

# Micro:bit

(page créée le 20 mai 2022, en cours de rédaction)

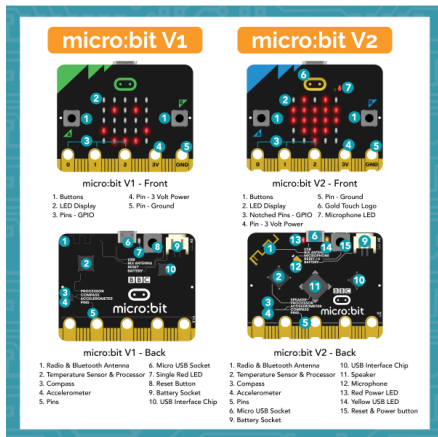
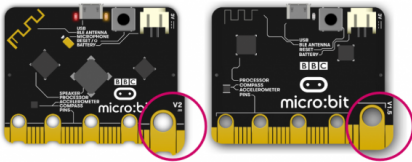
## Caractéristiques principales

Caractéristiques détaillées en fonction de la version : <https://tech.microbit.org/hardware/>

Il existe deux versions principales de micro:bit.

## Notes

- les connecteurs en anneaux sont utilisables avec des fiches bananes
- Les broches peuvent agir en bouton tactile capacitifs



## micro:bit V1.5 vs V2

Features/Specs	micro:bit v1.5	micro:bit v2
Release Date	NA	13th Oct 2022
MCU or Processor	Nordic Semiconductor nRF51822	Nordic Semiconductor nRF52833
MCU Core Architecture	ARM Cortex-M0 32-bit	ARM Cortex-M4 32-bit (FPVU)
MCU Flash Size	256KB	512KB
RAM Size	16KB	128KB
MCU Clock	16MHz	64MHz
USB Interface Processor	NXP KL26Z, 16KB RAM	NXP KL17Z, 32KB RAM
Microphone, MIC	None	MEMS Microphone, LED Indicator
Speaker	None	Onboard Piezo Buzzer
Touch Sensitive Logo	None	Touch Sensitive Logo Pad
Wireless	2.4GHz micro bit radio/BLE Bluetooth 4.0	2.4GHz micro bit radio/BLE Bluetooth 5.1
Power	3V via USB, 3V via edge connector or battery port	3V via USB, 3V via edge connector or battery port
Power Indicator LED	NA	Onboard Power Indicator LED
Power Off Button	NA	Onboard Power Button (Push and Hold)
Current for External	3V, 50mA	3V, 200mA
Motion Sensor	ST LSM303	ST LSM303
Edge Connector	25-pin, 3 dedicated GPIO, PWM, I2C, SPI, Power, and etc	25-pin, 4 dedicated GPIO, PWM, I2C, SPI, Power, and etc
Ring Connector	3 (GPIO) + 2 (Power) ring connectors	3 (GPIO) + 2 (Power) ring connectors, notched edge
I2C	Shared I2C Bus	Dedicated I2C Bus for peripherals
Software/IDE	C++, makecode, Python, Scratch	C++, makecode, Python, Scratch
Size	5cm (w) x 4cm (h)	5cm (w) x 4cm (h)

