

# Chap 13 – Informatique embarquée

## Objectifs

1. Périphériques d'entrée et de sortie Interface Homme-Machine (IHM) (capteurs et actionneurs systèmes embarqués ou robots)

## II. Programmons (suite)

### B. Programmez en Python (suite)

#### A faire vous-même 10

1. Saisissez le programme suivant :

```
from microbit import *
while True:
    face1 = Image("00000:09090:00000:90009:09990")
    face2 = Image("00000:09090:00000:99999:00000")
    face3 = Image("00000:09090:00000:09990:90009")
    face4 = Image("90009:99099:00000:09990:90009")
    face5 = Image("00000:00000:00000:00000:00000")
    all_faces = [face1, face2, face3, face4, face5, ]
    display.show(all_faces, delay=200)
```

2. Que fait ce programme ?

.....

#### A faire vous-même 11

1. Saisissez le programme suivant :

```
from microbit import *
while True:
    if button_a.is_pressed():
        display.show(Image.ARROW_W)
    elif button_b.is_pressed():
        display.show(Image.ARROW_E)
    else:
        display.show(Image.HEART)
        display.clear()
```

2. Que fait ce programme ?

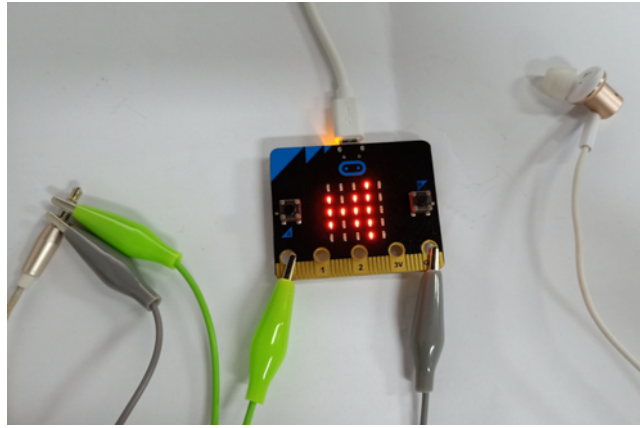
.....

#### A faire vous-même 12

1. Saisissez le programme suivant :

```
from microbit import *
import music
boat = Image("00090:90990:99990:90990:00090")
display.show(boat)
music.play(music.BIRTHDAY)
```

2. Réalisez le branchement suivant avec des écouteurs :



### 3. Que fait ce programme ?

.....

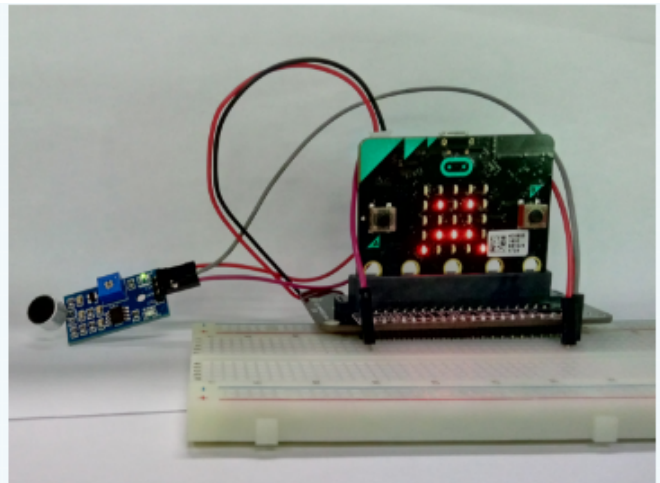
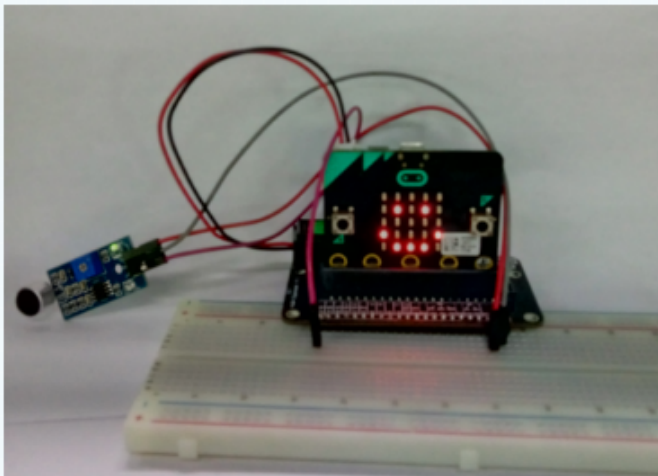
Voici une liste des mélodies disponibles :

- music.DADADADUM
- music.ENTERTAINER
- music.PRELUDE
- music.ODE
- music.NYAN
- music.RINGTONE
- music.FUNK
- music.BLUES
- music.BIRTHDAY
- music.WEDDING
- music.FUNERAL
- music.PUNCHLINE
- music.PYTHON
- music.BADDY
- music.CHASE
- music.BA\_DING
- music.WAWAWAWAA
- music.JUMP\_UP
- music.JUMP\_DOWN
- music.POWER\_UP
- music.POWER\_DOWN

### A faire vous-même 13



Objectif :



Hello, I'm micro:bit. I'm a coward. I'm always smiling like the first picture. If you talk to me loudly, I'll be scared into second pictures.

