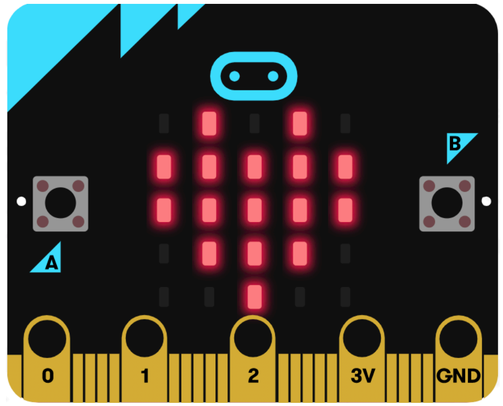
Programmer la carte micro:bit

avec python

DRANE
délégation académique
au numérique éducatif
académie Normandie
RÉGION ACADÉMIQUE



YES WE CODE !

Programmer la carte micro:bit

- Le dispositif *Yes we code!* de la fondation CGénial
- Programmer avec Mu
- Les fonctionnalités de la carte
- Premiers exemples
- Le ruban de led Neopixel
- Le kit Grove
- Le robot Maqueen
- Idées de projet

Lien vers le site académique SNT-NSI :

Philippe Leclerc et Nathalie Weibel

avec python

Le dispositif *Yes we code!*

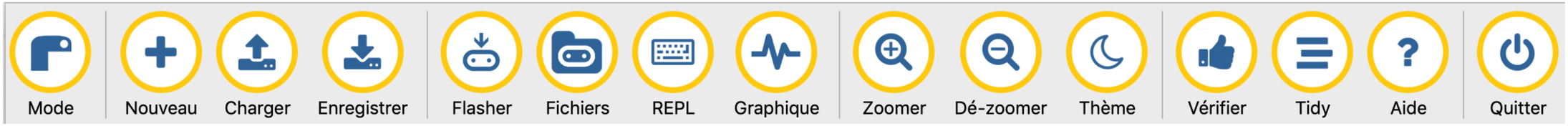
Richard Fuentes et Alexandra Costrachevici

- Informations : le site de la fondation CGénial
- Yes We Code! en Normandie : carte géographique
- Le padlet Yes We Code!

FONDATION CGENIAL

YES WE CODE !

Programmer avec Mu

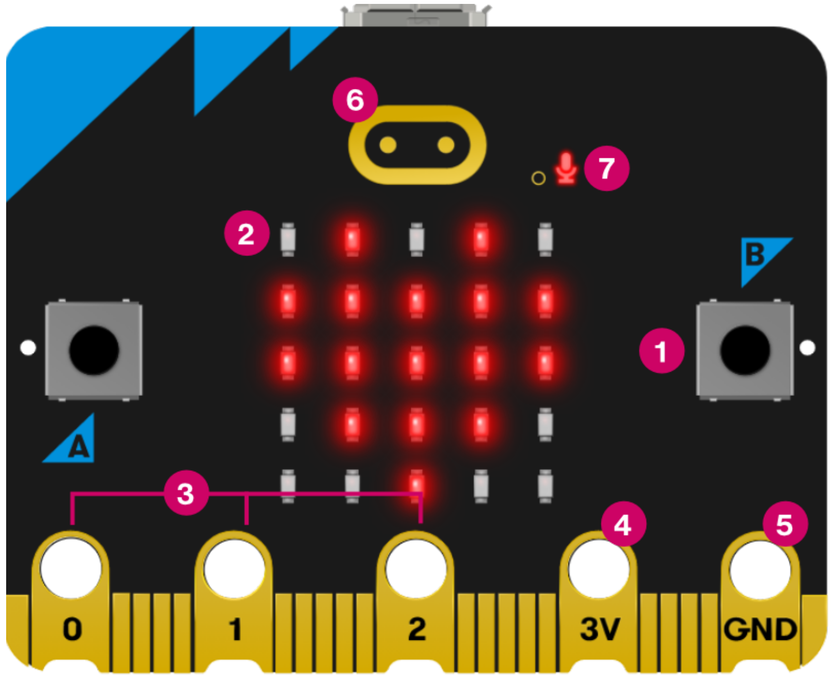


v. 1.1.0

- Mu : éditeur de code multiplateforme, avec un mode micro:bit
- auto-complétion, indentation automatique, vérification du code
- flashage des programmes sur la carte
- affichage graphique des données, console interactive REPL

Pour certains usages on pourra préférer : Vittascience, Edublocks ou Create with code

Les fonctionnalités de la carte



à l'avant :

1 : deux boutons poussoirs A et B programmables

2 : 25 LEDs rouges, programmables

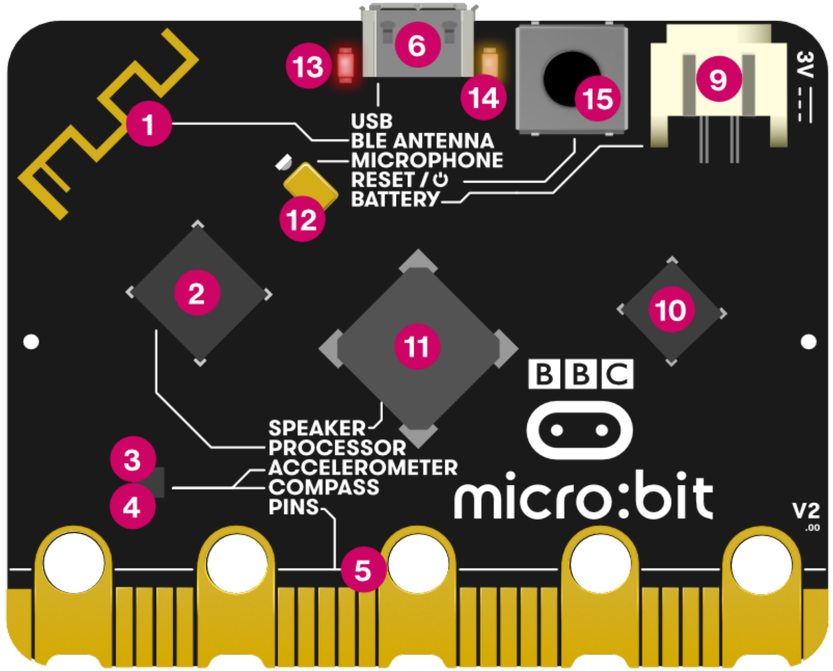
3, 4, 5 : des broches de connexion, d'alimentation.

