



L'IOT POUR 5 ANS ET +

APPRENDRE À  
CODER AVEC  
MICRO:BIT



# SOMMAIRE

La carte micro:bit

---

Usages

---

Exercices

---

Conclusion

BBC

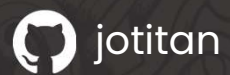


micro:bit



A photograph of ancient Mayan stone ruins, including a tall cylindrical tower and various carved structures, partially covered in green moss and vines, set against a backdrop of dense tropical forest and a cloudy sky.

**Jonathan BARANZINI**  
Développeur







**Thomas CAMI**  
Développeur



A vintage-style silver microphone with a black grille is mounted on a black tripod. It stands on a mossy log in a dense, sun-dappled jungle. The background is filled with various tropical plants and trees, with sunlight filtering through the canopy. A bright green rectangular box is overlaid on the bottom left of the image.

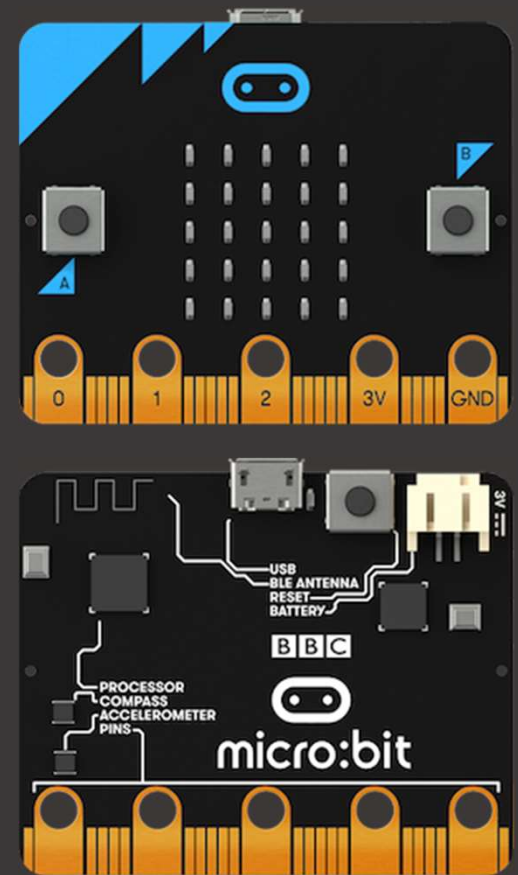
# LA CARTE MICRO:BIT

01



# LA CARTE MICRO:BIT

<b>Date</b>	06 / 07 / 2015
<b>Fabricant</b>	BBC (British Broadcasting Corporation), Micro:bit Educational Fundation
<b>Objectif</b>	Former les jeunes au développement informatique
<b>Technologie</b>	<ul style="list-style-type: none"> <li>• Microcontrôleur 16Mhz, 32 bits</li> <li>• 256ko mémoire flash + 16ko mémoire vive</li> <li>• Matrice led 5 x 5</li> </ul>
<b>Capteurs</b>	<ul style="list-style-type: none"> <li>• 2 boutons + 1 bouton sensible</li> <li>• 1 thermomètre, 1 boussole</li> <li>• 1 capteur de mouvement 3D</li> <li>• 1 micro + haut-parleur</li> <li>• Ports GPIO (x20) + bluetooth</li> </ul>
<b>Système d'exploitation</b>	Zephyr OS
<b>Langages</b>	Bloc, Javascript, Python
<b>Alimentation</b>	<ul style="list-style-type: none"> <li>• USB 5V</li> <li>• Piles (3V)</li> </ul>
<b>Accessoires</b>	<ul style="list-style-type: none"> <li>• Lumière : bande led / matrice led, feu tricolore</li> <li>• Moteur : servo-moteur, barrière</li> <li>• Feu tricolore</li> <li>• Capteurs : ultrasons...</li> </ul>







# USAGES

02



# USAGES

## DÉCOUVRIR LE DÉVELOPPEMENT

-  Le langage block pour **néophyte**
-  IDE très visuel
-  Accessible aux enfants
-  Documentation

