

Document V1.0 – Pyqueen V1.0

# Utilisation du robot micro:Maqueen v4.0 de DFRobot avec une carte BBC micro:bit sous Python

THOMAS POCHART  
03/01/2020

# **Sommaire**

Préambule : descriptif du matériel – p.3

1. Liste des composants – p.3
  - a. Carte BBC micro:bit – p.3
  - b. Cable USB/micro-USB – p.4
  - c. Kit Micro:Maqueen – p.4
  - d. Piles AAA – p.8
2. Prérequis logiciel – p.8

I. Travail du professeur : installations préalables – p.9

1. Construction du robot Maqueen – p.9
2. Installation du module Python pyqueen sur la carte BBC micro:bit – p.14
  - a. Position du problème – p.14
  - b. Prérequis théoriques – p.14
  - c. Installation avec Mu – p.14

II. Travail des élèves : les outils à disposition des élèves – p.18

1. Le module pyqueen – p.18
  - a. Import du module pyqueen – p.18
  - b. La fonction motors – p.18
  - c. La fonction motorL – p.18
  - d. La fonction motorR – p.19
  - e. La fonction lineL – p.19
  - f. La fonction lineR – p.19
  - g. La fonction distance – p.19
  - h. Les fonctions permettant l'utilisation des LEDs avant – p.19
  - i. La fonction ledrgb – p.19
2. Utilisation de Mu – p.20
  - a. Démarrage de Mu – p.20
  - b. Connection de la carte BBC micro:bit – p.20
  - c. Ecriture d'un programme – p.22
  - d. Flash d'un programme sur la carte BBC micro:bit – p.22

III. Quelques exemples – p.23

1. Avance en ligne droite et arrêt face à un obstacle – p.23
  - a. Problème – p.23
  - b. Solution en langage naturel – p.23
  - c. Solution en Python – p.23
2. Avance en ligne droite et réorientation aléatoire face à un obstacle – p.24
  - a. Problème – p.24
  - b. Solution en langage naturel – p.24
  - c. Solution en Python – p.24
3. Suivi d'une ligne – p.25
  - a. Problème – p.25
  - b. Solution en langage naturel – p.25
  - c. Solution en Python – p.25

IV. Dépannage – p.26

1. La carte BBC micro:bit affiche une erreur du type OS Error – p.26
2. La carte BBC micro:bit affiche une erreur du type Name Error – p.26
3. La carte BBC micro:bit n'affiche pas d'erreur, mais le robot reste inerte – p.26

Annexes – p.27

1. Listing du programme pyqueen.py – p.27

2. Code permettant l'import des fonctions liées à l'utilisation du Maqueen – p.28
  3. Solution Python à l'exemple 1 – p.28
  4. Solution Python à l'exemple 2 – p.28
  5. Solution Python à l'exemple 3 – p.29
- Bibliographie – p.30

# Préambule : descriptif du matériel

## 1. Liste des composants

### a. Carte BBC micro:bit

Fournie dans un package en carton vert (fig. 1, fig. 2).



Fig. 1 : Devant du package de la carte BBC micro:bit



Fig. 2 : Arrière du package de la carte BBC micro:bit

La BBC micro:bit (fig. 3, fig. 4) est un microordinateur comportant, entre autres, connectiques micro-USB et Bluetooth.

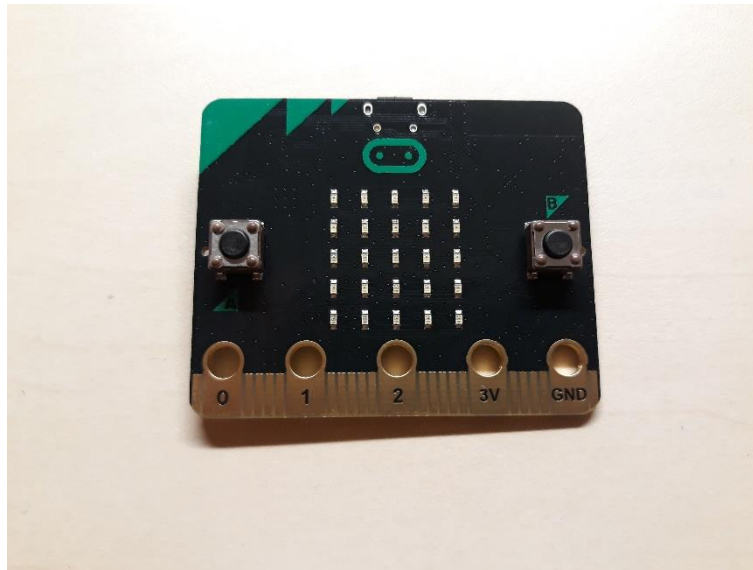


Fig. 3 : Avant de la carte BBC micro:bit



Fig. 4 : Arrière de la carte BBC micro:bit

#### b. Cable USB/micro-USB

Tout câble ayant pour fiches un USB mâle et un micro-USB mâle et pouvant transférer des données est compatible.

Le kit d'origine ne comprend pas ce câble.

#### c. Kit Micro:Maqueen

Contient une carte électronique (fig. 5, fig. 6), un capteur à ultrasons (fig. 7, fig. 8), un adhésif double-face (fig. 9), un boîtier à piles (fig. 10, fig. 11), deux roues en plastique (fig. 12) et deux pneus (fig. 13).

