**", l'initialiser à 0, puis programmer l'image_lutin de telle sorte à chaque clic dessus le score s'incrémente d'un point.

! Vous pouvez utiliser ces blocs :



- 5) Programmer le bouton **Reset** de telle sorte lorsqu'on clique dessus, le score sera mis à zéro.

! Deux instructions nécessaires.

- 6) Tester l'application avec l'émulateur.

Exercice 5 : Deviner un nombre secret

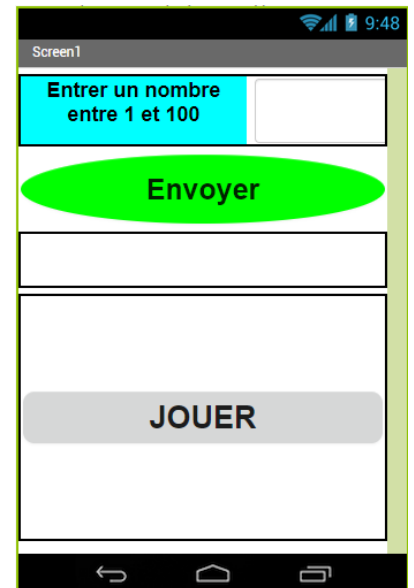
Réaliser une application dont le principe est de deviner un nombre choisis par l'ordinateur d'une manière aléatoire.

1) Créer l'interface utilisateur ayant comme design :

2) Coder les deux boutons « Envoyer » et « Jouer » en utilisant les deux événements :



3) Remettre dans l'ordre les instructions ci-contre, afin de réaliser le programme qui permet de deviner un nombre :



.....

.....

.....

.....

.....

.....

.....

.....

4) Créer le fichier APK et tester l'application sur votre smartphone.

PROJET 1 : Ne touche pas mon téléphone

Objectif : Créer une alarme qui se déclenche lorsqu'on approche la main du téléphone.

Vous aurez besoin de :

- Un bouton pour activer le capteur.
- Un capteur qui détecte la présence d'objets.
- Un lecteur qui contient le son Alarme.Wav.
✍ Vous trouvez l'image et le son sous D:\ 9ème\ Alarme



Bloc :

.....

.....

.....

.....

PROJET 2 : Piloter une lampe

Objectif : Piloter une lampe en prononçant les ordres "allumer" et "éteindre". Cette application simule l'état de la lampe en **interchangeant l'image** correspondante et un **message vocal** « lampe allumée » ou « lampe éteinte » se déclenche.

1) Créer l'interface contenant :

- Deux arrangements
- Deux images (lampe - éteinte / allumée)
- Bouton ayant comme image de fond l'image de micro
- Reconnaissance vocale
- Texte à parole
- Extension TaifunFlashlight

✍ Vous trouvez l'image et l'extension dans D:\ 9ème\ Lampe

2) Ajouter une option à votre application de telle sorte la torche de votre Smartphone s'allume en même temps que la lampe.

