

Objet connecté

Un **objet connecté** est un **objet** électronique capable de communiquer avec un autre **objet** (souvent un smartphone, une tablette ou un ordinateur). Cette communication permet à l'**objet** d'envoyer ou de recevoir des informations via une connexion Internet. On parle alors d'Internet des **objets** ou d'Internet of Things (IoT).

• À faire vous-même 1 :

Quels objets connectés connaissez-vous ?

1. Communication entre cartes micro:bits

Nous pouvons créer un réseau de cartes Micro:bits pour dialoguer entre elles afin d'envoyer la même donnée à toutes les cartes sur le réseau sans préciser le destinataire. Les cartes communiquent via la radio.

On décidera avec les élèves les canaux utilisés par chaque couple de carte émetteur-récepteur. Numéros des canaux : 0 à 83.

• À faire vous-même 2 :

Afin de tester la communication radio entre les 2 cartes, **programmez** et **simulez** les programmes suivants :

Carte émettrice

```
au démarrage
radio définir groupe 101

Quand une donnée est reçue par radio receivedNumber
montrer nombre receivedNumber
pause (ms) 500
efface l'écran
```

Carte réceptrice

```
au démarrage
radio définir groupe 101

lorsque secouer
envoyer le nombre 1 par radio
montrer l'icône
efface l'écran
```

a) Création d'un groupe
Il faut commencer par créer un groupe (ensemble de cartes Microbit qui échangent des données)

b) Envoi d'une donnée.
Lorsque tu envoies des informations, toutes les cartes qui font partie de ton réseau reçoivent toutes la même information.

c) Réception d'une donnée :
Dès que l'événement de réception est actif, il y a lecture de la donnée reçue

La donnée à lire est stockée dans la variable : **receivedNumber**

• À faire vous-même 3 :

Modifiez les programmes précédents pour afficher sur la carte micro:bit réceptrice la température mesurée sur la carte émettrice quand la carte est secouée.

Si tu veux le programmer en python

❖ **Programme à mettre dans les cartes micro:bits (ne pas écrire les commentaires)**

```
# Emetteur radio
from microbit import *
import radio

radio.on()
radio.config(channel=1) # choisir un numéro de canal radio
radio.config(power=7) # mettre le signal à puissance maxi

while True: # boucle infinie
    message_envoye = "OK" # message à envoyer
    radio.send(message_envoye) # envoi du message
    sleep(2000) # attendre 2s
```

```
# Recepteur radio
from microbit import *
import radio

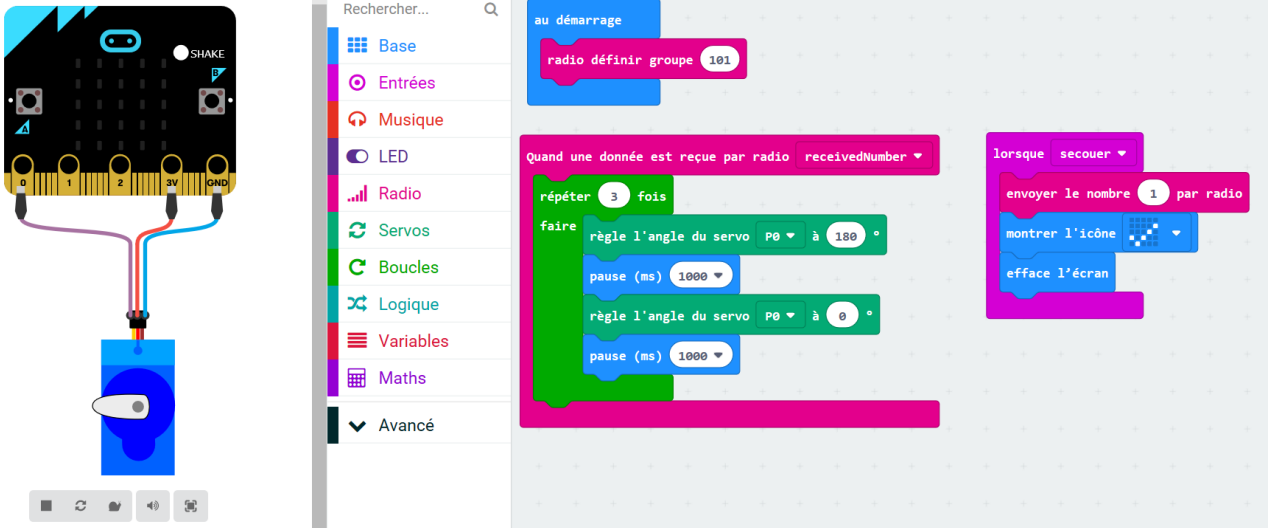
radio.on()
radio.config(channel=1) # choisir un numéro de canal radio
radio.config(power=7) # mettre le signal à puissance maxi

while True :
    message_recu = radio.receive() # récupérer le message reçu
```

```
if message_recu is not None: # si le message
reçu n'est pas vide alors l'afficher
display.scroll(message_recu)
```