Et sur la V2 :

6 : un logo tactile

7 : une led témoin du microphone

Les fonctionnalités de la carte



à l'arrière :

1. une antenne radio et bluetooth,
2. un micro-processeur et capteur de température,
3. un magnétomètre (boussole),
4. un accéléromètre,
5. des broches de connexion,
6. un port micro-USB,
9. un connecteur d'alimentation (2 piles AAA - 3V),
10. une puce d'interface USB

Et sur la V2 uniquement :

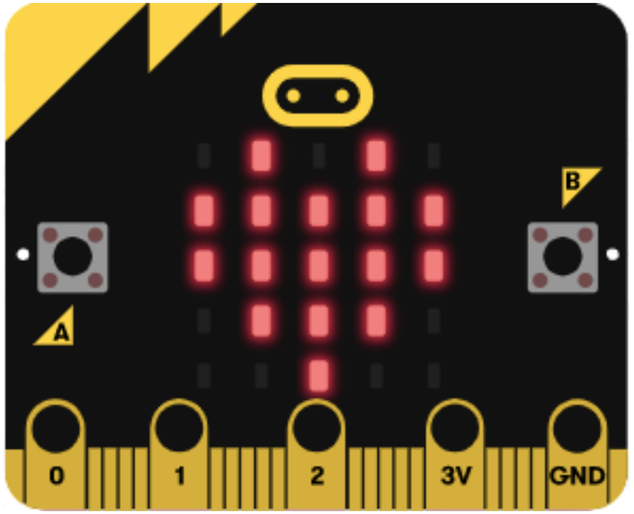
11. un haut-parleur
12. un microphone
13. une LED rouge, témoin d'alimentation
14. une LED jaune, témoin de communication USB
15. un bouton de réinitialisation et de marche/arrêt

Parcours de découverte

Exercice 1

```
display.scroll("Hello, World!")
```

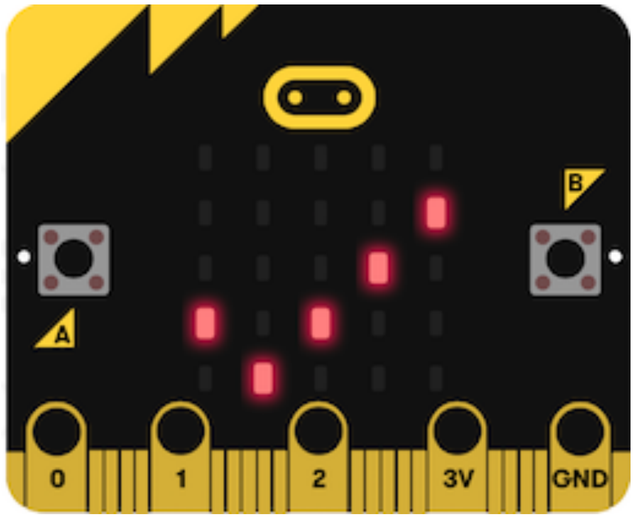
```
display.show(Image.HEART)
```



Parcours de découverte

Exercice 2

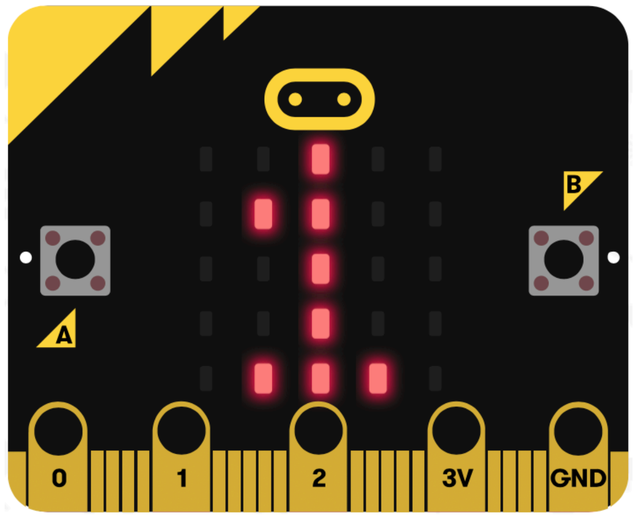
```
from microbit import *
display.show("?")
while True:
    if button_a.is_pressed():
        display.show(Image.YES)
    elif button_b.is_pressed():
        display.show(Image.NO)
```



Parcours de découverte

Exercice 3

```
from microbit import *
while True :
    display.show(1)
    sleep(500)
    display.clear()
    sleep(500)


for k in range(10):
    display.show(k)
    sleep(500)
```

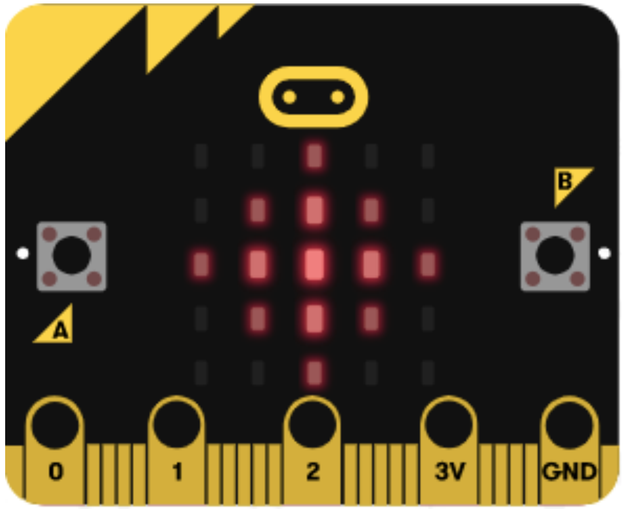


Faire clignoter un affichage. Utiliser des boucles

Parcours de découverte

Exercice 4

```
from microbit import *
eclat1 = Image("00300:03630:36963:03630:00300")
eclat2 = Image("00300:03330:33333:03330:00300")
while True:
    if button_a.is_pressed():
        display.show(eclat1)
        sleep(1000)
        display.show(eclat2)
```