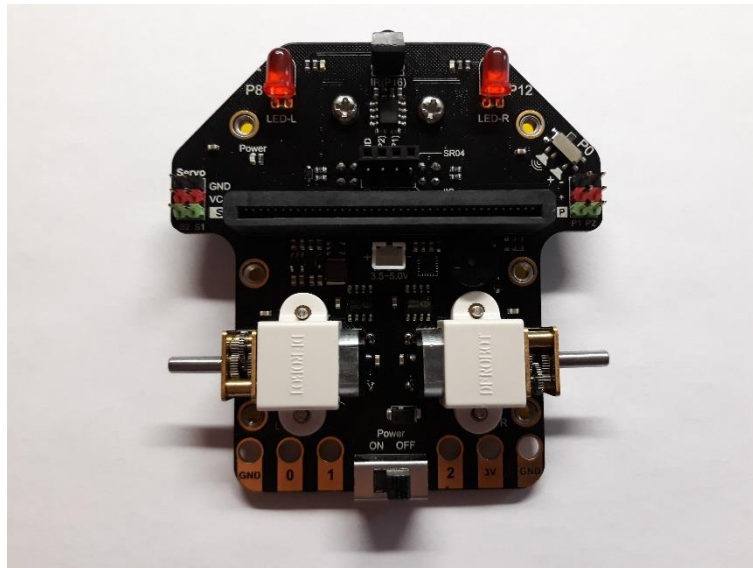


Fig. 5 : Avant de la carte électronique constituant le corps du Maqueen

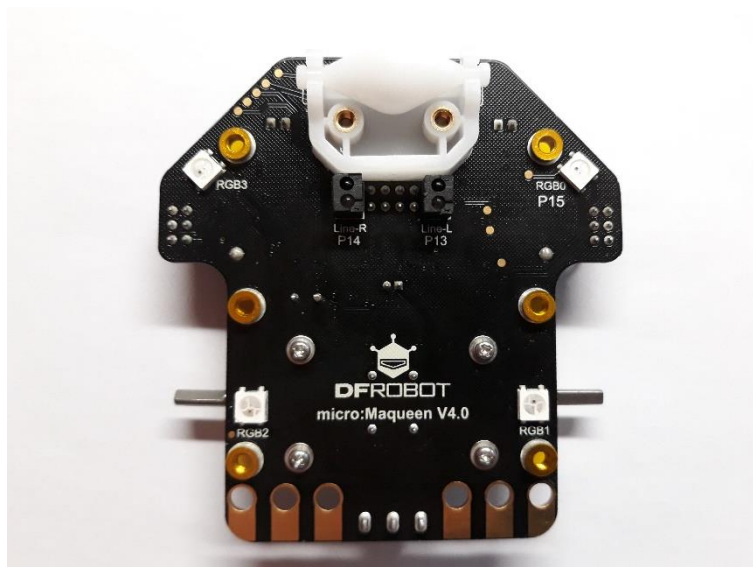


Fig. 6 : Arrière de la carte électronique constituant le corps du Maqueen



Fig. 7 : Avant du capteur à ultrasons

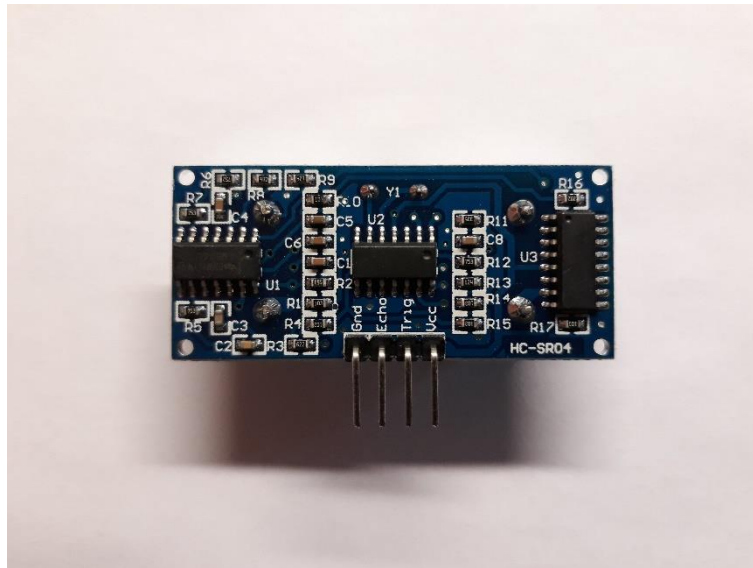


Fig. 8 : Arrière du capteur à ultrasons



Fig. 9 : Adhésif double-face



Fig. 10 : Avant du boîtier à piles



Fig. 11 : Arrière du boîtier à piles



Fig. 12 : Roues



Fig. 13 : Pneus



#### d. Piles AAA

L'alimentation du Maqueen est assurée par trois piles AAA, non fournies dans le kit d'origine.

## 2. Prérequis logiciel

L'interaction avec le robot se fera avec l'éditeur Python Mu, dont les instructions de téléchargement sont disponibles sur le site officiel [3].

La ressource logiciel pyqueen sera également utilisée.

# **I. Travail du professeur : installations préalables**

## **1. Construction du robot Maqueen**

1. Brancher le boîtier à piles au corps du robot (fig. 14).

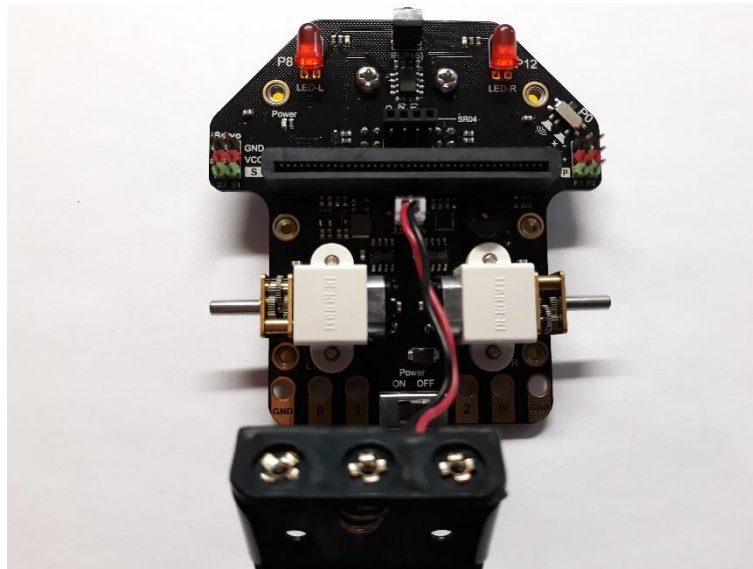


Fig. 14 : Branchement du boîtier à piles au corps du robot.

2. Couper un carré d'environ 5 mm de côté en bas à droite de l'adhésif (fig. 15).

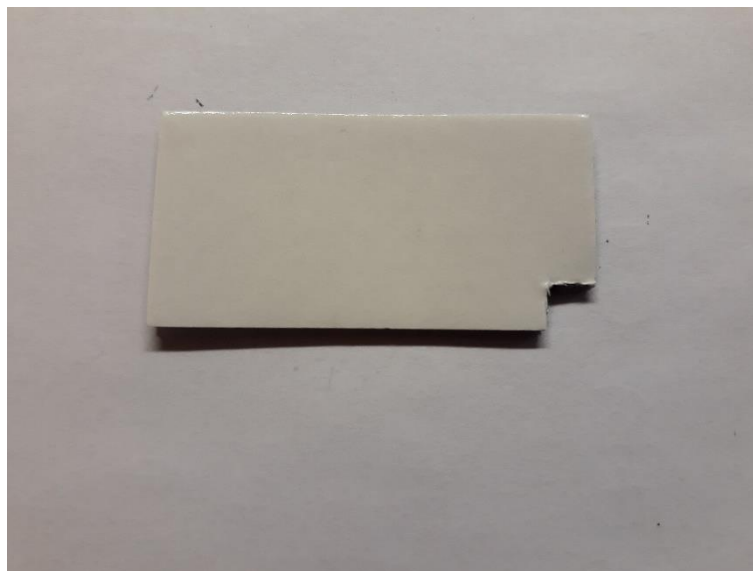


Fig. 15 : Découpe de l'adhésif.

3. Coller l'adhésif au corps avec le carré découpé à l'arrière du moteur droit du robot (fig. 16). Le câble d'alimentation du boîtier à piles doit passer sous l'adhésif (fig. 17).

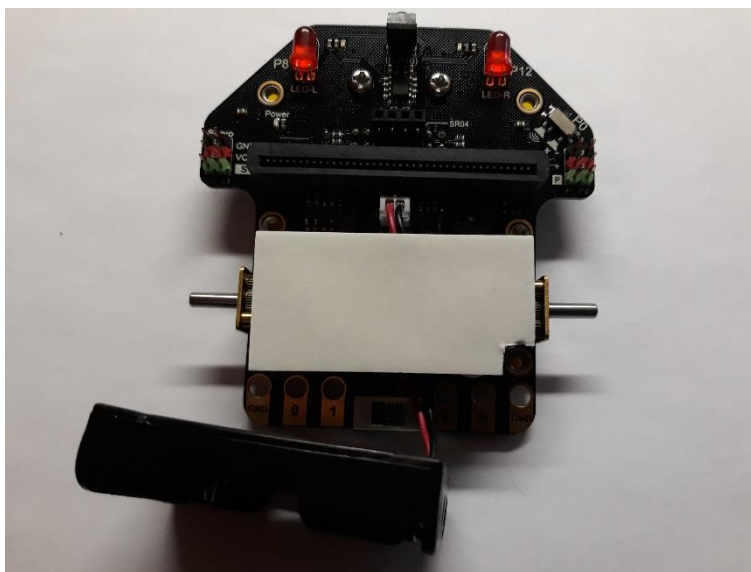


Fig. 16 : Adhésif collé au corps du robot.

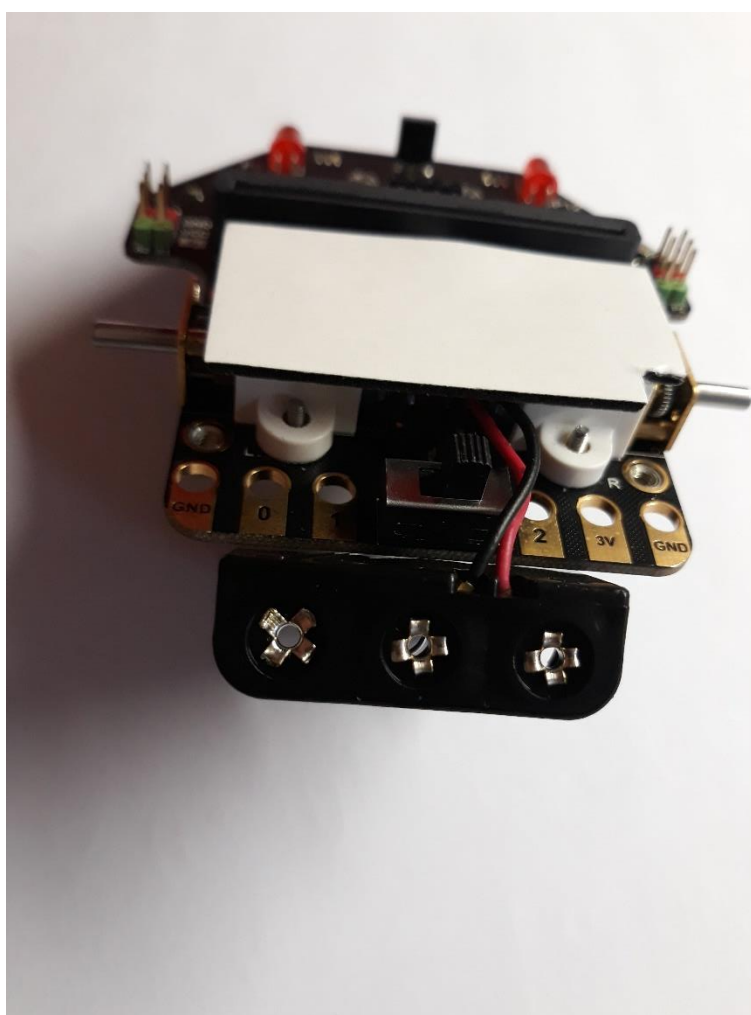


Fig. 17 : Position sous l'adhésif du câble d'alimentation.

