

Nom :	Prénom :	Classe :
-------	----------	----------

NSI 1re — Créer une IHM à partir d'un cahier des charges

Objectifs :

- Savoir créer une interface homme-machine (IHM) à partir d'un cahier des charges.
- Découvrir la carte programmable micro:bit.
- Programmer un objet réel avec le langage Python.

Introduction

Créer une IHM à partir d'un cahier des charges, c'est concevoir une **interface homme-machine (IHM)** à partir d'un document qui décrit ce que l'on veut obtenir : fonctionnalités, écrans, boutons, messages, contraintes, ergonomie¹, etc. Une IHM sert à permettre à une personne d'interagir facilement avec une machine, un logiciel ou un système.

Une IHM, c'est tout ce qui rend possible l'échange entre l'utilisateur et la machine : écran tactile, boutons, menus, alertes sonores, affichages, ou logiciel de commande.

Le **cahier des charges**, lui, est le document qui fixe les attentes du projet, y compris l'interface utilisateur, les contenus à afficher et parfois même l'ergonomie.

Exemple de cahier des charges

Projet : création d'un compteur de votes pour une classe.

Fonctionnalités :

- L'appui sur un bouton "A" ajoute 1 au premier choix d'un vote.
- L'appui sur un bouton "B" ajoute 1 au deuxième choix d'un vote.
- Un appui simultané sur les boutons "A" et "B" remet les scores à zéro.

Affichages :

- Le score s'affiche sur l'écran sous forme de chiffres.

Contrainte :

- L'utilisatrice ou l'utilisateur doit comprendre rapidement quelle touche utiliser.

