

1. BIBLIOTHÈQUES

Import des bibliothèques pour utiliser pleinement la carte du robot Maqueen :

```
from microbit import *
import time
import machine
import music
```

Attention, il faut flasher le programme avec la carte allumée.

2. MOTEURS

Commandes	Instruction Python	Remarques
Mettre en mouvement le robot (I2C)		
Arrêter le moteur gauche	<code>i2c.write(0x10, bytearray([0x00, 0x0, 0]))</code>	
Arrêter le moteur droit	<code>i2c.write(0x10, bytearray([0x02, 0x0, 0]))</code>	
Faire tourner le moteur gauche (ou droit)	<code>i2c.write(0x10, bytearray([0x00, 0x0, 0]))</code>	
	Adresse I2C 0x10 par défaut	Moteurs gauche : 0x00 droit : 0x02
		Sens horaire : 0x0 trigo : 0x1
		Vitesse de 0 à 255

3. CAPTEURS INTÉGRÉS

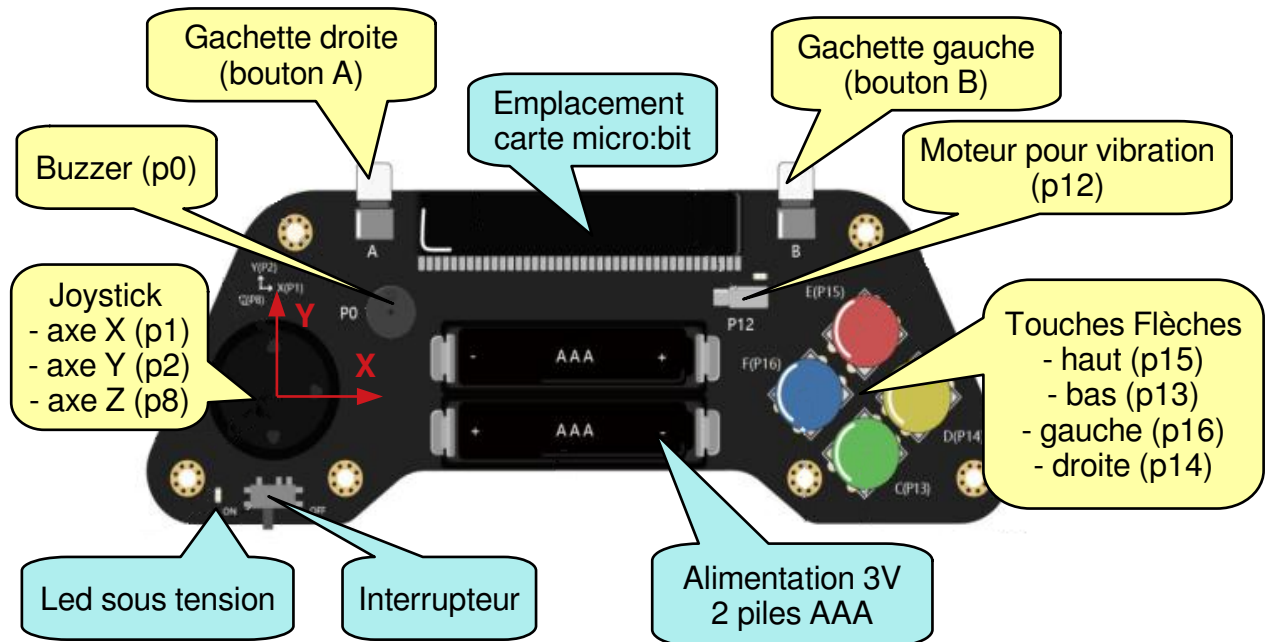
Commandes	Instruction Python	Remarques
Mesure de la distance avec un obstacle avec le capteur ultra-son (p1, p2)		
Émission d'une onde pendant 10 ms (Trigger)	<code>pin1.write_digital(1)</code> <code>sleep(10)</code> <code>pin1.write_digital(0)</code>	
Réception de l'écho et mesure du temps d'attente	<code>pin2.read_digital()</code> <code>t = machine.time_pulse_us(pin2, 1)</code>	
Calcul de la distance	<code>d = 340 * t / 20000</code>	distance en cm
Suiveur de ligne		
Détection d'une ligne à gauche	<code>if pin13.read_digital():</code> <code>display.show("gauche")</code>	Détection de l'absorption de lumière par une ligne noire
Détection d'une ligne à droite	<code>if pin14.read_digital():</code> <code>display.show("droite")</code>	