4. Coller le boîtier à piles à l'autre face de l'adhésif (fig. 18). Le câble d'alimentation passe par le carré découpé sur l'adhésif (fig. 19).

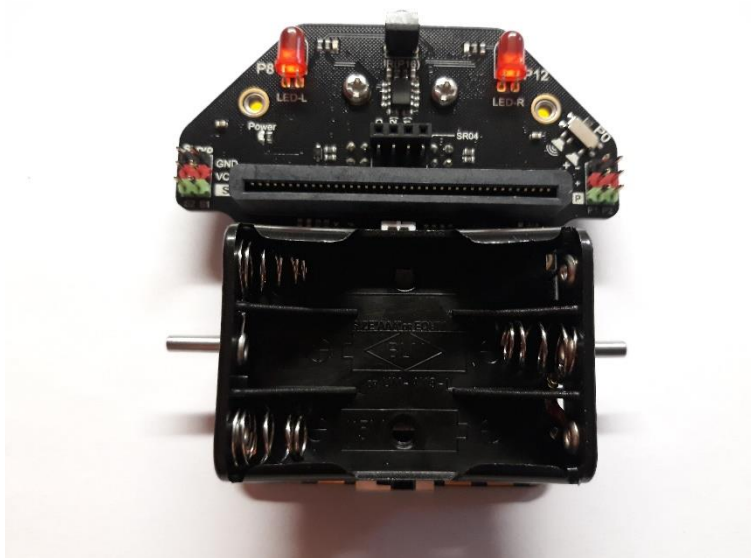


Fig. 18 : Boîtier à piles collé au corps du robot.

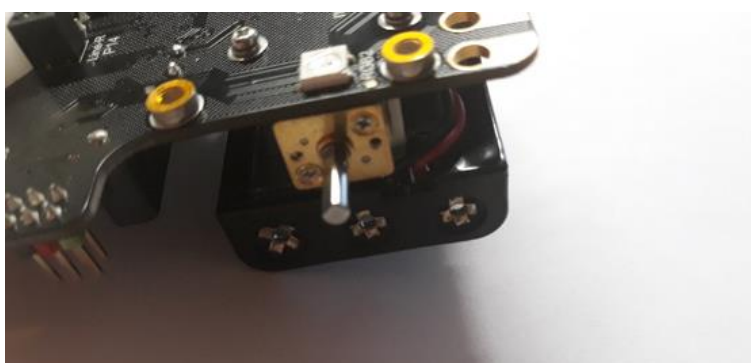


Fig. 19 : Câble d'alimentation passant par le carré découpé sur l'adhésif.

5. Placer les pneus sur les roues (fig. 20).



Fig. 20 : Roue avec pneu.

6. Monter les roues sur le corps du robot. On remarque que les rainures des pneus pointent dans deux directions opposées (fig. 21).

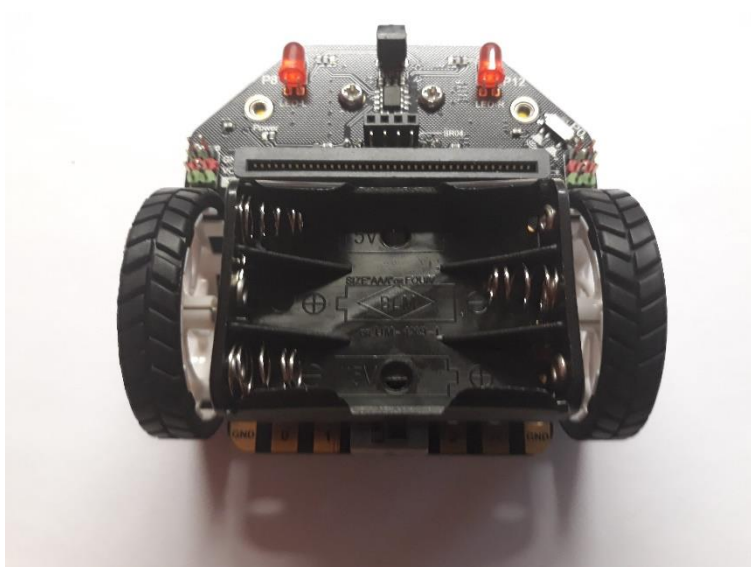


Fig. 21 : Corps du robot avec roues.

7. Connecter le capteur à ultrasons (fig. 22).

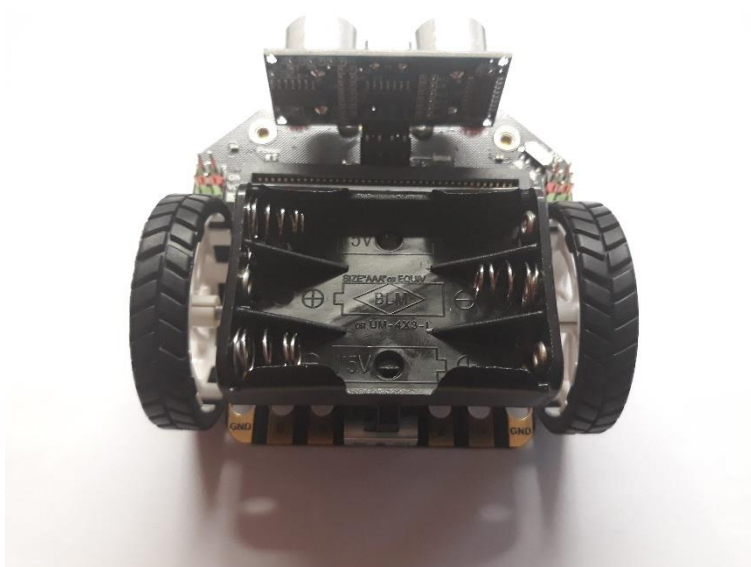


Fig. 22 : Capteur à ultrasons branché au corps du robot.

8. Insérer les piles dans le boîtier à piles (fig. 23).

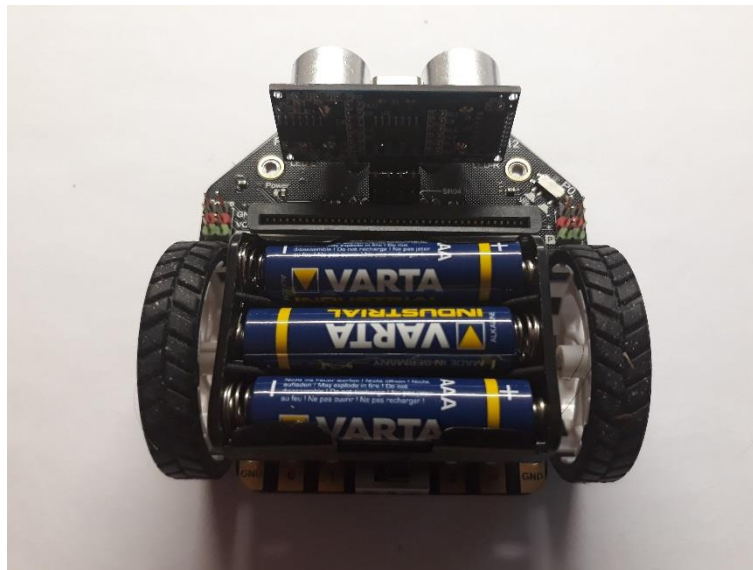


Fig. 23 : Position des piles dans leur boîtier.

