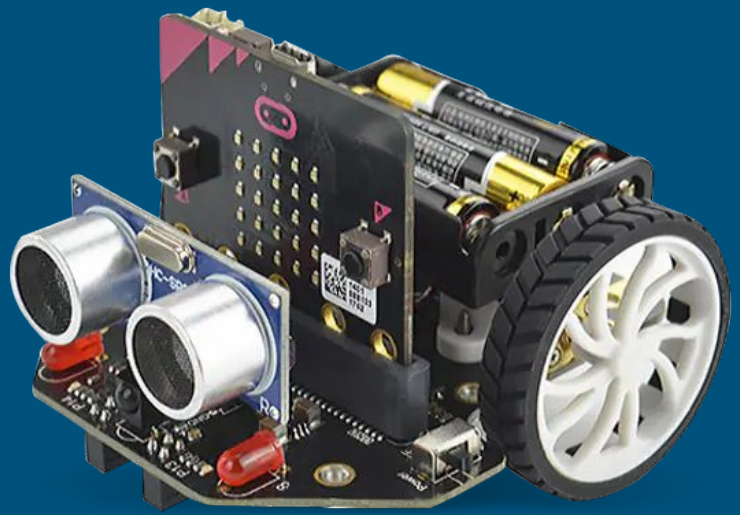
Projet McQueen

Le robot programmable MacQueen, basé sur la carte microbit permet le contrôle de deux moteur intégrés. Son châssis comporte également un capteur à ultrasons, deux suiveurs de lignes, 4 leds RGB, un buzzer, deux leds rouges et un récepteur IR pour le pilotage via une télécommande IR.



Caractéristiques techniques

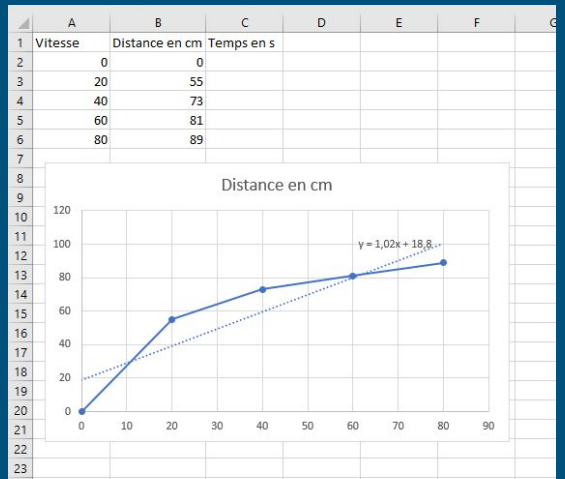
Alimentation :
3,5 à 5 Vcc (via 3 piles/accus AAA non inclus)
3,7 Vcc via accu LiPo (non inclus, support non inclus)
Motoréducteur :
rapport de réduction: 1:150
vitesse de rotation maxi: 133 tr/min
piloté par une liaison PWM
2 x modules suiveurs de lignes (P13 et P14)
1 x buzzer (P0, désactivable via un inter)
4 x leds RGB (P15)
2 x leds rouges (P8 et P12)
1 x récepteur IR (P16)
1 x détecteur US HC-SR04 à enficher sur le connecteur 4 broches prévu (Echo: P2 et Trig: P1)
1 x interface I2C (SCL: P19 et SDA: P20)
2 roues en caoutchouc : ⌀ 43 mm
1 roue avant en plastique
Dimensions : 81 x 85 x 44 mm
Poids : 75,55 g

Déterminer la vitesse réelle du robot:

Permettant de définir une règle d'entrée sortie entre la commande et la vitesse réelle du robot.

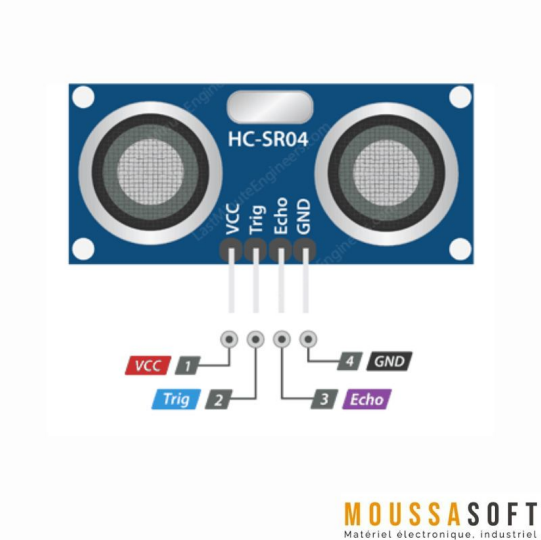
- Ecrire un programme permettant de faire tourner les moteurs à 20 unité de vitesse moteur pendant 2s
- Mesurer la distance qu'il a parcouru
- Calculer sa vitesse
- Répéter ce protocole toute les 20 unités de vitesse moteur
- Rentrer dans Excel les données et créer un graphique avec en abscisse la vitesse du moteur et en ordonnée la vitesse du robot
- Conclure sur le rapport entre vitesse du moteur et la vitesse du robot.

