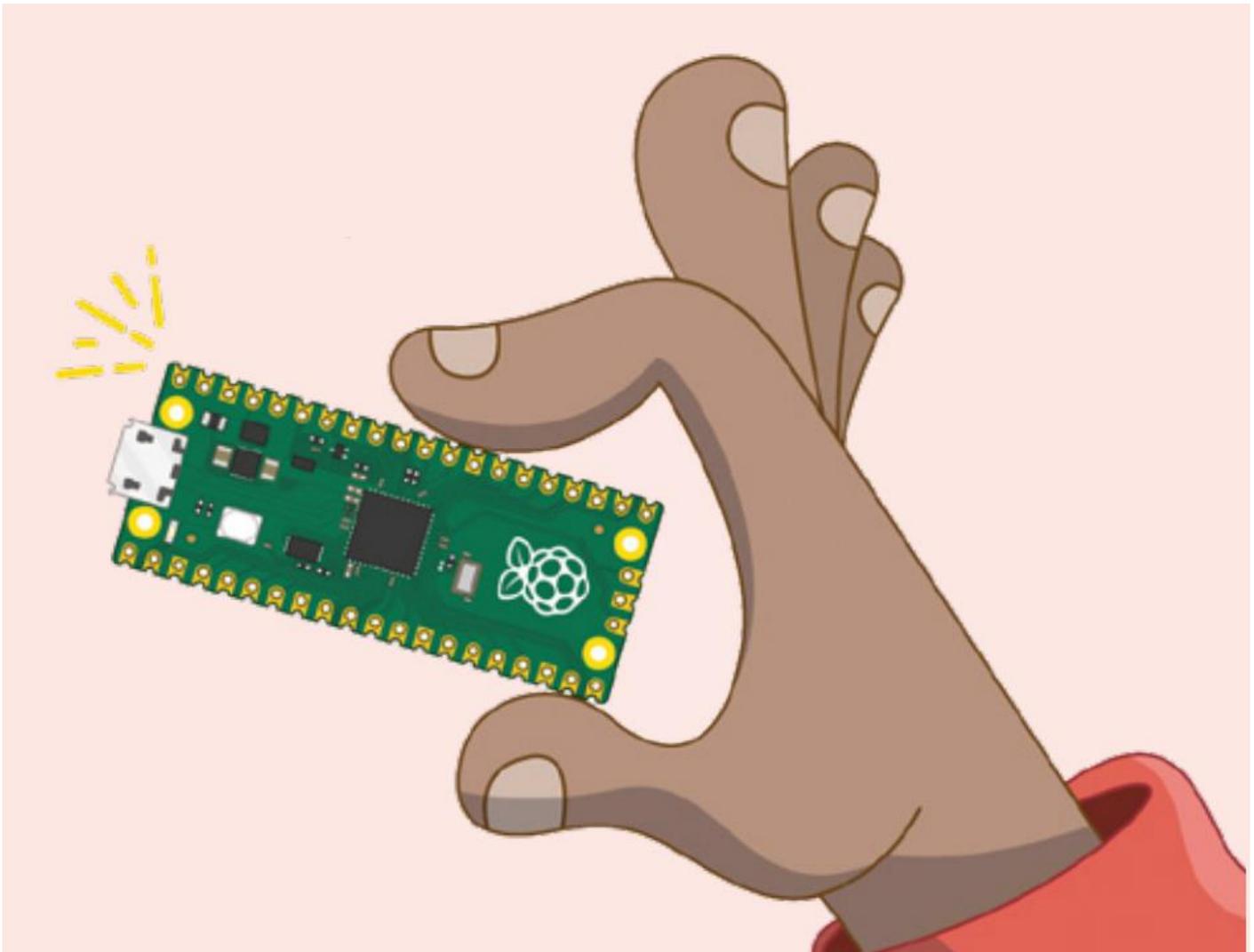




## TP N°1 :

# Raspberry Pi Pico

## Configurations, Leds et BP





## I. CircuitPython

### 1. Présentation de CircuitPython



Comme pour la Raspberry Pi Pico, il est possible de programmer en Python dans une version adaptée aux microcontrôleurs : micropython

Je préfère, dans un premier temps, utiliser une version dérivée de micropython : circuitpython.

Circuitpython a plusieurs avantages :

- Prise en charge par Adafruit
- Communauté nombreuse et réactive
- Prise en charge de nombreux composants et capteurs

### 2. Téléchargement de circuitpython

CircuitPython est régulièrement mise à jour. Pour obtenir la dernière version cliquez sur ce bouton

[https://circuitpython.org/board/raspberry\\_pi\\_pico/](https://circuitpython.org/board/raspberry_pi_pico/)

#### Pico

by Raspberry Pi



#### CircuitPython 7.0.0

This is the latest **stable** release of CircuitPython that will work with the Pico.

