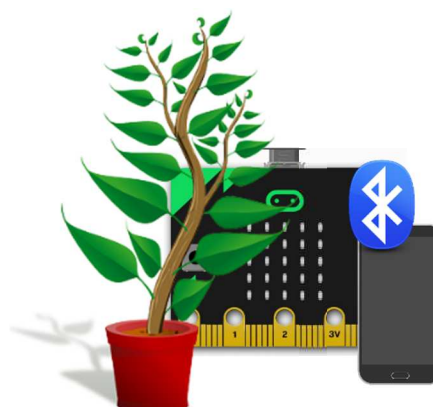


PLANTE CONNECTÉE



Thème du programme : Informatique embarquée et objets connectés

| Extrait du programme | Contenus | Capacités attendues |
|----------------------|---|---|
| | Systèmes informatiques embarqués | 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. |
| | 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 <ul style="list-style-type: none"> 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 : Programmer l'acquisition de la température et de l'humidité à l'aide d'une carte Microbit (équipée de ses propres capteurs : pas de capteurs à ajouter). Réaliser un affichage des valeurs sur la matrice de LED présente sur la carte pour fonctionner de façon autonome. Établir une communication Bluetooth entre la carte et un smartphone. Réaliser une IHM sur le smartphone afin d'acquérir en temps réel les valeurs de température et d'humidité et d'envoyer un signal si l'arrosage est nécessaire.

Matériel : 1 carte Microbit (environ 20 euros), 2 barres métalliques, 2 fils, 1 smartphone ou tablette et 1 plante.

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. 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.

Durée : 1h30 * 3

Période : Deuxième ou troisième trimestre

Bibliographie et Sitographie :

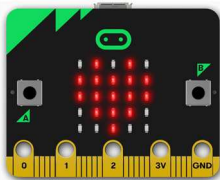
- http://iot.appinventor.mit.edu/assets/howtos/MIT_App_Inventor_Microbit_IOpin_potentiometer.pdf
- http://iot.appinventor.mit.edu/assets/howtos/MIT_App_Inventor_Microbit_LED.pdf
- http://iot.appinventor.mit.edu/assets/howtos/MIT_App_Inventor_Microbit_Temperature.pdf
- <https://microbit.org/guide/hardware/pins/>
- <https://makecode.microbit.org/projects/soil-moisture/code>
- <https://microbit.org/fr/guide/features/>

PRÉSENTATION DE LA CARTE

Cette présentation des éléments de la carte Micro:bit, que nous allons utiliser, est extraite de la présentation de tous les éléments disponibles sur le site (<https://microbit.org/fr/guide/features/>).

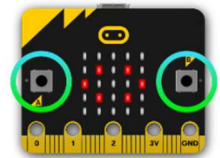
Nous allons utiliser, sur cette carte :

- 25 LED programmables individuellement
- 2 boutons programmables
- Des broches de connexion
- Un capteur de température
- Une communication sans fil, via Radio et Bluetooth
- Son Interface USB



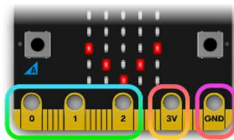
Les LED

LED signifie Light Emitting Diode, Diode électroluminescente. La carte dispose de 25 LED, toutes programmables individuellement, ce qui permet d'afficher du texte, des nombres et des images.



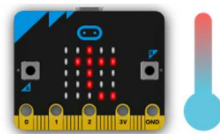
Les boutons

Il y a deux boutons sur la face avant (étiquetés A et B).



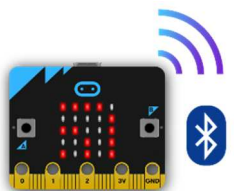
Les broches

Il y a 25 connecteurs externes sur la tranche, que l'on nomme 'broches'. On peut programmer des moteurs, des LED, ou tout autre composant électrique à l'aide de ces broches. Nous allons les utiliser dans la suite pour réaliser un capteur d'humidité.



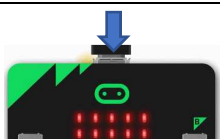
Capteur de température

Ce capteur permet de mesurer la température ambiante, en degrés Celsius.