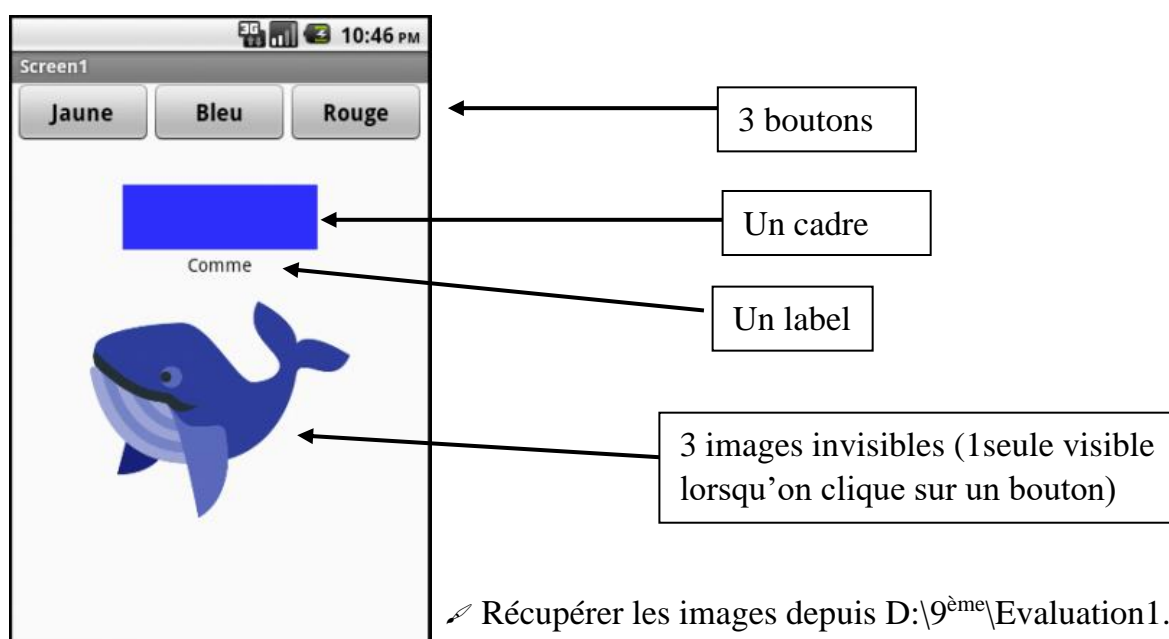


/20

EVALUATION 1 :

Objectif : Réaliser une application nommée « **devoir_prénom_nom** » dont le principe est d'apprendre les couleurs aux bébés.

- 1) Connectez-vous en local à App Inventor avec l'adresse email **9B..G...@example.com**
- 2) Créer l'interface ci-contre :



- 3) Programmer : Coder chaque bouton de telle sorte lorsqu'on clique dessus :
 - a. Le composant cadre prend la couleur de fond correspondante.
 - b. L'image correspondante s'affiche à l'écran.
 - c. Ajouter le composant **texte à parole** pour parler la couleur correspondante.

! Tester l'application avec l'émulateur.

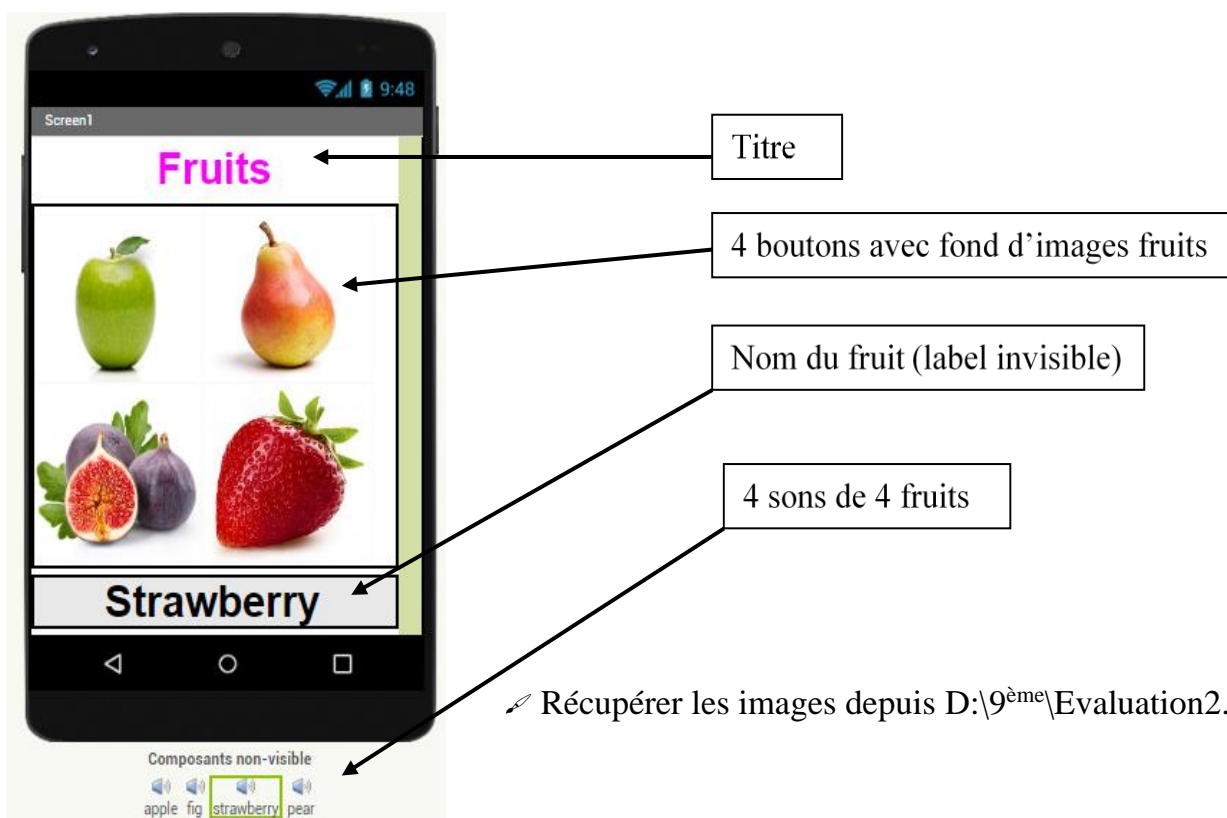
	Lancement	Design					Blocs
		Cadre et label	Création des boutons	Insertion des images	Mise en forme	Nom des composants	
Barème	1	2	3	3	2	2	7
Note							

