

Notificateur connecté

Auteur : Fabien · **Publié le** 16/01/2023 · 4 vues · 3 téléchargements PDF

[FabLab Maillol](#)

Ce simple interrupteur vous permettra de notifier les autres utilisateurs que votre FabLab est ouvert ou fermé.

Matériel :

- * Un ESP8266 (NodeMCU ou Wemos Mini D1)
- * Un interrupteur (ou un poussoir mais le code est à adapter)
- * Un anneau de Leds de type NéoPixel

Étapes du projet

ÉTAPE 1

Préparer l'ESP à Micropython

Pour cela je vous propose de commencer par consulter la page :

<https://fablab66.fr/#!/projects/python-sur-des-micro-controleurs-esp8266>

Vous flasherez alors votre ESP avec la dernière version de micropython avec μ pycraft.

Vous aurez besoin de la librairie urequests. (présente dans Ntfy_python.zip)

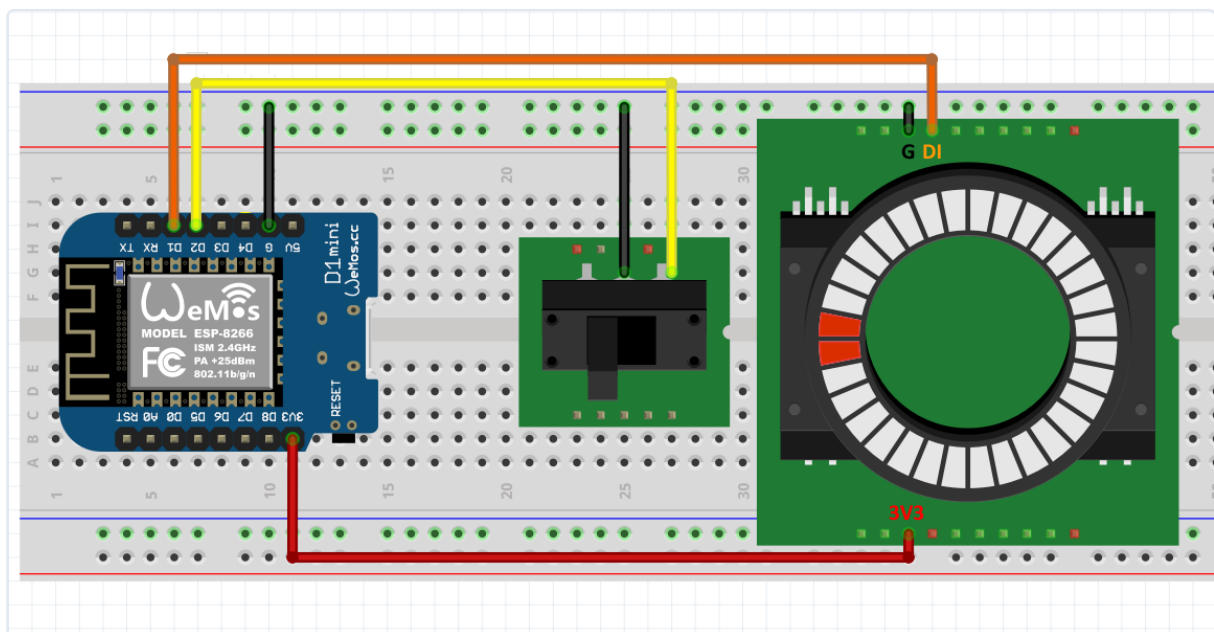
ÉTAPE 2

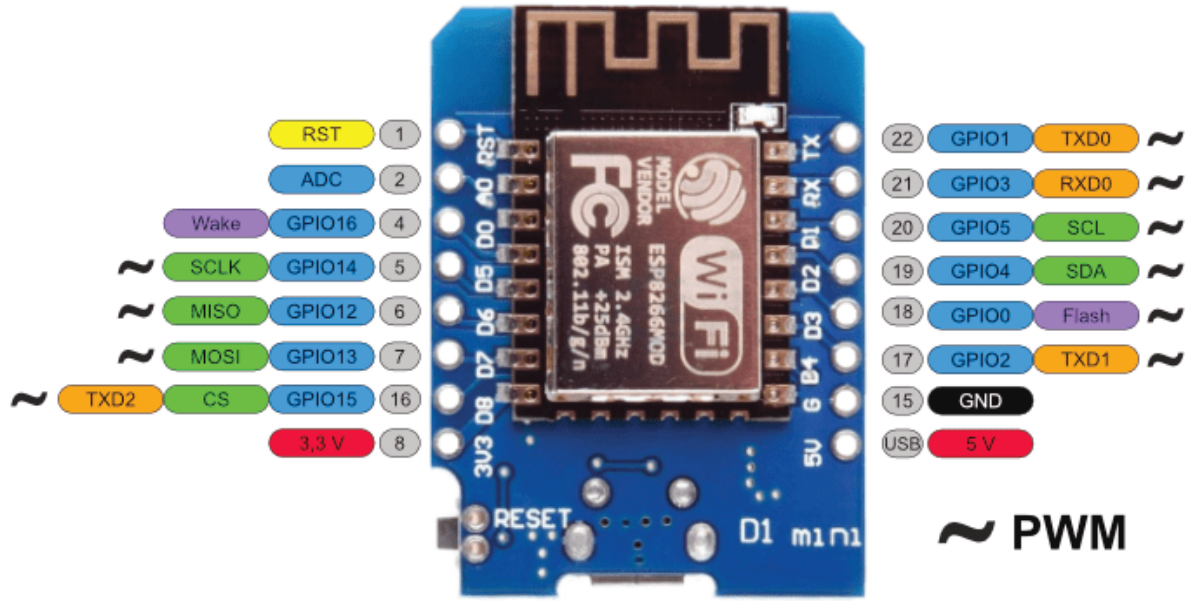
Schéma de câblage

Attention, les LEDs adressables sont sensées fonctionner en 5V mais le signal de commande des ESP est en 3,3V donc si vous connectez la sortie 5V à VCC des LEDs alors celles-ci ne fonctionneront pas.

La Pin DO (ou DOUT) des Leds Néopixel ne nous intéressera pas ici.

MiniD1	3V3	VCC	LEDs
MiniD1	G	G	LEDs
MiniD1	D2 (GPIO4)	D In	LEDs
MiniD1	G	G	Interrupteur
MiniD1	D1 (GPIO5)	1	Interrupteur





ÉTAPE 3

Code Python

Situé dans le fichier Ntfy_python.zip, il contient la librairie urequest qui permet de faire les requêtes POST vers le site ntfy.sh

Le code [main.py](#) qui est lancé automatiquement après le fichier [boot.py](#) sur l'ESP.

```
import network, machine, time
import urequests as requests
import neopixel