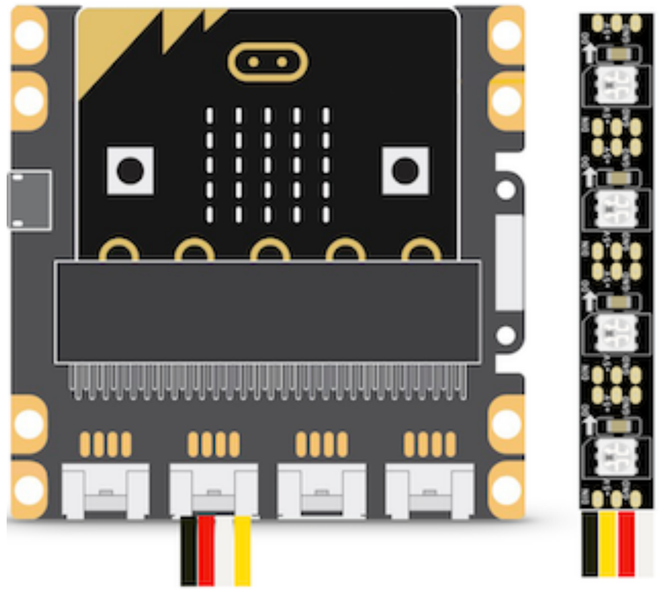
Parcours de découverte

Exercice 5

```
from microbit import *
import radio
radio.on()
vote_ok = True
while vote_ok:
    display.show(Image.SQUARE_SMALL)
    sleep(100)
    if button_a.is_pressed():
        display.show(Image.YES)
    elif button_b.is_pressed():
        display.show(Image.NO)
    if button_a.was_pressed():
        radio.send("A")
        vote_ok = False
    elif button_b.was_pressed():
        radio.send("B")
        vote_ok = False
```

Exercice 6

```
from microbit import *
import neopixel
np = neopixel.NeoPixel(pin0, 30)
while True:
    for i in range(30):
        np[i] = (255, 0, 0)
         sleep(200)
         np.show()
    np.clear()
```



Défi : Jeu de Nim sur un ruban NeoPixel

Matériel : 3 cartes micro:bit, 1 ruban neopixel

Fonctionnement attendu :

- Deux joueurs s'affrontent au jeu de Nim. Chacun joue à l'aide d'une carte micro:bit. L'un des joueurs a la couleur rouge, l'autre la couleur bleu.
- Le ruban NeoPixel est piloté par une 3e carte : il est totalement allumé et toutes ses leds sont éclairées en blanc.
- Les joueurs doivent s'approprier tour à tour entre une et trois leds qui s'afficheront en rouge ou bleu selon le joueur. Celui qui s'empare de la dernière led du ruban gagne la partie.

Fonctionnement

Défi : Jeu de Nim sur un ruban NeoPixel

- Chacun leur tour, les joueurs appuient 1, 2 ou 3 fois sur le bouton A, et valident leur choix avec le bouton B. La carte affiche les chiffres 1, 2 ou 3 lors des appuis sur le bouton A et l'image `Image.YES` lors de la validation avec le bouton B.
- La valeur choisie est envoyée par radio à la carte qui pilote le ruban de led. Le nombre correspondant de led(s) s'allume en couleur sur le ruban, en rouge ou bleu selon le joueur.
- Lorsque la dernière led est atteinte, le ruban fait clignoter ses leds de la couleur du joueur gagnant.

Fonctionnement

Défi : Jeu de Nim sur un ruban NeoPixel

Programme sur la carte du joueur A

```
from microbit import *
import radio
radio.on()

nombre = 0
while True:
    if button_a.was_pressed() and nombre < 3:
        nombre = nombre + 1
        display.show(nombre)
        sleep(100)
    if button_b.was_pressed():
        radio.send('R' + str(nombre))
        display.show(Image.YES)
        nombre = 0
        sleep(500)
```

Programme sur la carte du joueur B

Programme identique en remplaçant R par B

cartes des joueurs

Défi : Jeu de Nim sur un ruban NeoPixel

Programme sur la carte pilotant le ruban NeoPixel

```
from microbit import *
import neopixel
import radio

radio.on()
np = neopixel.NeoPixel(pin0, 30)

rouge = (240, 0, 0)
bleu = (0, 0, 240)
blanc = (200, 200, 200)

for x in range(30):
    np[x] = blanc
np.show()
sleep(500)

position = 0
code = ""
nombre = 0
```

```
def clignoter(couleur):
    for x in range(30):
        np[x] = couleur
    np.show()
    sleep(500)
    np.clear()
    sleep(100)

while True:
    info = radio.receive()
    if info:
        code = info[0]
        nombre = info[1]
        if code == "R":
            couleur = rouge
        elif code == "B":
            couleur = bleu
        coup = 0
        while position < 30 and coup < int(nombre):
            np[position] = couleur
            np.show()
            sleep(100)
            position = position + 1
            coup = coup + 1
    if position == 30:
        clignoter(couleur)
    sleep(500)
```