**Start here** if you are new to CircuitPython.

[Release Notes for 7.0.0](#)

FRENCH

DOWNLOAD .UF2 NOW

Built-in modules available: bleio, adafruit\_bus\_device, adafruit\_pixelbuf, aesio, alarm, analogio, atexit, audiobusio, audiocore, audiomixer, audiomp3, audiopwmio, binascii, bitbangio, bitmaptool, bitops, board, busio, countio, digitalio, displayio, errno, fontio, framebufferio, getpass, imagecapture, json, keypad, math, microcontroller, msgpack, neopixel\_write, nvm, onewireio, os, paralleldisplay, pulseio, pwmio, qrio, rainbowio, random, re, rgbmatrix, rotaryio, rtc, sdcardio, sharpdisplay, storage, struct, supervisor, synthio, terminalio, time, touchio, traceback, ulab, usb\_cdc, usb\_hid, usb\_midi, vectorio, watchdog

#### CircuitPython 7.1.0-beta.3

1. Choisissez le français

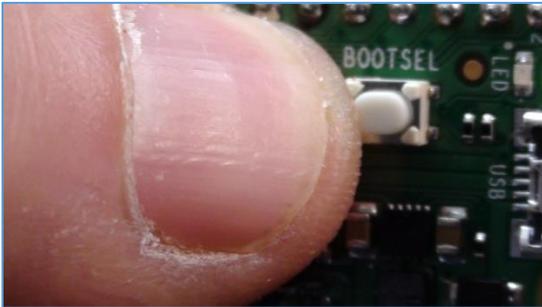
Tâche N°1. 2. Téléchargez le fichier sur votre ordinateur

Vous avez récupéré un fichier au format UF2 (USB Flashing Format développé par Microsoft) qui contient le programme à implémenter dans le microcontrôleur.



### 3. **Branchement de la carte**

Tâche N°2. Appuyez et maintenez le bouton *BOOTSEL*



Tâche N°3. Branchez-le pico sur le PC

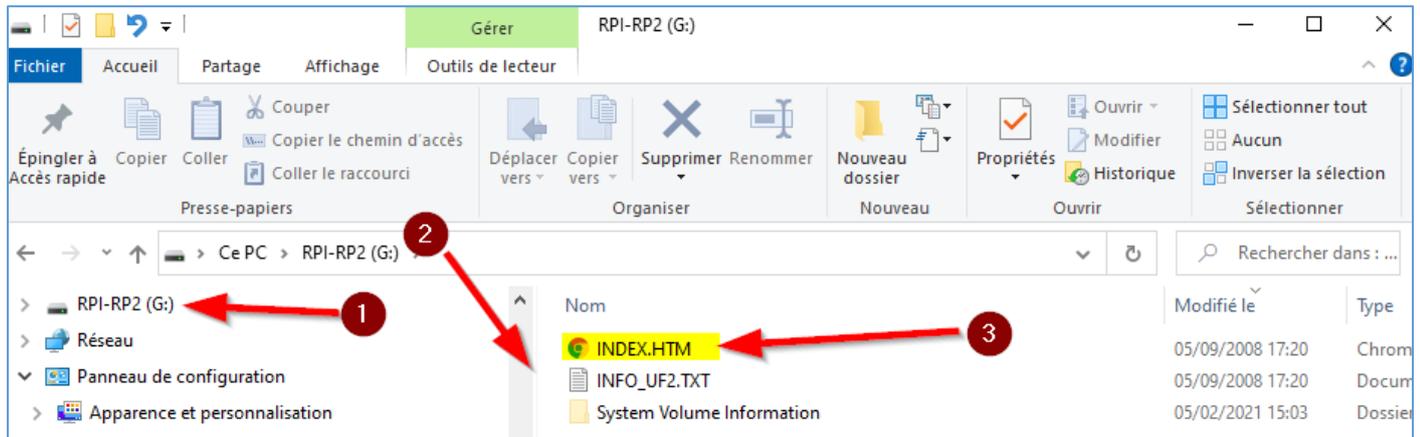


Tâche N°4. Comptez trois secondes puis relâchez le bouton



Et voilà on peut commencer à programmer.

Vous devez voir une sorte de clé usb connecté sur un des ports (à la lettre G dans mon cas). Le nom est RPI-RP2 (1) et à droite (2) mes fichiers s'y trouvent :



#### 4. Installation sur la pico

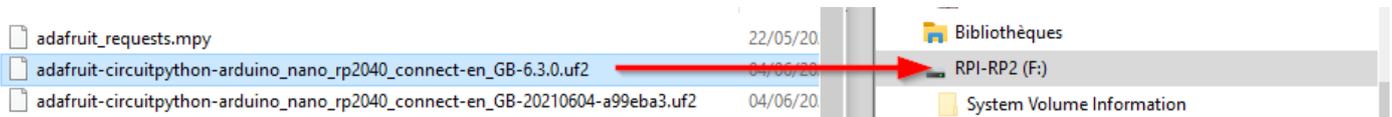
Pour déposer et exécuter le programme contenu dans le fichier précédent, il faut mettre la carte Arduino en mode UF2.