• À faire vous-même 4 :

Dès qu'une carte micro:bit 1 est secouée, un drapeau associé à une servomoteur, raccordé sur la patte 0 bouge 3 fois.



• À faire vous-même 5 : Mini-projet

Dès que la température atteint 22°C, orienter les lames du store de 60°, par commande d'un actionneur servomoteur sinon les lames du store restent ouvertes. Si la lumière est trop faible, le store se ferme.



Programmez et **simulez** un programme permettant d'obtenir ce fonctionnement.

Evolution du produit, le propriétaire de la maison veut pouvoir piloter à distance l'inclinaison des lames en appuyant sur des boutons poussoirs montées et descentes. **Programmez** et **simulez** un programme permettant d'obtenir ce fonctionnement avec deux cartes micro:bits.

2. Communication cartes micro:bits et smartphones

- ❖ Activité proposée par Damien Iceta de créteil : **Plante Connectée** extrait de magistère : <https://magistere.education.fr/dgesco/course/view.php?id=1538>

Extrait ci-après de la partie à exploitée pour notre activité.

3. Étape : Programmer la carte Micro:bit afin de lui permettre de communiquer en bluetooth et appairer smartphone et carte

Objectifs

Objectifs : Etablir une connexion entre la carte et le smartphone.

Capacités attendues :

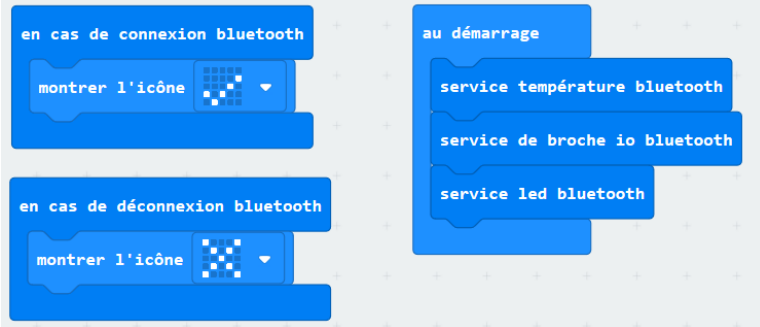
- Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs, l'IHM et les actions des actionneurs dans des systèmes courants.

Téléverser le programme permettant la connexion

Charger le programme Bluetooth.hex dans la carte micro:bit (glisser déposer).

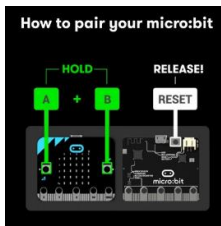
Celui-ci contient les blocs ci-contre (les blocs Bluetooth apparaissent lorsque l'on va dans extensions Bluetooth).

Quel est le symbole qui s'affiche lorsque l'on est connecté et déconnecté ?



Remarque : Sur la droite on distingue les services disponibles au démarrage. Nous n'avons besoin pour cette application que de la température, mais les deux autres services seront utiles pour la suite.

Appairer la carte et le smartphone



Appairer la carte micro:bit et le smartphone :

- Appuyer sur les boutons A et B.
- Maintenir A et B appuyés et appuyer et relâcher le bouton reset.
- Un symbole Bluetooth apparaît puis une figure caractéristique de votre carte permettant l'appairage que nous n'utiliserons pas ici.
- Aller sur le smartphone dans les paramètres Bluetooth, dans la liste des appareils sélectionner celui nommé micro:bit.

Vous devriez alors voir le symbole correspondant à la connexion.

4. Réaliser une interface graphique permettant d'afficher la température en temps réel

Objectifs

Objectif : Réaliser une interface graphique permettant d'afficher la température en temps réel sur le smartphone.