9. Connecter la carte BBC micro:bit en prenant garde au sens de celle-ci (fig. 24).

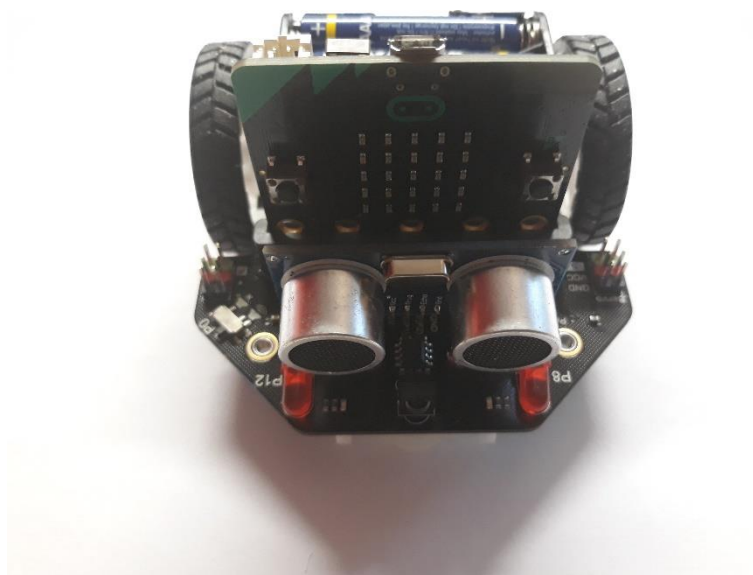


Fig. 24 : Position de la carte BBC micro:bit sur le corps du robot.

## 2. Installation du module Python pyqueen sur la carte BBC micro:bit

### a. Position du problème

Le Maqueen n'est pas nativement compatible avec Python sous forme de code [1] : il est donc nécessaire d'installer des ressources Python supplémentaires afin de pouvoir programmer le Maqueen via un éditeur textuel Python.

Ces ressources ne devant pas être modifiées pour assurer le bon fonctionnement du robot, leur installation doit être faite afin que les élèves n'aient pas ce droit de modification.

La présente partie décrit ladite installation.

### b. Prérequis théoriques

La carte BBC micro:bit présente une mémoire flash de 256 ko, dans laquelle est stockée le code régissant le fonctionnement de la carte.

Ce code peut être réparti en plusieurs fichiers. Au minimum, la carte présente un fichier `main.py`, qu'elle lit et exécute lorsqu'elle est alimentée. Ce fichier `main.py` peut lui faire appel aux autres fichiers stockés sur la carte.

Dans le cas présent, les ressources nécessaires à l'utilisation du Maqueen vont être stockées dans un fichier séparé, nommé `pyqueen.py`. Lorsqu'ils programmeront, les étudiants auront accès au fichier `main.py`, mais n'auront pas la possibilité de modifier le fichier `pyqueen.py`.

### c. Installation avec Mu

Pour accéder aux fichiers stockés dans la mémoire de la carte BBC micro:bit, nous allons utiliser Mu, éditeur Python ciblant les programmeurs débutants. Il faut donc que ce dernier soit installé sur l'ordinateur servant à préparer les cartes BBC micro:bit. Il existe de nombreuses façons d'installer Mu sur un ordinateur, détaillées sur le site de Mu [3].

Il sera également nécessaire de disposer du fichier `pyqueen.py` contenant les ressources nécessaires à l'utilisation de la machine Maqueen par la carte BBC micro:bit. Ce fichier est inspiré d'un programme disponible sur le site personnel d'Olivier Lecluse, professeur de mathématiques [2]. Un listing de `pyqueen.py` est donné en annexe 1.

Ces éléments réunis, il est possible d'installer le fichier `pyqueen.py` dans la carte BBC micro:bit en procédant comme suit :

1. Brancher la carte BBC micro:bit à l'ordinateur.
2. Lancer Mu.
3. Choisir le module BBC micro:bit (fig. 25).

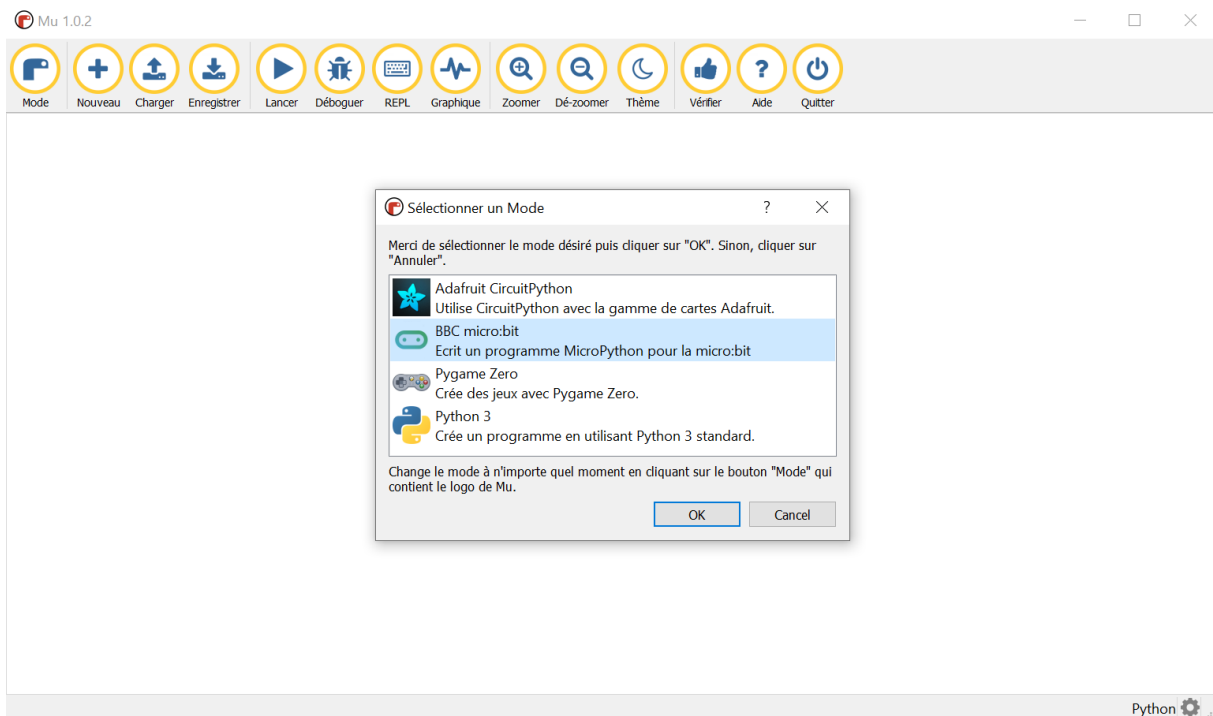


Fig. 25 : Choix du module BBC micro:bit sous Mu.

4. Placer le fichier pyqueen.py dans le dossier mu\_code situé dans le dossier de l'utilisateur actuel sous Windows (fig. 26).