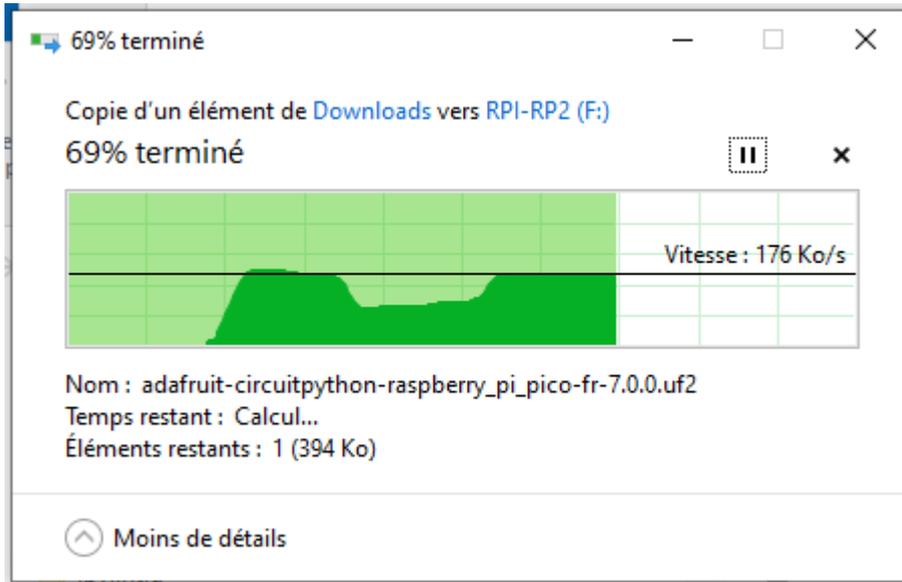
**Tâche N°5.** Pour cela, appuyez rapidement deux fois sur le bouton blanc (Bootselect)

Après une petite musique vous devez voir dans vos répertoires, la transformation de votre carte en lecteur nommé **RPI-RP2** :

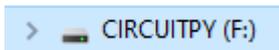
System Volume Information	07/06/2021 10:41
INDEX.HTM	05/09/2021 16:20
INFO_UF2.TXT	05/09/2021 16:20

Déplacez le fichier UF2 téléchargé du site d'Adafruit vers le lecteur RPI-RP2





Et le lecteur se transforme en CIRCUITPY



C'est terminé.

#### Remarques :

- Je préfère l'installer la version anglaise, pour rechercher les erreurs c'est plus facile sur Internet
- Si le lecteur n'apparaît pas en RPI-RP2, essayez plusieurs fois les deux appuis
- Si vous après avoir mis le fichier UF2, le lecteur CIRCUITPY n'apparaît toujours pas, ou si vous rencontrez des problèmes par la suite, essayez de télécharger essayez de télécharger un fichier UF2 qui va effacer et remettre à zéro votre carte. Attention : toutes vos données seront définitivement perdues !!!

Lien : [https://cdn-learn.adafruit.com/assets/assets/000/102/609/original/flash\\_nuke.uf2?1622831270](https://cdn-learn.adafruit.com/assets/assets/000/102/609/original/flash_nuke.uf2?1622831270)

## II. Premier essai : blink encore et toujours

### 1. Le programme complet

```
# -*- coding: utf-8 -*-  
from time import sleep  
import board  
from digitalio import DigitalInOut, Direction  
  
# LED qui se trouve sur la carte  
led = DigitalInOut(board.LED)  
led.direction = Direction.OUTPUT  
  
while True:  
    led.value = True  
    print("led on : allumé")  
    sleep(1)  
  
    led.value = False  
    print("led off :: éteinte")  
    sleep(1)
```



## 2. Explications du code

J'utilise des accents dans les print, donc format utf-8

```
# -*- coding: utf-8 -*-
```

La fonction sleep permet de mettre en pause le programme, elle se trouve dans time

```
from time import sleep
```

J'ai besoin de la bibliothèque board pour récupérer le port de la led se trouvant sur la carte

```
import board
```

DigitalInOut va permettre de créer un objet sur une broche numérique et Direction le sens.

```
from digitalio import DigitalInOut, Direction
```

Création d'une variable, plus précisément d'un objet qui va représenter la led de la carte. Le sens est entrant.

Led est donc une broche d'entrée numérique.

```
# LED qui se trouve sur la carte  
led = DigitalInOut(board.LED)  
led.direction = Direction.OUTPUT
```

Une boucle infinie

```
while True:
```

Pour mettre la broche à l'état haut il faut mettre value à True

```
led.value = True  
print("led on : allumé")
```

Attente d'une seconde :

```
sleep(1)
```

Puis on recommence avec éteint

```
led.value = False  
print("led off :: éteinte")  
sleep(1)
```