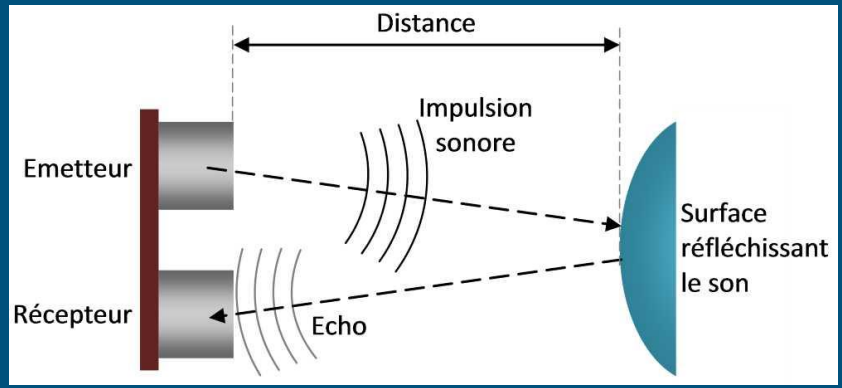
En suivant ce protocole, on sera alors capable de déterminer la vitesse réelle du robot en fonction de la valeur sans unité donnée dans les fonctions moteurDroit(), moteurGauche() et setSpeed() de la librairie Maqueen.oo

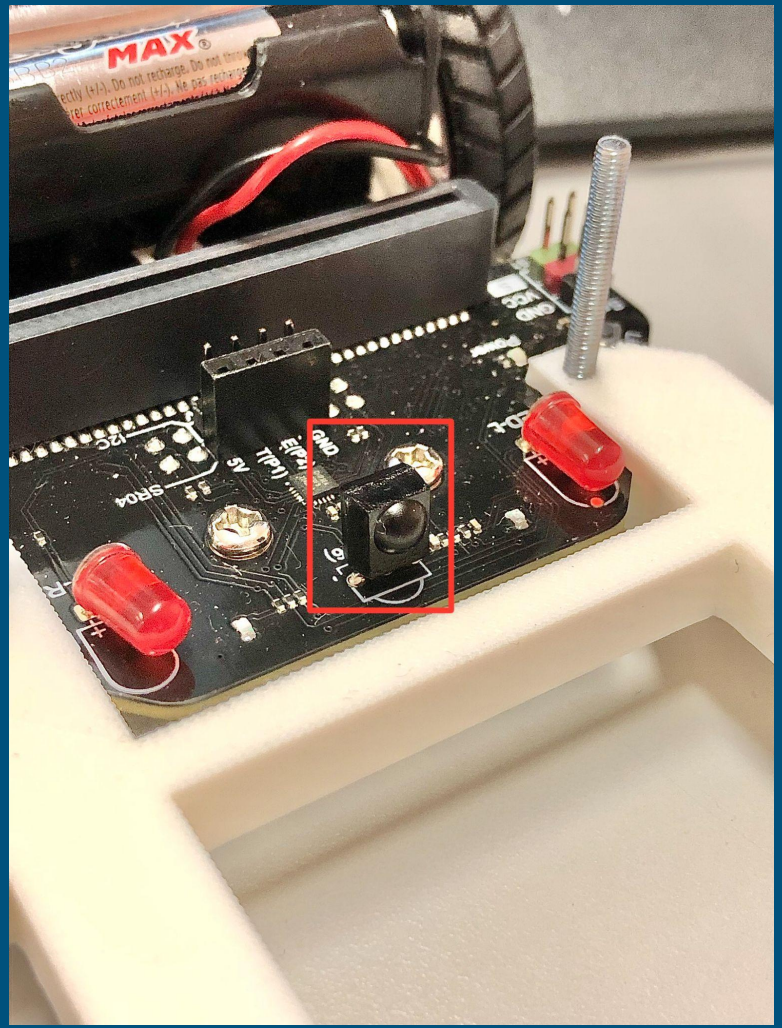


Capteur ultrason

Le capteur ultrason HC-SR04 mesure une distance en émettant et recevant des ondes ultrasons. Il a une portée de 2 cm à 4 m. Il est souvent utilisé sur des systèmes Arduino.