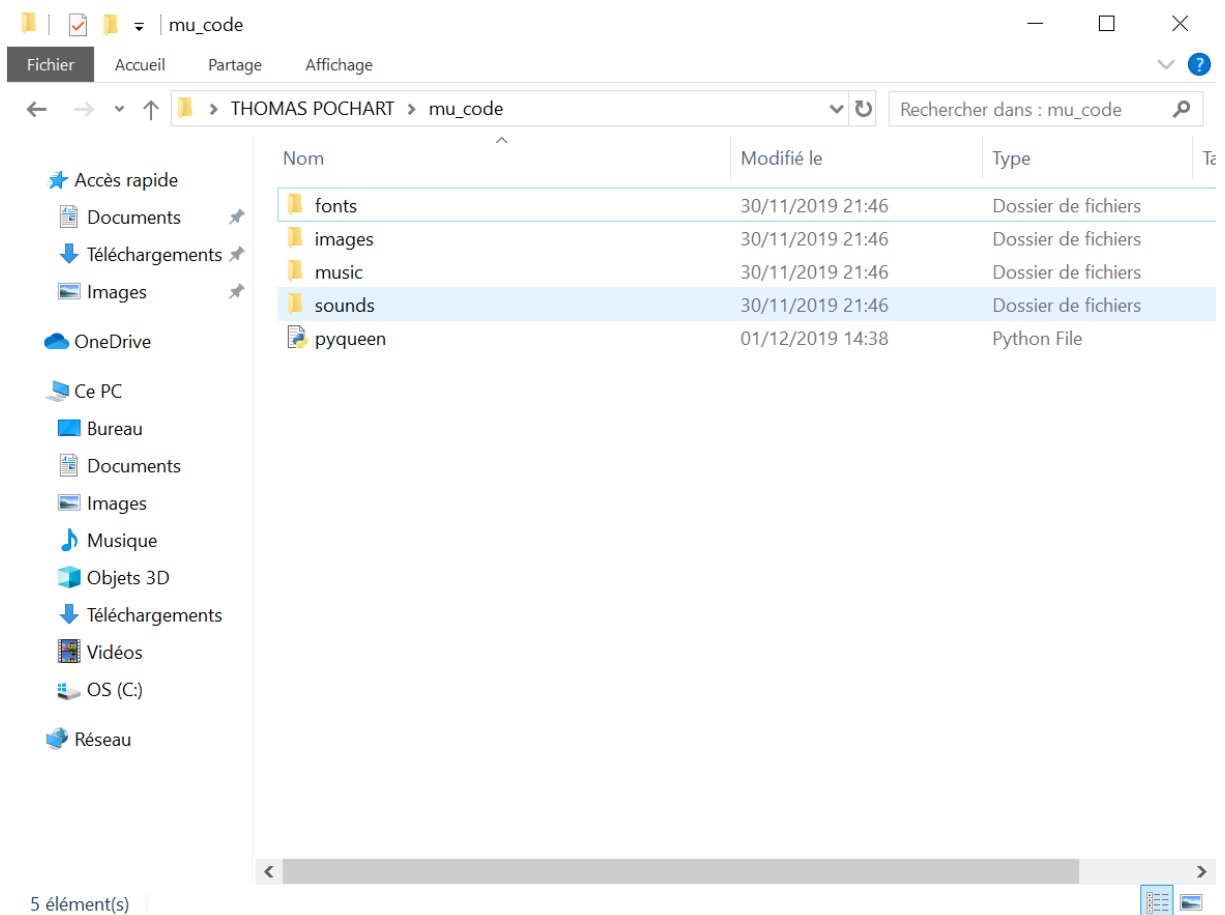


Fig. 26 : Placement du fichier pyqueen.py dans le dossier mu\_code.



5. Dans Mu, cliquer le bouton Fichiers du menu supérieur. Un menu interactif doit s'ouvrir (fig. 27).

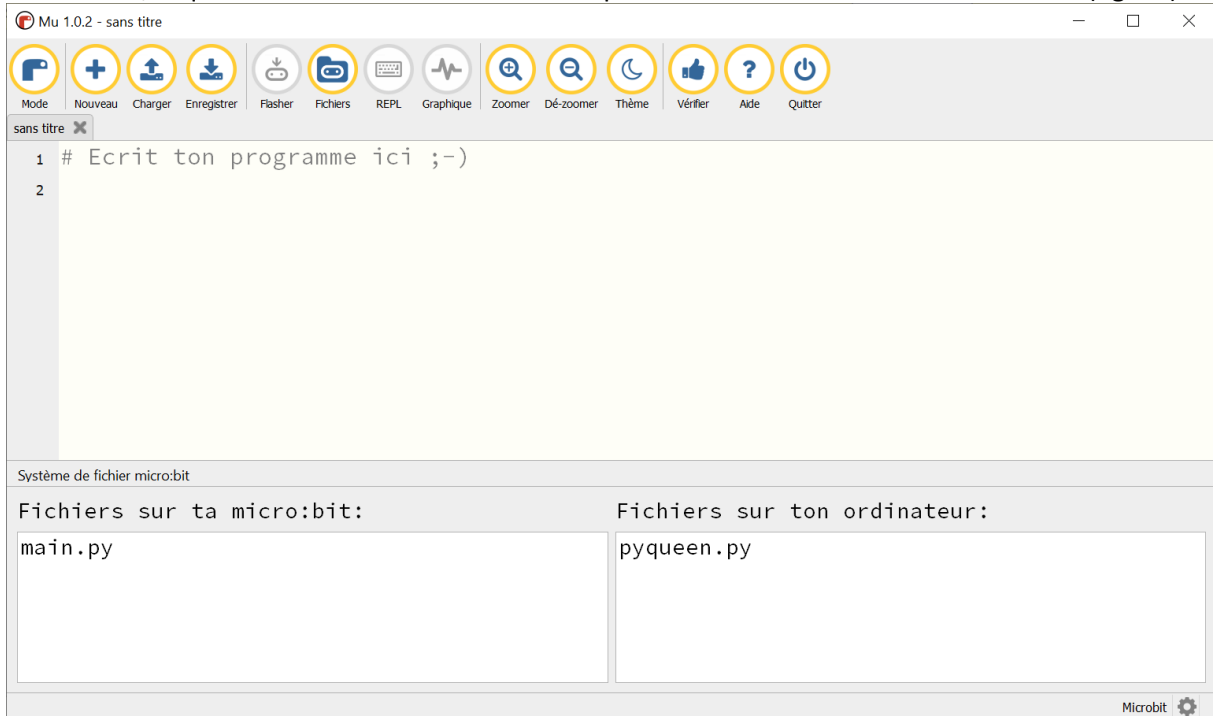


Fig. 27 : Menu interactif obtenu en cliquant sur le bouton Fichiers de Mu.

6. Via un clic long, glisser le fichier pyqueen.py de la zone « Fichiers sur ton ordinateur » vers la zone « Fichiers sur ta micro:bit » (fig. 28). Le temps de la copie, le voyant flash de la BBC micro:bit clignote (fig. 29).

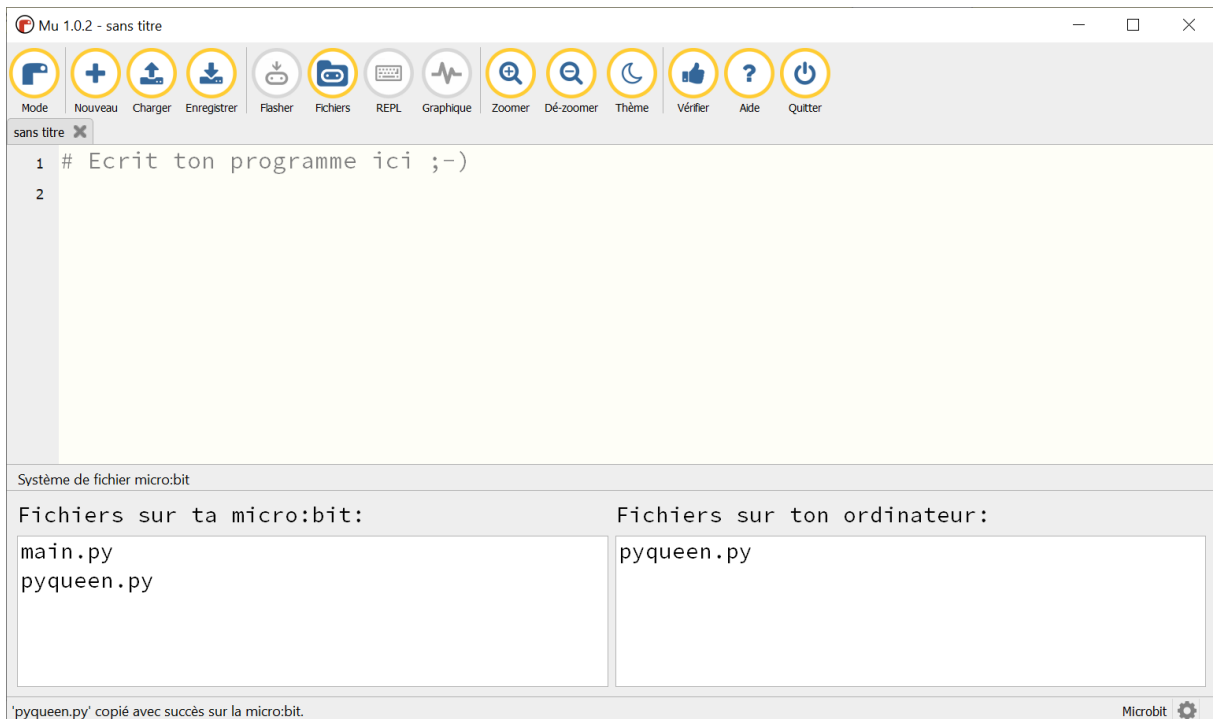


Fig. 28 : Résultat de la copie du fichier pyqueen.py sur la carte BBC micro:bit.



Fig. 29 : Voyant flash de la carte BBC micro:bit.

Désormais, la carte BBC micro:bit peut importer les fonctions nécessaires à l'interaction avec le Maqueen.