## 3. Lancement du programme

Tâche N°6. Exécutez mu-editor

Tâche N°7. Créez un nouveau script avec Nouveau

Mu 1.1.0.beta.2



Tâche N°8. Copier-coller le code précédent



Mu 1.1.0.beta.2 - sans titre •

Mode Nouveau Charger Enregistrer Série Graphique Zoomer Dé-zoomer Thème

```

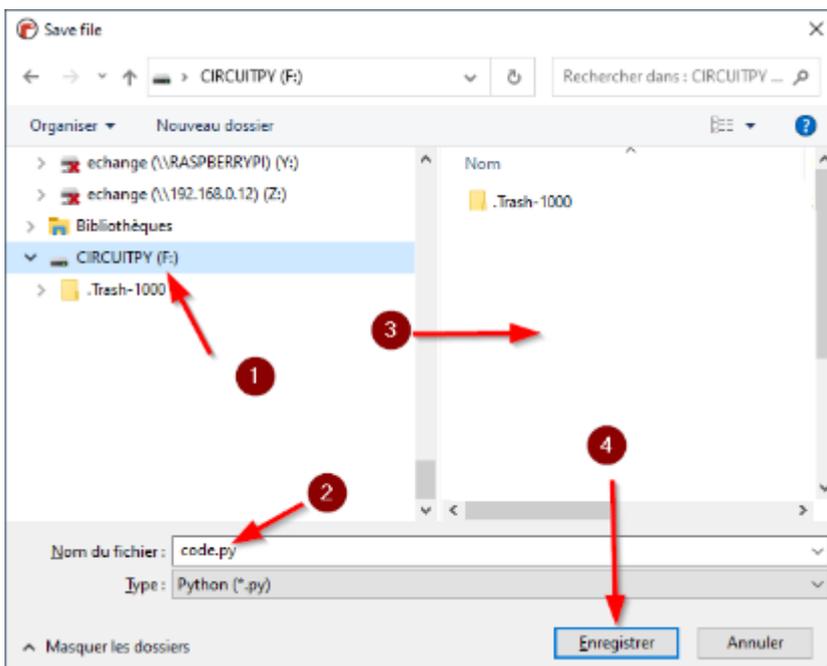
1 # -*- coding: utf-8 -*-
2 from time import sleep
3 import board
4 from digitalio import DigitalInOut, Direction
5
6 # LED qui se trouve sur la carte
7 led = DigitalInOut(board.LED)
8 led.direction = Direction.OUTPUT
9
10 while True:
11     led.value = True
12     print("led on : allumé")
13     sleep(1)
14
15     led.value = False
16     print("led off :: éteinte")
17     sleep(1)

```

Tâche N°9. Cliquez sur CIRCUITPY

Tâche N°10. Modifiez le nom en **code.py**

Tâche N°11. Enregistrez-le sur le lecteur/Répertoire







## 2. Afficher les broches

```
import board
dir(board)
```

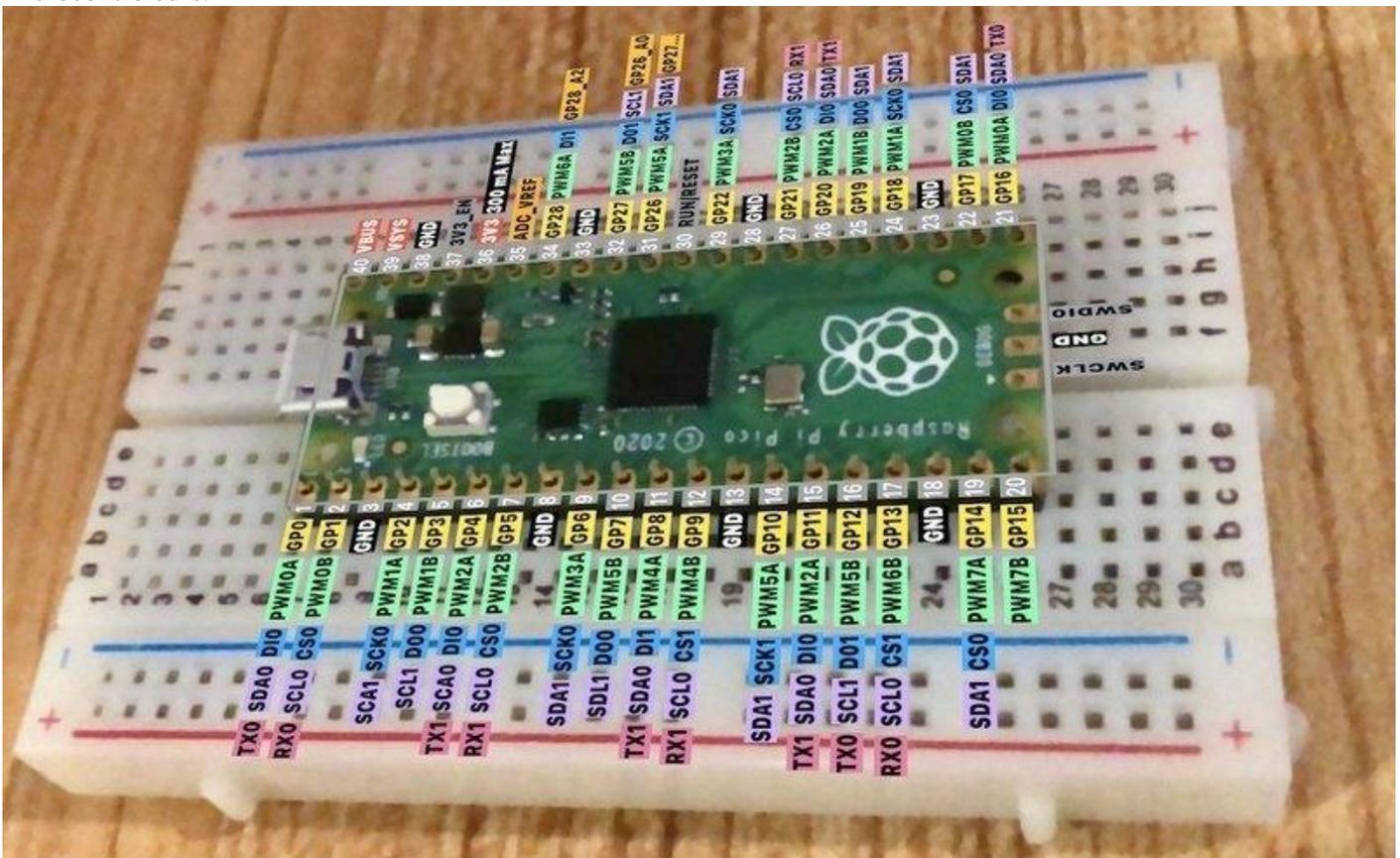
Tâche N°13.