Do you want to make a coward with micro:bit? Follow me.

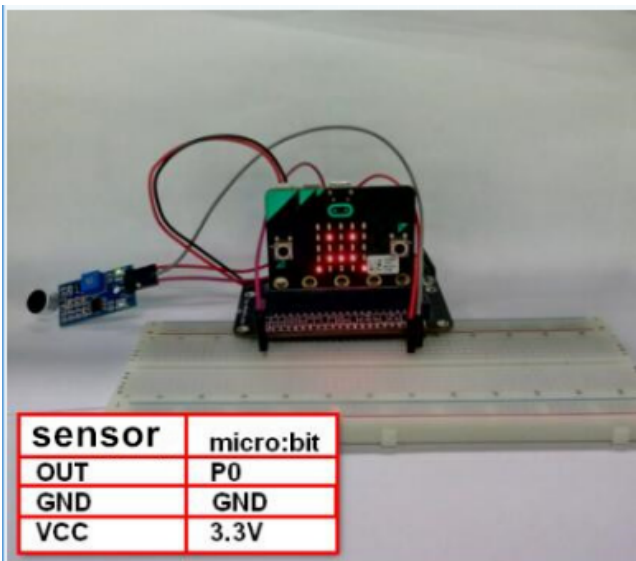
Préparation :

## Hardware:

- 1 X Micro: bit Board
- 1 X Micro:bit breakout
- 1 X 830 holes breadboard
- 1 X Micro USB Cable
- 3 X Female to male cable
- 1 X Sound sensor

Then the micro:bit is connected to the computer through USB, and the computer will pop up a U disk and click the URL in the U disk to enter the programming interface. Input this URL [https://github.com/lzty634158/yahboom\\_mbit\\_en](https://github.com/lzty634158/yahboom_mbit_en) to get the package.

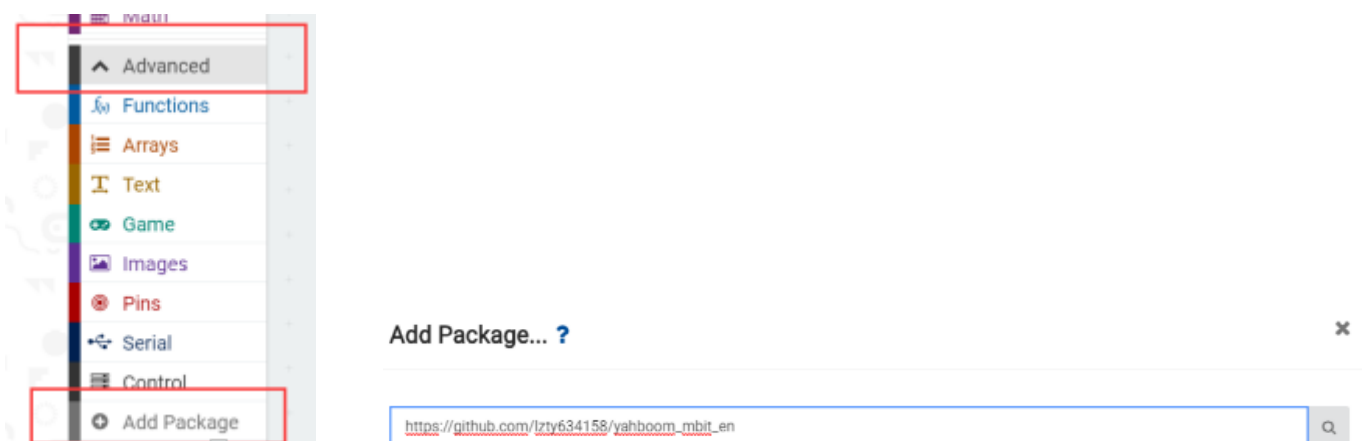
## Connection :