L'antenne Bluetooth

Une antenne BLE (Bluetooth Low Energy) permet d'envoyer et de recevoir des signaux Bluetooth. Cela permet à la carte de communiquer sans fil avec des PC, des smartphones, des tablettes. (Gamme de fréquences : 2402MHz à 24801MHz - Version Bluetooth: V4.0 Bluetooth basse consommation)



L'interface USB

Sur le haut de la carte, vous trouvez l'interface USB qui vous permet de connecter la carte à votre ordinateur via un câble USB. Cela fournit l'alimentation et vous permet de télécharger des programmes.

MESURER L'HUMIDITÉ DU SOL

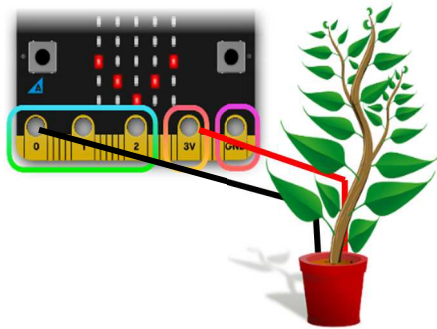
Objectifs

Objectifs : Afficher sur la carte l'humidité du sol

Capacités attendues :

- Écrire des programmes simples d'acquisition de données
- Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs

Brancher



Relier une des barres métalliques à la broche 3V et l'autre sur la broche nommée P0.

Insérer la carte dans un port USB. Une fenêtre s'ouvre, cliquer sur MICROBIT.HTM puis aller sur « let's Code ».

Il est possible de programmer en python ou avec un langage graphique. Nous utiliserons ici le langage graphique.

Aller sur « MakeCode Editor » puis « let's code ».

Coder et téléverser



Le sol a une conductivité électrique qui dépend de la quantité d'eau et de nutriments. Pour mesurer cette conductivité, on lit la tension sur la broche P0 avec « analog read pin ». La valeur obtenue sera comprise entre 0 et 1023. On utilise alors « plot bar graph » pour réaliser l'affichage.

Réaliser le graphe ci-dessus utilisant les menus « Basic, LED et Advanced/Pins ».

Cliquez sur le bouton **Télécharger dans l'éditeur**. Ceci va télécharger un fichier 'hex', qui est un format compact de votre programme que votre micro:bit peut lire. **Une fois le fichier hex téléchargé, copiez-le sur votre micro:bit.** La carte micro:bit se met en pause et la LED jaune à l'arrière clignote. Une fois que c'est fini, le code s'exécute automatiquement.

Tester le capteur dans différents sols. Obtient-on la valeur maximale (1023)?

On n'obtient pas une valeur précise, mais uniquement une représentation graphique. **On souhaiterait maintenant que lorsque l'on appuie sur le bouton A la valeur en volt s'affiche.**

Écrire ce que fait chacun des blocs ci-dessous :

Modifier (ajouter une interaction)



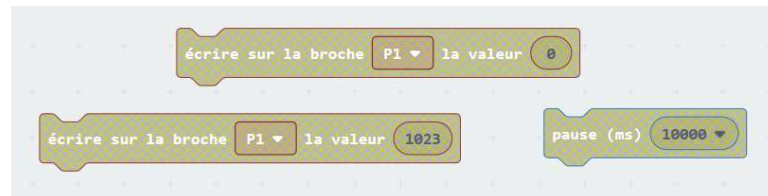
Tester le programme (Télécharger puis glisser-déposer dans la carte Micro:bit).

Quelles sont les valeurs minimum et maximum obtenues ? Comment peut-on utiliser la valeur maximum pour obtenir un graphique prenant l'amplitude maximum sur les LED ?

Modifier (gestion de l'énergie)

On souhaite **éviter de gaspiller de l'énergie**. Actuellement le circuit est connecté directement sur la broche 3V donc il consomme en permanence. On souhaite à la place utiliser la broche P1 pour l'alimentation et mettre cette broche à l'état haut uniquement lors de la mesure. De plus l'évolution de l'humidité étant lente, une pause dans la mesure toutes les dix secondes peut être ajoutée.

Où faut-il ajouter les trois blocs suivants au programme déjà réalisé ? Tester ce nouveau programme.