/20

EVALUATION 2 :

Objectif : Ton frère n'arrive pas à apprendre les noms des fruits en anglais, pour l'aider créer une application mobile nommée « **devoir_prenom_nom** » qui contient les images et les noms de quelques fruits.

- 1) Connectez-vous en local à App Inventor avec l'adresse email **9B..G...@example.com**
- 2) Créer l'interface ci-contre :



- 3) **Programmer :** Coder chaque bouton de telle sorte lorsqu'on clique dessus :
 - a. On entend le son correspondant au fruit
 - b. Le nom du fruit correspondant apparait en-dessous de l'écran
- ! Tester l'application avec l'émulateur.

	Lancement	Design					Blocs
		Création des labels	Création des boutons	Insertion des images	Insertion des sons	Nom des composants	
Barème	1	3	4	2	2	2	6
Note							

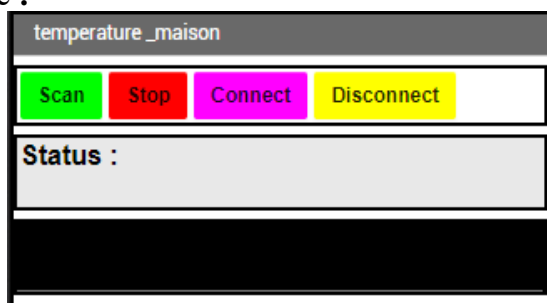
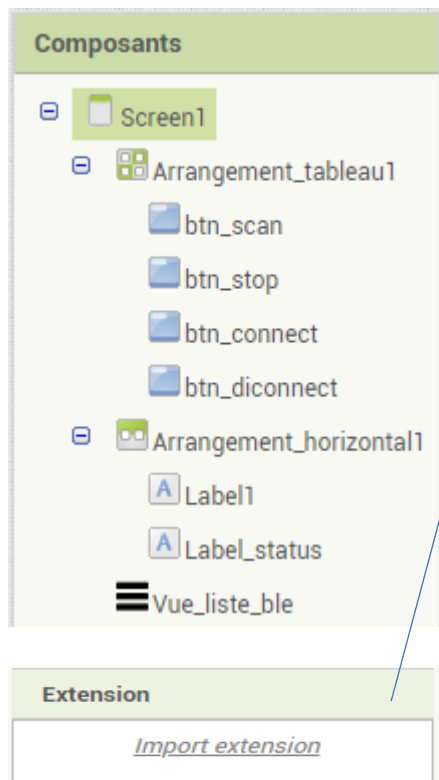
EXEMPLE 1