Dans le REPL, testez

. Que se passe-t-il ?

# IV. Utilisation de l'application Réalité Augmentée

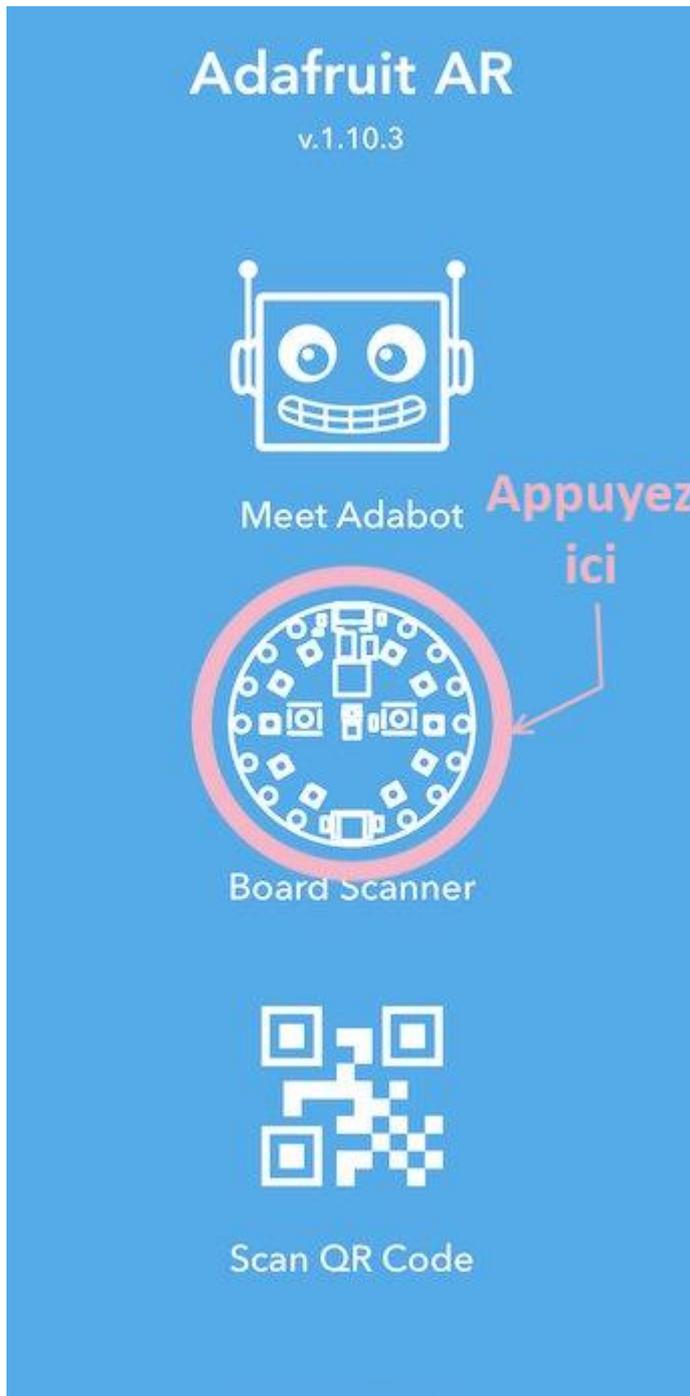
Uniquement sur iOS : une application vous permet de visualiser grâce à votre téléphone Apple les broches de votre microcontrôleurs.



Pour cela :

