

1. Le robot Maqueen

Le robot Maqueen micro:bit est un robot contrôlé par la carte micro:bit. Il est petit, maniable et facile d'utilisation.

La carte microbit embarque une version Python appelé MicroPython qui a été développée pour les microcontrôleurs.



2. Caractéristiques

Le robot Maqueen possède beaucoup de fonctionnalités :

- 2 capteurs de suivi de ligne
- 2 LED rouges
- 4 LED RVB Neopixel pour éclairage d'ambiance
- 1 capteur de distance ultrason
- 1 buzzer pour effets sonores
- 2 moteurs à engrenage contrôlables séparément par i2c
- 1 alimentation par pack de 3 piles AAA
- 1 capteur infrarouge permettant au robot d'être télécommandé



3. Affectation des entrées-sorties

Sorties			Entrées		
Désignation	Repère	Pin Microbit	Désignation	Repère	Pin Microbit
LED rouge gauche	LED-L	8	Capteur de ligne gauche	Line-L	13
LED rouge droite	LED-R	12	Capteur de ligne droite	Line-R	14
LED néopixel	RGB0, RGB1, RGB2, RGB3	15	Capteur infrarouge	IR	16
Buzzer		0	Capteur ultra-sons	Trig	1
			Echo	2	

Bus I2C					
Désignation				Repère	Pin Microbit
I2C adresse : 0x10				SCL	19
				SDA	20
Bytearray de trois octets					
Moteur	Moteur Moteur gauche : 0 Moteur droit : 2	Sens Avancer : 0 Reculer : 1	Vitesse 0 - 255		

4. Module microbit de MicroPython

Le module `microbit` de MicroPython permet de contrôler la carte `micro:bit`.

```
from microbit import *
```

4.1. Méthode `write_digital()`

La méthode `write_digital()` permet de spécifier l'état logique (0 ou 1) d'une sortie (`pin`).

```
pin0.write_digital(1) # met la sortie 0 à l'état logique 1
pin2.write_digital(0) # met la sortie 2 à l'état logique 1
```

4.2. Méthode `read_digital()`

La méthode `read_digital()` permet de lire l'état logique (0 ou 1) d'une entrée (`pin`).

```
pin0.read_digital() # retourne l'état logique (0 ou 1) de l'entrée 1
```

4.3. Protocole I2C

Le module I2C permet de communiquer avec le module de commande des moteurs connectés sur le bus I2C.

La méthode `init` permet de définir la fréquence d'horloge des signaux `sda` et `scl`, ainsi que la définition des bornes.
`microbit.i2c.init(freq=100000, sda=pin20, scl=pin19)`

La méthode `i2c` permet d'écrire d'une information sur le bus I2C en spécifiant l'adresse (`addr`) et le mot (`buf`).
`microbit.i2c.write(addr, buf)`

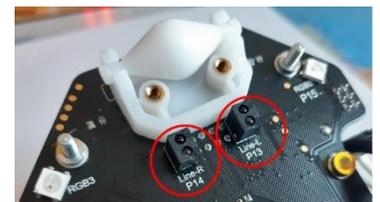
```
#écriture à l'adresse 0x10 de [moteur gauche, avancer, vitesse : 50]
microbit.i2c.write(0x10, bytearray([0, 0, 50]))
```

5. Capteur optique infra-rouge

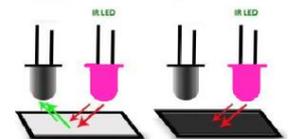
Le robot dispose à l'avant de 2 capteurs optiques infrarouge.

Le capteur optique infrarouge est constitué :

- d'une LED infrarouge ;
- d'un photo-transistor.



Le capteur envoie un signal haut lorsqu'il détecte le blanc et un signal bas quand il détecte le noir.



Programme qui allume une LED quand le blanc est détecté.

```
from microbit import *
while True:
    if pin13.read_digital():
        pin8.write_digital(1)
    else:
        pin8.write_digital(0)
```