url='http://ntfy.sh/mon-fil-de-notification'

#Nbre de led Neopixels
nbr_leds=16
#Broche de l'interrupteur
pin = machine.Pin(5, machine.Pin.IN, machine.Pin.PULL_UP)
#OnBoard led pour vérifier si l'esp se connecte
led = machine.Pin(2, machine.Pin.OUT)
#NeoPixel:
ring = neopixel.NeoPixel(machine.Pin(4), nbr_leds)

sta_if=network.WLAN(network.STA_IF)
if not sta_if.isconnected():
    sta_if.active(True)
    sta_if.connect('Reseau Wifi','MotDePasseDuReseau')
    while not sta_if.isconnected():
        pass
print(sta_if.ifconfig())

couleurs=[(255, 102, 0), # Orange
(255, 0, 102), # Rose bonbon
(153, 51, 255), # Violet
(0, 0, 255), # bleu pastel
(63, 255, 63), # vert pastel
(255,0,0)]

#Petite animation de lancement
for c in couleurs:
    for k in range(nbr_leds):
        ring[k] = c
        ring.write()
        time.sleep(0.1)
#Etat de l'interrupteur
```

```

state=0
#On fait une boucle infinie
while True:
    #Si on observe un changement d'état
    if pin.value() != state:
        if state==0:
            post_data="[] Le FabLab vient d'ouvrir".encode('utf-8')
            state=1
            couleur=(0,255,0) #Vert
        else:
            post_data='[] Le FabLab vient de fermer'.encode('utf-8')
            state=0
            couleur=(255,0,0) #Rouge
        res = requests.post(url, data = post_data) #On notifie via le site internet
        #On passe toutes les leds a la couleur souhaitee l'une apres l'autre
        for i in range(2):
            for k in range(nbr_leds):
                ring[k] = couleur
                ring.write()
                time.sleep(0.1)
            time.sleep(1)