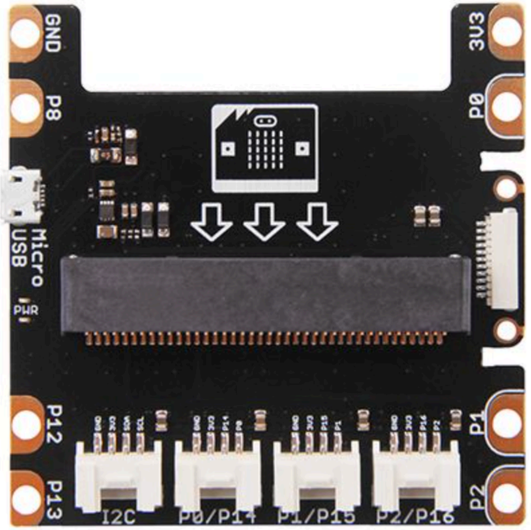
carte pilotant le ruban

Le kit Grove

- Remarques sur le matériel
- Ressources pour exploiter le kit
  - Articles sur site disciplinaire SNT/NSI de Normandie
  - Livret
  - Autres tutoriels
- Démos
  - Éclairage automatique “intelligent”
  - Télémètre numérique

Le kit Grove

Alimentation externe 5V
pour l'utilisation de composants
"gourmands" en énergie










Remarques sur le matériel

Le kit Grove

Articles sur site disciplinaire : https://cutt.ly/dhEQnWY

Ressources pour exploiter le kit

Le kit Grove

Adaptation du livret fourni avec le kit Grove

- Librairies de fonctions en micro-python
- Mini-projets développés pour mu-editor

