



# MicroPython - Entrées, Sorties (GPIO)



[Mise à jour le : 1/6/2023] **En cours de rédaction**

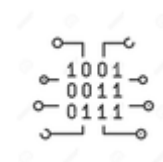
- **Ressources**

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

- **Lectures connexes**

- [ESP32/ESP8266 Digital Inputs and Digital Outputs with MicroPython](#)
- [MicroPython with ESP32 and ESP8266: Interacting with GPIOs](#)
- [ESP32/ESP8266 PWM with MicroPython - Dim LED](#)
- [ESP32/ESP8266 Analog Readings with MicroPython](#)
- [MicroPython: Interrupts with ESP32 and ESP8266](#)

## 2. Entrées, sorties numériques



## 3. Entrées analogiques

- **Ressource**

- [Quick reference for the RP2, Pins and GPIO](#) sur Micropython.org., potentiomètre

10kOhm.

### 3.1 Présentation

- **Ressource**

- [Un signal analogique : petits rappels](#) sur le site Zeste de savoir.



### 3.2 Exemples de code

- [RPi Pico](#)
- [ESP32](#)

- **Ressource**

- [ADC \(analog to digital conversion\) | class Pin – control I/O pins](#) sur Micropython.org.

Exemple de code pour un **Raspberry Pi Pico**

\*.py

```
# -----  
-----  
# Lecture et affichage dans la console de la tension issue d'un  
# potentiomètre  
# Date : 22/5/2023  
# Matériels : Raspberry Pi Pico, Shield Grove, pot. 10k  
# ADC accessibles sur le shield Grove pour RP2 :  
# Connecteur: ADC      : GPIO  
#           A0 : ADC0      : 26  
#           A1 : ADC0,ADC1: 26,27  
#           A2 : ADC1,ADC2: 27,28  
# IDE : Thonny  
# -----  
-----  
from machine import ADC, Pin  
import time  
  
# Le potentiomètre 10kOhm est connecté à l'entrée analogique A0 du  
# shield.  
# Attention : La tension doit être comprise entre 0 - 3,3V (3,6V max !)  
# sur une entrée analogique.  
# Configuration  
pot = ADC(Pin(26))  
  
while (True):
```

```
val=pot.read_u16() # lecture de l'ADC
U = val*3.3/65535 # Calcul de la tension
print("%.2f" % U) # Affichage dans la console (formaté à 2
décimales)
time.sleep(1)
```

- **Ressource**

- [ADC \(analog to digital conversion\)](#) sur Micropython.org.

Exemple de code pour un **ESP32 Feather Huzzah**

\*.py

```
# ADC accessibles 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

# Le potentiomètre 10k0hm est connecté à l'entrée analogique A2 de
l'ESP32.
# Configuration
adc = ADC(Pin(34))
# 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=1692516734>

Last update: **2023/08/20 09:32**

