

# Micro:Bit

## C'est quoi cette carte ?



Vous pouvez utiliser votre BBC micro:bit pour toutes sortes de créations géniales, allant des robots aux instruments de musique – les possibilités sont infinies.

### Ce qu'en dit le créateur

La BBC déclare : *“Le Micro:bit est un ordinateur de poche que vous pouvez programmer, personnaliser et contrôler afin de rendre concrets vos idées numériques, des jeux et des applications. Mesurant 4 cm par 5 cm, il est conçu pour être amusant et facile à utiliser ; les utilisateurs peuvent créer toutes sortes de choses, que ce soient des jeux, des animations ou des messages défilants aussi bien à l'école, à la maison qu'en déplacement. Tout ce dont vous avez besoin est de l'imagination et de la créativité. Le BBC Micro:bit est entièrement programmable. Cela signifie que chacune de ses diodes électroluminescentes peut être programmée individuellement comme peuvent également l'être les boutons, les entrées et sorties, l'accéléromètre, le magnétomètre et la connexion Bluetooth”*

### Programmer le Micro:bit

Le Micro:bit se programme de façon simple dans un environnement d'édition simple via un site Web. Les langages utilisés par les éditeurs sont : Le JavaScript, Microsoft Block, Microsoft Touch Develop et Python.

L'utilisateur crée et écrit le programme sur l'éditeur de son choix. Il l'envoie à un serveur distant qui se chargera de le compiler pour être ensuite téléchargeable par l'utilisateur et transféré sur le Micro:bit. Point fort de cette innovation, la programmation via Android et IOS par Bluetooth.

[Pour commencer avec un Micro:bit FR](#)

[Fonctionnalités](#)



Ton micro:bit dispose des spécificités techniques suivantes :

Les principaux composants :

1. Microcontrôleur Nordic nRF51822
2. Processeur Cortex ARM-M0 (architecture ARMv6-M) RISC 32-bits à 16 MHz
3. Mémoire RAM : 256kB flash
4. Mémoire statique : 16kB
5. Connectique bluetooth 4.0 basse énergie/2.4 GHz maître/esclave ;
6. Un connecteur pour piles ;
7. Un porte-piles (2 LR03/AAA) ;
8. Une prise femelle micro-USB (permet l'alimentation du Micro:bit ainsi que le transfert des codes);
9. Un bouton de ré-initialisation, libellé "R" pour "Reset" ;
10. Une LED jaune du système ;
11. Deux boutons programmables, libellés "A" et "B" sur la carte ;
12. Une matrice carrée de 25 LED (5 x 5) rouges programmables pouvant servir d'affichage, notamment pour des motifs animés, du texte alphanumérique déroulant ;
13. Un capteur de mouvement 3D Freescale MMA8652 via le bus I2C ;
14. Un magnétomètre 3D Freescale MAG3110 via le bus I2C ;
15. 5 ports d'entrée-sortie en forme d'anneau. Chacune des 5 E/S est programmable, pour être traitée soit en analogique, soit en numérique. Les anneaux sont compatibles avec des prises crocodile ou des fiches banane 4mm ;
16. 2 ports : Puissance référencé "PWR" & masse référencé "GND" fournissant 3 Volts, éventuellement pour alimenter un autre appareil, libellés "3V" et "GND" sur la carte ;
17. et 3 autres, référencés "Pins" P1 à P3, libellés "1", "2", "3" sur la carte ;
18. Un connecteur latéral à 20 broches, à connexion standard, référencés "Pins" P3 à P22. Ceci permet de connecter à un appareil, comme un Arduino, Galileo, Kano et Raspberry

[Connecteur Micro:Bit V1 et V2](#)

[Connecteur Micro:Bit](#)

## Pour coder

[En javascript Block et texte FR](#)

[En Python Texte](#)

[MicroPython FR](#)

[Créer des Scripts](#)

[Micro:bit : Blocks pour Python EN](#)

## Tutoriels

[Tutos Micro bit](#)

[Liens Micro:Bit avec un RaspberryPi](#)