4. LEDS

Commandes	Instruction Python	Remarques
Leds Rouges avants (p8 et p12)		
Allumer led gauche	<code>pin8.write_digital(1)</code>	<code>pin12</code> pour la led droite
Éteindre led gauche	<code>pin8.write_digital(0)</code>	
Leds d'ambiance RVB (p15)		
Définir une couleur sur une led RVB	<code>bnpMaq[1] = (51, 255, 255)</code> <code>npMaq.show()</code>	<code>NbpMaq[1]</code> est la led n°1, les numéros des leds vont de 0 à 3.
Le codage de la couleur est basé sur 3 octets (de 0 à 255) définissant les quantités de Rouge, de Vert et de Bleu. Ici la couleur est turquoise. Blanc = (255, 255, 255), Rouge = (255, 0, 0), Noir (éteint) = (0, 0, 0), ...		

5. SON

Commandes	Instruction Python	Remarques
Jouer de la musique (p0)		
Définis et joue une mélodie	<code>tune=["A7:0", "G7:0", "E7:0", "C7:0", "D7:0", "B7:0", "F7:0", "C8:0", "A7:0", "G7:0", "E7:0", "C7:0", "D7:0", "B7:0", "F7:0", "C8:0"]</code> <code>music.play(tune)</code>	
Le codage d'une note une chaîne de caractère avec la note, l'octave et la durée. La note est le système anglo-saxon C D E F G A B (C) pour DO RÉ MI FA SOL LA SI (DO).		
Joue une mélodie prédéfinie	<code>music.play(DADADADUM)</code>	
'DADADADUM', 'ENTERTAINER', 'PRELUDE', 'ODE', 'NYAN', 'RINGTONE', 'FUNK', 'BLUES', 'BIRTHDAY', 'WEDDING', 'FUNERAL', 'PUNCHLINE', 'PYTHON', 'BADDY', 'CHASE', 'BA_DING', 'WAWAWAWAA', 'JUMP_UP', 'JUMP_DOWN', 'POWER_UP'		
Arrêter la musique	<code>music.stop()</code>	
Effet sonore (p0)		
Émet un son	<code>music.pitch(440, 500)</code>	Ici 440 est la fréquence (Hz) et 500 la durée (ms)



6. MANETTE

Commandes	Instruction Python	Remarques
Boutons et joystick (p1, p2, p8, p13, p14, p15 et p16)		
Test d'un bouton pressé (ici l'axe X)	<code>if pin1.read_digital() == 0: display.show("X")</code>	Valable aussi pour <code>pin2</code> , <code>pin8</code> , <code>pin13</code> , <code>pin14</code> , <code>pin15</code> , et <code>pin16</code>
Au repos la broche délivre un signal, il faut donc détecter l'absence de signal.		
Bouton A pressé	<code>if button_a.is_pressed(): display.show("A")</code>	Valable aussi pour <code>button_b</code>
Valeur de l'axe X	<code>pin1.read_analog()</code>	Valeur entre 0 et 1018, 525 pour la position au repos
Valeur de l'axe Y	<code>pin2.read_analog()</code>	
Vibration (p12)		
Faire vibrer la manette	<code>pin12.write_digital(1)</code>	
Arrêter les vibrations	<code>pin12.write_digital(0)</code>	
Son (p0) – similaire au Robot Maqeen (voir page 2)		