## APPRENDRE UN LANGAGE INFORMATIQUE

-  Python  
Javascript  
Typescript
-  Strict minimum

## SE PERFECTIONNER EN ALGO

-  Environnement minimal
-  Faibles capacités (CPU + mémoire)
-  Entretien d'embauche

## EXPÉRIMENTER SES IDÉES

-  Nombreux capteurs
-  Ports GPIO
-  Pas cher
-  Librairies disponibles



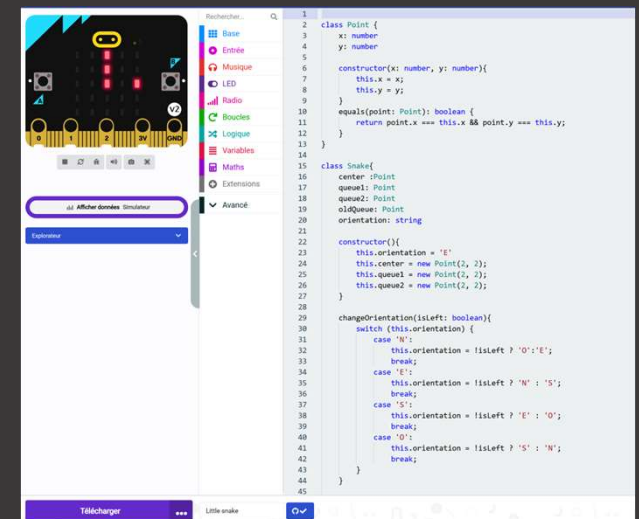
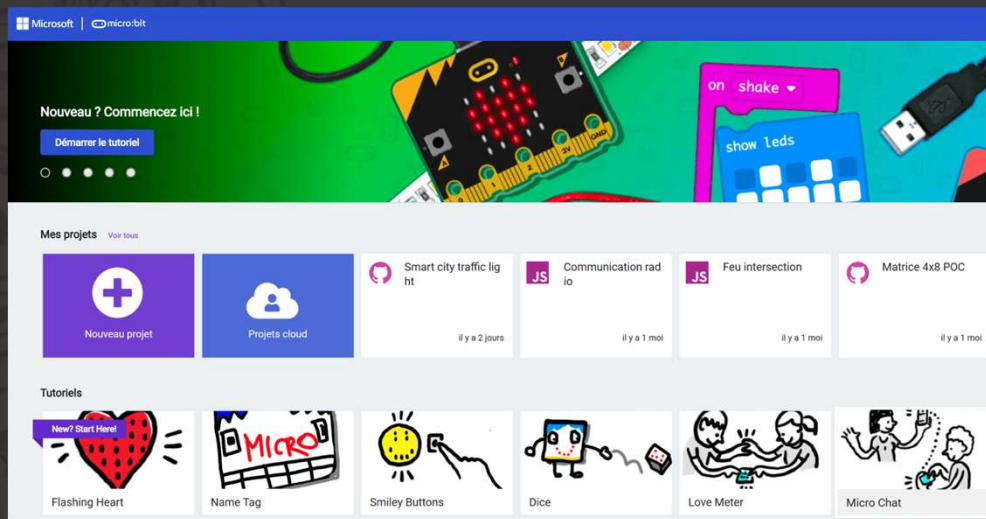
# EXERCICES

03



# PLATEFORME DE DÉVELOPPEMENT

- <https://makecode.microbit.org>
- Compte pour sauvegarder son code
- Simulateur de carte affiché pour tester le code
- Envoi du code sur la carte en USB
- Passage du bloc au javascript ou python
- Librairie de fonctions et auto-complétion
- Intégration de bibliothèques externes
- Création d'une librairie facilitée

