

# Utilisation du Raspberry Pi Pico avec MicroPython

Dans cet article de cours, nous allons explorer comment utiliser le Raspberry Pi Pico avec MicroPython et vous montrer des exemples de programmes pour lire et écrire des valeurs sur les broches GPIO (General-Purpose Input/Output).

3GMS

6GMS

5GMS

5TTR

6TTR

4TTR

 Découverte

Le Raspberry Pi Pico est un microcontrôleur puissant et abordable qui peut être programmé avec MicroPython, un langage de programmation convivial pour les débutants.

## Prérequis

Avant de commencer, assurez-vous d'avoir les éléments suivants à votre disposition :

- Un Raspberry Pi Pico
- Un ordinateur avec MicroPython installé (vous pouvez télécharger MicroPython sur le site officiel)
- Un câble micro USB pour la connexion entre l'ordinateur et le Raspberry Pi Pico

## Installation de MicroPython

Ceci est une procédure d'installation manuelle que nous n'avons pas vue en classe. Je la laisse pour que vous sachiez que ça existe, mais il ne faut pas la connaître pour le cours.

1. Téléchargez la dernière version de MicroPython depuis le site officiel (<https://micropython.org/>).
2. Connectez votre Raspberry Pi Pico à votre ordinateur à l'aide du câble micro USB. Le Pico devrait être reconnu comme un périphérique de stockage USB.
3. Copiez le fichier MicroPython que vous avez téléchargé sur le Pico. Il doit avoir une extension ".uf2".
4. Déconnectez le Pico de votre ordinateur, puis reconnectez-le. Vous verrez un nouveau périphérique de stockage appelé "RPI-RP2".
5. Supprimez le fichier "CURRENT.UF2" s'il existe et renommez votre fichier MicroPython téléchargé en "CURRENT.UF2". Cela permettra au Pico de démarrer automatiquement avec MicroPython.
6. Déconnectez à nouveau le Pico de l'ordinateur, puis reconnectez-le. Il devrait maintenant exécuter MicroPython.

# Programme de base

---

Commençons par un exemple simple de programme en MicroPython pour allumer et éteindre une LED connectée à une broche GPIO du Raspberry Pi Pico.

```
1 | import machine
2 | import utime
3 |
4 | # Définir la broche GPIO à utiliser
5 | led = machine.Pin(25, machine.Pin.OUT)
6 |
7 | # Boucle infinie pour clignoter la LED
8 | while True:
9 |     led.toggle() # Inverse l'état de la LED
10 |    utime.sleep(1) # Attendre 1 seconde
```

Dans ce programme, nous utilisons le module `machine` pour gérer les broches GPIO. Nous définissons la broche 25 comme une sortie (`machine.Pin.OUT`). Ensuite, nous utilisons une boucle infinie pour alterner l'état de la LED entre allumé et éteint toutes les secondes à l'aide de la méthode `toggle()`.

## Lecture de valeurs analogiques

---

Vous pouvez également lire des valeurs analogiques à partir de broches GPIO pour mesurer des grandeurs telles que la luminosité avec un capteur de lumière. Voici un exemple pour lire une valeur analogique à partir de la broche 26 :

```
1 | import machine
2 |
3 | # Définir la broche GPIO à utiliser
4 | analog_pin = machine.Pin(26)
5 |
6 | # Configurer l'ADC (Convertisseur Analogique-Numérique)
7 | adc = machine.ADC(analog_pin)
8 |
9 | # Lire et imprimer la valeur analogique
10 | while True:
11 |     valeur = adc.read_u16() # Lecture de la valeur analogique (0-65535)
12 |     print("Valeur analogique : ", valeur)
```

Dans cet exemple, nous utilisons le module `machine` pour configurer l'ADC (Convertisseur Analogique-Numérique) sur la broche 26. La valeur analogique est lue à l'aide de la méthode `read_u16()` et affichée dans la console.

## Conclusion

---

Le Raspberry Pi Pico avec MicroPython offre une excellente plateforme pour l'apprentissage de la programmation et de l'électronique. Vous pouvez créer une variété de projets intéressants en utilisant les broches GPIO pour contrôler des composants électroniques. N'hésitez pas à explorer davantage et à créer vos propres projets en utilisant ces concepts de base.