[Le Micro:bit Comment l'utiliser ?](#)

[Micro:bit with Arduino EN](#)

[BBC micro:bit + Arduino IDE\\_EN](#)

[Guide référence du langage des blocs FR](#)

## Démarrage

[Prise en main rapide de Micro:bit](#)

[Utilisation de la carte Micro:bit](#)

[Pilotage Mini-Move Micro:bit](#)

[Programmer un Micro:bit avec Arduino IDE](#)

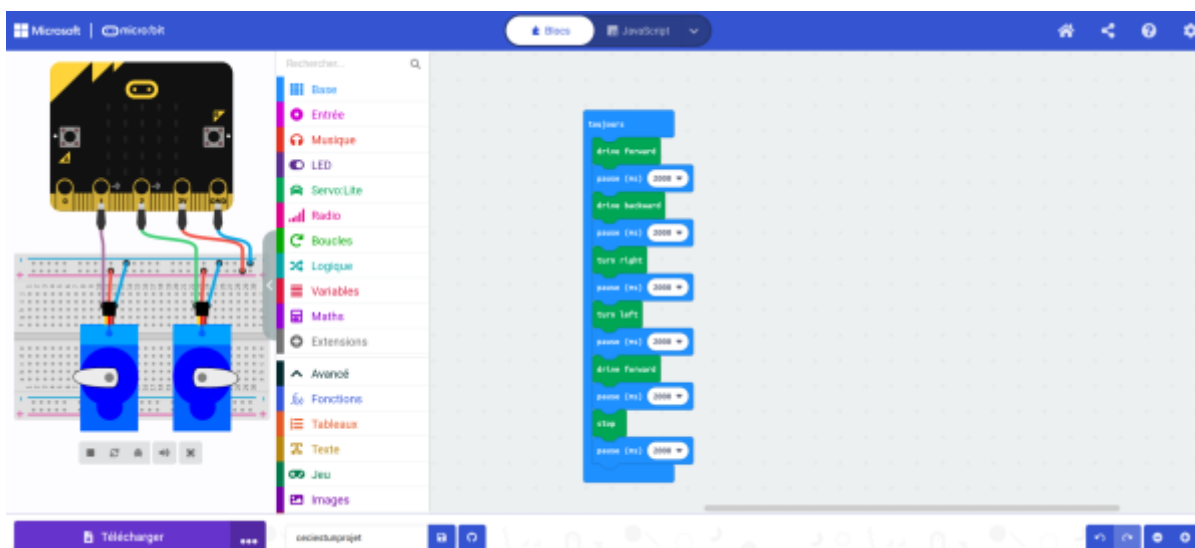
[test Micro:bit leds Neopixel](#)

## Tutos FR

[Tuto doc FR](#)

## Exemples

## Exemples001



## Bitcar



Bitcar

[GitHub BitCar](#)

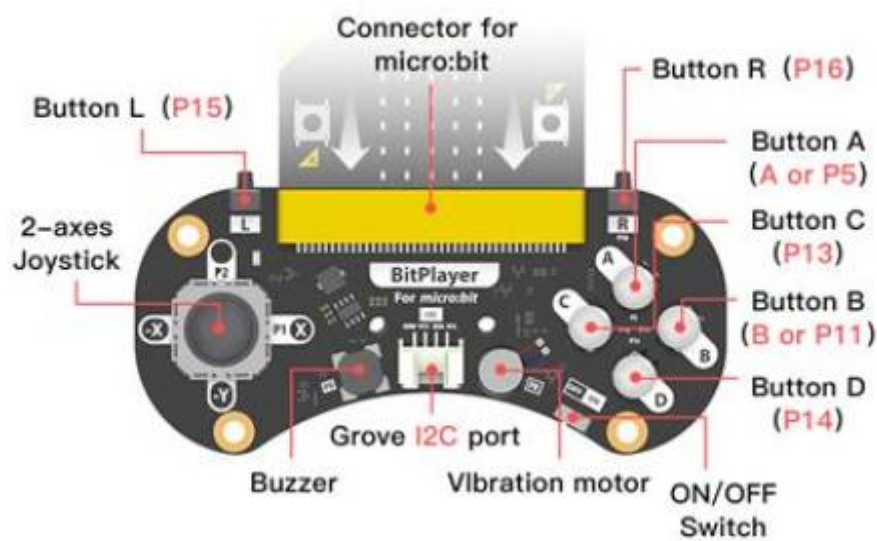
[Bitcar doc](#)