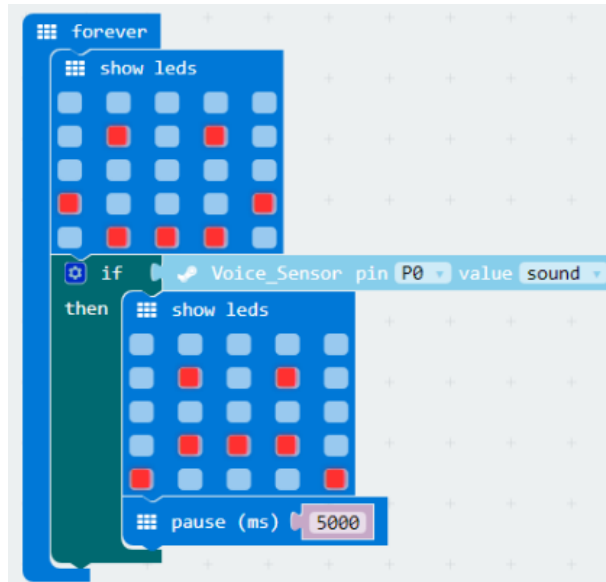
You can connect the cable according to the connection table above. A sound sensor needs to be debugged before use, and a screwdriver is needed to adjust the sensitivity of the sound sensor (a ring of small circles around). The lattice has always been a smiling face or a dot matrix and often turns into a crying face because the sensitivity is not well adjusted.

## Ajout du module spécifique :

Enter the URL in the input column : [https://github.com/lzty634158/yahboom\\_mbit\\_en](https://github.com/lzty634158/yahboom_mbit_en)

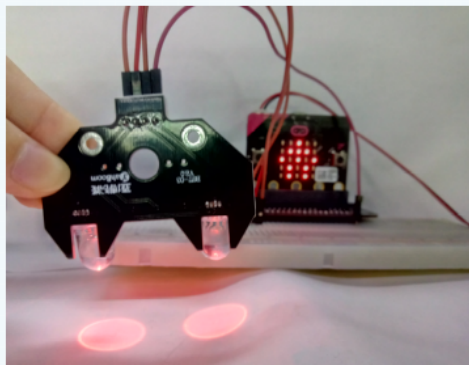


Programme :



A faire vous-même 14

Objectif :



Do you know what the traffic rules on the road are, boys and girls? Red is stop, green is go. When children cross the street, they must have adults around. And the red light is to let the car pass through, the green light to cross the road.

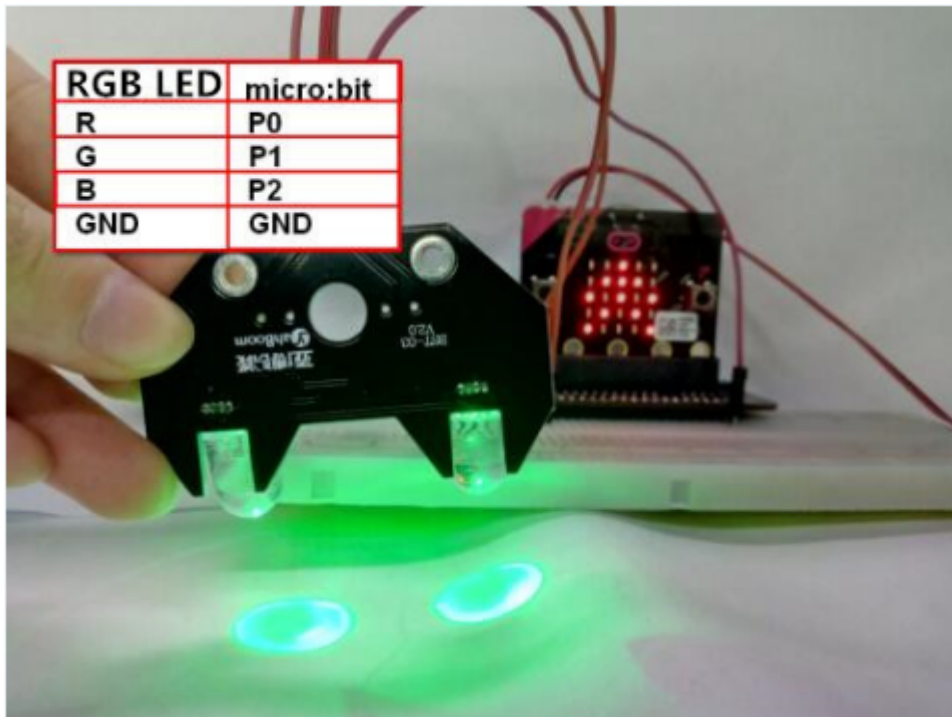
Préparation :

Hardware:

- 1 X Micro: bit Board
- 1 X Micro USB Cable
- 1 X Micro:bit breakout
- 4 X Male to male cable
- 1 X 830 holes breadboard
- 1 X RGB LED

Then the micro:bit is connected to the computer through USB, and the computer will pop up a U disk and click the URL in the U disk to enter the programming interface. Input this URL [https://github.com/lzty634158/yahboom\\_mbit\\_en](https://github.com/lzty634158/yahboom_mbit_en) to get the package.

Connection :



Programme :

```
forever loop
  RGB
  pin1 P0
  pin2 P1
  pin3 P2
  value red
  show leds
  pause (ms) 10000
  RGB
  pin1 P0
  pin2 P1
  pin3 P2
  value green
  show leds
  pause (ms) 10000
```