Le kit Grove

Articles

Mesurer la vitesse du son avec un microcontrôleur
27 mars 2019

Mini-projets Grove/Micro:bit
5 février

Kit Grove pour micro:bit et applications en SNT
5 mars 2020

Documents joints

Initiation Mu-editor avec Microbit

Découverte de la carte microbit avec l'éditeur MU

Radar de recul

Activité Objets Connectés

Clignotants vélo

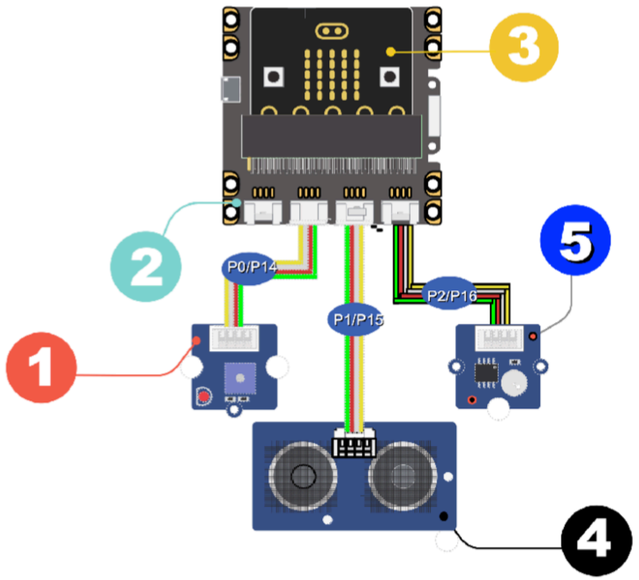
Activité Objets Connectés

Autres tutoriels

Le kit Grove

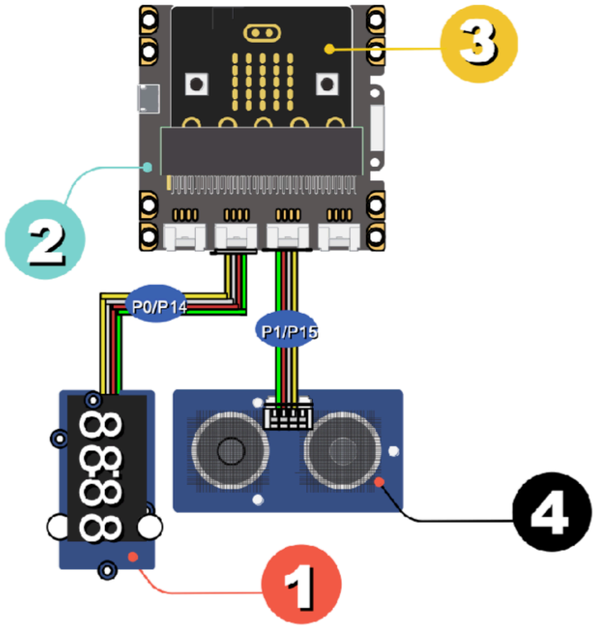
Eclairage automatique

1. Led
2. Interface Grove
3. Micro:bit
4. Capteur à ultra-sons
5. Capteur de luminosité



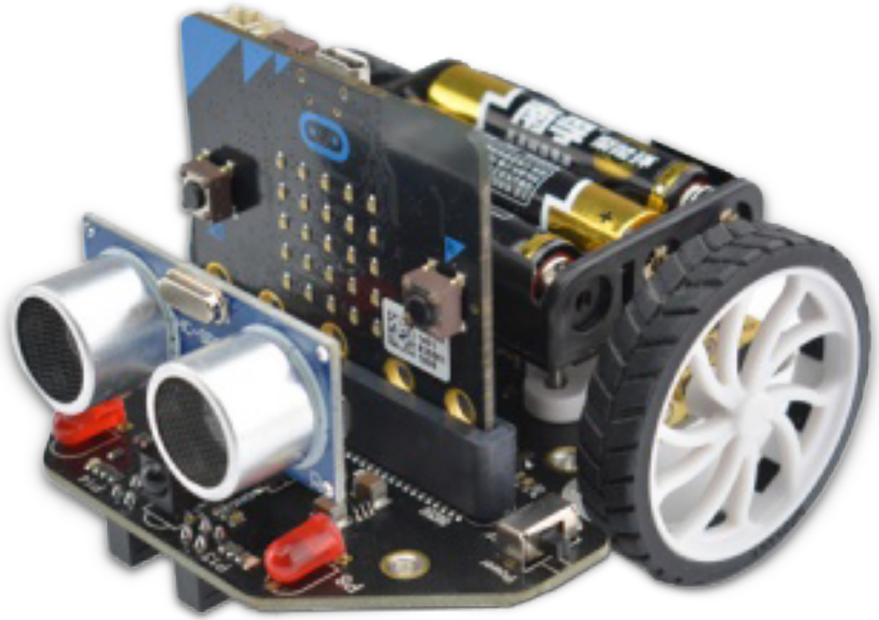
Télémètre

1. Afficheur 4 digits
2. Interface Grove
3. Micro:bit
4. Capteur à ultra-sons

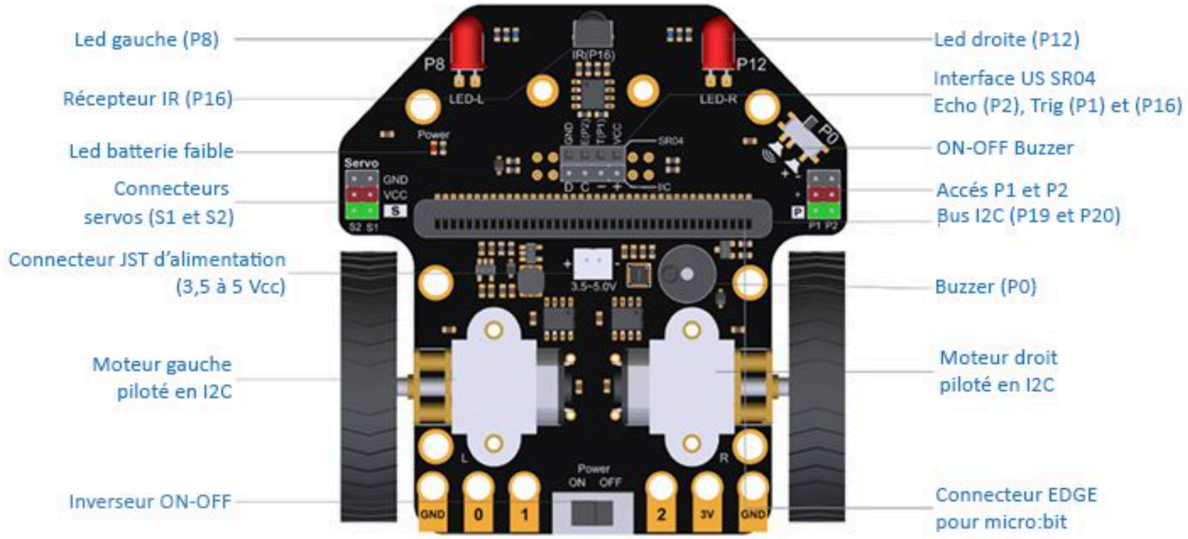


Démos

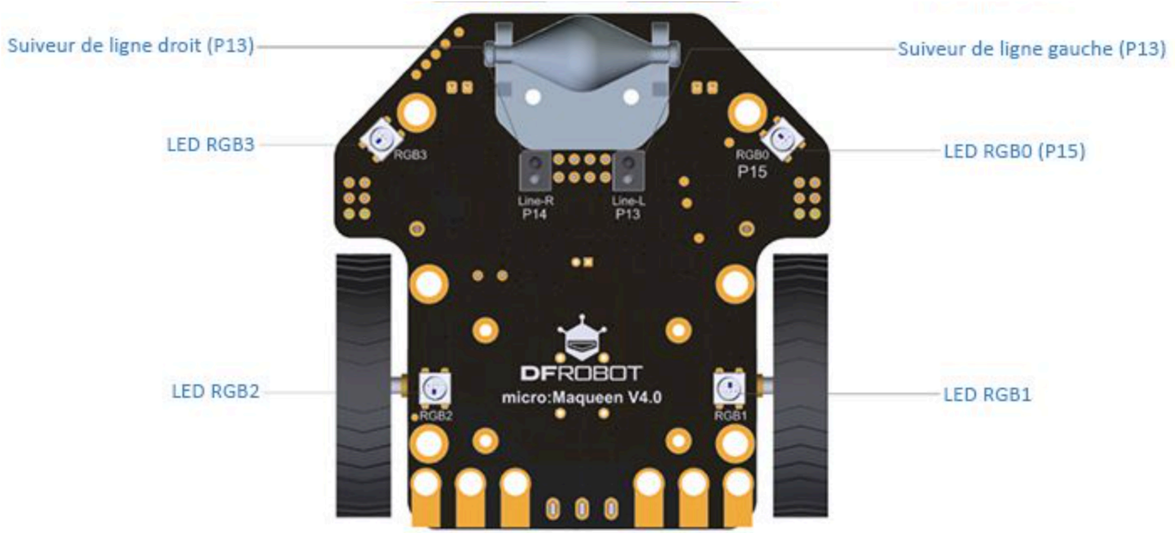
Le robot Maqueen



Le robot Maqueen



Dessus



Dessous

Composants de la carte

Le robot Maqueen

Articles

Mesurer la vitesse du son avec un microcontrôleur
27 mars 2019

Mini-projets Grove/Micro:bit
5 février

Kit Grove pour micro:bit et applications en SNT
5 mars 2020

Dispositif Yes We Code 2020/21
8 décembre 2020

Maqueen avec mu-editor
18 janvier

Documents joints

Programme de démonstration

Librairie Maqueen

Ressources

Le robot Maqueen

mesure_distance() : **Renvoie la distance en cm**

capteurD : Renvoie l'état du capteur droit

capteurG : Renvoie l'état du capteur gauche

allume_led('D') : Allume la led Droite ou Gauche

eteint_led('D') : Éteint la led Droite ou Gauche

moteurD(vitesse) : fait tourner le moteur droit

moteurG(vitesse) : fait tourner le moteur gauche

servo1(angle) : positionne le servo1 ($0 < \text{angle} < 180$)