```

```

Badger_Ntly.py
1 import network, machine, time
2 import requests as requests
3 import neopixel
4
5 url='http://ntfy.sh/mon-fil-de-notification'
6
7 #On fait une boucle infinie
8 nbr_leds=16
9
10 pin = machine.Pin(5, machine.Pin.IN, machine.Pin.PULL_UP)
11
12 led = machine.Pin(2, machine.Pin.OUT)
13
14 ring = neopixel.NeoPixel(machine.Pin(4), nbr_leds)
15
16 sta_if=network.MLAN(network.STA_IF)
17 if not sta_if.isconnected():
18     sta_if.active(True)
19     sta_if.connect('Reseau Wifi', 'MotDePasseDuReseau')
20 while not sta_if.isconnected():
21     pass
22 print(sta_if.ifconfig())
23
24 couleurs=[(255, 102, 0), 1 change
25 (255, 0, 102), 1 change
26 (153, 51, 255), 1 change
27 (0, 0, 255), 1 change
28 (0, 255, 0), 1 change
29 (255, 0, 0)]
30
31 #On passe toutes les leds a la couleur souhaitee
32 for c in couleurs:
33     for k in range(nbr_leds):
34         ring[k] = c
35         ring.write()
36         time.sleep(0.1)
37     time.sleep(1)

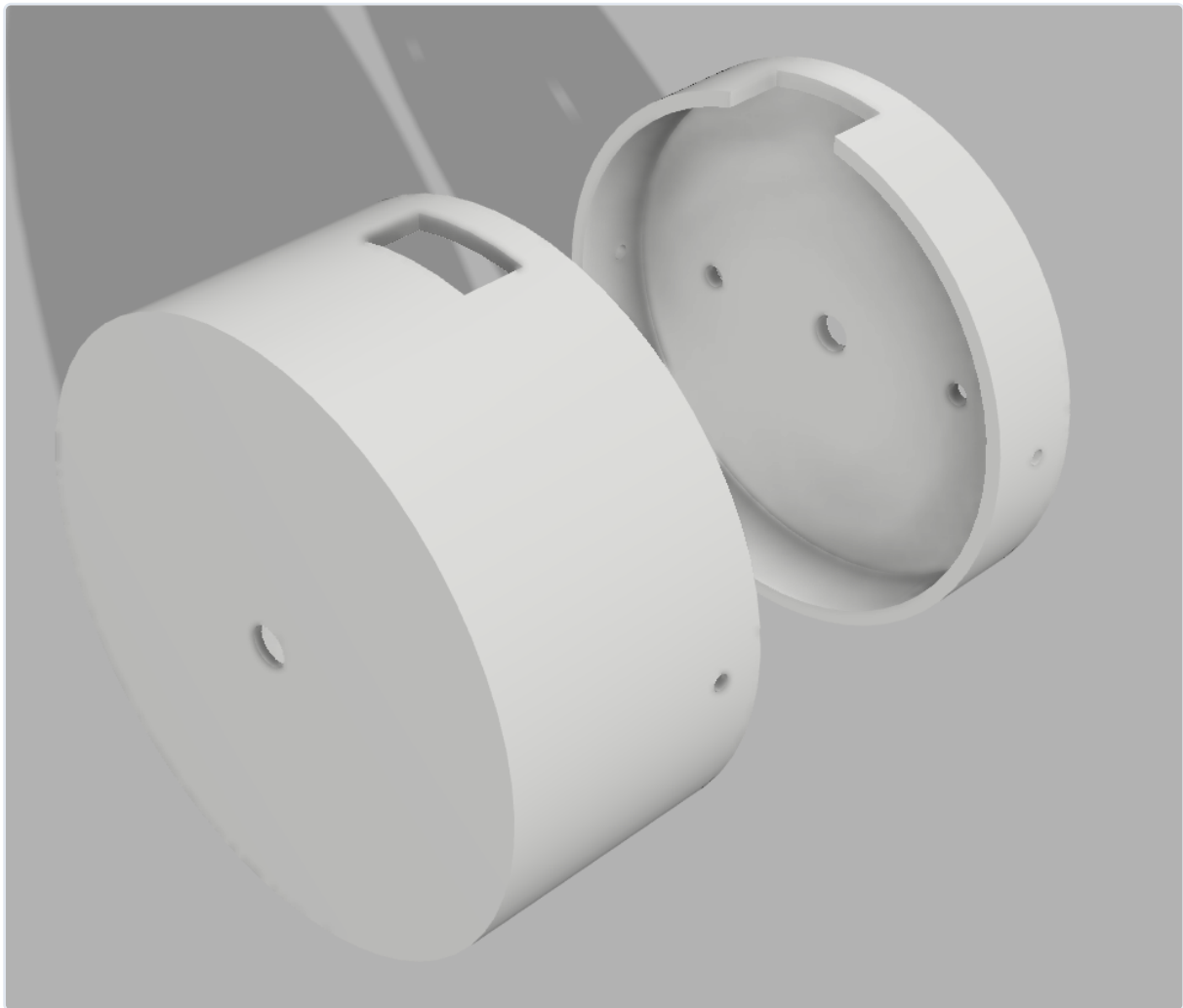
```