Capteur infrarouge

Le robot possède également un récepteur infrarouge TSOP34838. Il est utile pour pouvoir être commandé grâce à sa télécommande.

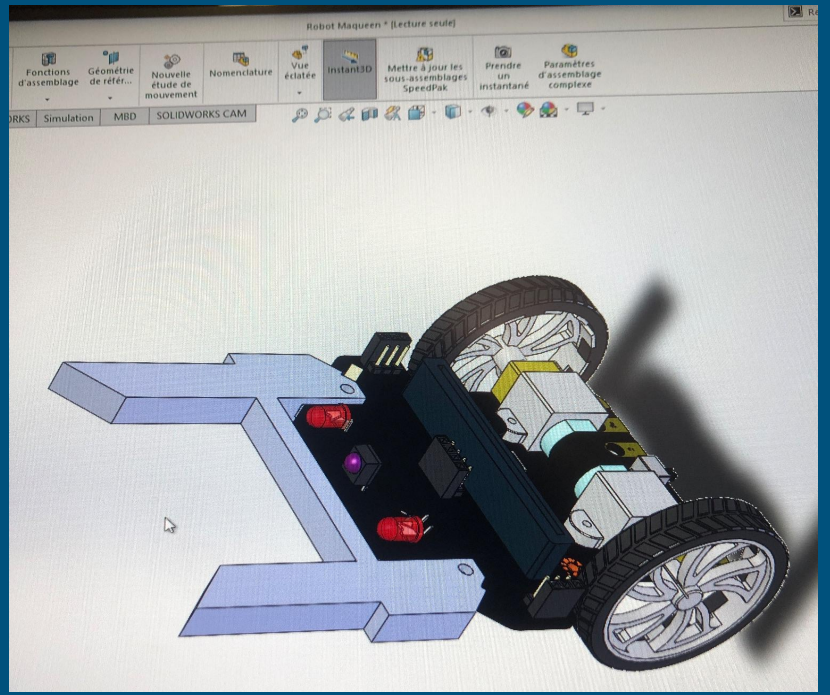
La carte micro:bit est équipée d’une boussole et d’un accéléromètre.

Deux suiveurs de ligne sont présents sous le robot (pin 13 et 14) pour détecter les lignes que l’on peut tracer au marqueur par exemple.

Mécanique

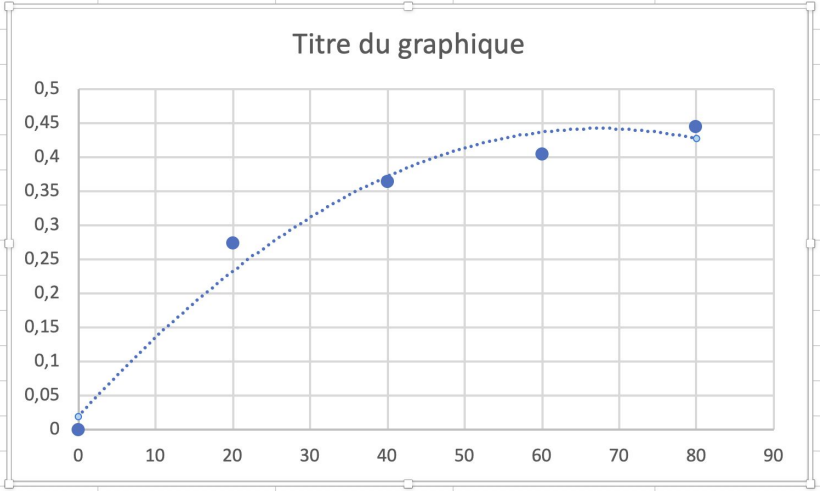
Réaliser une pince

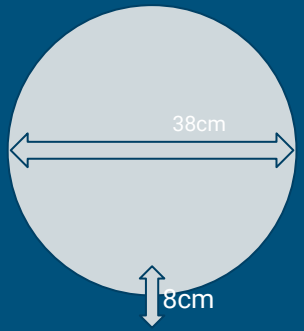
Piloter en vitesse le robot McQueen



Vitesse entrée/sortie

Temps en s	Distance en cm	Vitesse moteur	Vitesse en m/s
2	0	0	0
2	55	20	0,275
2	73	40	0,365
2	81	60	0,405
2	89	80	0,445