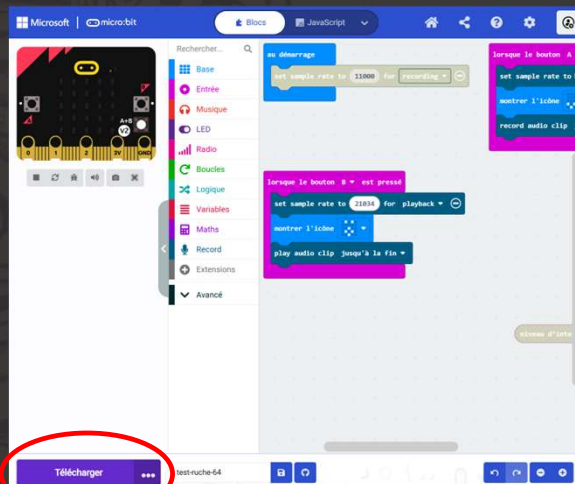




# DÉPLOYER SUR LA CARTE

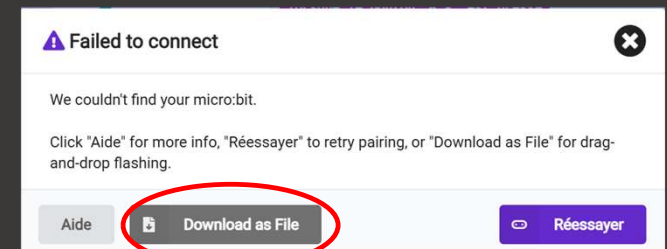
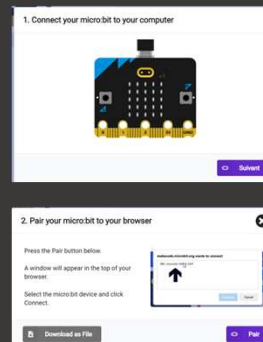
## Windows

- Cliquer sur Télécharger
- Suivre les étapes
- Carte reconnue lors de l'appairage
- La carte est à jour



## Linux

- Carte non reconnue par la plateforme
- Télécharger le binaire
- Le déplacer directement sur la carte (UMS)
- La carte redémarre et exécute le nouveau code