## Extensions

BitCar

<https://github.com/TinkerGen/pxt-BitCar>

## BitPlayer



[Wiki Seedstudio Bit Player EN](#)

[Doc Micro Bit Bit Player EN](#)

[Exemple :Code Bit Player EN](#)

[Doc Seedstudio Bit Player EN](#)

## Extensions

BitPlayer

<https://github.com/TinkerGen/pxt-BitPlayer>

## MicroBit et Arduino



Saviez-vous que l'IDE Arduino peut être utilisé pour programmer le micro:bit ? Vous disposez désormais d'une autre façon d'utiliser cette carte géniale ! Apprenez à configurer Arduino pour programmer votre micro:bit, faire clignoter quelques LED, lire le capteur de température interne, envoyer et recevoir des données via Bluetooth - et même enregistrer des données sur Adafruit.IO !

Le micro:bit est une petite plate-forme d'apprentissage pour enfants alimentée par nRF51 - vous pouvez l'utiliser avec Microsoft MakeCode (programmation par blocs glisser-déposer ou Javascript), micropython ou mbed. Mais nous aimons vraiment utiliser l'IDE Arduino, d'autant plus qu'il existe des milliers de projets existants que vous pouvez utiliser et adapter. De plus, vous pouvez avoir des projets beaucoup plus avancés car vous ne manquerez pas de mémoire (comme vous le feriez avec micropython) et vous pouvez écrire à peu près n'importe quel code que vous voulez (avec MakeCode, vous êtes plus limité à ce qui vous a déjà été fourni, un compromis pour la facilité d'utilisation).

Prenez un microbit et suivez-nous pour découvrir comment vous pouvez faire des choses assez avancées avec votre « bit ! »

## Installer la carte et blink!

### Installer le pilote Windows

Si vous utilisez Mac, Win 10+ ou Linux, ce n'est pas obligatoire ! Ignorez cette étape

### Télécharger Arduino IDE

Vous devrez utiliser l'IDE de bureau. Assurez-vous que vous utilisez la dernière version.

[IDE Arduino](#)

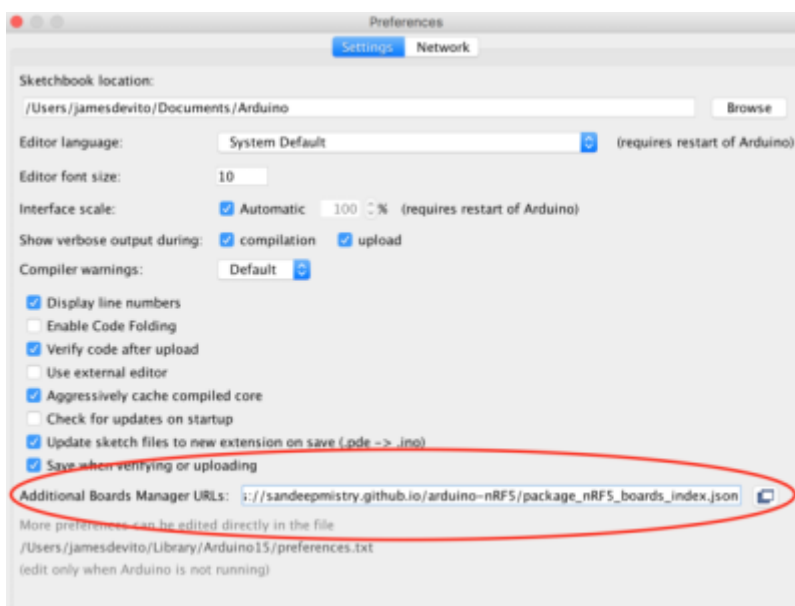
## Installer SoftDevice sur MicroBit

Arduino suppose qu'une radio « softdevice » est déjà installée. Si vous avez utilisé MicroPython avec votre microbit, ce softdevice a été effacé.

