GOTRONIC
ROBOTIQUE ET COMPOSANTS ÉLECTRONIQUES

Le guide pour bien démarrer sur Micro:Bit

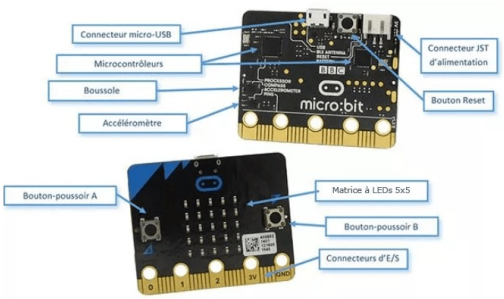
Présentation de la carte

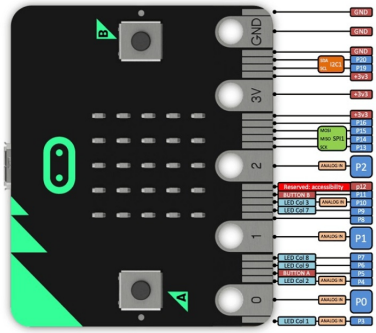
La carte micro:bit est un nano-ordinateur créée pour la promotion de la programmation dans l'éducation .

Elle est équipée d'un processeur ARM, de deux boutons, d'une matrice à LEDs 5×5, de 22 broches,
d'un accéléromètre, d'un magnétomètre, d'un capteur de température et d'une antenne Bluetooth,
ce qui en fait une carte parfaite pour les objets connectés. Elle peut également communiquer avec d'autres cartes.

Elle est très simple d'utilisation, mais peut également s'intégrer dans des applications complexes comme des robots.

Caractéristiques générales	
CPU	ARM Cortex M0 - 32 bits
Mémoire flash	256 ko
Mémoire RAM	16 ko
Bluetooth	4.0
Cadencement	16 MHz
Capteurs	
Accéléromètre	MMA8652
Magnétomètre	MAG3110
Température	De -5 à 50°C
Entrée / Sorties	
Alimentation	3V (USB, broches, connecteur
Micro USB	Pour la programmation
E/S numériques	19 broches
E/S analogiques	6 broches
Autres E/S	Série, I2C, SPI





Programmation

La carte micro:bit se programme à l'aide de différentes interfaces et différents langages, ce qui la rend facile d'accès.

Elle peut se programmer en blocs, en JavaScript et en Python.

Pour coder en blocs ou en JavaScript, il faut utiliser l'éditeur MakeCode.

Pour coder en Python, il faut utiliser l'éditeur Python de micro:bit ou le logiciel Mu.

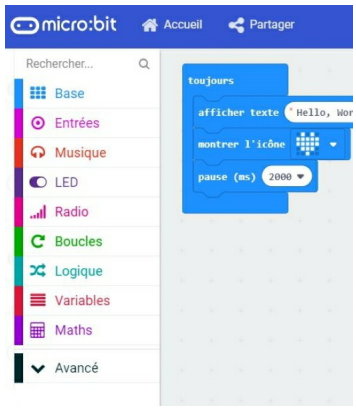
L'éditeur MakeCode

Programmation en blocs

En haut au centre de la page se trouve deux boutons : ***Blocs*** et ***JavaScript***. Assurez-vous d'être dans ***Blocs***.

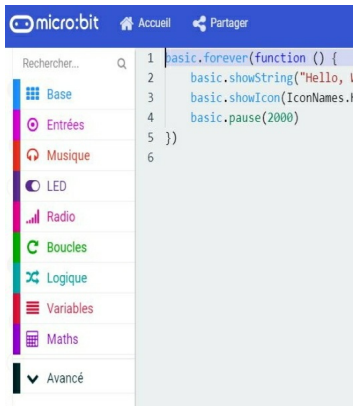
Pour construire votre programme, il suffit de glissez le bloc qui vous intéresse dans la zone de programmation. Pour savoir comment hiérarchiser et configurer les blocs, référez-vous à notre **notice en bas de page**.

Si l'un de vos blocs apparaît grisé, c'est qu'il a été placé à un mauvais endroit.
Si l'un de vos blocs est marqué par le symbole ⚠, c'est qu'il y a une erreur.



Programmation en JavaScript

En haut au centre de la page se trouve deux boutons : ***Blocs*** et ***JavaScript***.
Assurez-vous d'être dans ***JavaScript***.
Vous pouvez directement écrire votre code en JavaScript.
Vous pouvez également vous aider du bandeau de blocs pour que l'éditeur intègre directement le code qui vous intéresse. L'éditeur est doté d'un système d'auto-complétion du code.
Si votre code apparaît souligné en rouge, c'est que votre code est erroné.



L'éditeur gère automatiquement la conversion de la programmation en blocs vers le JavaScript et inversement.
Si vous voulez voir à quoi ressemble votre code en blocs en JavaScript, il suffit de cliquer sur le bouton ***JavaScript*** en haut au centre (cliquez sur ***Blocs*** si vous voulez le cas inverse).
Cela peut être un bon moyen d'apprendre le JavaScript si vous êtes néophyte avec ce langage.

Lorsque votre programmation est terminée, cliquez sur ***Télécharger*** pour télécharger votre programme. Votre fichier sera au format **.hex**.

Procédez au transfert [voir plus bas].

L'éditeur Python

Rendez-vous sur l'éditeur Python de micro:bit.

