

Les Broches du Raspberry Pi Pico en MicroPython

Sur cette page, nous allons explorer les différentes broches du Pico, leur numéro, leur fonction (analogique, PWM, numérique), et comment les utiliser en MicroPython.

3GMS

6GMS

5GMS

5TTR

6TTR

4TTR

📶 Découverte

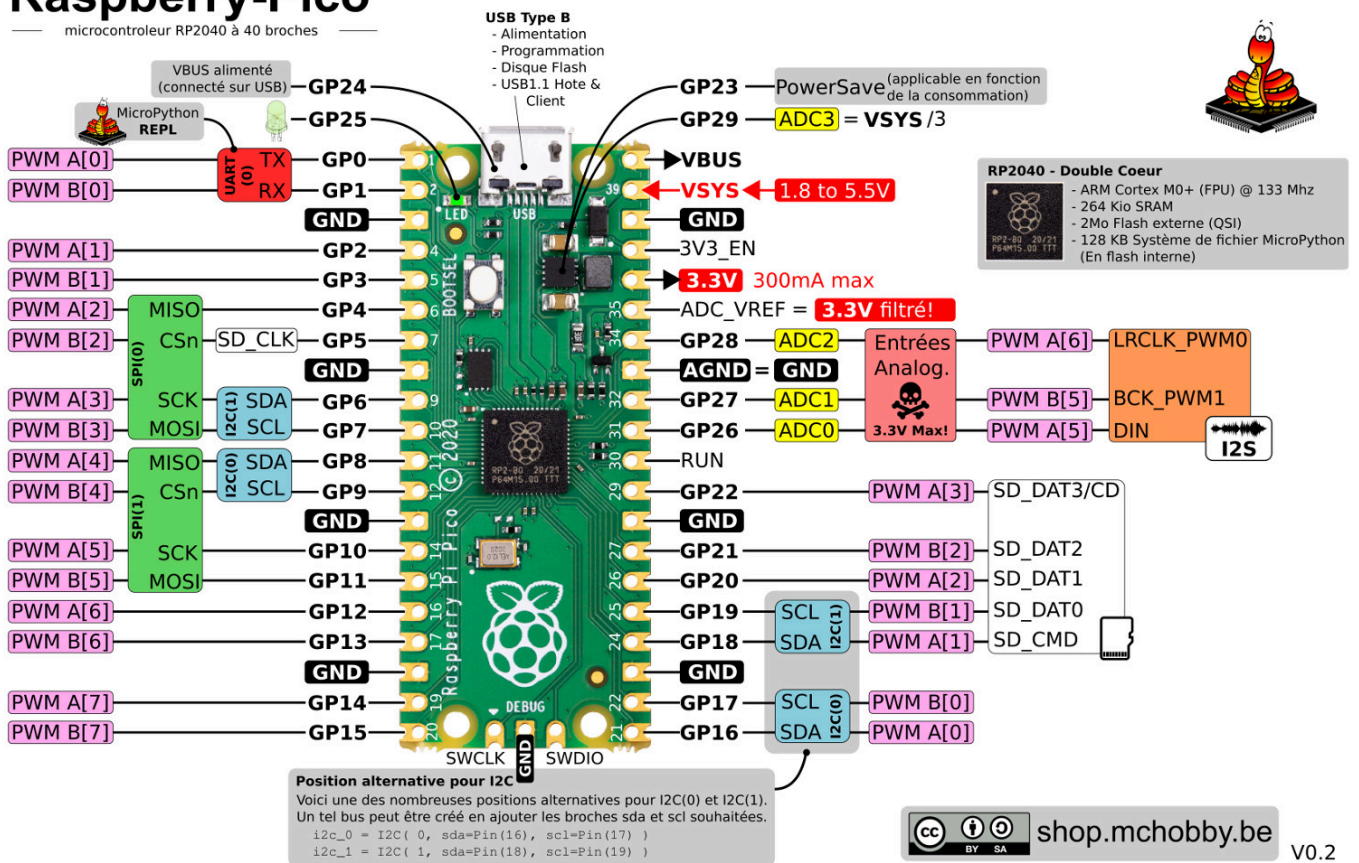
Le Raspberry Pi Pico dispose de plusieurs broches GPIO (General-Purpose Input/Output) qui peuvent être utilisées pour diverses tâches en MicroPython.