ÉTAPE : PROGRAMMER LA CARTE MICROBIT AFIN DE LUI PERMETTRE DE COMMUNIQUER EN BLUETOOTH ET APPAIRER SMATPHONE ET CARTE

Objectifs

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

Capacités attendues :

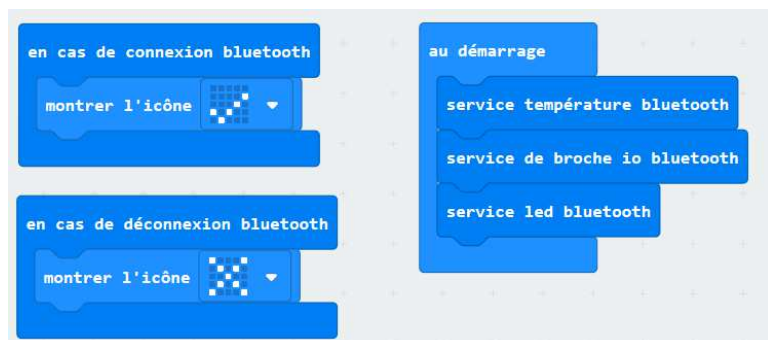
- Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs, l'IHM et les actions des actionneurs dans des systèmes courants.

Téléverser le programme permettant la connexion Bluetooth

Charger le programme Bluetooth.hex dans la carte microbit (glisser déposer).

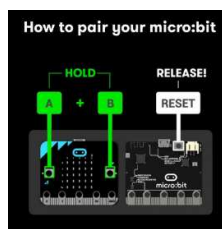
Celui-ci contient les blocs ci-contre (les blocs Bluetooth apparaissent lorsque l'on va dans extensions Bluetooth).

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 microbit 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 microbit.

Vous devriez alors voir le symbole correspondant à la connexion.

RÉALISER UNE INTERFACE GRAPHIQUE PERMETTANT D’AFFICHER LA TEMPÉRATURE EN TEMPS REEL

Objectifs

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

Capacités attendues :

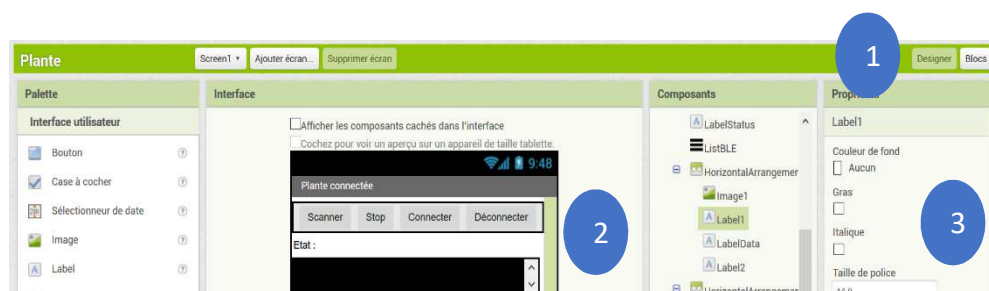
- Réaliser une IHM simple d’un objet connecté
- Écrire des programmes simples d’acquisition de données

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 requiert 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.

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

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

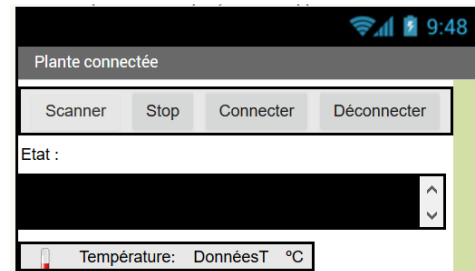
En cliquant sur l'image dans « Composants » régler la hauteur et largeur à « 10 percent », dans image sélectionner l'image importée ImageT.png.

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

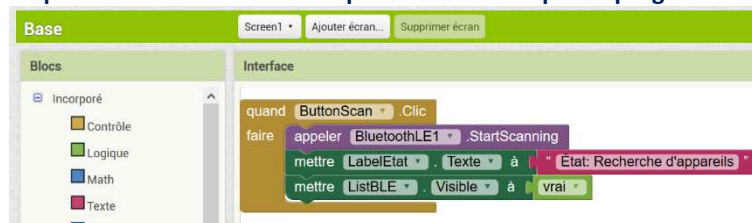
Dans « Composants » cliquer sur Label2 et dans les propriétés remplir le champ TEXT avec « DonnéesT ».