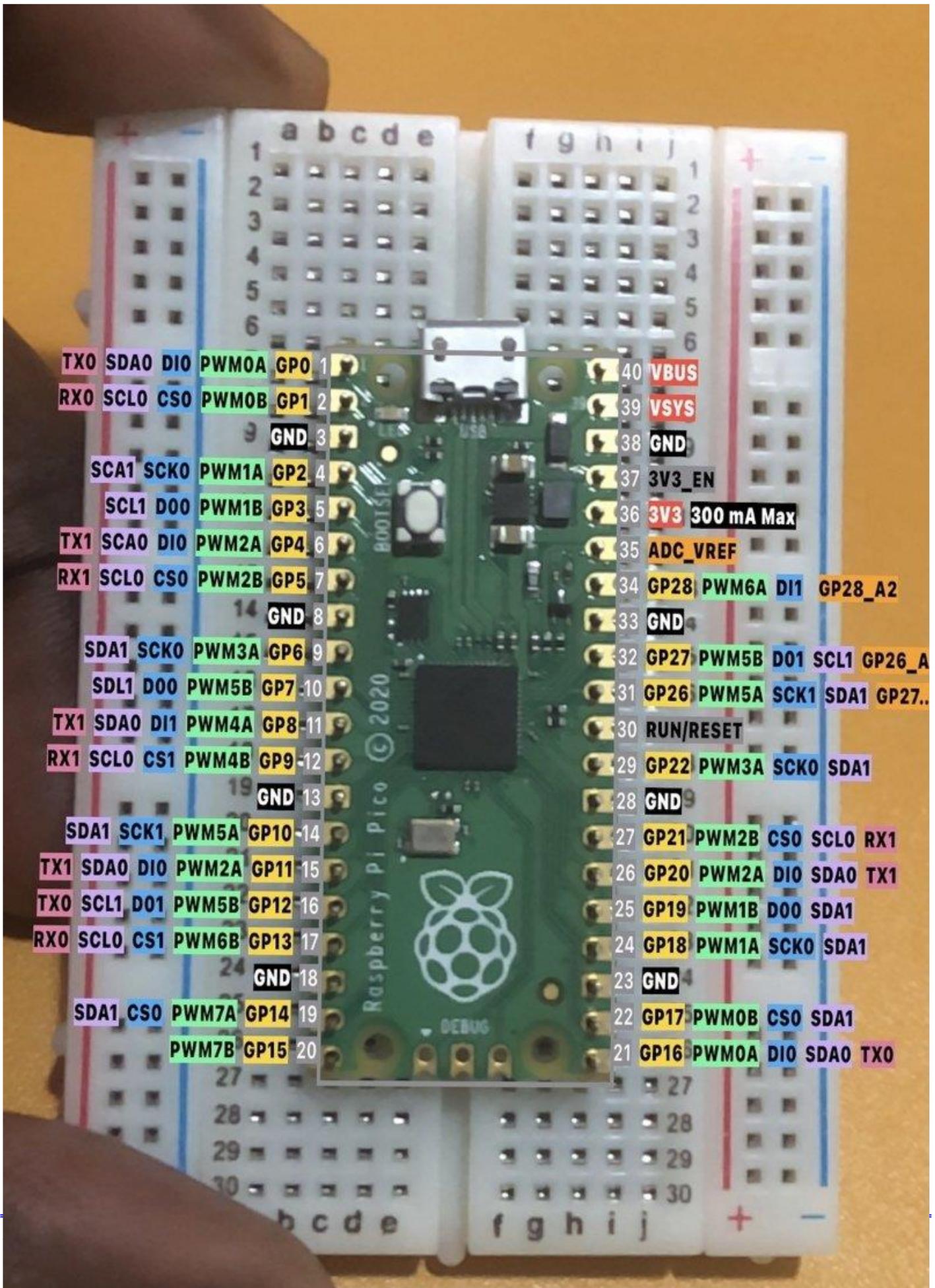
Tâche N°14. Téléchargez l'application [Adafruit AR sur the AppStore](https://adafru.it/Ptb) (<https://adafru.it/Ptb>)

Tâche N°15. Lorsque vous l'avez lancée, appuyez sur le bouton *Board Scanner* pour détecter votre carte



Tâche N°16. Scannez le dessus de la carte Raspberry Pi Pico assez proche de l'objectif de votre téléphone.

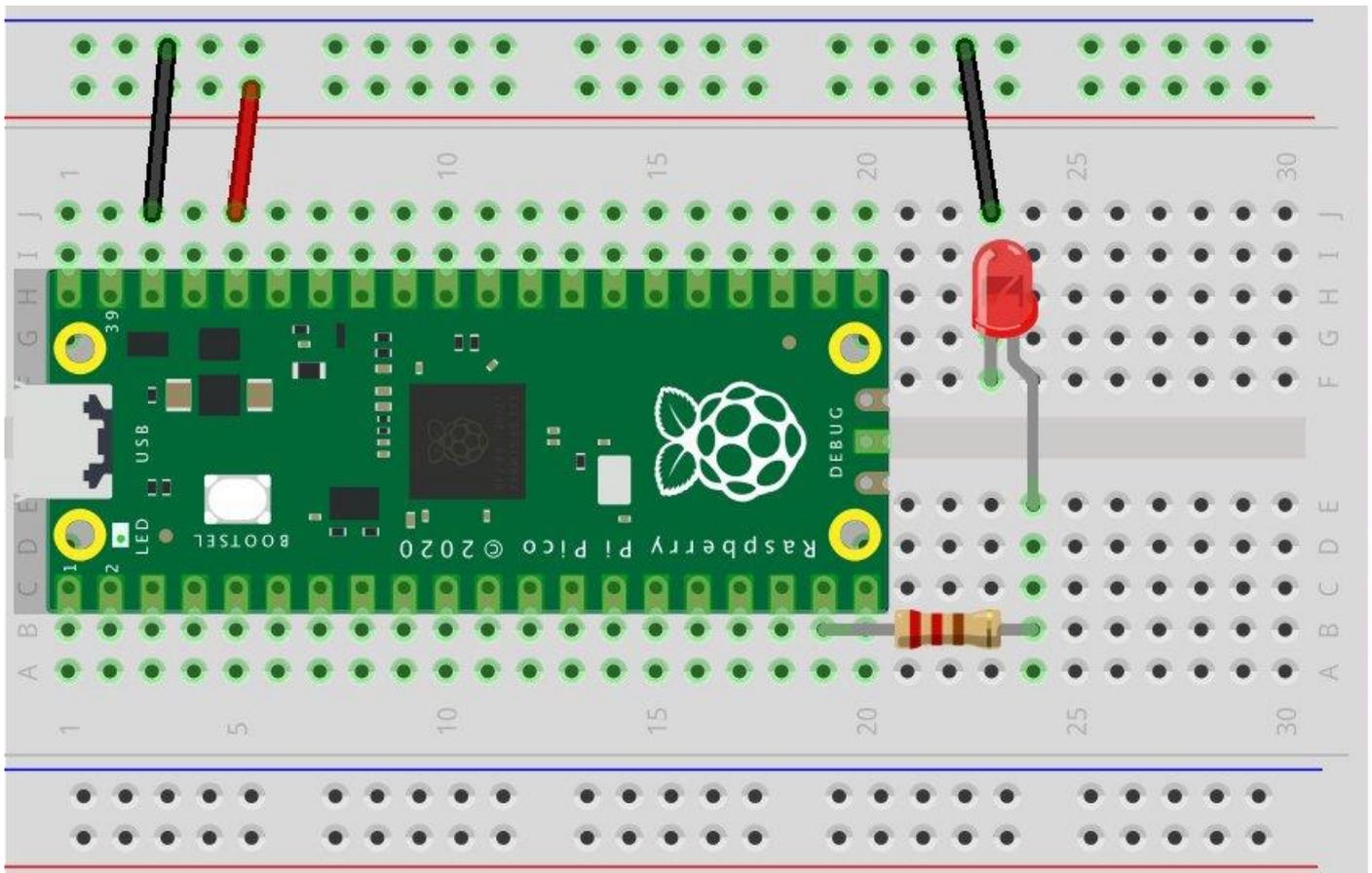
Vous devriez voir virtuellement les broches de votre carte ainsi :





