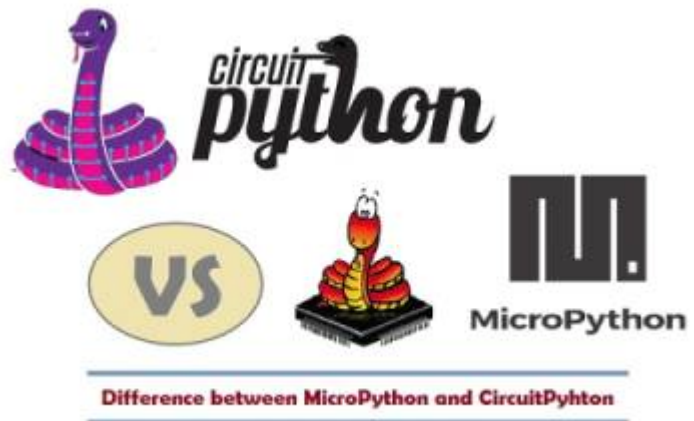




MicroPython - Circuit Python - Généralité



[Mise à jour le : 3/6/2023]

- **Ressources**





- [Site MicroPython](#)
- [Site CircuitPython](#)





1. Micropython vs CircuitPython

MicroPython est une implémentation simple et efficace du langage de programmation **Python 3**, qui inclut un petit sous-ensemble de la bibliothèque standard Python et qui est optimisée pour fonctionner sur des microcontrôleurs. Il est suffisamment compact pour s'adapter à **256 ko d'espace de code** et à **16 ko de RAM**.

CircuitPython est un **dérivé open source** du langage de programmation **MicroPython** destiné aux étudiants et aux débutants. Le développement de CircuitPython est soutenu par **Adafruit Industries**. Il s'agit d'une implémentation logicielle du langage de programmation Python 3, écrit en C.

2. Quelques cartes à microcontrôleur programmables en MicroPython

| ESP8266 | ESP32 | Raspberry Pi Pico W (RP2W) | BrainPad Pulse et Tick |
|---|---|--|---|
|  |  |  |  |
| Doc. MicroPython | Doc. MicroPython | Doc. MicroPython | Site BrainPad |

| Pyboard v1.1 | Wipy | Flip&Click SAM3X | Micro bit |
|--|--|---|--|
|  |  |  |  |
| Doc. MicroPython | Doc. MicroPython | MikroElektronica | BBC |

3. Les bibliothèques

3.1 Le module machine

Le module **machine** contient des fonctions spécifiques au matériel d'un microcontrôleur particulier. Ces fonctions permettent d'accéder aux composants matériels. En particulier : - l'unité centrale, - les temporisateurs, - les bus et - les broches d'entrée/sortie.

Exemples

*.py

```
import machine # Importe tout le module
from machine import Pin # Importe uniquement la classe Pin pour accéder
aux E/S

led = Pin(2, Pin.OUT)
led.value(1)
```

3.2 Le module time

Le module **time** fournit des fonctions liées au temps. La classe **sleep** de ce module met en pause l'exécution du programme en cours pendant le nombre de secondes spécifiées.

Exemples

*.py

```
from machine import Pin
from time import sleep

led = Pin(2, Pin.OUT)

while True:
    led.value(1)
    sleep(0.5)
    led.value(0)
    sleep(0.5)
```

4. WebREPL

- **Source** : [Installer micropython sur un ESP8266/32](#)

From:
<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:
<http://webge.fr/dokuwiki/doku.php?id=python:micropython:generalites&rev=1685785254>

Last update: **2023/06/03 11:40**