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

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

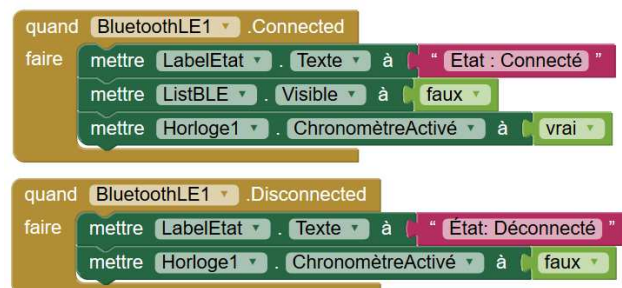


Cliquer maintenant sur Blocks pour accéder à la partie programmation graphique.

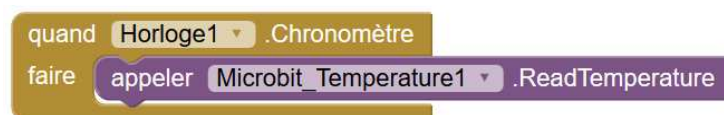


On retrouve à nouveau une zone de visualisation du code graphique et une zone Blocks pour choisir les méthodes, variables... pour programmer. Chaque bloc

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).**




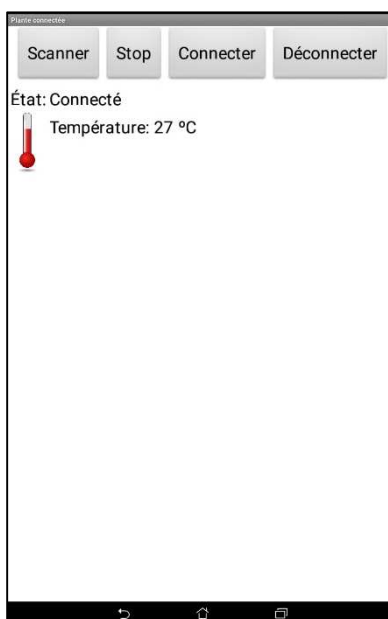
À chaque fois qu'une température est reçue, il faut mettre l'affichage à jour correspondant au LabelT. **Ajouter les blocs suivants.**



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="text-align: right;">  <p>MIT AI2 Companion MIT Center for Mobile Computing</p> </div> <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'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> |



AJOUTER LE CAPTEUR D'HUMIDITÉ (UTILISATION DES BROCHES)

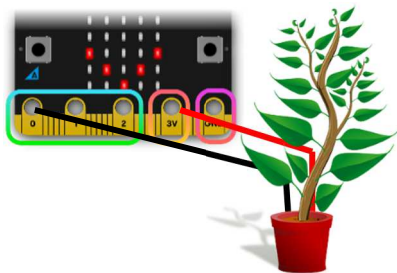
Objectifs

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

Capacités attendues :

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

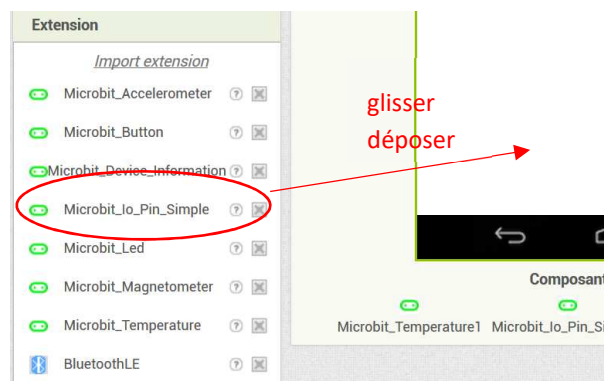
Brancher



Utiliser le même montage que précédemment afin d'obtenir une mesure de l'humidité.

Connecter une des barres métalliques sur la broche 3V et l'autre sur la broche P0.

Réaliser l'interface graphique

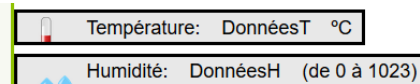


Reprendre le projet précédent avec le capteur de température et ajouter l'extension nécessaire à la gestion des broches : **Microbit_Io_Pin_Simple**.

Cliquer sur l'icône **Microbit_Io_Pin_Simple** dans l'interface puis renseigner les propriétés avec **BluetoothLE1** :



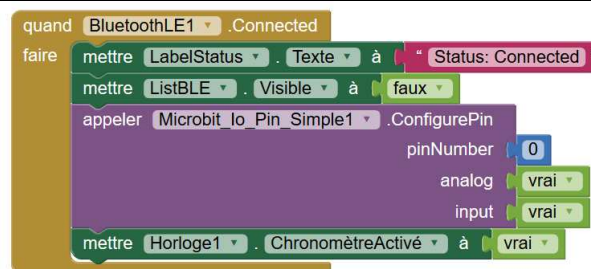
Compléter l'interface en ajoutant un élément horizontal comportant une image et trois labels nommés respectivement label3, labelH et label4 et dont les textes correspondent à la figure de droite.