La réinstallation est facile, téléchargez [ce fichier hex](#) et faites-le glisser sur votre lecteur MICROBIT

## Ajout de la prise en charge de la carte NRF5x

Le microbit utilise le nRF51 qui n'est pas pris en charge « nativement ». Mais il est facile d'ajouter un support ! micro-ordinateurs\_



Dans Arduino, accédez à Préférences et ajoutez-les :

[https://sandeepmistry.github.io/arduino-nRF5/package\\_nRF5\\_boards\\_index.json](https://sandeepmistry.github.io/arduino-nRF5/package_nRF5_boards_index.json)

dans la zone de texte URL du gestionnaire de cartes supplémentaires. Si ce n'est pas votre premier, assurez-vous de séparer les URL par une virgule.



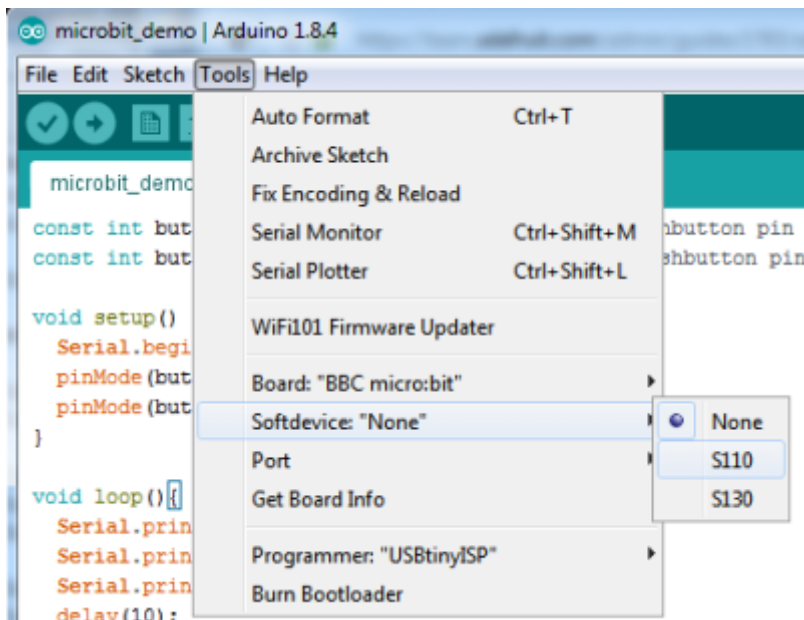
Ouvrez Outils > Carte > Gestionnaire de cartes dans la barre de menu, recherchez nRF5 et installez « Nordic Semiconductor nRF5 Boards » par Sandeep Mistry

### Sélectionnez le tableau et téléchargez

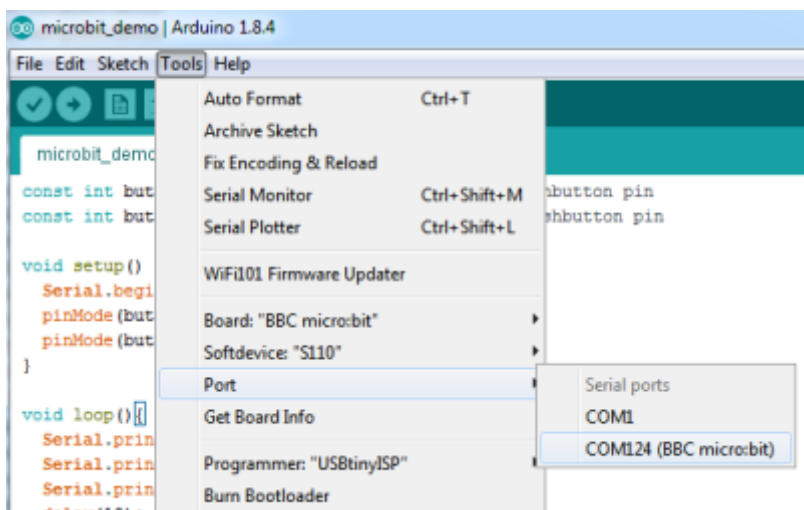
Sélectionnez BBC micro:bit dans le menu Cartes