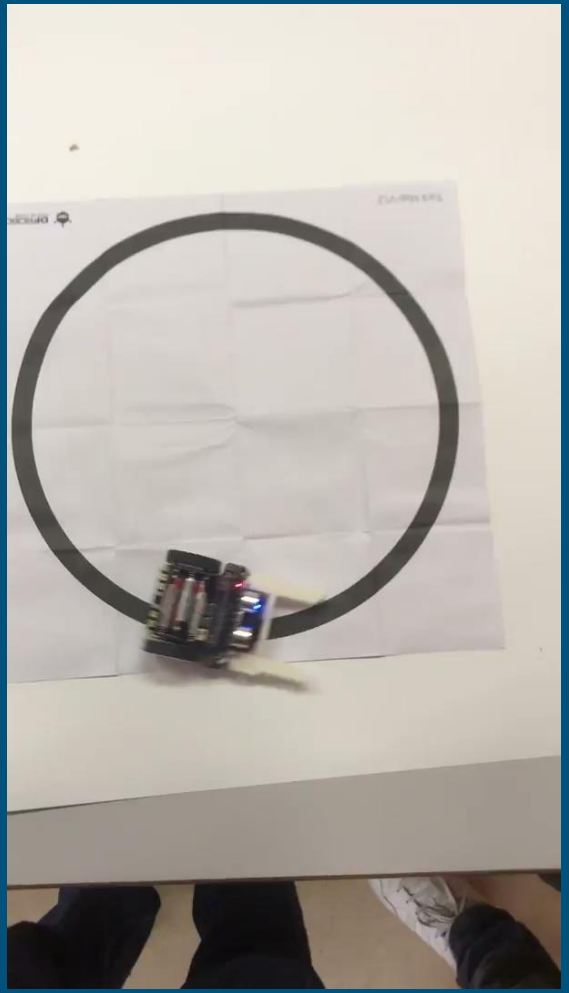
$\omega = 2\pi \times 14/10s = 9$ rad/s
Ri =38/2 -8/2 =15 cm
Re =38/2 +8/2 =23 cm