ÉTAPE 4

Conception boîtier

Le fond du boîtier présente deux trous pour la fixation au mur et une encoche pour le passage du câble USB.

L'avant du cache est percé à la dimension de l'interrupteur et la profondeur permet de mettre tout le matériel tout en laissant apparaître le port USB du microcontrôleur. Il est également percé de deux petits trous latéraux pour placer deux petites vis pour permettre de rendre solidaires le fond et l'avant.



ÉTAPE 5

Préparer l'application mobile

Vous avez deux possibilités : [l'application mobile](#) ou [l'application internet](#).

L'application mobile :

Rendez-vous sur le store de votre téléphone et installé l'application [ntsy.sh](https://ntfy.sh).

Au lancement vous trouverez un signe "+" permettant de rajouter un fil à suivre, rentrez celui que vous avez déclaré dans votre programme Python.

N'oubliez pas d'activer les notifications en haut.

La web app (ou application internet):

Vous pouvez directement vous rendre sur la page <https://ntfy.sh/app> choisir "S'abonner à un sujet" en bas à gauche de la page et saisir le fil que vous avez intégré dans votre script Python.





ntfy.sh/ [redacted].



16/01/2023 15:22

 Le FabLab vient de fermer

16/01/2023 13:54

 Le FabLab vient d'ouvrir

16/01/2023 13:54

 Le FabLab vient de fermer

16/01/2023 13:40

 Le FabLab vient d'ouvrir

16/01/2023 13:40

 Le FabLab vient de fermer

16/01/2023 08:07

 Le FabLab vient d'ouvrir

ÉTAPE 6

Vous voilà fin prêt

Votre premier objet connecté est prêt, vous devez attendre au lancement que votre microcontrôleur se soit connecté à votre réseau wifi. Cette étape est réalisée quand la diode bleue présente sur l'ESP s'allume et que la séquence "Petite animation du lancement" commence.

ÉTAPE 7

Evolutions envisagées

La prochaine étape sera d'y inclure un lecteur NFC avec un code propre à chaque utilisateur de façon à savoir qui est présent au FabLab, qui a ouvert et fermé le lab.