



PREMIERE GENERALE

NUMERIQUE ET SCIENCES INFORMATIQUES

CARTE RASPBERRY PICO : GESTION DES SORTIES PWM

1 – PWM SUR LA CARTE RASPBERRY PICO

Un signal modulé en largeur d'impulsion (MLI, ou PWM pour Pulse Width Modulation) permet de contrôler l'intensité lumineuse d'une LED, la vitesse de rotation d'un moteur ou encore la position angulaire d'un servomoteur.

Dans la carte Raspberry PICO, les signaux PWM sont générés par des blocs matériels spécifiques. Une fois configurés dans le script, ces blocs génèrent les signaux PWM en permanence. Les ressources du processeur sont alors disponibles pour exécuter d'autres tâches.

Le Raspberry Pi Pico possède 8 paires de blocs PWM (PWM 0 à PWM 7), soit **16 sorties PWM** au total. Ces sorties PWM peuvent, théoriquement, être rattachées à n'importe quelle broche de la carte. Chaque bloc PWM peut avoir une fréquence indépendante. Sur la base d'un μC RP2040 fonctionnant à 125 MHz, les fréquences PWM possibles sur les sorties d'un Raspberry Pico sont comprises entre 7.5 Hz à plusieurs Mégahertz.

MicroPython fait abstraction du hardware et sélectionne automatiquement un bloc PWM disponible. La configuration consiste à associer un objet PWM à une broche et de choisir la fréquence PWM.

```
from machine import PWM, Pin
nom_pwm = PWM(Pin(numero_broche, mode=Pin.OUT))
nom_pwm.freq(frequence)
```

Pour générer le signal PWM, on dispose de la méthode **duty_u16(rapport_cyclique)**. Le paramètre rapport_cyclique doit être égal à 0 pour un rapport cyclique de 0% et 65535 pour un rapport cyclique de 100%.

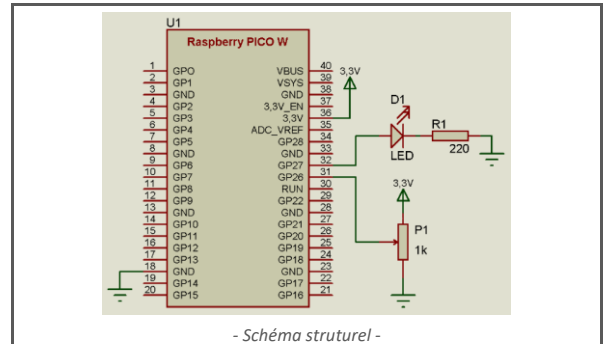
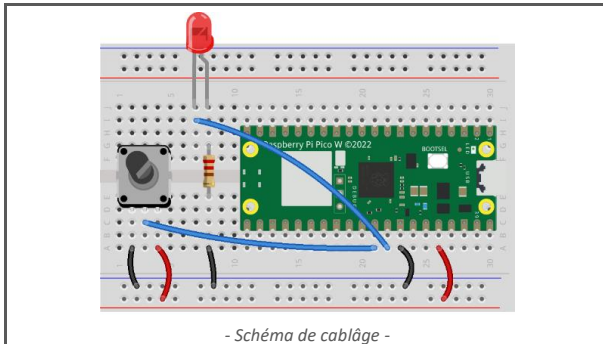
2 – COMMANDER L'ÉCLAIREMENT D'UNE LED PAR PWM

Le programme ci-dessous permet de régler l'éclairement de la LED en fonction du réglage du potentiomètre :

```
from machine import ADC, PWM, Pin
import time

adc = ADC(0)
led_pwm = PWM(Pin(27, mode=Pin.OUT))
led_pwm.freq(50)

while True:
    N = adc.read_u16()
    led_pwm.duty_u16(N)
    time.sleep_us(20)
```



3 – COMMANDER LA POSITION D'UN SERVOMOTEUR

Pour commander un servomoteur, il faut lui fournir des impulsions avec une période de 20 ms. La position du servomoteur sera fonction de la durée de l'impulsion :

- durée d'impulsion égale à 0,5 ms (rapport cyclique = 2.5%), le servomoteur se met en position 0° ;
- durée d'impulsion égale à 1.5 ms (rapport cyclique = 7.5%), le servomoteur se met en position 90° ;
- durée d'impulsion égale à 2,4 ms (rapport cyclique = 12%), le servomoteur se met en position 180°.

Le servomoteur utilisé est un servomoteur HITEC HS-422 qui nécessite une alimentation de 4,8 à 6V. Il est donc nécessaire d'utiliser une alimentation externe pour pouvoir l'alimenter avec une tension suffisante.



Le programme ci-dessous permet de positionner un servomoteur connecté sur la broche GPIO27 en fonction de la tension générée par un potentiomètre connecté sur la broche GPIO26.

```
from machine import ADC, PWM, Pin
import time

adc = ADC(0)
servo_pwm = PWM(Pin(27, mode=Pin.OUT))
servo_pwm.freq(50)
rcy_min = 1640          # rcy = 2,5%
rcy_max = 7864          # rcy = 12%

while True:
    N = adc.read_u16()          # Lecture de l'entrée GPIO26
    angle = N * 180 // 65535    # angle = f(réglage potentiomètre)
    rcy = int((rcy_max - rcy_min) * angle / 180 + rcy_min)  # Calcul Rapport Cyclique
    pwm_servo.duty_u16(rcy)    # Génération signal PWM
    time.sleep_ms(10)
```

