



# MicroPython - Entrées, Sorties



[Mise à jour le : 12/7/2021] **En cours de rédaction**

- **Ressources**

- [MicroPython.org](https://micropython.org)
- [MicroPython documentation](#)
- [IDE Thonny](#)

## 1. Présentation

Les exemples de code de cette page ont été testés sur une carte ESP32 Feather Huzzah. Voir ses caractéristiques sur la page "[La carte ESP32 Feather Huzzah](#)"

## 2. Entrées, sorties numériques



### 2.1 Sortie numérique

- **Ressource** : [Pins and GPIO](#) sur Micropython.org.
- Exemple pour un **ESP32 Feather Huzzah** (LED #13)

[helloesp32.py](#)

```
# Faire clignoter la led de la carte

from machine import Pin
import time

# Led de la carte
Led = Pin(13, Pin.OUT)
```

```
while(True):  
    Led.on()  
    time.sleep(1)  
    Led.off()  
    time.sleep(1)
```

- Exemple pour un **Raspberry Pi Pico** (LED #25)

[hellopico.py](#)

```
# Faire clignoter la led de la carte  
# A modifier dans le code ci-dessus  
...  
Led = Pin(25, Pin.OUT)  
...
```



## 2.2 Entrée numérique

- **Ressource** : [Pins and GPIO](#) sur Micropython.org.
- **Matériel** : [Digilent Pmod BTN: 4 User Pushbuttons \[Schéma\]](#)
- Exemple pour un **ESP32 Feather Huzzah**

[\\*.py](#)

```
# Configuration en entrée des broches connectées  
# à deux boutons-poussoir  
# A modifier dans le code ci-dessus  
...  
button_min = Pin(25, Pin.IN)  
button_hr = Pin(26, Pin.IN)  
...
```

- Exemple pour un **Raspberry Pi Pico**

[hellopico.py](#)

```
# A modifier dans le code ci-dessus
...
button_min = Pin(20, Pin.IN)
button_hr = Pin(21, Pin.IN)
...
```

## 2.3 Interruption

- **Ressource** : [GPIO Pins, External interrupts](#) sur Micropython.org.
- **Matériel** : ESP32 Feather Huzzah, [Digilent Pmod BTN: 4 User Pushbuttons](#) [Schéma]
- Exemple

\*.py

```
# Code partiel du programme HORLOGE

# Réglage de l'heure à la mise sous tension
time_offset=12*3600+0*60+0 # hh+mm+ss

# Routines de service d'interruption (ISR)
def handle_interrupt_min(pin):
    global time_offset
    time_offset+=60
    time.sleep(.2)

def handle_interrupt_hr(pin):
    global time_offset
    time_offset+=3600
    time.sleep(.2)

# Réglage des minutes
# Ajout de 60s à l'heure initiale
button_min = Pin(25, Pin.IN)
# Gestionnaire d'interruption
button_min.irq(trigger=Pin.IRQ_RISING, handler=handle_interrupt_min)

# Réglage des heures
# Ajout de 3600s à l'heure initiale
button_hr = Pin(26, Pin.IN)
# Gestionnaire d'interruption
button_hr.irq(trigger=Pin.IRQ_RISING, handler=handle_interrupt_hr)
```

Th



**Télécharger** le projet MICROPYTHON\_ESP32\_HORLOGE pour Thonny et la **vidéo** de la démo.



### 3 Entrée analogique

- **Ressource** : [ADC \(analog to digital conversion\)](#) sur Micropython.org.
- **Matériels** : potentiomètre connecté à l'entrée analogique A2 de l'ESP32 Feather Huzzah.
- *Exemple*

[analog.py](#)

```
# ADC accessible en Python sur la carte ESP32 Feather Huzzah :  
#  
# ADC:GPIO  
# A2 : 34  
# A3 : 39  
# A4 : 36  
# A7 : 32  
# A9 : 33  
  
from machine import ADC, Pin  
  
adc = ADC(Pin(34)) # A2  
# Sur une entrée analogique, la tension doit  
# être comprise entre 0 - 3,3V (3,6V max !)  
adc.atten(ADC.ATTN_11DB) # voir doc  
# Mesure  
value = adc.read()  
  
print(value) # affichage dans la console
```

From:

<http://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<http://webge.fr/dokuwiki/doku.php?id=python:micropython:es&rev=1628666361>

Last update: **2021/08/11 09:19**