# PROGRAMME

<https://jotitan.github.io/microbit-devox-2025>

Découvrir la programmation 

Démo : faire des katas 

La bille qui roule 

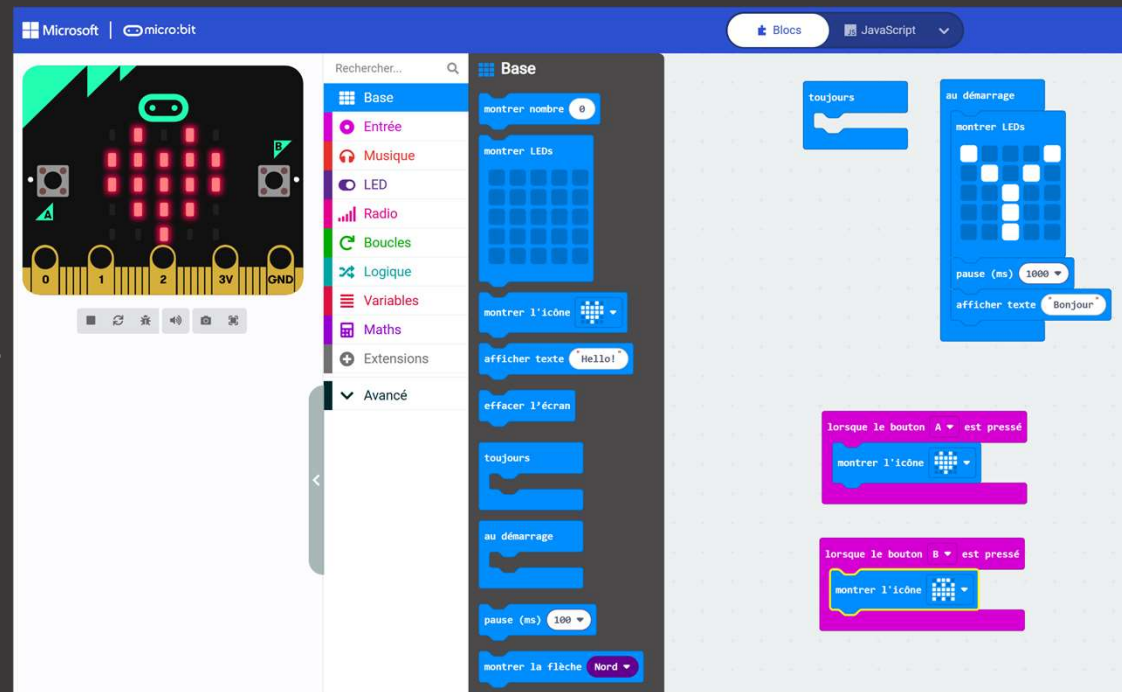
Icônes par radio 

Tu chauffes tu brules  




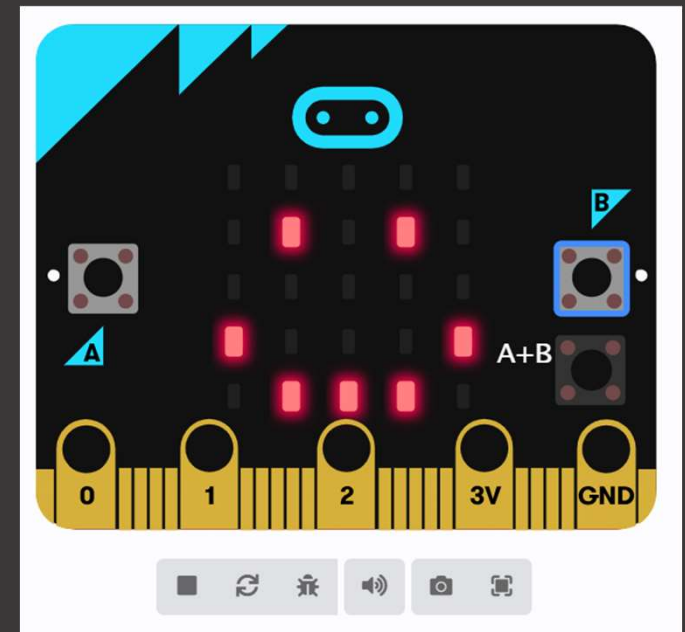
# DÉCOUVRIR LA PROGRAMMATION

- **Langage** : bloc
- **Objectif** : manipuler du bloc comme un enfant
- **Exercice** :
  - Au démarrage, afficher une icône de cœur
  - Quand on appuie sur le bouton A, incrémenter un compteur et l'afficher
  - Quand on appuie sur le bouton B, réduire ce compteur et l'afficher
  - Quand on appuie sur les deux boutons, remettre le compteur à 0 et l'afficher



# KATA D'ALGORITHMIE / TEST ENTRETIEN

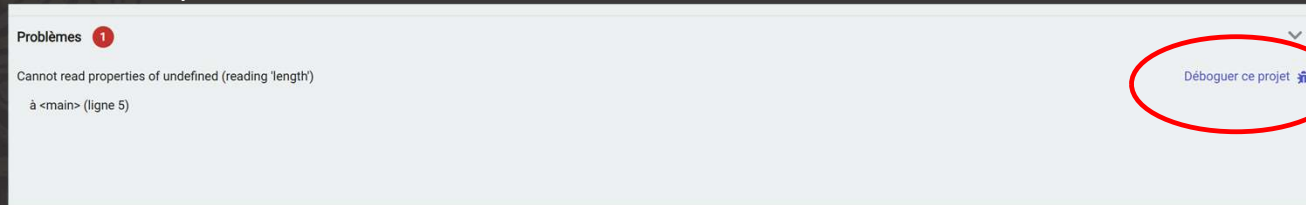
- **Langage** : Javascript
- **Lien** :  <https://github.com/jotitan/microbit-tests-kata>
- **Objectif** : kata court pour s'améliorer en algorithmie
- **Contexte** :
  - La carte micro:bit dispose d'une matrice Led 5x5 permettant de facilement visualiser un tableau
- **Démarrage** :
  - Librairie Github à importer (implémente le bouton A et B)
  - Une fonction à implémenter *loadImplementations*
- **Exercice** :
  - Lire le readme 😊
- **Lancement**
  - Le bouton A affiche une nouvelle entrée
  - Le bouton B lance le test en retournant, si ça marche 😊 + résultat, sinon ☹️



