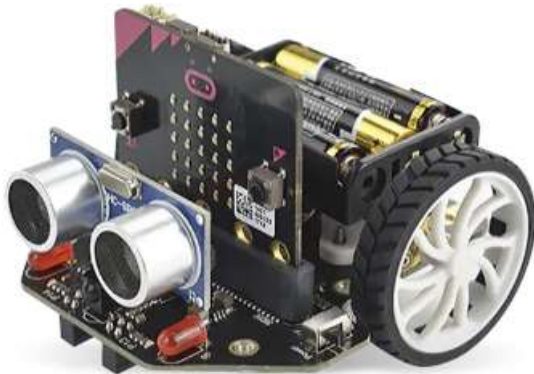
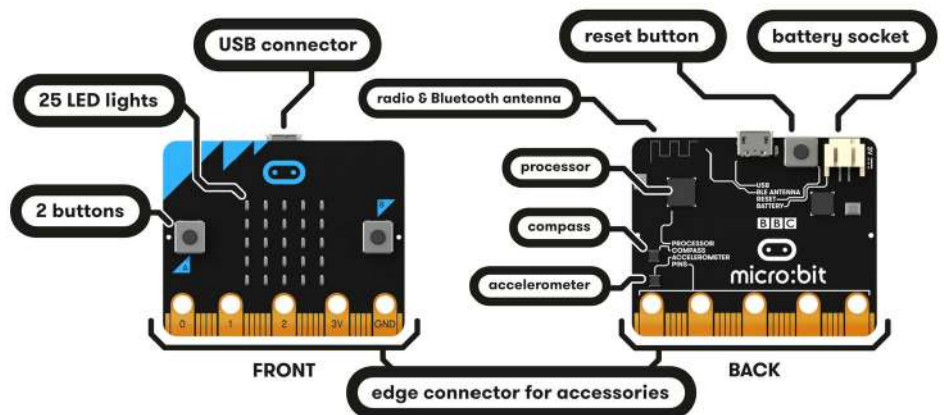


1. Contextualisation

La carte micro:bit, éditée par la BBC, est un microcontrôleur (microprocesseur avec mémoire et entrées/sorties). Elle est munie d'un processeur ARM et de plusieurs capteurs et interfaces de connexion. Il n'a donc pas de système d'exploitation mais permet d'exécuter des programmes.

Fonctionnalités incluses :

Capteurs de lumière, de température, broches de connexion, communication sans fil (Radio et Bluetooth), interface USB.



Le robot micro:Maqueen pour carte micro:bit est basé sur une carte permettant le contrôle des deux moteurs intégrés. Ce châssis comporte également un détecteur à ultra son, deux suiveurs de lignes ...

Il est alimenté à 3,5 V par un bloc de 3 piles AAA.

Prise en main :

✂ À Faire : Effectuer les actions suivantes :

1. Télécharger le fichier [maqueen.py](#) sur le site de M. Ramstein
2. Se rendre à l'adresse <https://python.microbit.org/v/3>
3. Dans l'interface Web, nous allons importer un fichier nécessaire au fonctionnement du robot

1. Cliquer sur



2. Cliquer sur open

Open...

3. Sélectionner le fichier maqueen.py téléchargé à l'étape 1

4. Cliquer sur puis cliquer sur ☒ Add file maqueen.py puis **Confirm**

4. Connecter la carte micro:bit à l'ordinateur via le câble USB,

5. Cliquer sur **Send to micro:bit** ; cela a pour effet de charger le code python sur la carte.

2. Faire avancer le robot

```
1 from microbit import *
2 from maqueen import Maqueen
3
4 robot = Maqueen()
5 robot.avance(40) # Valeur de 0 à 100
6 sleep(500)      # Valeur en ms
7 robot.stop()
```

✂ À Faire : Effectuer les actions suivantes :

1. Modifier les valeurs des lignes 5 et 6 pour étudier le comportement du robot.
2. À chaque modification de code, ne pas oublier de cliquer sur 

3. Quel est l'effet de la ligne 5 ?

4. Quel est l'effet de la ligne 6 ?

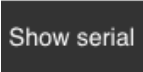
3. Faire tourner le robot sur lui-même

```
1 from microbit import *
2 from maqueen import Maqueen
3
4 robot = Maqueen()
5 robot.moteurGauche(40) # Vitesse du moteur gauche en % de la puissance
6 sleep(500)
7 robot.stop()
```

✂ À Faire : Modifier le code pour faire tourner le robot sur lui-même, i.e à 360°.