Broches GPIO du Raspberry Pi Pico

Le Raspberry Pi Pico possède 26 broches GPIO que vous pouvez utiliser pour interagir avec des composants électroniques et des capteurs. Voici une liste des broches GPIO disponibles sur le Pico :

Raspberry-Pico

microcontrôleur RP2040 à 40 broches



Utilisation des Broches en MicroPython

Configuration d'une broche en mode numérique

Pour configurer une broche en mode numérique (entrée ou sortie), vous pouvez utiliser la bibliothèque `machine` en MicroPython. Par exemple, pour configurer la broche GP0 en tant que sortie, vous pouvez utiliser le code suivant :

```
1 | import machine
2 |
3 | # Configurer GP0 en sortie
4 | broche_gp0 = machine.Pin(0, machine.Pin.OUT)
```

MicroPython offre plusieurs méthodes pour contrôler la tension de sortie :

```
1 | from machine import Pin
2 |
3 | Pin.on() et Pin.off()
4 | Pin.high() et Pin.low()
5 | Pin.value(valeur)
```

Les noms de fonctions sont explicites :

```
1 | pin_led = Pin(16, mode=Pin.OUT)
2 |
3 | # Impose une tension de 3.3V en sortie (état logique haut)
4 | pin_led.on()
5 | pin_led.high()
6 | pin_led.value(1)
7 |
8 | # Impose une tension de 0V en sortie (état logique bas)
9 | pin_led.off()
10 | pin_led.low()
11 | pin_led.value(0)
```

Utilisation des broches en mode PWM

Le Raspberry Pi Pico prend en charge la modulation de largeur d'impulsion (PWM) sur certaines broches. Cela permet de contrôler la vitesse des moteurs, la luminosité des LED, etc. Voici comment configurer une broche en mode PWM :

```
1 | import machine
2 |
3 | # Configurer GP2 en mode PWM
4 | broche_gp2 = machine.Pin(2)
5 | pwm = machine.PWM(broche_gp2)
```

Lecture de valeurs analogiques

Pour lire des valeurs analogiques à partir des broches GPIO, vous pouvez utiliser l'ADC (Convertisseur Analogique-Numérique) intégré au Pico.

Note/Rappel: Le RPi Pico possède 4 broches analogiques: GP26->29

Voici comment configurer une broche pour lire une valeur analogique :

```
1 | import machine
2 |
3 | # Configurer GP26 pour la lecture analogique
4 | broche_gp26 = machine.Pin(26)
5 | adc = machine.ADC(broche_gp26)
6 |
7 | # Lire la valeur analogique
8 | valeur_analogique = adc.read_u16()
```

Le code suivant permet de lire la valeur d'un potentiomètre analogique:

```
1 | from machine import ADC, Pin
2 | from utime import sleep
3 | pot = ADC(26)
4 | conversion_factor = 3.3 / 65535
5 |
6 | while True:
7 |     raw = pot.read_u16()
8 |     volts = raw * conversion_factor
9 |     print('Raw: {} '.format(raw), 'Voltage {:.1f}V'.format(volts))
10 |     sleep(1)
```

Et ce code permet de changer la fréquence de clignotement d'une LED en fonction de la valeur lue sur un potentiomètre:

```
1 | from machine import ADC, Pin
2 | from utime import sleep
3 |
4 | pot = ADC(26)
5 | led = Pin(25, Pin.OUT)
6 |
7 | def map(s, a1, a2, b1, b2):
8 |     return b1 + (s - a1) * (b2 - b1) / (a2 - a1)
9 |
10 | while True:
11 |     raw = pot.read_u16()
12 |     delay = map(raw, 0, 65535, 0, 1)
13 |     print('Raw: {} '.format(raw), 'Delay {:.1f}s'.format(delay))
14 |     led.value(1)
15 |     sleep(delay)
16 |     led.value(0)
17 |     sleep(delay)
```

Conclusion

Les broches GPIO du Raspberry Pi Pico offrent de nombreuses possibilités pour interagir avec le monde physique en utilisant MicroPython. Que ce soit pour contrôler des LED, des moteurs, ou pour lire des capteurs analogiques,

vous avez maintenant les connaissances de base pour commencer à créer vos propres projets avec le Pico. Explorez davantage et expérimentez avec les différentes broches pour donner vie à vos idées.