servo2(angle) : positionne le servo2

ledRGB(numéro, R,V,B) : Allume la(es) led(s) RGB

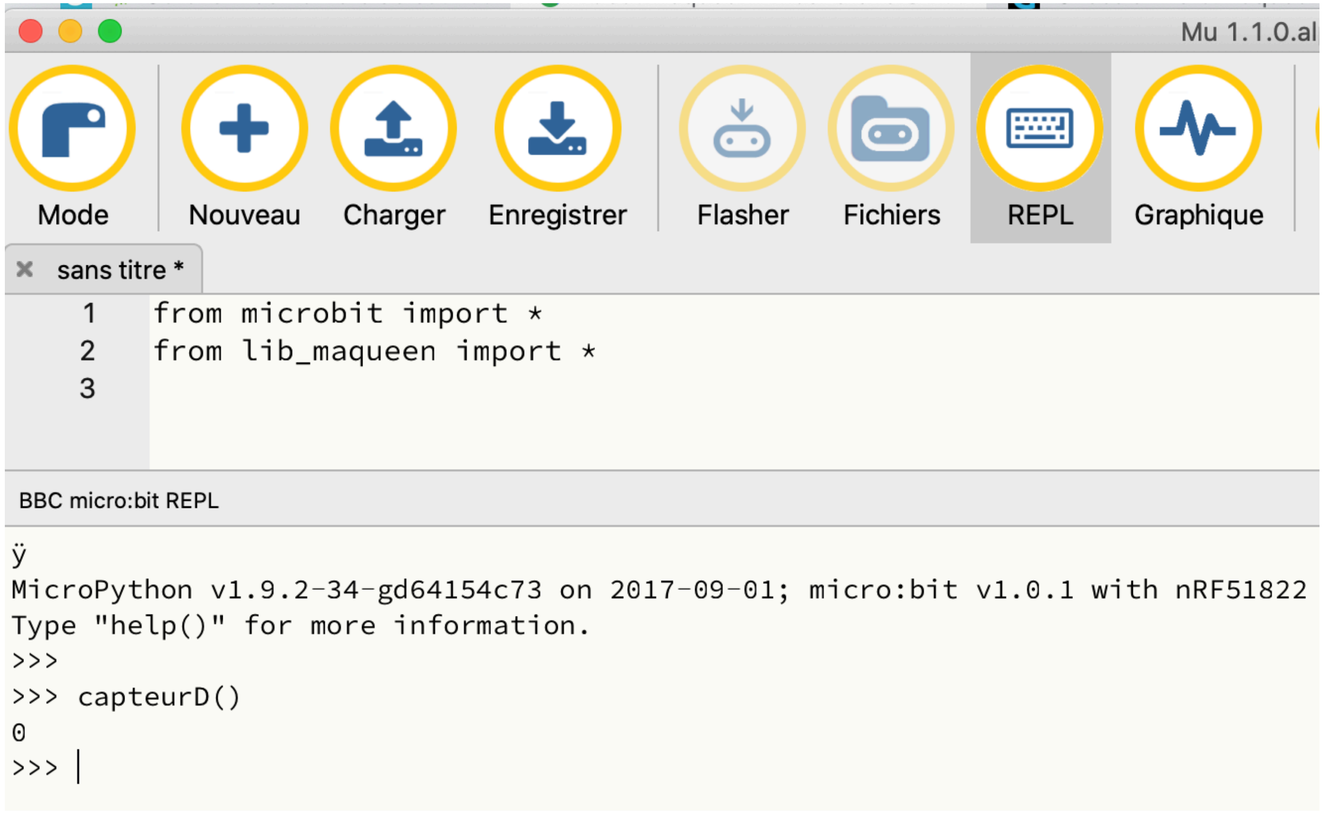
eteintledRGB(numéro, R,V,B) : Éteint la(es) led(s) RGB

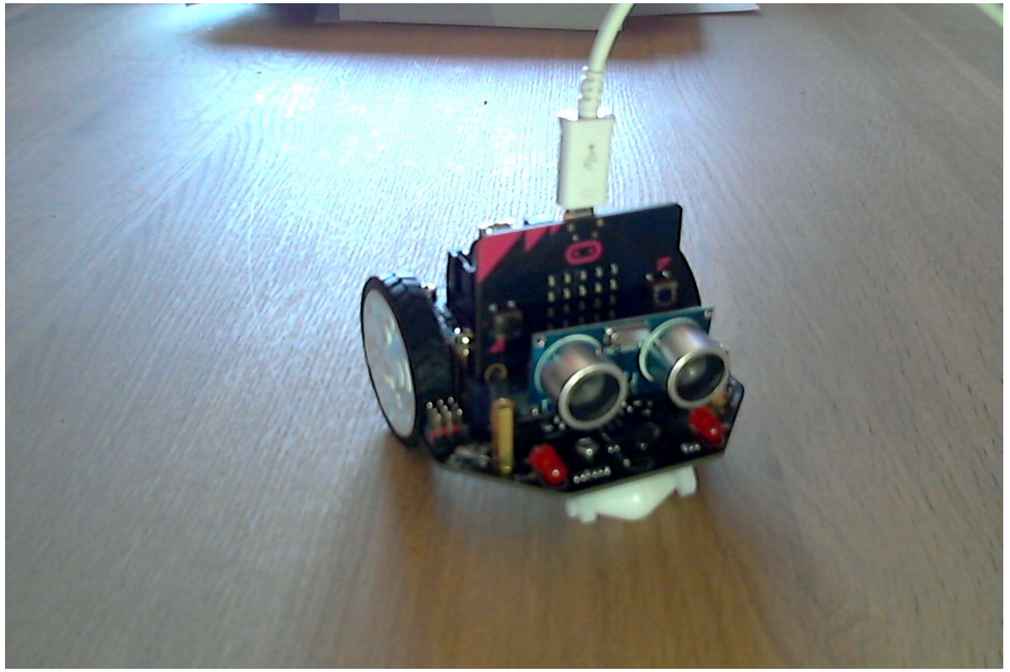
sirene() : Joue une mélodie sirène de police