4. Avancer le robot jusqu'à détection d'un obstacle

```
1 from microbit import *
2 from maqueen import Maqueen
3
4 robot = Maqueen()
5 while True:
6     distance_obstacle = robot.distance() # Mesure la distance d'un éventuel obstacle
7     print(distance_obstacle)           # Affiche la distance
8     sleep(500)                          # Temporise de 500 ms
```

✂ À Faire : Modifier le code pour qu'il affiche un message "Attention, obstacle" lorsque le robot détecte un obstacle à moins de 15 cm. Aide : Cliquer sur  pour afficher les valeurs.

✂ À Faire : Modifier le code pour que le robot avance à 40 % de sa puissance de manière infinie et s'arrête lorsqu'il détecte un obstacle à moins de 15cm.

5. Suivre une ligne

```

1 from microbit import *
2 from maqueen import Maqueen
3
4 robot = Maqueen()
5
6 while True:
7     detecteur_gauche = pin13.read_digital() # Mesure la valeur du capteur de suivi gauche
8     detecteur_droit = pin14.read_digital()   # Mesure la valeur du capteur de suivi droit
9     print(detecteur_gauche, "-", detecteur_droit) # Affiche les valeurs
10    sleep(500)

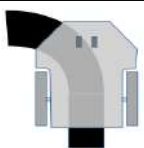
```

✂ À Faire : Effectuer les actions suivantes :

1. Exécuter le code et tester sur un support comportant une ligne noire sur fond blanc.
2. Quel est le domaine de valeur des capteurs de suivi ?
3. Quelle est la correspondance entre les valeurs mesurées et la couleur du support détectée ?

Couleur détectée	Valeur mesurée
Blanc	
Noir	

✂ À Faire : Modifier le code pour que le robot suive une ligne noire selon les règles suivantes :

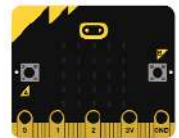
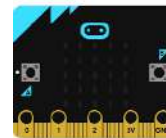
	Quand les 2 capteurs suiveurs de ligne voient du noir, le robot doit aller tout droit. Les moteurs sont à la même vitesse.
	Lors d'un virage à droite, le capteur de gauche voit du blanc et le capteur de droite voit du noir. Il faut que le robot tourne à droite, ce qui implique que le moteur de droite s'arrête pour que le moteur de gauche tourne plus vite.
	Lors d'un virage à gauche, le capteur de gauche voit du noir et le capteur de droite voit du blanc. Il faut que le robot tourne à gauche, ce qui implique que le moteur de gauche s'arrête pour que le moteur de droite tourne plus vite.
	Quand les 2 capteurs suiveurs de ligne voient du blanc, le robot s'arrête.

6. Contrôler le robot via une télécommande micro:bit

La carte micro:bit est équipée d'un émetteur/récepteur radio qui permet à deux cartes de communiquer à distance.

Durant cette activité :

- Vous travaillerez en binôme avec chacun une carte.
- Un code canal vous est attribué. Notez-le ici



À Faire : Effectuer les actions suivantes :

	Carte « Émetteur »		Carte « Récepteur »
1	from microbit import *	1	from microbit import *
2	import radio	2	import radio
3		3	
4	radio.config(channel=X) # Active le canal	4	radio.config(channel=X) # Active le canal
5	radio.on() # Active la connexion radio	5	radio.on() # active la connexion radio
6	while True:	6	while True:
7	if button_a.was_pressed():	7	message = radio.receive()
8	radio.send("Hello !")	8	if message:
		9	display.scroll(message)

1. Copier les codes ci-dessus dans les interfaces Web et cliquer sur 
2. Décrire l'effet du programme sur la carte. Une interaction est-elle possible ?

3. Quelles lignes permettent d'envoyer et recevoir un message ? Quelle type d'information est échangée ?

À Faire : Modifier les codes des cartes « Émetteur » et « Récepteur » pour que les règles suivantes s'appliquent.

Action de l'« Émetteur »	Effet sur le « Récepteur »
Appui sur le bouton A	Le robot tourne à gauche
Appui sur le bouton B	Le robot tourne à droite