Pi =2π xRi =94cm
Pe =2π xRe =145 cm

Vi =Pi/10s =9,4 cm/s =0,094 m/s
Ve =Pe/10s =14,5 cm/s =0,145 m/s

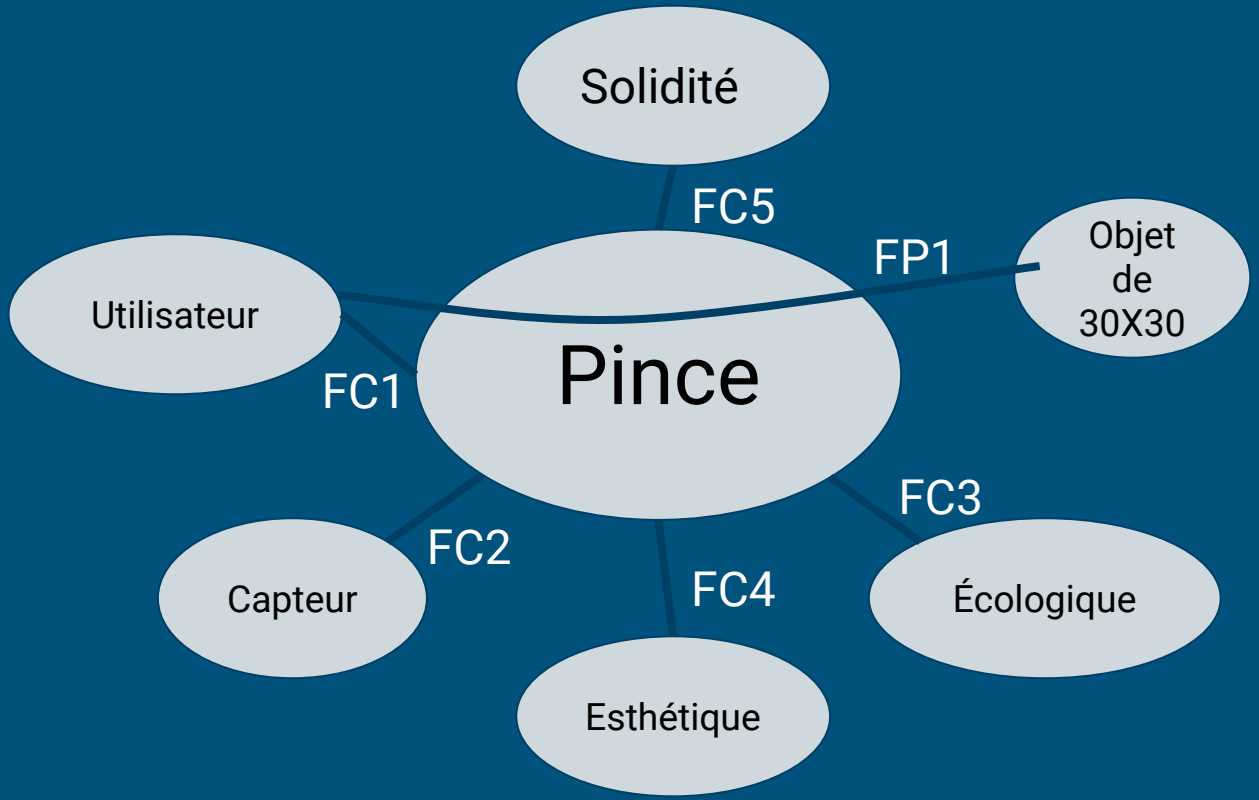
Plage d'utilisation

- Vitesse très peu impactée par le type de sol
- Vitesse non modifiée par la pose de la pince
- Vitesse robot en m/s= 0,0125 x Vitesse moteur