## V. Led externe et bouton

### 1. Led



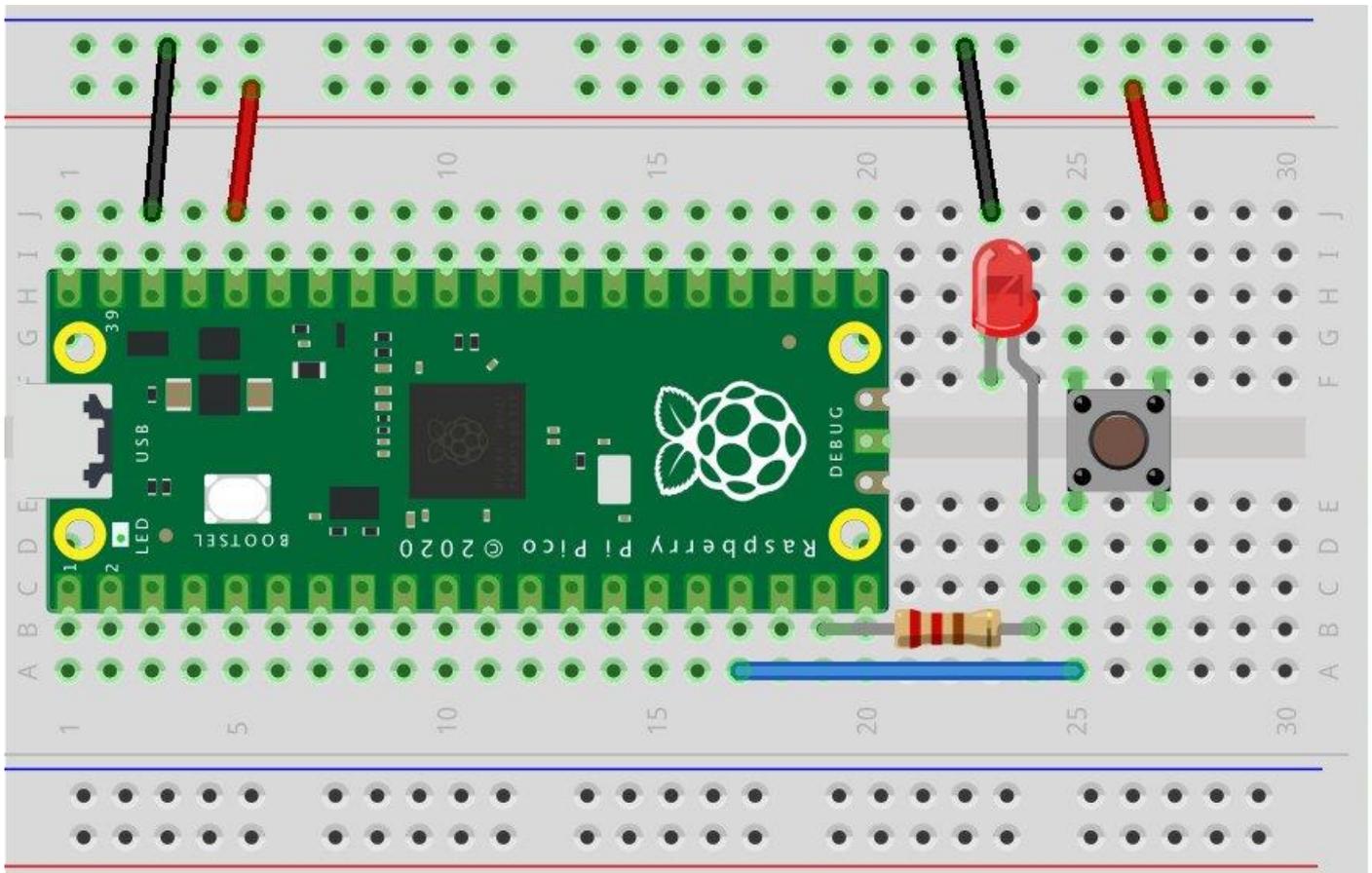
fritzing

Tâche N°17. Effectuez ce montage, la led étant sur le GP14

Tâche N°18. Reprenez le programme de la led interne puis modifiez-le pour faire clignoter la led port 14