Configuration de l'application mobile pour rechercher les appareils Bluetooth disponibles

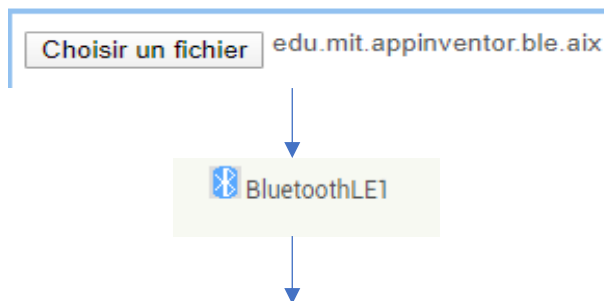
Objectif : Détecter le Bluetooth disponible à proximité (carte micro:bit).

ETAPE 1 : La création de l'interface de pilotage

- 1) Lancer app inventor, puis créer l'application suivante :
- 2) Ajouter les composants nécessaires :



Remarque : On doit importer l'extension Bluetooth pour pouvoir communiquer donc il suffit de suivre les étapes suivantes :



Le composant BluetoothLE1 sera ajouté aux composants de l'application

- 3) Programmer les boutons

- **Le bouton scan :**

Pour configurer le composant Bluetooth pour démarrer la numérisation, et changer l'étiquette d'état (label_status) et rendre la vue_liste_ble visible qui va contenir l'ensemble des appareils bluetooth.

```

quand btn_scan .Clic
faire
  appeler BluetoothLE1 .StartScanning
  mettre Label_status .Texte à " Scan "
  mettre Vue_liste_ble .Visible à vrai
  
```

- **Le bouton stop :** arrêter l'analyse de l'application et changer le libellé d'état lorsqu'on appui sur Bouton Stop.

```

quand btn_stop .Clic
faire
  appeler BluetoothLE1 .StopScanning
  mettre Label_status .Texte à " Stop scan "

quand BluetoothLE1 .DeviceFound
faire
  mettre Vue_liste_ble .Éléments de la chaîne à BluetoothLE1 .DeviceList
  
```

- **Le bouton connect :**

Pour se connecter à la carte micro bit via Bluetooth.

```

quand btn_connect .Clic
faire
  appeler BluetoothLE1 .Se connecter
  index Vue_liste_ble . Indice sélection
  mettre Label_status .Texte à " Connection en cours "
  mettre Vue_liste_ble .Visible à faux
  
```

- **Le bouton deconnect :** Pour se déconnecter de l'appareil Bluetooth.

```

quand btn_deconnect .Clic
faire
  appeler BluetoothLE1 .Déconnecter
  mettre Label_status .Texte à " deconnecter "
  