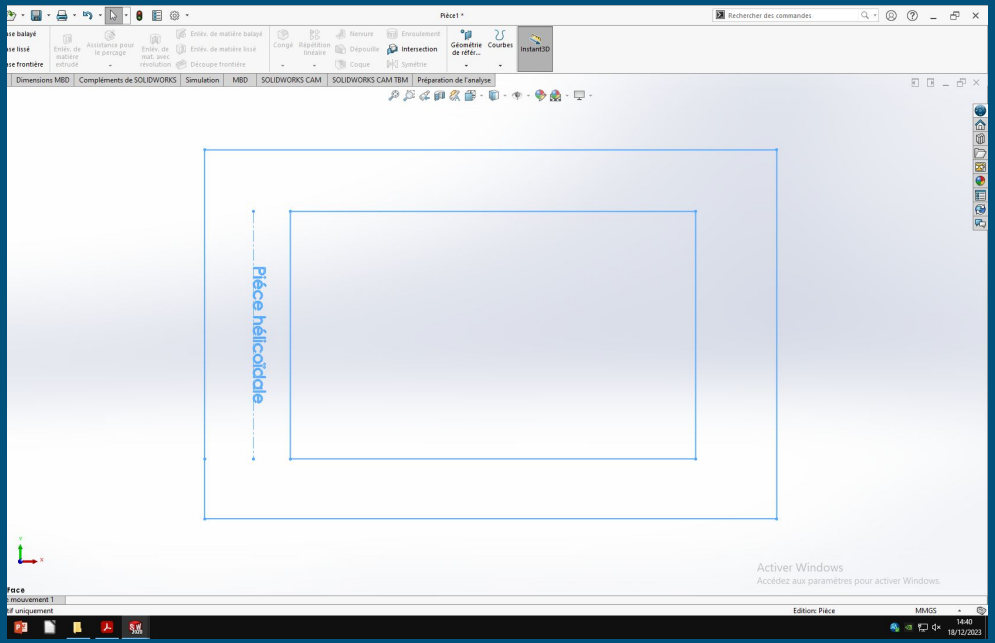


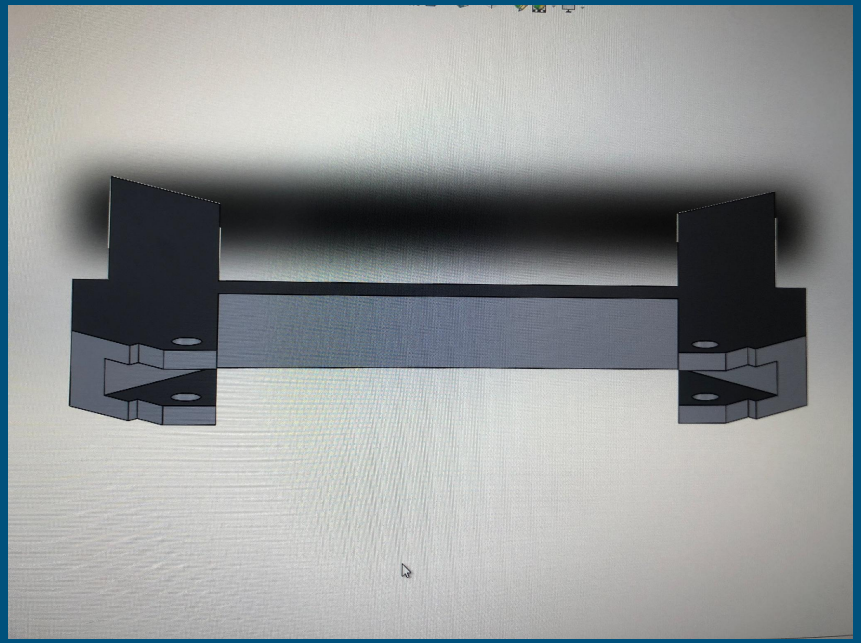
Adaptation robot : Objectif pince

Diagramme bête à corne pour définir besoin



Cahier des charges de la pince





La pince en action

```python
from microbit import *
from maqueen import Maqueen
import radio

# constante
FIND = False
SPEED = 20

# instanciation de la classe Maqueen dans mq
mq = Maqueen()

# modification de la variable _vitesse de la classe Maqueen
mq.setVitesse(SPEED)

# Mode chercheur d'objet
def search_obstacle(_time: int, num: int):
    # num 1 : tourne à gauche
    # num -1 : tourne à droite
    t = running_time()
    mq.moteurGauche(-num * SPEED)
    mq.moteurDroit(num * SPEED)
    while running_time() - t < _time:
        if mq.distance() <= 20:
            break
    mq.avance()  # reset moteurD et moteurG pour aller tout droit

mq.avance()
while not FIND:
    if pin13.read_digital() == 0: # tourne à droite
        search_obstacle(900, -1)
    elif pin14.read_digital() == 0: # tourne à gauche
        search_obstacle(900, 1)
    if mq.distance() < 2: # trouve l'objet
        mq.stop()
        FIND = True
```

Mode 1: détecteur d'objets

Cette partie du code permet au robot de détecter un objet (ici un bouchon en liège) dans un parcours limité par des lignes noires qu'il doit éviter. Quand il touche une ligne il tourne sur lui même, le sens de sa révolution est déterminé par les capteurs de lignes. Si le capteur droit (pin 14) renvoie 0, le robot sort par la droite donc il tournera à gauche pour rester dans le terrain, même protocole pour le capteur gauche (pin 13). Si le robot trouve un objet à moin de 20cm de lui pendant sa révolution, il s'arrête de tourner et avance tout droit jusqu'à l'objet, il s'arrête quand l'objet est à moin de 2 cm de lui pour passer à la suite du code…

Code de la télécommande

```
from microbit import *
from maqueen import Maqueen
import radio

SPEED = 20
mq = Maqueen()
radio.on()
radio.config(group=8)

while True:
    if button_a.is_pressed() and not button_b.is_pressed(): # left
        radio.send("l")
    if button_b.is_pressed() and not button_a.is_pressed(): # right
        radio.send("r")
    if button_b.is_pressed and button_a.is_pressed():
        radio.send("s")
```

Mode 2 et 3: robot télécommandable et suiveur de ligne

```
# Mode télécommande
radio.on()
radio.config(group=8)

while pin13.read_digital() == 1 and pin14.read_digital() == 1: # tant qu'on ne touche pas de bordure
    incoming = radio.receive()
    if incoming == "r":
        mq.moteurDroit(0)
        mq.moteurGauche(SPEED)
    elif incoming == "l":
        mq.moteurGauche(0)
        mq.moteurDroit(SPEED)
    elif incoming == "s":
        mq.avance()
    else:
        mq.stop()

# Mode suiveur de ligne
while True:
    if pin14.read_digital() == 1: # il dévie à droite donc active le moteur droit
        mq.moteurDroit(SPEED)
        mq.moteurGauche(0)
    elif pin13.read_digital() == 1: # il dévie à gauche donc on active le moteur gauche
        mq.moteurGauche(SPEED)
        mq.moteurDroit(0)
    else: # il est au milieu du tracé donc il va tout droit
        mq.avance()
```

Cette partie du code permet dans un premier temps au robot d’être télécommandé par une autre carte Microbit et dans un deuxième temps, une fois la bordure du terrain touchée, de suivre la ligne noire jusqu’à ce qu’on l’arrête.

Le robot en fonctionnement