## II. Travail des élèves : les outils à disposition des élèves

### 1. Le module pyqueen

#### a. Import du module pyqueen

Il est nécessaire d'importer les fonctions nécessaires au fonctionnement de la machine Maqueen au début de chaque programme. Ceci se fait via les instructions fournies en annexe 2, à placer dans l'onglet de Mu où l'élève écrit son fichier main.py (fig. 30).

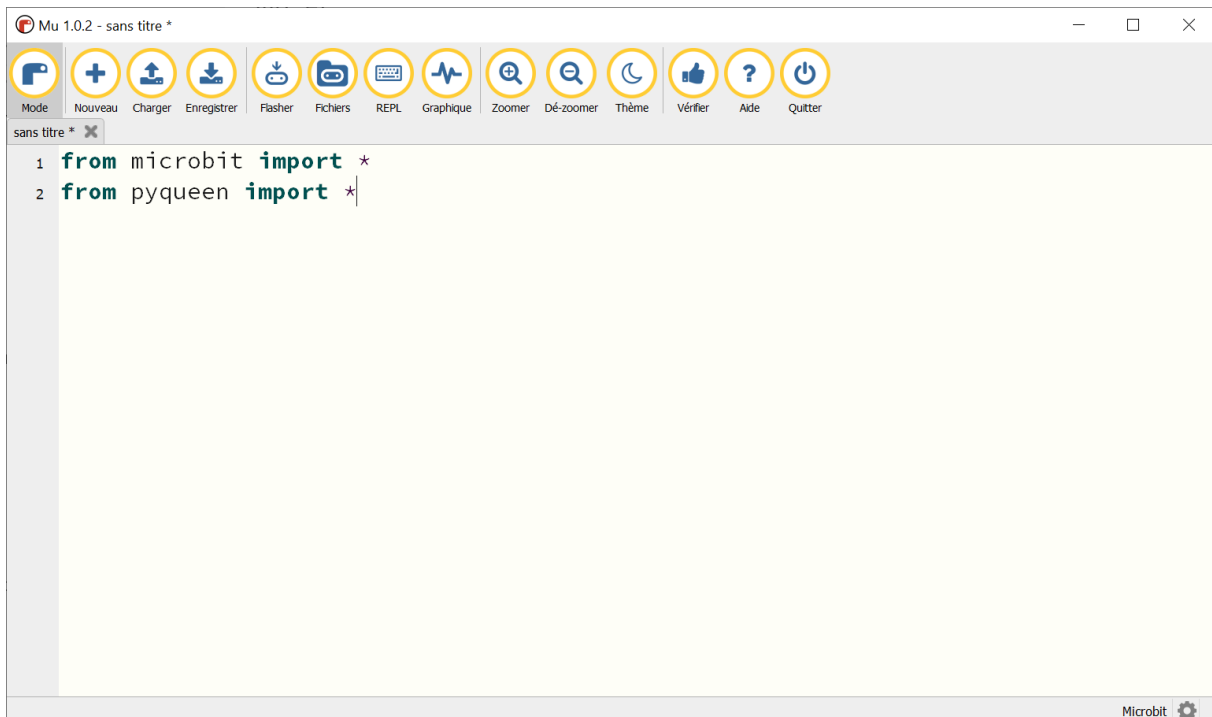


Fig. 30 : Lignes préalables à l'écriture d'un fichier main.py compatible avec le Maqueen dans Mu.

#### b. La fonction motors

`motors` est la fonction faisant avancer en ligne droite le Maqueen.

Elle prend en argument une vitesse, donnée sous la forme d'un entier compris entre -100 et 100.

L'appel de `motors(n)` met les moteurs en rotation à la vitesse angulaire associée à `n`, et ce jusqu'à nouvel ordre.

Les vitesses angulaires associées aux entiers de -100 à -1 correspondent à un recul du robot.

La vitesse angulaire associée à l'entier 0 correspond à l'immobilité du robot.

Les vitesses angulaires associées aux entiers de 1 à 100 correspondent à une avance du robot.

#### c. La fonction motorL

`motorL` est la fonction mettant en rotation le moteur gauche du Maqueen.

Elle prend en argument une vitesse, donnée sous la forme d'un entier compris entre -100 et 100.

L'appel de `motorL(n)` met le moteur gauche en rotation à la vitesse angulaire correspondant à `n`, et ce jusqu'à nouvel ordre.

#### d. La fonction motorR

motorR est la fonction mettant en rotation le moteur droit du Maqueen.

Elle prend en argument une vitesse, donnée sous la forme d'un entier compris entre -100 et 100.

L'appel de motorR(n) met le moteur droit en rotation à la vitesse angulaire correspondant à n, et ce jusqu'à nouvel ordre.

#### e. La fonction lineL

La fonction lineL renvoie l'information acquise par le capteur line-L sous forme de booléen.

Elle ne prend pas d'arguments.

#### f. La fonction lineR

La fonction lineR renvoie l'information acquise par le capteur line-R sous forme de booléen.

Elle ne prend pas d'arguments.

#### g. La fonction distance

La fonction distance renvoie la distance en centimètres séparant le capteur ultrasons de l'obstacle qu'il détecte.

Elle ne prend pas d'arguments.

#### h. Les fonctions permettant l'utilisation des LEDs avant

L'utilisation des LEDs avant se fait à l'aide des fonctions suivantes, qui ne prennent pas d'arguments :

- ledL\_on : Allume la LED avant gauche.
- ledL\_off : Eteint la LED avant gauche.
- ledR\_on : Allume la LED avant droit.
- ledR\_off : Eteint la LED avant droit.
- leds\_on : Allume les LEDs avant.
- leds\_off : Eteint les LEDs avant.

#### i. La fonction ledrgb

La fonction ledrgb prend en entrée :

- n : Entier entre 0 et 3 indiquant le numéro de la LED à allumer (les LEDs sont numérotées sous le châssis)
- r : Entier entre 0 et 255 correspondant à la valeur de la composante rouge du code RGB de la couleur à faire afficher par la LED.
- g : Entier entre 0 et 255 correspondant à la valeur de la composante verte du code RGB de la couleur à faire afficher par la LED.
- b : Entier entre 0 et 255 correspondant à la valeur de la composante bleue du code RGB de la couleur à faire afficher par la LED.

Et allume la LED numéro n à la couleur définie par le code RGB (r,g,b)

## 2. Utilisation de Mu

### a. Démarrage de Mu

Ouvrir Mu et choisir le module BBC micro:bit.

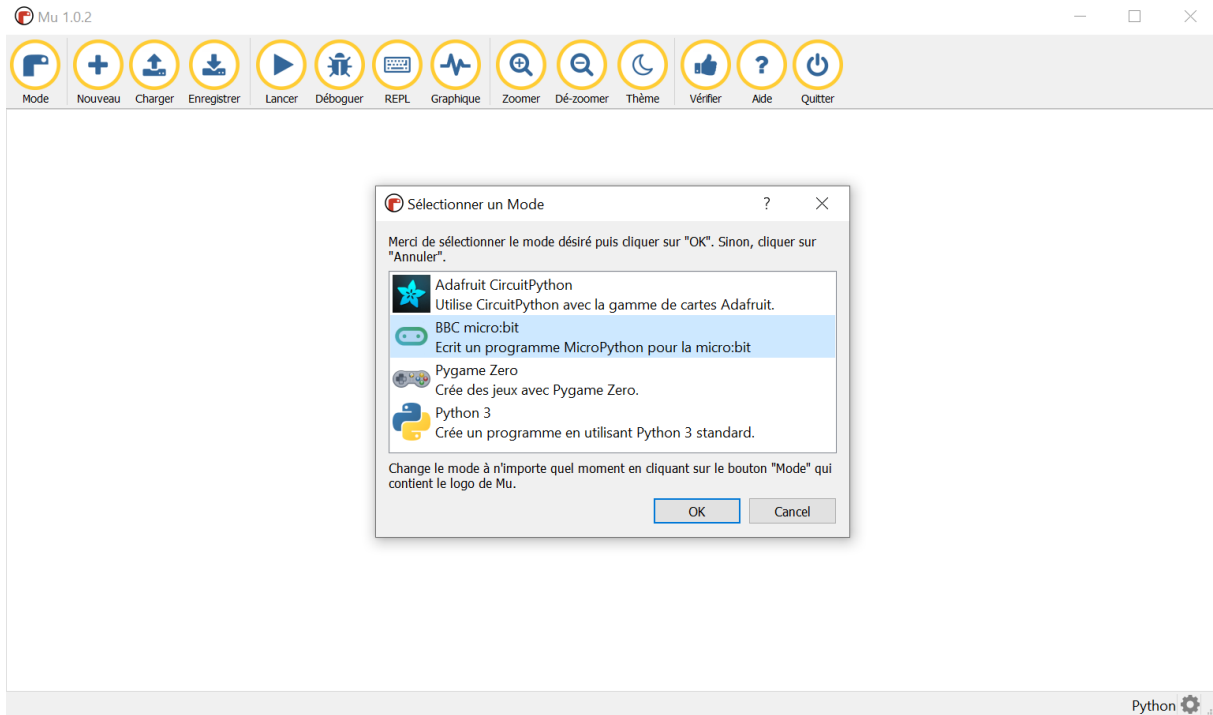


Fig. 31 : Choix du module BBC micro:bit sous Mu.