Programmer les différentes interactions

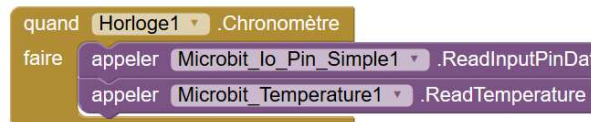
Initialisation de la lecture des broches

Ajouter le bloc issu de la palette **Microbit_Io_Pin_Simple1** permettant d'initialiser la lecture de la broche : on donne le numéro de la broche, si c'est une broche analogique ou numérique et si c'est une entrée (C'est le cas pour un capteur).



Lecture des données

On a mis en place précédemment un bloc permettant d'effectuer une mesure toutes les 100 ms pour la température.



Ajouter ici la mesure relative à l'humidité en utilisant la palette **Microbit_Io_Pin_Simple1**.

Affichage des données

InputPinDataReceived renvoie une liste. Pour réaliser l'affichage des données, il nous faut sélectionner la deuxième donnée de cette liste.



Le fichier **Plante_Temperature_Humidite.aia** contient l'ensemble des données complétées.

Compléter le programme avec les blocs correspondant à l'image.

Créer la nouvelle application et la tester. En cas de besoin cette application se trouve dans le dossier **Corrigé/Plante_Temperature_Humidite.apk**

AJOUTER L'AFFICHAGE LED

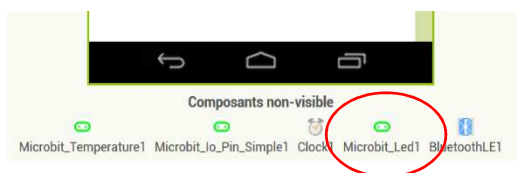
Objectifs

Objectif : Afficher sur les LED un signal demandant l'arrosage à partir du smartphone.

Capacités attendues :

- Réaliser une IHM simple d'un objet connecté
- Écrire des programmes simples de commande d'un actionneur

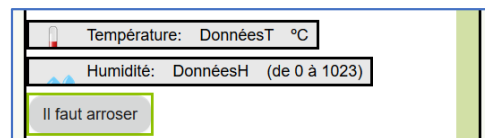
Réaliser l'interface graphique



De plus un bouton a été ajouté pour envoyer le signal d'arrosage.

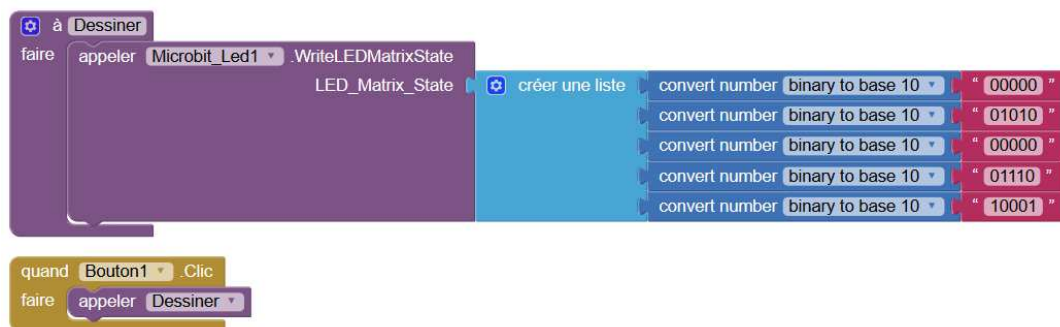
Charger le fichier **Plante_total.aia** (projet/importer projet aia de mon ordinateur)

L'extension nécessaire à la gestion des broches a déjà été ajoutée : **Microbit_Led**.



Des blocs ont été ajoutés : lorsque l'on clique sur le bouton un signal demandant l'arrosage s'affiche sur les LED.

Programmer les différentes interactions



Que font chacun de ces blocs ? Représenter l'image affichée.

Modifier les 0 et 1 pour afficher une image correspondant mieux à une demande d'arrosage.

Créer la nouvelle application et la tester. En cas de besoin cette application se trouve dans le dossier **Corrigé/Plante_total.apk**