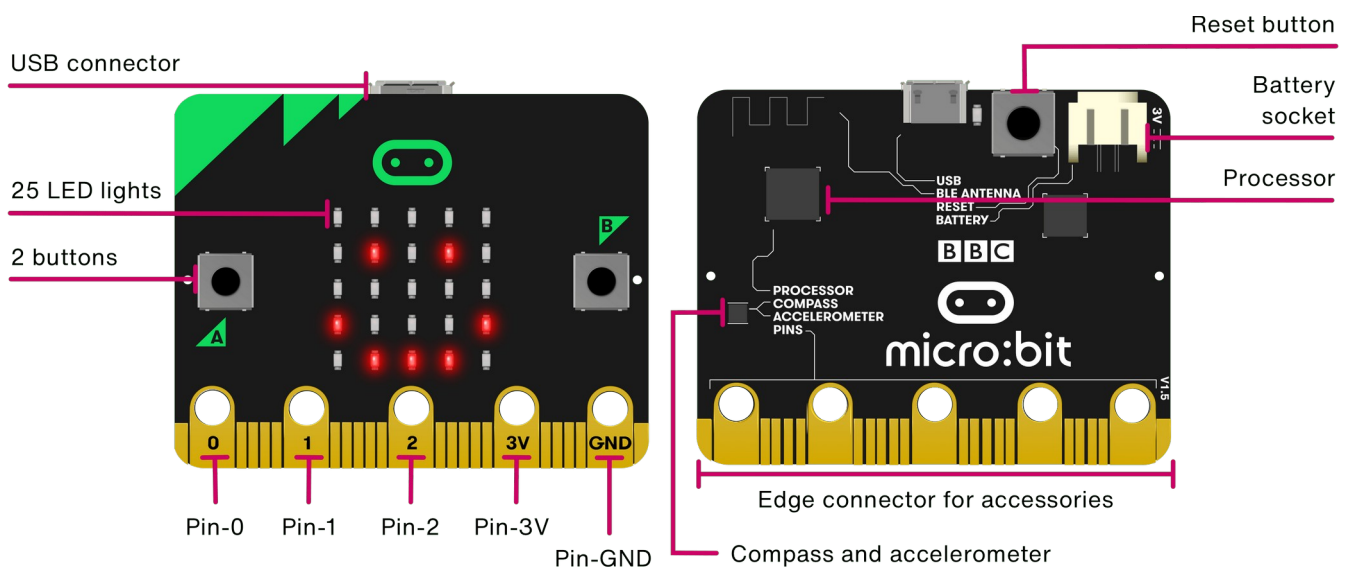
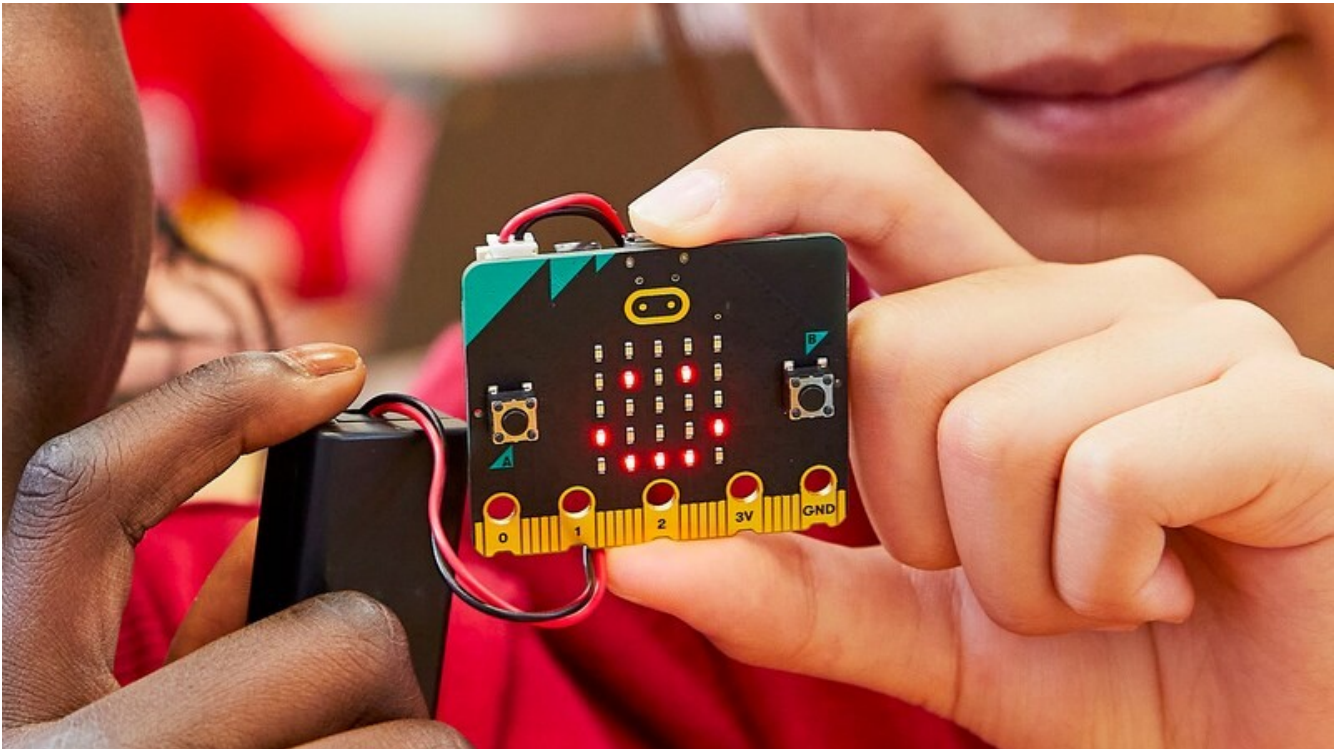
Limite technique :

- Le programme doit fonctionner de façon indépendante sur un micro:bit², sans écran externe, ni ordinateur branché.

1 L'ergonomie est la discipline qui consiste à adapter un outil, un poste ou un environnement à l'être humain pour qu'il soit plus confortable, plus sûr et plus efficace.

2 Le micro:bit est une petite carte électronique programmable conçue au Royaume-Uni par la BBC (service public britannique de radio et de télévision) avec l'objectif de rendre l'apprentissage du codage simple, concret et ludique pour les jeunes, notamment à l'école. Elle sert à créer facilement des objets connectés, des jeux minimalistes et des expériences interactives grâce à ses boutons, ses capteurs et son écran LED.