```

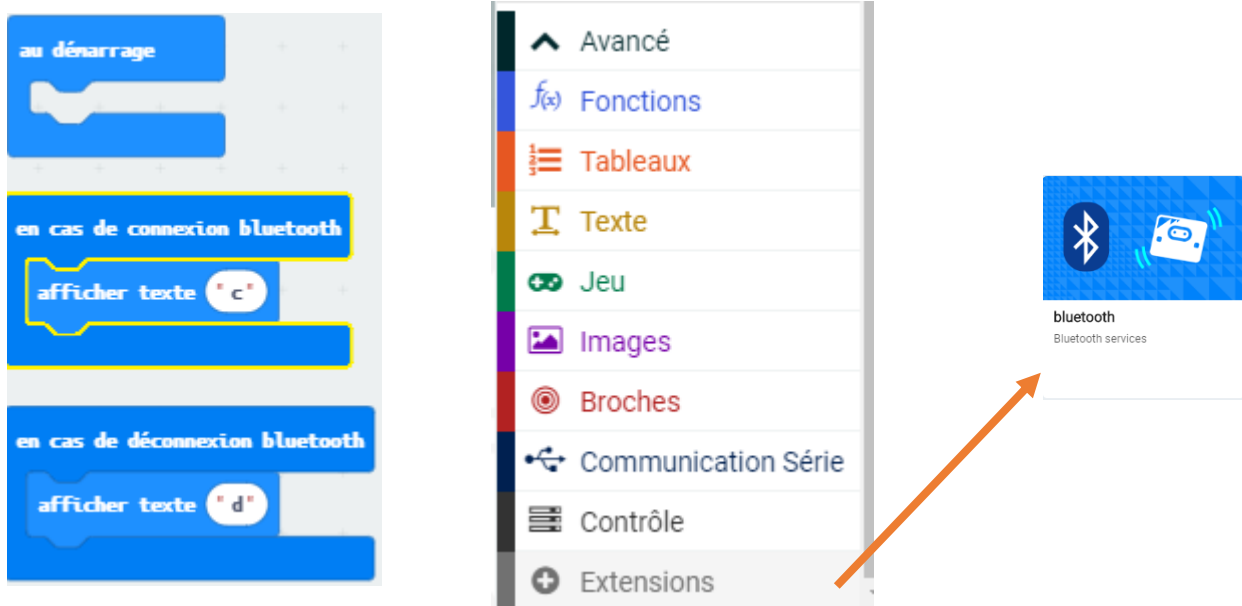
Maintenant, testons notre application en utilisant le **MIT AI2 compagnon**. Une fois que vous avez connecté votre appareil sur votre ordinateur, testez l'application à l'aide des étapes suivantes:

- Cliquez sur le bouton scan, Vous devriez voir une liste des appareils BLE.
- Lorsque vous voyez votre appareil, cliquez sur arrêter l'analyse.
- Cliquez sur le nom de votre appareil dans la liste.
- Cliquez sur Connecter, si votre appareil se connecte avec succès votre Label_Status devrait devenir "Connecter".

Configuration de la carte Microbit pour rechercher les appareils Bluetooth disponibles

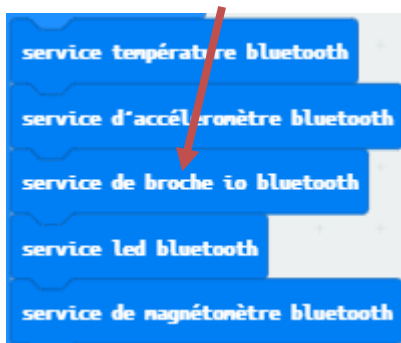
ETAPE 2 : La configuration de la carte pour pouvoir communiquer avec le smart phone

Lancer MakeCode :

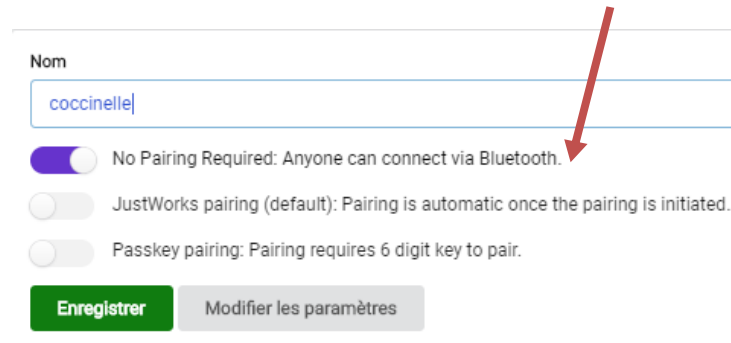


Au niveau de la programmation de la carte microbit, on doit ajouter au niveau du bloc de démarrage l'un **des services Bluetooth** suivant :

Au niveau Mackode



Au niveau App inventor on doit ajouter, on doit activer le paramètre :



