

MACE ROBOTICS

Documentation de la carte MR-4

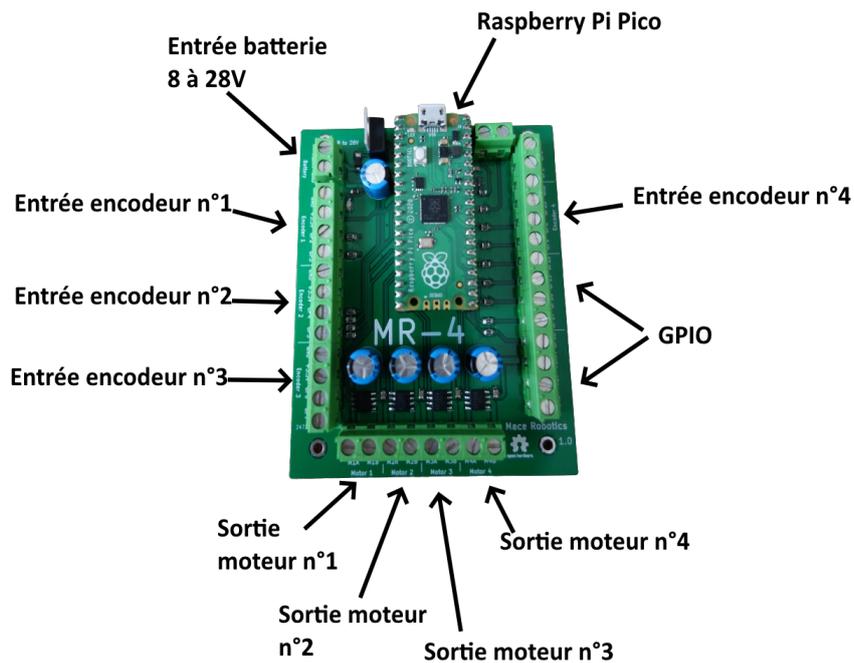
Mise à jour : 13/07/23

Table des matières

1) Présentation de la carte MR-4.....	3
2) Logiciels & matériels nécessaire.....	4
3) Contrôles des sorties moteurs.....	5
1. Gestion direction et vitesse :.....	5
2. Gestion de la direction:.....	6
3. Brake moteur.....	7
4. Stop moteur.....	8
4) Contrôles des GPIO.....	9
5) Lecture d'encodeurs.....	11
6) Lecture de la tension batterie.....	12
7) Dimensions.....	13

1) Présentation de la carte MR-4

- Entrée batterie :
 - Tension minimale : 8V
 - Tension maximale : 28V
- 4 entrée encodeurs (tension de +3,3V)
- GPIO
 - Tension max : +3,3V
- 4 sorties moteurs (2A max)



2) Logiciels & matériels nécessaire

Le langage MicroPython est préinstallé sur le robot MR-Pico.

Lien pour télécharger l'IDE : <https://thonny.org/>

3) Contrôles des sorties moteurs

La fonction pour le contrôle d'un moteur est :

```
➤ MR4.motor1Control(direction, vitesse)
```

Avec :

- direction : valeur entière de 0 ou 1
- vitesse : de 0 à 100 %

1. Gestion direction et vitesse :

Contrôle de la sortie moteur n°1 et n°3 :

```
import time
import MR4

# contrôle sortie 1
MR4.motor1Control(1, 95) # direction =1, vitesse à 95%

# contrôle sortie 3
MR4.motor3Control(1, 45) # direction =0, vitesse à 45%

# end of file
```

2. Gestion de la direction:

Contrôle des 4 sorties moteurs (direction 0) pendant 3 secondes et dans l'autre direction pendant 3 secondes.

```
import time
import MR4

while 1:

    # direction 0
    MR4.motor1Control(0, 50) # vitesse à 50%
    MR4.motor2Control(0, 50)
    MR4.motor3Control(0, 50)
    MR4.motor4Control(0, 50)

    time.sleep(3)

    # direction 1
    MR4.motor1Control(1, 30) # vitesse à 30%
    MR4.motor2Control(1, 70) # vitesse à 70%
    MR4.motor3Control(1, 50)
    MR4.motor4Control(1, 80)

    time.sleep(3)

# end of file
```

3. Brake moteur

Contrôle de la sortie n°1 avec une vitesse de 50 % et brake du moteur après 10 secondes.

```
import time
import MR4

print("start")

# contrôle sortie 1
MR4.motor1Control(1, 50)

time.sleep(10)

MR4.motor1Brake()

print("end")
# end of file
```

4. Stop moteur

Contrôle de la sortie n°1 avec une vitesse de 50 % et stop du moteur après 5 secondes.

```
import time
import MR4

print("start")

# contrôle sortie 1
MR4.motor1Control(1, 50)

time.sleep(5)

MR4.motor1Stop()

print("end")
# end of file
```

4) Contrôles des GPIO

La carte MR-4 a plusieurs GPIO utilisable en entrées ou sorties. Ces GPIO ont une tension maximale de +3,3V.

Exemple de contrôle de 4 pins en sorties de la carte MR-4 :

```
import time
import MR4
from machine import Pin

# Configuration des gpio en sorties
gpio16 = Pin(16, mode=Pin.OUT)
gpio17 = Pin(17, mode=Pin.OUT)
gpio18 = Pin(18, mode=Pin.OUT)

while 1:

    # Tension de 3.3V en sortie (état logique haut)
    gpio16.on()
    gpio17.on()
    gpio18.on()

    # pause de 1 seconde
    time.sleep(1)

    # Tension de 0V en sortie (état logique bas)
    gpio16.off()
    gpio17.off()
    gpio18.off()

    time.sleep(1)

# end of file
```

MACE ROBOTICS

Exemple de lecture d'une entrée :

```
import time
import MR4
from machine import Pin

# Configuration gpio 16 en entrée
gpio16 = Pin(16, mode=Pin.IN, pull=Pin.PULL_UP)

valeur = 0

while 1:

    # Lecture de la gpio16
    valeur = gpio16.value()
    print("La valeur est de : ", valeur)

    time.sleep(1)
# end of file
```

5) Lecture d'encodeurs

Exemple de lecture de quatre encodeurs. Les encodeurs sont connectés sur les entrées encodeur de la carte MR-4. Il y a 4 entrées encodeurs sur la carte.

```
import time
import encoder

# encoder init
encoder1 = encoder.Encoder(2,3)
encoder2 = encoder.Encoder(4,5)
encoder3 = encoder.Encoder(6,7)
encoder4 = encoder.Encoder(8,26)

while True:

    # Lectures des encodeurs
    print("Encoder1 = ", encoder1.read())
    print("Encoder2 = ", encoder2.read())
    print("Encoder3= ", encoder3.read())
    print("Encoder4 = ", encoder4.read())
    time.sleep(1)

# end of file
```

6) Lecture de la tension batterie

Exemple pour la lecture de la tension de la batterie en volt:

```
import time
import MR4

while 1:
    tension = MR4.batteryTension()
    print("Tension batterie = ",tension)
    time.sleep(1)

# end of file
```

7) Dimensions

Voici les dimensions mécanique de la carte MR-4 :