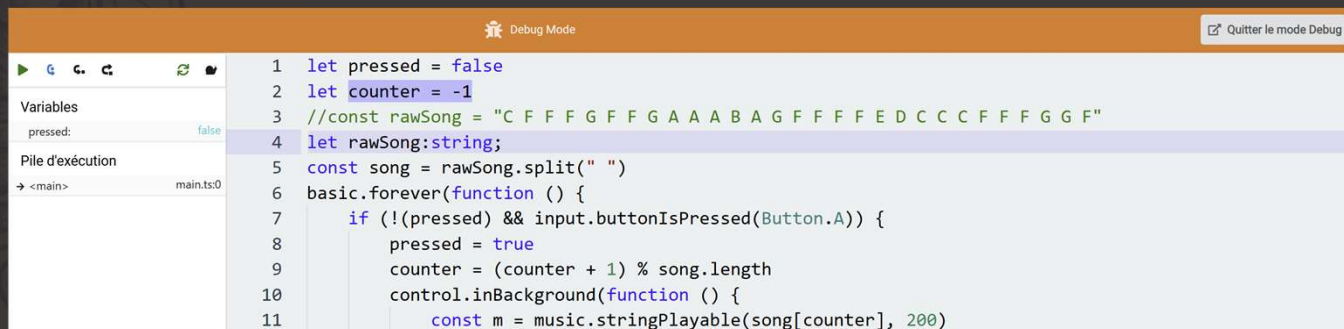


# DEBUGGER SUR MICRO:BIT

- Le Classique : F12 et console.log(message);
- Mode débog :
  - Quand ça plante



- L'option « déboguer ce projet apparaît »
- Mode debug normal : points d'arrêt, pile d'exécution et variables



# IOT : LA BILLE QUI ROULE

- **Langage** : Javascript
- **Objectif** : Simuler une bille sur un plateau qui bouge
- **Lien** :  <https://github.com/jotitan/microbit-rolling-ball>
- **Exercice** :
  - Utiliser la détection de l'orientation de la carte
  - Etablir





# RADIO SUR MICRO:BIT

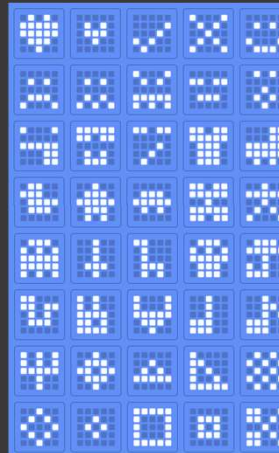
- Fonctionne en radio et bluetooth
- ⚠ Attention aux interférences
- Fréquence
  - Bande de fréquence : entre 2.4Ghz et 2.5Ghz
  - 84 pas de fréquences
  - `radio.setFrequency(freq) // 0-83`
- Group
  - Permet le filtrage des messages
  - 256 valeurs
  - `radio.setGroup(group); // 0-255`
- Fonctions de communication
  - Chaîne de caractères : 19 caractères max
  - Nombre : nombre décimal ou entier
  - Couple chaîne de caractères (8 caractères max) et un nombre
  - Buffer (19 octets max)
  - Emission : `sendXXXXX` où xxxx est le type
  - Réception : `onReceivedXXXXX` où xxxx est le type
  - Détail du paquet reçu : `receivedPacket`



Documentation : <https://makecode.microbit.org/reference/radio>

# IOT : COMMUNICATION RADIO

- **Langage** : Javascript
- **Objectif** : Faire communiquer les micro:bit par radio. Chacun peut choisir son canal, son icône, et l'envoyer aux autres.
- **Lien** :  <https://github.com/jotitan/microbit-radio-icons>
- **Etapes** :
  - Sélection d'un canal de communication
  - Sélection d'une icône
  - Envoi de l'icône
  - Affichage d'une icône reçue sur le canal



# IOT : JEU DU « TU CHAUFFES TU BRULES »

- **Langage** : Javascript
- **Objectif** : Trouver les balises cachées dans la salle
- **Lien** : <https://github.com/jotitan/microbit-burn-cold-game>
- **Exercice** :
  - Plusieurs émetteurs sont présents dans la salle et envoient des messages sur les canaux 4 à 9 toutes les 500 ms
  - Ecrire un récepteur qui utilise la puissance de réception du message pour estimer relativement la distance
  - Afficher sur l'écran des indices pour évaluer la distance de la balise
  - Utiliser les boutons pour changer de groupe radio pour trouver les autres balises







**Merci**