## Brochage / Pinout

# micro:bit

## PINOUT

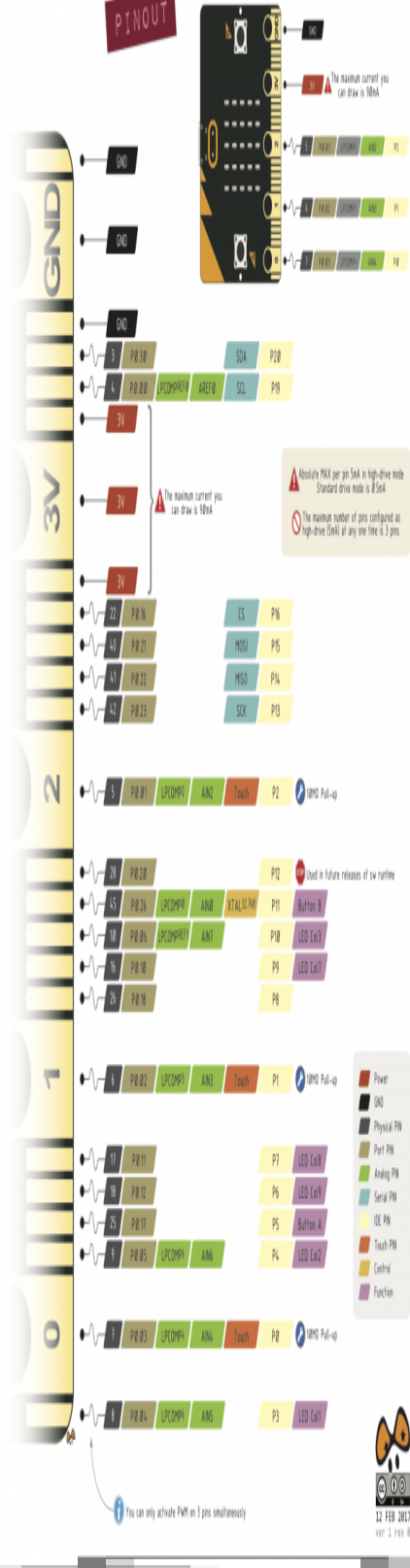


Schéma par PighiXXX, CC BY-SA

## Programmer la carte

En ligne, par bloc, en javascript ou micropython, avec l'éditeur makecode

- <https://makecode.microbit.org/?lang=fr>

En ligne / en local, par bloc avec Scratch

- <https://scratch.mit.edu/microbit>

En local, en code micropython, avec l'éditeur Mu

- <https://codewith.mu/>
- <https://microbit-micropython.readthedocs.io/en/latest/index.html>

En local, en code C, avec l'IDE arduino

- <https://learn.adafruit.com/use-micro-bit-with-arduino?view=all>

## Utilisation

### Connexion série sur linux

A minima, on peut utiliser screen pour recevoir les données

```
ls /dev/ttyACM*          # chercher le port
screen /dev/ttyACM0 115200
# on arrête screen avec CTRL-A, puis K (pour Kill)...
# sur l'utilisation de screen, voir https://www.tecmint.com/screen-command-examples-to-manage-linux-terminals/
```

Des exemples pour une réception dans processing ici : [https://github.com/emoc/microbit\\_utile](https://github.com/emoc/microbit_utile)

## Extensions

micro:bit + shield grove : [https://wiki.seeedstudio.com/Grove\\_Inventor\\_Kit\\_for\\_microbit/](https://wiki.seeedstudio.com/Grove_Inventor_Kit_for_microbit/)

## Ressources

<https://github.com/carlosperate/awesome-microbit>

Utilisation micro:bit + processing :

- [https://github.com/AtelierNum/workshop\\_code\\_creatif\\_1819/blob/master/Floody/Floody\\_Code/Floody\\_Code.pde](https://github.com/AtelierNum/workshop_code_creatif_1819/blob/master/Floody/Floody_Code/Floody_Code.pde)
- [https://github.com/AtelierNum/workshop\\_code\\_creatif\\_1819/blob/master/Floody/Floody\\_Code/microsensor.pde](https://github.com/AtelierNum/workshop_code_creatif_1819/blob/master/Floody/Floody_Code/microsensor.pde)

Compteur de pas avec l'accéléromètre intégré :

<https://microbit.org/fr/projects/make-it-code-it/low-energy-step-counter/?editor=python>

Data logging avec microbit v2 : <https://microbit.org/get-started/user-guide/data-logging/>

Article extrait de : <http://lesporteslogiques.net/wiki/> - **WIKI Les Portes Logiques**

Adresse : <http://lesporteslogiques.net/wiki/materiel/microbit>

Article mis à jour: **2022/05/24 15:21**