## 2. BP



fritzing

Tâche N°19. Testez le code ci-dessous

```
import time
import board
import digitalio

button = digitalio.DigitalInOut(board.GP13)
button.switch_to_input(pull=digitalio.Pull.DOWN)

while True:
    print(button.value)
    time.sleep(0.5)
```

Tâche N°20. Expliquez ces lignes

```
button = digitalio.DigitalInOut(board.GP13)
button.switch_to_input(pull=digitalio.Pull.DOWN)
```

Tâche N°21. Modifiez le code pour afficher « bouton appuyé » ou « bouton non appuyé » avec un if



### 3. Led et BP

Tâche N°22. Ecrivez un code qui lorsque le BP est appuyé, allume la led

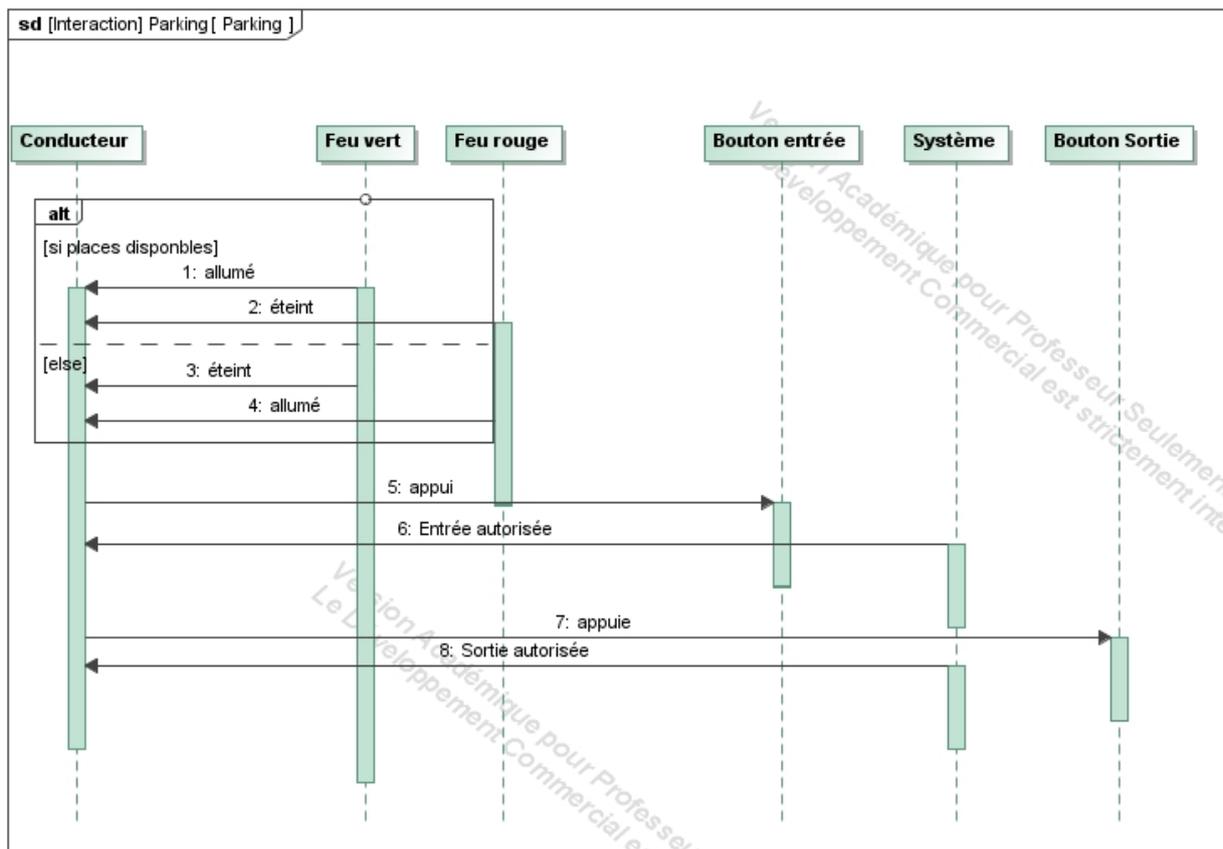
## VI. Exercice : Places de parking

### 1. Cahier des charges



Le gérant d'un parking voudrait mettre en place un système permettant de prévenir les usagers lorsque le parking est complet : le feu vert indique des places libres et le feu rouge est allumé dans le cas contraire.

Un conducteur, désirant utiliser ce parking, arrive à l'entrée, les feux lui indiquent si c'est complet ou non. Ensuite pour entrer, il doit appuyer sur un bouton. De même pour sortir, il doit appuyer sur le bouton « sortie »



### 2. A nous de jouer

Tâche N°23. Implémentez à l'aide de votre pico, ce système de parking