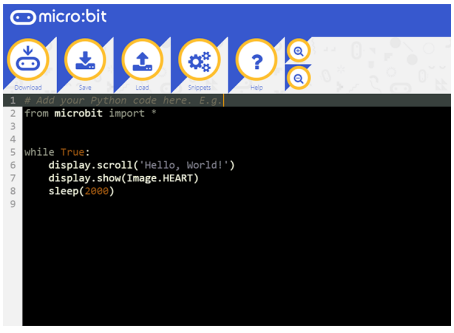
Vous pouvez directement écrire votre code en Python. Vous pouvez également vous aider du bouton ***Snippets***
pour que l'éditeur intègre directement le code qui vous intéresse.

Si vous avez des doutes sur une syntaxe de code, cliquez sur ***Help*** puis sur ***Documentation***.

Vous avez la possibilité de sauvegarder votre code code en fichier Python (**.py**) en cliquant sur ***Save***.
Lors du téléchargement, il se peut que votre navigateur vous demande une confirmation d'enregistrement.

Pour pouvoir avoir le fichier **.hex** qui permettra de programmer votre carte micro:bit, vous devez cliquer sur ***Download***.

Procédez au transfert [voir plus bas].



Le logiciel Mu

Télécharger le logiciel Mu ici en choisissant la version adaptée à votre système d'exploitation et installez-le.

Lancez le logiciel, cliquez sur ***BBC micro:bit*** puis sur ***OK***.

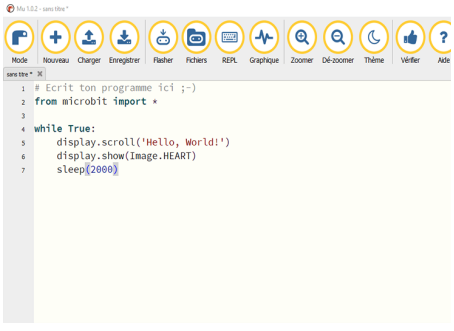
Vous pouvez directement écrire votre code en Python. Le logiciel est doté d'un système d'auto-complétion du code.

Pour vérifier qu'il n'y a aucune erreur dans votre code, cliquez sur ***Vérifier.***

Si vous avez besoin d'un moniteur série, cliquez sur ***REPL***.

Si vous avez besoin d'un retour graphique, cliquez sur ***Graphique.***

Pour sauvegarder votre code en fichier Python (**.py**), cliquez sur ***Enregistrer.***



Pour programmer votre carte mico:bit avec votre code, reliez votre carte avec un câble micro-USB / USB à votre ordinateur et cliquez sur ***Flasher***. **Le transfert a été effectué**, vous pouvez déconnecter votre carte et l'utiliser.

Transfert

Reliez votre carte micro:bit avec un câble micro-USB / USB à votre ordinateur, celui-ci va reconnaître votre carte comme étant une clé USB. La LED orange de votre carte est allumée.

Copier-coller votre fichier **.hex** dans votre carte.

Votre carte va se déconnecter momentanément de votre ordinateur avant de se reconnecter. Pendant ce laps de temps, la LED orange va clignoter rapidement puis redeviendra fixe.

Une fois ce processus terminé, votre carte est programmée. Vous pouvez la déconnecter de votre ordinateur et l'utiliser.

Notice

Les blocs peuvent être liés entre eux de 2 manières :

- Les uns en dessous des autres. Dans ce cas, l'ordre dans lequel les blocs sont empilés est également l'ordre dans lequel ils seront exécutés.
- Les uns dans les autres. Dans ce cas, les blocs se complémentent.

Il existe 2 façons de personnaliser un bloc :

- la zone de texte libre
- la liste déroulante

Un bloc qui comporte une zone de texte vous permet d'y écrire ce que vous voulez. Cela peut être un nombre, un caractère ou un texte.

Un bloc qui comporte une liste déroulante vous donne un certain nombre de choix prédéfinis en rapport avec ce bloc. Par exemple, dans les fonctions mathématiques, vous y trouverez sinus – cosinus – tangente – et leurs inverses.

Présentation du bandeau de programmation :

- **Base*** = gestion de la matrice à LEDs, gestion du temps, boucles de base (démarrage + répète à l'infini)
- **Entrées*** = instructions pour les différents capteurs de la carte
- **Musique** = gère l'interface audio de la carte
- **LED*** = fonctions avancées de la matrice à LEDs
- **Radio*** = gère la communication radio
- **Boucles=** gère les répétitions d'instructions (avec compteur ou suivant une condition)
- **Logique** = comprend toutes les conditions logiques (ET, OU, NON, , SI, SINON, ...)
- **Variables** = permet la création et la gestion de variables
- **Maths** = comprend toutes les opérations mathématiques (+, -, sinus, PI, max, random, ...)
- **Avancé** = affiche plus de catégories dans le bandeau
- **Fonctions** = permet la création, l'appel et le retour de fonction
- **Listes** = permet la création et la gestion de liste d'éléments
- **Texte** = permet la création et la gestion de textes
- **Jeu*** = gère la plupart des manettes conçues pour micro:bit
- **Images** = autres fonctions avancées de la matrice à LEDs
- **Broches*** = gestion des entrées / sorties
- **Communication Série*** = pour utiliser le moniteur série
- **Contrôle*** = pour gérer l'électronique interne de la carte (détection de front, reset, ...)
- **Extensions** = permet d'ajouter des bibliothèques de blocs / code JavaScript pour certains composants
- ***Plus** : affiche la suite d'une catégorie