Régler SoftDevice sur S110



Et définissez le port sur le microbit



Et créez un nouveau croquis avec cette démo clignotante

[demo001MicroBit\\_arduino\\_clign.ino](#)

```
const int COL1 = 3; // Column #1 control
const int LED = 26; // 'row 1' led

void setup() {
  Serial.begin(9600);

  Serial.println("microbit is ready!");

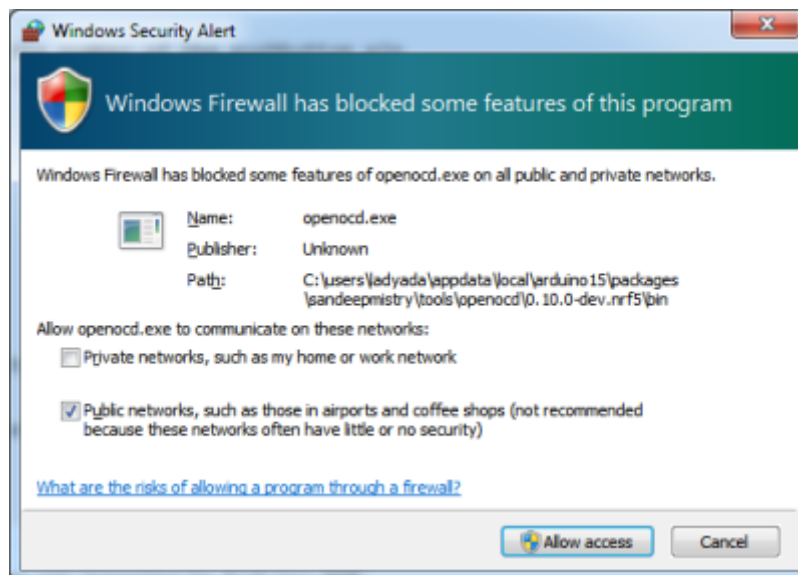
  // because the LEDs are multiplexed, we must ground the opposite side
  // of the LED
  pinMode(COL1, OUTPUT);
  digitalWrite(COL1, LOW);

  pinMode(LED, OUTPUT);
}
```

```
}  
  
void loop(){  
  Serial.println("blink!");  
  
  digitalWrite(LED, HIGH);  
  delay(500);  
  digitalWrite(LED, LOW);  
  delay(500);  
}
```

Cliquez sur Télécharger !

Si vous recevez un avertissement concernant openocd , approuvez l'accès afin qu'il puisse télécharger le code



## Spécifique à Linux

La version openocd du paquet sandeepmistry semble être 32 bits. Si vous utilisez une installation Linux 64 bits, vous pouvez obtenir cette erreur lorsque vous essayez de télécharger des croquis :

```
{dossier Arduino}/packages/sandeepmistry/tools/openocd/0.10.0-dev.nrf5/bin/openocd : erreur lors du chargement des bibliothèques partagées : libudev.so.1 : impossible d'ouvrir le fichier objet partagé : aucun fichier ou répertoire de ce type
```

Pour contourner ce problème, installez la version 32 bits de la bibliothèque libudev avec :

# Statistiques

Cet page a été consultée : Aujourd'hui: 7 Hier: 9 Jusqu'à maintenant: 3861

From:

<https://www.magenealogie.chanterie37.fr/www/fablab37110/> - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:

<https://www.magenealogie.chanterie37.fr/www/fablab37110/doku.php?id=start:micro-bit&rev=1728656581>

Last update: **2024/10/11 16:23**