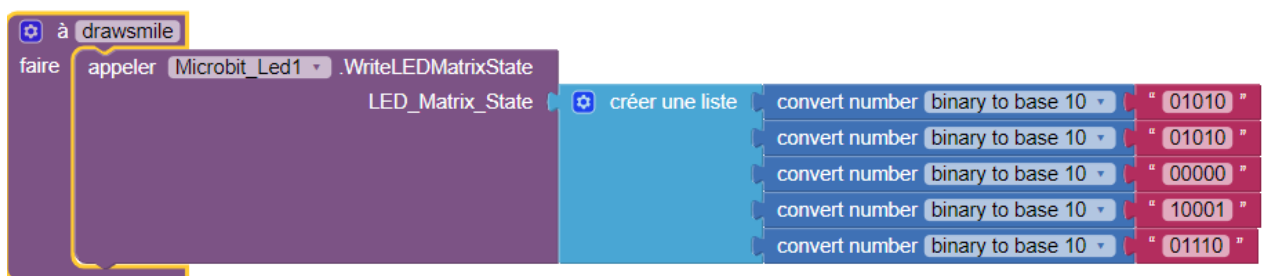
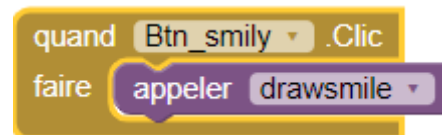
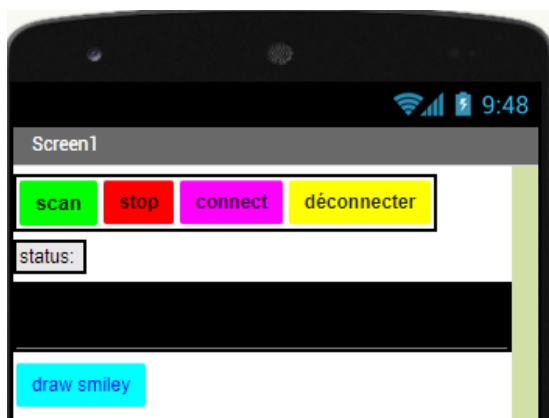
EXEMPLE 2

Piloter la matrice LED du Microbit à partir d'une application Mobile

Afficher un smile sur la carte Microbit lorsqu'on appuie sur un bouton draw Smile.

Modifier l'atelier 1 au niveau de l'application app inventor et le code de la carte **Microbit** :

Au niveau app inventor :



Au niveau de Makecode :

- Lorsqu'on clique sur le **bouton drawsmile**, elle envoie un smile sur la carte Microbit.

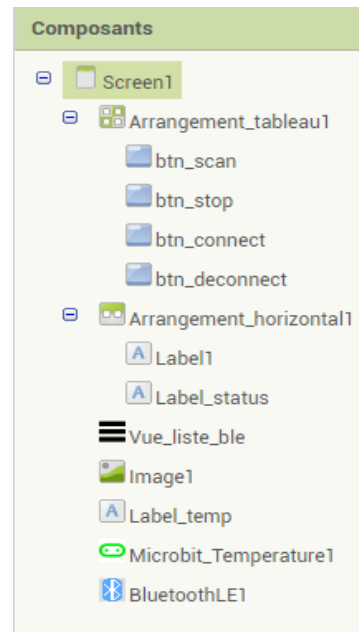


On doit ajouter au niveau du code de la carte micro bit le service de **led bluetooth**

EXEMPLE 3

Création d'une application app inventor qui reçoit la température de la maison par bluetooth

Au niveau de l'app inventor :



On garde la même partie de détection de la carte micro bit et on lui ajoute, le bloc quand l'application de température reçoit la valeur de température détecter de

```

quand Microbit_Temperature1 .TemperatureReceived
  température_value
  faire mettre Label_temp . Texte à obtenir température_value
  
```

```

quand BluetoothLE1 .Connected
  faire mettre Label_status . Texte à "Connecté"
  mettre Vue_liste_ble . Visible à faux
  appeler Microbit_Temperature1 .RequestTemperatureUpdates
  
```

Au niveau de MakeCode :



On doit ajouter au niveau de la carte micro bit le **service de température**