son(fréquence, durée) : Joue un son

power_is_on(): Renvoie 1 si le robot est sous tension

Le robot Maqueen





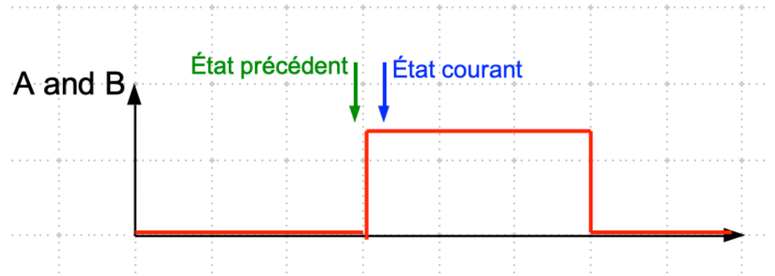
Démo en mode interactif

Le robot Maqueen

Programme sur la carte radio-commande

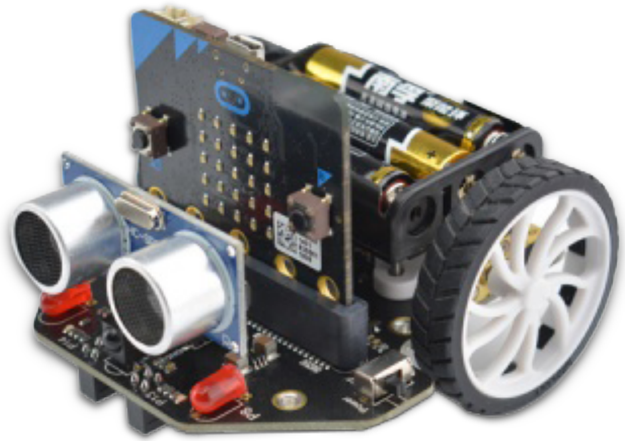
```
from microbit import *
import radio

radio.config(channel=22)
radio.on()

# appuyer sur les 2 boutons pour demarrer
while not (button_a.is_pressed() and button_b.is_pressed()):
    pass

etat_precedent = False
marche = False

while True:
    # on gere la vitesse du robot en inclinant la carte microbit
    consigne_vitesse = accelerometer.get_y()

    # limitation inclinaison  a ± 765 -> vitesse = ± 255
    vitesse_base = int(max(-765, min(consigne_vitesse, 765)) / 3)
    # on adapte le sens d'inclinaison au sens de marche
    vG = vD = - vitesse_base

    # gestion des boutons
    # Commutation Marche/arret en appuyant sur les 2 boutons
    # detection front montant A.B
    etat_courant = bool(button_a.is_pressed() and
button_b.is_pressed())
    if (etat_courant is True and etat_courant != etat_precedent):
        marche = not marche
    etat_precedent = etat_courant
```

```
# modifications des vitesses en fonction des boutons
if not marche :
    vG = vD = 0
# bouton A -> tourne a gauche
elif (button_a.is_pressed() and not button_b.is_pressed()):
    vG = int(vG / 10)
# bouton B -> tourne a droite
elif (button_b.is_pressed() and not button_a.is_pressed()):
    vD = int(vD / 10)

# on envoie les consignes
datas = "{},{}".format(vG, vD)
radio.send(datas)
sleep(100)
```



Le robot Maqueen

Programme sur la carte du robot



La carte reçoit par radio, les consignes de vitesse pour les 2 moteurs

```
from microbit import *
from lib_maqueen import *
import radio

# test si le bus i2c est actif
while not power_is_on():
    pass

# parametrages
radio.config(channel=22)
radio.on()

vG = vD = 0
ordre = ordre_courant =  "{},{}".format(vG, vD)

while True :
    # attente ordre radio
    ordre = radio.receive()

    if ordre != None and ordre != ordre_courant :
        # traitement
        vG, vD = ordre.split(',')
        vitesseG, vitesseD = int(vG), int(vD)

        # execution
        moteurG(vitesseG)
        moteurD(vitesseD)
        ordre_courant = ordre
```

La carte Enviro:bit

- La carte Enviro:bit permet la mesure de plusieurs données environnementales :
  température, humidité, pression, lumière, couleur et son.

- Pour une utilisation avec mu-Editor, il faut télécharger 3 librairies ici : micropython-envirobit

La carte Enviro:bit

```
import bme280
b = bme280.bme280
```

- `b.temperature()`
- `b.pressure()`
- `b.humidity()`
- `b.altitude()`
- `b.set_qnh(value)`

```
import tcs3472
t = tcs3472.tcs3472()
```

- `t.rgb()`
- `t.scaled()`
- `t.light()`
- `t.set_leds()`

```
import sound
s = sound.sound()
```

- `sound.read()`
- `sound.wait_for_double_clap()`
- `sound.wait_for_clap()`

Méthodes

La carte Enviro:bit

Mesure de la couleur d'un éclairage

Éclairage Led RGB	Envirobit	Couleur
0, 0, 0	156 , 123 ,118	
255, 0, 0	225 , 23 ,30	rouge
0, 255, 0	30 , 163 ,67	vert
0, 0, 255	8 , 68 ,192	bleu
255, 255, 0	99 , 111 ,51	jaune
255, 0, 255	83 , 50 ,135	magenta
0, 255, 255	14 , 106 ,142	turquoise

- Pour les couleurs primaires, on observe une dominance de la composante principale
- Pour les couleurs secondaires, difficile d'obtenir une correspondance

Mesure de la couleur d'un objet

Leds allumées : éclairage ambiant → 138 , 113 ,112

Objet	mesure1	mesure2
Balle blanche	67 , 93 ,103	60 , 92 ,104
Balle orange	122 , 86 ,68	116 , 89 ,63
Balle marron	80 , 93 ,89	85 , 96 ,90
Post it vert	101 , 120 ,93	115 , 120 ,99
Post-it jaune	112 , 106 ,69	114 , 106 ,72