La carte micro:bit



Brève description de la carte programmable micro:bit.

- Deux boutons “A” et “B” (utilisables séparément ou ensemble pour déclencher des actions).
- Un écran composé de 25 LED rouges (intensité réglable).
- Un connecteur USB (micro USB).
- Un bouton “Reset” (remise à zéro).
- Un connecteur pour piles.
- Un processeur.
- Des capteurs : luminosité, température, boussole (mesure les champs magnétiques et détecte le nord magnétique), accéléromètre (mesure les mouvements et l'orientation de la carte en 3 dimensions).
- Une interface Bluetooth, grâce à une puce Bluetooth Low Energy (BLE) qui émet et reçoit des ondes radio.

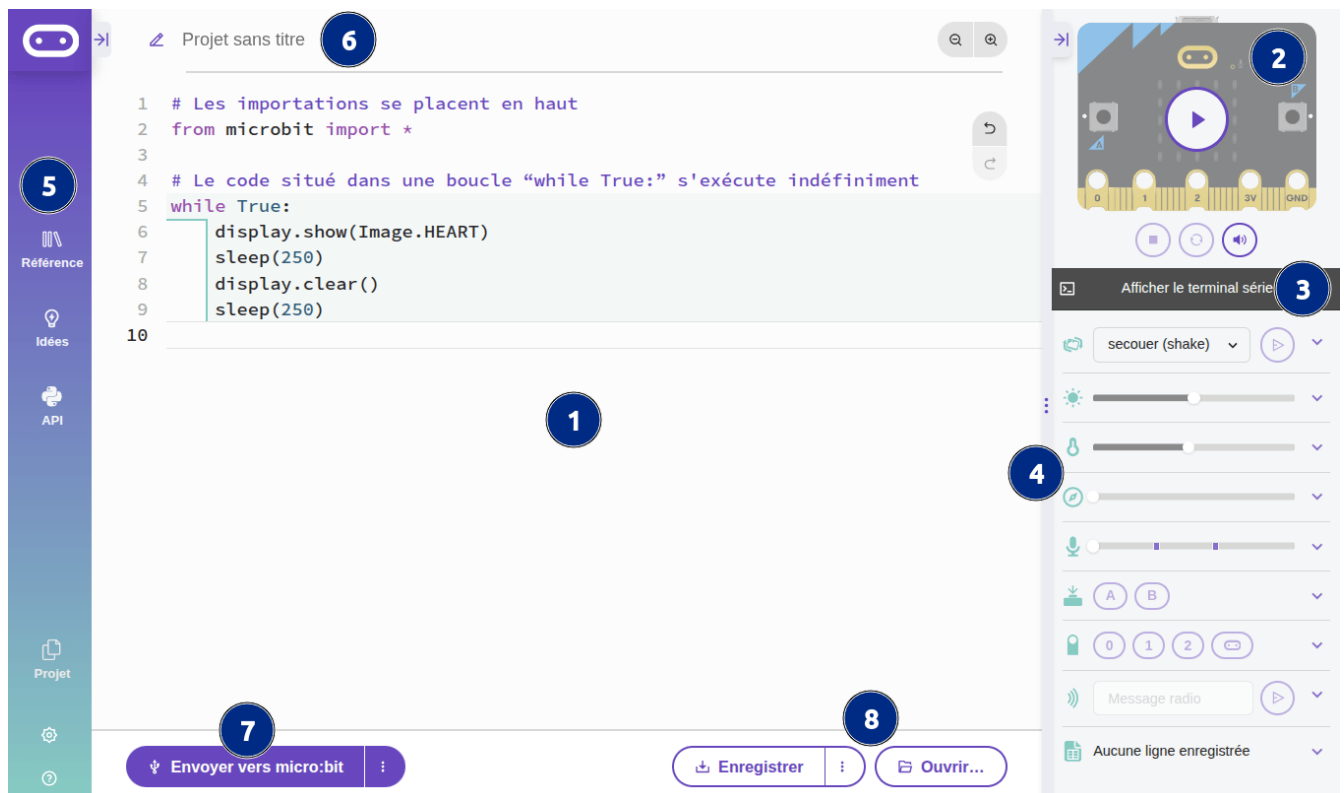
Entrées et sorties

La carte micro:bit aide à comprendre le fonctionnement des ordinateurs. Lorsque nous tapons sur notre ordinateur ou que nous touchons l'écran de notre téléphone, nous utilisons un périphérique **d'entrée**. Les entrées permettent aux ordinateurs de détecter les choses qui se passent dans le monde réel, afin qu'ils puissent y réagir et faire se produire quelque chose, généralement sur une **sortie** comme un écran ou des écouteurs.

Entre l'entrée et la sortie, il y a le **processeur**. Il récupère des informations à partir d'entrées, comme des boutons par exemple, et fait se produire quelque chose sur les sorties, comme jouer de la musique dans nos écouteurs.

Comment programmer un micro:bit ?

Une carte micro:bit peut être programmée *visuellement*³ par des “blocs” (comme Scratch) ou par du Python, à l'adresse suivante : <https://python.microbit.org>



N°	Description	N°	Description
1	Éditeur de code	5	Aide en ligne (Référence)
2	Simulation de l'affichage sur la carte	6	Titre du projet en cours
3	Affichage de la console Python	7	Envoi du programme sur la carte branchée
4	Simulation des capteurs de la carte	8	Enregistrement du programme en fichier

3 La programmation par blocs se fait en ligne, grâce à des sites Web comme makecode.microbit.org ou fr.vittascience.com/microbit/

Let's code !

Maintenant que les présentations sont faites, il est temps de coder notre micro:bit !