### b. Connection de la carte BBC micro:bit

Connecter la carte BBC micro:bit à l'ordinateur via le câble USB/micro-USB et attendre quelques secondes qu'elle soit détectée. Mu indique avoir détecté la carte via la mention « Nouveau périphérique BBC micro:bit connecté » en bas à gauche de sa fenêtre (fig. 32).

Ne pas connecter la carte BBC micro:bit si elle est encore sur le Maqueen allumé.

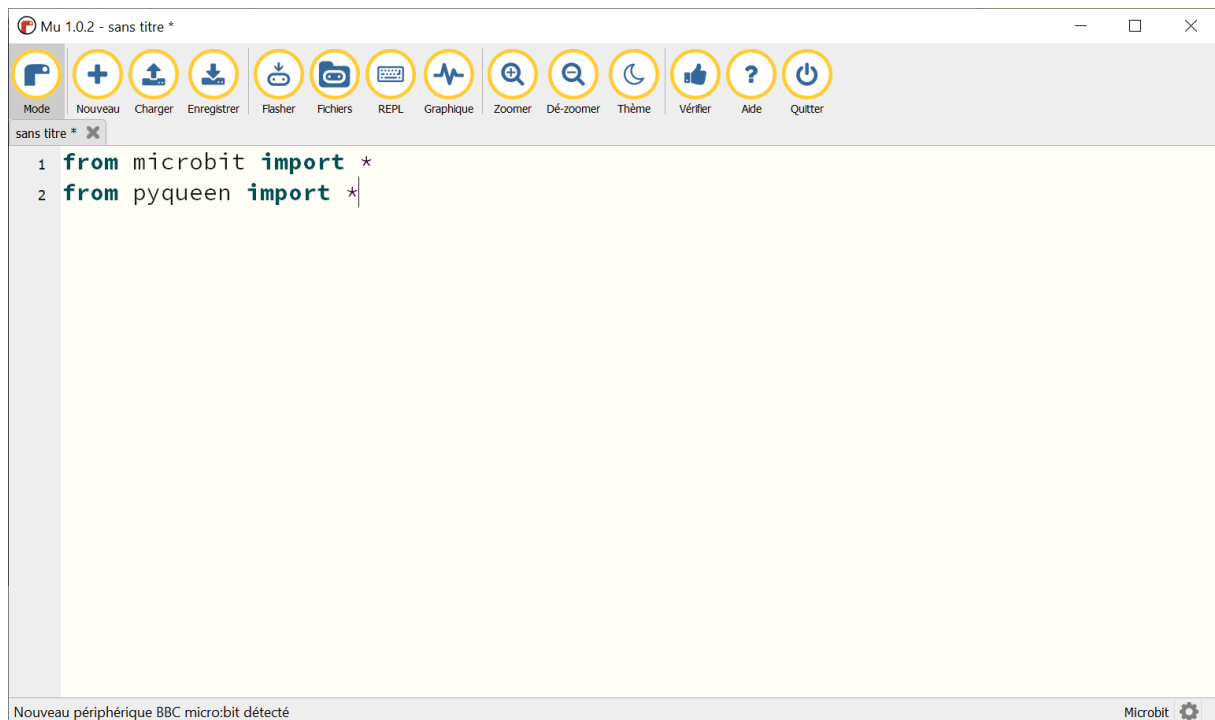


Fig. 32 : Mention « Nouveau périphérique BBC micro:bit connecté » sur Mu.

### c. Ecriture d'un programme

Tout programme doit commencer par les lignes décrites en Annexe 2 (fig. 33).  
À la suite de cela, l'élève peut taper son code.

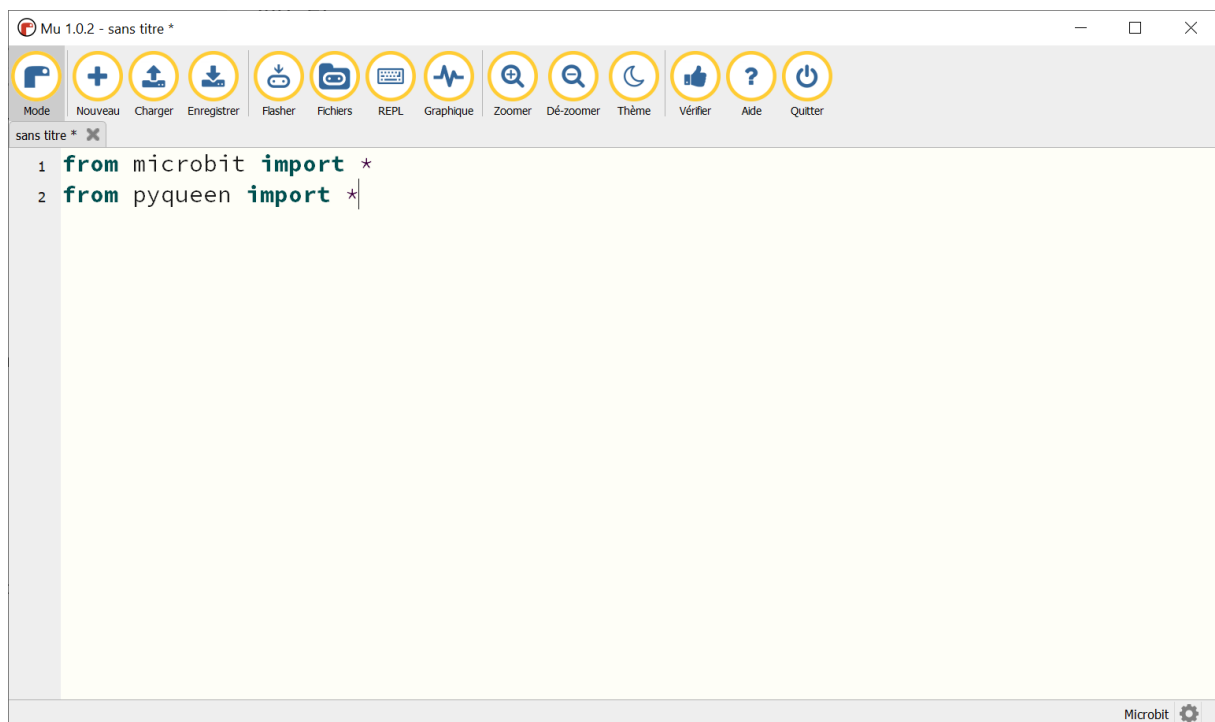


Fig. 33 : Lignes préalables à l'écriture d'un fichier compatible avec le Maqueen dans Mu.

#### d. Flash d'un programme sur la carte BBC micro:bit

Une fois un programme écrit, il suffit de cliquer sur le bouton Flasher de Mu pour que ce programme soit écrit en tant que main.py sur la carte BBC micro:bit.

Lors du flash, le voyant flash de la carte clignote (fig. 34). Ne pas déconnecter la carte tant que le voyant clignote.



Fig. 34 : Voyant flash de la carte BBC micro:bit.

### **III. Quelques exemples**

#### **1. Avance en ligne droite et arrêt face à un obstacle**

##### **a. Problème**

Programmer le robot Maqueen de sorte qu'il avance en ligne droite en affichant un smiley HAPPY sur la carte BBC micro:bit tant qu'il n'est pas frontalement à moins de 5 cm d'un obstacle, et s'arrête en affichant un smiley SAD lorsqu'un obstacle lui fait face à moins de 5 cm.

##### **b. Solution en langage naturel**

On retient l'algorithme suivant :

Faire prendre à la variable affichée par la carte BBC micro:bit la valeur smiley HAPPY.

Fixer la vitesse des moteurs à 10 unités.

Tant que vrai : #Boucle infinie que l'on arrêtera manuellement

    Si la distance à l'objet le plus proche frontalement est inférieure à 5 cm :

        Fixer la vitesse des moteurs à 0 unités.