Capacités attendues :

- Réaliser une IHM simple d'un objet connecté
- Écrire des programmes simples d'acquisition de données

Remarque APP Inventor sans compte en local

Remarque à destination des collègues : Il est possible d'utiliser APP inventor en ligne, mais aussi hors ligne. Bien que la version en ligne <http://ai2.appinventor.mit.edu/> possède de nombreux avantages en termes de simplicité d'utilisation puisqu'aucune installation n'est requise, elle demande de créer un compte ou d'utiliser un compte Google pour s'identifier, ce qui pose problème pour une utilisation en classe. **Une procédure d'installation hors ligne est disponible pour résoudre ce problème. Cette solution permet de ne pas recourir à un compte Google, et aussi d'éviter de divulguer des données personnelles et ne requière pas de connexion internet. Cette solution permet également de créer des comptes locaux pour les élèves. Dans les annexes vous trouverez la procédure simplifiée d'installation de APP inventor portable et de création des comptes locaux.** Celle-ci a été réalisée à partir de la procédure d'installation disponible sur le site de l'académie de Poitiers <http://ww2.ac-poitiers.fr/sciences-ingenieur-sti/spip.php?article189> ainsi que d'un PDF de l'académie de Nantes donné également en annexe.

Réaliser l'interface graphique



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 Interface (2) puis paramétrer cet élément dans la zone Propriétés (3).

Importer la base déjà préparée Base.aia en cliquant sur Projets / Importer le projet de mon ordinateur.

Glisser à partir de la palette Disposition un arrangement horizontal.

Réaliser l'interface graphique

Glisser une image puis 3 labels à l'intérieur de cet élément.

Dans « Média » charger l'image ImageT.png.

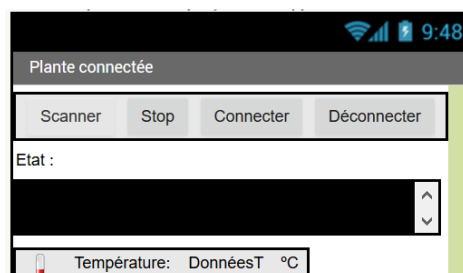
En cliquant sur l'image dans « Composants » régler la hauteur et largeur à « 10 percent », dans image sélectionner l'image importée ImageT.png.

Dans « Composants » cliquer sur Label1 et dans les propriétés remplir le champ TEXT avec « Température: ».

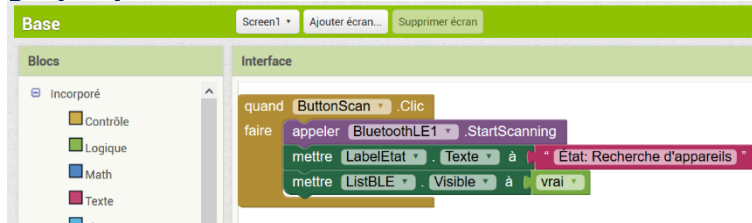
Dans « Composants » cliquer sur Label2 et dans les propriétés remplir le champ TEXT avec « DonnéesT ».

Dans « Composants » cliquer sur Label3 et dans les propriétés remplir le champ TEXT avec « °C ».

Dans « Composants » cliquer sur Label2 le renommer en « LabelT ».



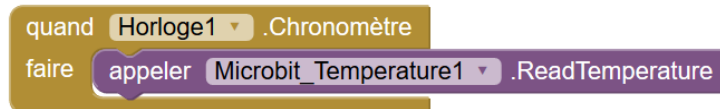
Cliquer maintenant sur Blocks pour accéder à la partie programmation graphique.



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. Les différents éléments permettant la connexion Bluetooth ont déjà été ajoutés. Un élément Horloge a également été ajouté et l'intervalle de temps a été réglé à 100 ms dans les propriétés. On peut remarquer que le chronomètre est activé lorsque l'on est dans l'état connecté et désactivé sinon.

On souhaite effectuer une mesure toutes les 100 ms. **Ajouter les blocs suivants (palette Horloge et Microbit_Température).**




À chaque fois qu'une température est reçue, il faut mettre l'affichage à jour correspondant au LabelT. **Ajouter les blocs suivants.**



Remarque à destination des collègues : Il y a plusieurs manières possibles de tester l'application créée. Le plus rapide est d'utiliser l' « AI2 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.

Charger l'application dans la tablette ou smartphone

Sans AI2 companion	Avec AI2 companion
<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>	<div style="display: flex; align-items: center;"> <div style="flex: 1;"> <p>Installer AI2 companion (cf. AI2 companion) – 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'AI2 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> </div> <div style="flex: 0.5; text-align: center;">  <p style="font-size: small;">MIT AI2 Companion MIT Center for Mobile</p> </div> </div>