Dans ce TP, nous utiliserons le navigateur Chrome (ou “Chromium”, son équivalent open-source) car il est — à ce jour — un des seuls à prendre en charge la technologie WebUSB permettant de transférer directement le programme Python depuis l'éditeur de code (python.microbit.org) vers le micro:bit (branché en USB à l'ordinateur).

► Exercice 1 — LEDs et boutons

Choisir un des projets ci-dessous (ou tous) puis le(s) programmer sur l'éditeur en ligne python.microbit.org :

Projet 1A : Création d'un badge animé

Fonctionnalités :

- Le badge fait défiler notre prénom en continu.

Affichages :

- Le prénom s'affiche sur l'écran en défilant.

Contrainte :

- Aucune contrainte.

Limite technique (optionnelle) :

- Le programme doit fonctionner de façon indépendante un micro:bit, sans écran externe, ni ordinateur branché.

Projet 1B : Cœur battant

Fonctionnalités :

- Le programme montre un cœur battant à l'aide de deux images intégrées, un grand et un petit cœur, sur l'écran de LEDs du micro:bit.

Affichages :

- Les différentes images présentées dans une séquence créent l'illusion d'un mouvement : un cœur de plus en plus grand et de plus en plus petit.
- Après avoir affiché chaque image, le programme s'arrête pendant une demi-seconde (500 millisecondes) avant d'afficher l'image suivante.

Contrainte :

- L'animation est maintenue en continu en utilisant une boucle infinie : elle répète la séquence d'affichage de ces deux images et fait une pause jusqu'à ce que nous débranchions le micro:bit.

Limite technique (optionnelle) :

- Le programme doit fonctionner de façon indépendante un micro:bit, sans écran externe, ni ordinateur branché.

► Exercice 2 — Capteurs

Apprendre à programmer le micro:bit pour mesurer l'environnement en utilisant ses capteurs intégrés...

Choisir un des projets ci-dessous (ou tous) puis le(s) programmer sur l'éditeur en ligne python.microbit.org :

Projet 2A : Compteur de pas

Fonctionnalités :

- La carte compte combien de fois elle a été secouée. Elle stocke ce nombre dans une variable.

Affichages :

- Le nombre de pas s'affiche sur l'écran en défilant.

Contrainte :

- Chaque fois que l'accéléromètre du micro:bit détecte un tremblement, le programme augmente la variable de 1, et affiche le nouveau nombre sur l'écran LED.