        Faire prendre à la variable affichée par la carte BBC micro:bit la valeur smiley SAD.

    Sortir de la boucle Tant Que.

##### **c. Solution en Python**

Un listing d'une implémentation Python de l'algorithme qui précède est donné en annexe 3.



## 2. Avance en ligne droite et réorientation aléatoire face à un obstacle

### a. Problème

Programmer le robot Maqueen de sorte qu'il avance tant qu'il n'est pas frontalement à moins de 5 cm d'un obstacle, et change d'orientation de façon aléatoire lorsqu'un obstacle lui fait face à moins de 5 cm.

### b. Solution en langage naturel

Fixer la vitesse des moteurs à 50 unités.

Tant Que Vrai :

    Si la distance de l'objet le plus proche frontalement est inférieure à 5 cm :

        Fixer la vitesse des moteurs à 0 unités.

        r est un entier aléatoire entre 0 et 1.

        Si r=1 :

            Fixer la vitesse du moteur droit à -50 unités.

            Attendre 1 s.

            Fixer la vitesse des moteurs à 0 unités.

        Sinon :

            Fixer la vitesse du moteur gauche à -50 unités.

            Attendre 1 s.

            Fixer la vitesse des moteurs à 0 unités.

        Fixer la vitesse des moteurs à 50 unités.

### c. Solution en Python

Un listing d'une implémentation Python de l'algorithme qui précède est donné en annexe 4.

## 2. Suivi d'une ligne

### a. Problème

Programmer le robot Maqueen de sorte qu'il avance en suivant une ligne noire.

### b. Solution en langage naturel

Fixer la vitesse des moteurs à 50 unités.

Tant Que Vrai :

    Si non((ligne sous capteur gauche) et (ligne sous capteur droit)) :

        Si non(ligne sous capteur gauche) et non(ligne sous capteur droit) :

            Arrêter les moteurs

        Sinon si (ligne sous capteur gauche) et non(ligne sous capteur droit) :

            Arrêter le moteur gauche.

            Fixer la vitesse du moteur droit à 10 unités.

            Tant que non((ligne sous capteur gauche) et (ligne sous capteur droit)) :

                Attendre.

            Fixer la vitesse des moteurs à 50 unités.

    Sinon :

        Arrêter le moteur droit.

        Fixer la vitesse du moteur gauche à 10 unités.

        Tant que non((ligne sous capteur gauche) et (ligne sous capteur droit)) :

            Attendre.

        Fixer la vitesse des moteurs à 50 unités.

### c. Solution en Python

Un listing d'une implémentation Python de l'algorithme qui précède est donné en annexe 5.

## IV. Dépannage

### 1. La carte BBC micro:bit affiche une erreur du type OS Error

Essayer de lancer le programme tout de même sur le Maqueen. La carte BBC micro:bit peut avoir du mal à reconnaître la viabilité du code servant à la définition des fonctions motorL, motorR et motors.

### 2. La carte BBC micro:bit affiche une erreur du type Name Error

Vérifier que les noms des fonctions employés sont tapés sans faute et que les imports ont bien été faits (cf. II.2.c).

Si c'est le cas, vérifier que pyqueen.py est bien présent sur la carte BBC micro:bit via le bouton Fichiers de Mu. Dans le cas où pyqueen.py n'est pas présent sur la carte BBC micro:bit, l'installer en se référant au I.2.

### 3. La carte BBC micro:bit n'affiche pas d'erreur de code, mais le robot Maqueen reste inerte

Vérifier que le robot Maqueen est bien alimenté : lorsque le bouton situé à l'arrière du robot est en position ON, un voyant rouge doit s'allumer (fig. 35).

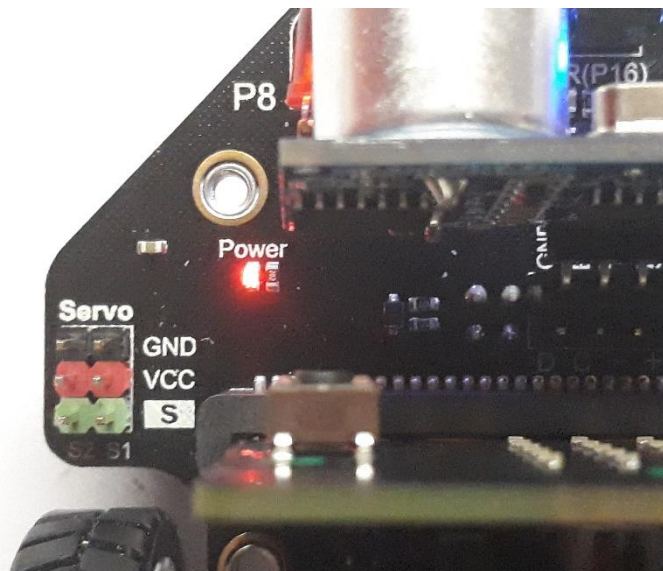


Fig. 35 : Témoin d'alimentation du robot Maqueen.

# Annexes

Pour obtenir des fichiers .py à partir des présentes annexes, il suffit de copier-coller les codes-sources ici fournis dans un éditeur Python et d'enregistrer le fichier sous le nom voulu.

## 1. Listing du programme pyqueen.py

```
#pyqueen v1.0
#Thomas Pochart, décembre 2019

from microbit import *
from machine import *
from music import *

addr=0x10

def motorL(n):
    if n>100:
        n=100
    if n<-100:
        n=-100
    if n>=0:
        direc=0
    else:
        direc=1
    v=abs(n)*255//100
    i2c.write(addr,bytearray([0,direc,v]))

def motorR(n):
    if n>100:
        n=100
    if n<-100:
        n=-100
    if n>=0:
        direc=0
    else:
        direc=1
    v=abs(n)*255//100
    i2c.write(addr,bytearray([2,direc,v]))

def motors(n):
    motorL(n)
    motorR(n)

def lineL():
    if pin13.read_digital():
        return True
    else:
        return False

def lineR():
    if pin14.read_digital():
        return True
    else:
        return False

def distance():
    pin1.write_digital(1)
```

```

    sleep(10)
    pin1.write_digital(0)
    pin2.read_digital()
    t2=time_pulse_us(pin2, 1)
    d=340*t2/20000
    return t2

def bip():
    tune=["A7:0","A7:0","A7:0","A7:0","A7:0","A7:0","A7:0"]
    play(tune)

```

## 2. Code permettant l'import des fonctions liées à l'utilisation du Maqueen

```

from microbit import *
from pyqueen import *

```

## 3. Solution Python pour l'exemple 1

```

from microbit import *
from pyqueen import *

motors(10)
display.show(Image.HAPPY)
while True:
    if distance()<5:
        motors(0)
        break
display.show(Image.SAD)

```

## 4. Solution Python pour l'exemple 2

```

from microbit import *
from pyqueen import *
from random import *

motors(50)
while True:
    if distance()<5:
        motors(0)
        r=randint(0,2)
        if r==1:
            motorR(-50)
            sleep(1000)
            motors(0)
        else:
            motorL(-50)
            sleep(1000)
            motors(0)
    motors(50)

```

## 5. Solution Python pour l'exemple 3

```
motors(50)
display.show(Image.HAPPY)
while True:
    if not (not lineL() and not lineR()):
        if lineL() and lineR():
            motors(0)
            display.show(Image.SAD)
        elif not lineL() and lineR():
            motorL(0)
            motorR(10)
            while not (not lineL() and not lineR()):
                sleep(1)
            motors(50)
        elif else:
            motorR(0)
            motorL(10)
            while not (not lineL() and not lineR()):
                sleep(1)
            motors(50)
```

## **Bibliographie**

- [1] Documentation DFRobot à propos du Maqueen :  
[wiki.dfrobot.com/micro:Maqueen\\_for\\_micro:bit\\_SKU:ROB0148-E\(ROB0148\)](http://wiki.dfrobot.com/micro:Maqueen_for_micro:bit_SKU:ROB0148-E(ROB0148))  
(consultée le 01/12/2019)
- [2] Page du site personnel d'Olivier Lecluse concernant l'utilisation de Python avec le Maqueen :  
[lecluseo.scenari-community.org/CircuitPython/co/g\\_maqueen.html](http://lecluseo.scenari-community.org/CircuitPython/co/g_maqueen.html)  
(consultée le 01/12/2019)
- [3] Site officiel de Mu :  
[codewith.mu](http://codewith.mu)  
(consulté le 01/12/2019)