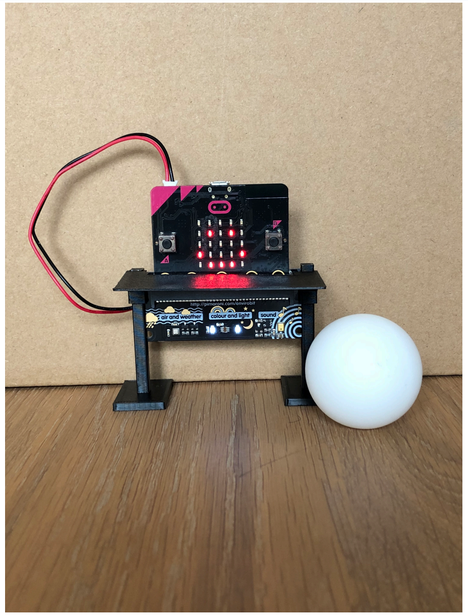
Bonne "répétabilité" des mesures (à distance constante)
On observe une variation modérée des composantes
Difficile de faire le lien entre la couleur et les mesures.
Utilisation : Il est possible de différencier des objets de couleurs différentes (apprentissage), mais impossible d'identifier la couleur d'un objet à partir des mesures

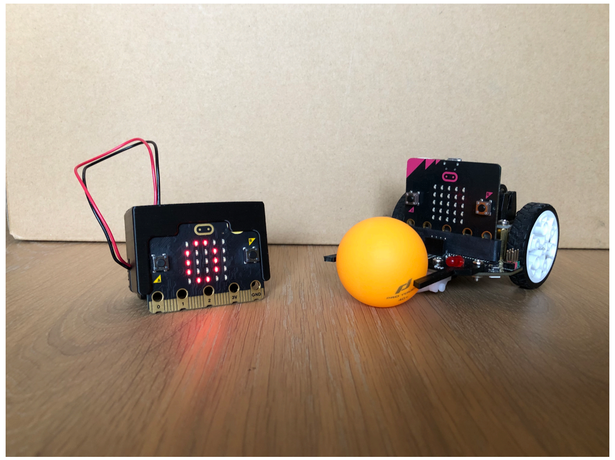
Quelques tests

Le robot footballeur

- 1 but qui reconnait et affiche la couleur de la balle qui entre dans le but

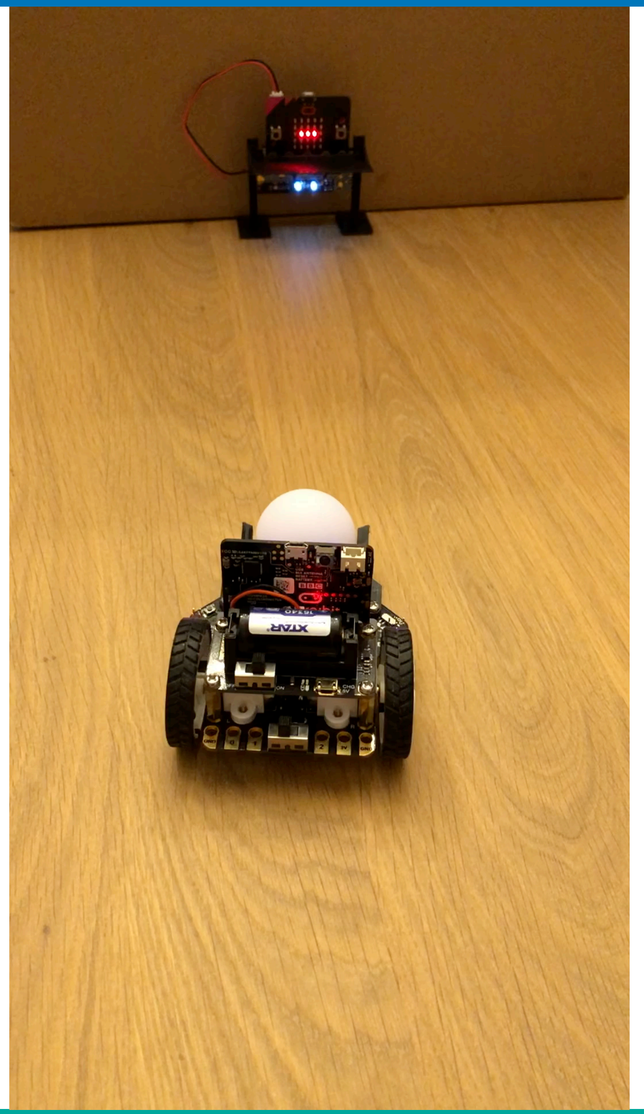
- 2 joueurs dotés chacun de :
  - 1 robot radio commandé
  - 1 radio commande qui affiche le score
  - 1 balle de couleur spécifique





Le robot footballeur

- Le joueur doit amener sa balle dans le but avec le robot
- Si la balle est reconnue, l'initiale de sa couleur est affichée sur le but et le score sur la radio concernée est incrémenté.
- l'objectif est de marquer un maximum de points dans un temps imparti
- Le bouton A du but, remet les scores à zéro pour une nouvelle partie



Démo

Idées de projet

Quelques suggestions

- ✓ Jeu de chasse au trésor connectée
- ✓ Réalisation de tutoriels vidéos pour utiliser la carte :le robot : https://www.youtube.com/watch?v=79Dmald6o_I
- ✓ Programmation d'un véhicule autonome
- ✓ Jardin et plantes connectés
- ✓ Dispositifs numériques connectés pour un escape game
- ✓ Parcours d'épreuves pour Maqueen
- ✓ Mini-golf : https://sites.google.com/view/imake-it/aktivitäten/minigolf-challenge (en allemand)
- ✓ Crazy basketball game : https://twitter.com/PinkyPepper_/status/1306606607829266434?s=20
- ✓ Yes We Play Soccer : https://www.youtube.com/watch?v=NSP1ke0COb8
- ✓ El encuentro : https://www.youtube.com/watch?v=hL464MVQdPQ
- ✓ E-videur : https://www.youtube.com/watch?v=n5MlW0zg97I

Quelques conseils

- Trouver une place pour tous
- Permettre des approches différentes
- Laisser une place à la créativité

Thématiques des projets

Prix thématiques 2022

- ✓ Environnement – développement durable
- ✓ Santé
- ✓ Social/ Aide à la personne
- ✓ Jeux/Loisirs
- ✓ Robotique
- ✓ Sécurité
- ✓ Sport
- ✓ Art/design
- ✓ Education tutoriels
- ✓ Expérimentation scientifique/technique