Limite technique :

- Le programme doit fonctionner de façon indépendante sur un micro:bit, sans écran externe, ni ordinateur branché.

Projet 2B : Thermomètre

Fonctionnalités :

- Lors de l'appui sur le bouton A, le micro:bit affiche la température actuelle du processeur sur son écran.

Affichages :

- Lorsque le bouton A n'est pas appuyé, l'écran est vide.
- Lorsque le bouton A est appuyé, la température s'affiche sur l'écran.

Contrainte :

- Aucune.

Limite technique :

- Le programme doit fonctionner de façon indépendante sur le micro:bit, sans écran externe, ni ordinateur branché.

Projet 2C : Veilleuse

Fonctionnalités :

- Une veilleuse qui allume l'écran LED du micro:bit dans l'obscurité.

Affichages :

- En plein jour, l'écran est automatiquement éteint.
- Dans l'obscurité, l'écran est automatiquement allumé.

Contrainte :

- Si le niveau de lumière tombe en dessous d'un certain seuil, alors on allume les LED sur l'écran du micro:bit. Sinon, on efface l'écran pour éteindre l'éclairage LED.

Projet 2D "chili pepper" : Thermomètre intérieur-extérieur

Coder deux programmes différents :

- Un **micro:bit extérieur** détecte la température et la transmet sur le groupe radio 23.
- Lorsque le **micro:bit intérieur** reçoit une température par le groupe radio 23, il la stocke dans une variable.
- Lorsque le bouton A de ce micro:bit intérieur est appuyé, il affiche sa propre température actuelle.
- Lorsque le bouton B de ce micro:bit intérieur est appuyé, il affiche la température stockée dans la variable (celle envoyée par le micro:bit extérieur).

► **Exercice 3** — Encore plus de projets

Quelques projets supplémentaires pour se faire plaisir ...

- 3.1. **Thermomètre Max-min** : suivre les températures les plus élevées et les plus basses en laissant le programme fonctionner sur un micro:bit. *Le bouton A affiche la température minimale enregistrée, le bouton B affiche la température maximale enregistrée. Le programme fait également clignoter un point sur l'écran LED chaque fois que la boucle infinie s'exécute, afin de savoir qu'il fonctionne.*
- 3.2. **Téléporteur de canard** : le programme est mis en place sur deux micro:bit. En secouant la première carte, le canard disparaît et apparaît sur la deuxième (et inversement). *L'illusion est rendue possible grâce à un message radio envoyé par la carte secouée. Si une des deux cartes reçoit ce message, elle affiche l'icône canard.*
- 3.3. **Réponse secrète** : utiliser la radio de deux micro:bit pour répondre secrètement à des questions. *Deux amis posent une question qui ont "oui" ou "non" comme réponse. Un appui sur le bouton A envoie "oui" comme réponse, un appui sur le bouton B envoie "non". Lorsqu'un message radio est reçu, si ce message vaut "oui" alors il affiche une coche sur l'écran LED. Si ce message vaut "non" il affiche une croix.*
- 3.4. **Chasse au trésor** : utiliser deux micro:bit pour faire un jeu de chasse au trésor en utilisant la communication radio. *Un programme "trésor" envoie des signaux radio, un autre "détecteur" cherche à les recevoir.*
- 3.5. **Course d'orientation** : utiliser plusieurs micro:bit pour faire une course d'orientation en utilisant la communication radio. *Deux programmes différents : un programme "balise" et un programme "récepteur". Installer le programme "balise" sur des micro:bit situés à différents endroits. Chaque balise a son identifiant unique. Les balises transmettent leur numéro d'identification toutes les 200 millisecondes. Le programme "récepteur" est installé sur autant de micro:bit que de joueuses / joueurs. Le programme "récepteur" affiche le numéro d'identification diffusé par la balise à proximité. L'affichage clignote lorsque la balise est encore loin, et se stabilise lorsqu'on se rapproche.*

Pour aller plus loin : la joueuse ou le joueur ayant collecté toutes les balises gagne et un dessin apparaît sur sa carte micro:bit.