

# Programmes en Micropython SmartHome

## Test Led pin 12

[exemple001.py](#)

```
from machine import Pin
import time

led = Pin(12, Pin.OUT)# Build an LED object, connect the external LED
light to pin 0, and set pin 0 to output mode
while True:
    led.value(1)# turn on led
    time.sleep(1)# delay 1s
    led.value(0)# turn off led
    time.sleep(1)# delay 1s
```

## Test Led PWM

[exemple002.py](#)

```
import time
from machine import Pin,PWM

#The way that the ESP32 PWM pins output is different from traditionally
controllers.
#It can change frequency and duty cycle by configuring PWM's parameters
at the initialization stage.
#Define GPIO 0's output frequency as 10000Hz and its duty cycle as 0,
and assign them to PWM.
p0 = Pin(012, Pin.OUT)
pwm = PWM(p0, freq=10000, duty_u16=8192)

try:
    while True:
#The range of duty cycle is 0-1023, so we use the first for loop to
control PWM to change the duty
#cycle value,making PWM output 0% -100%; Use the second for loop to
make PWM output 100%-0%.
        for i in range(0,1023):
            pwm.duty(i)
            time.sleep_ms(1)

        for i in range(0,1023):
```

```
        pwm.duty(1023-i)
        time.sleep_ms(1)
except:
    #Each time PWM is used, the hardware Timer will be turned ON to
    #cooperate it. Therefore, after each use of PWM,
    #deinit() needs to be called to turned OFF the timer. Otherwise, the
    #PWM may fail to work next time.
    pwm.deinit()
```

## Test des 2 Boutons

[exemple003.py](#)

```
from machine import Pin
import time

button1 = Pin(16, Pin.IN, Pin.PULL_UP)
button2 = Pin(27, Pin.IN, Pin.PULL_UP)

while True:
    btnVal1 = button1.value() # Reads the value of button 1
    btnVal2 = button2.value()
    print("button1 =", btnVal1) #Print it out in the shell
    print("button2 =", btnVal2)
    time.sleep(0.1) #delay 0.1s
```

## Test Bouton 1 M/A Led

[exempl004.py](#)

```
from machine import Pin
import time

button1 = Pin(16, Pin.IN, Pin.PULL_UP)
led = Pin(12, Pin.OUT)
count = 0

while True:
    btnVal1 = button1.value() # Reads the value of button 1
    #print("button1 =", btnVal1) #Print it out in the shell
    if(btnVal1 == 0):
        time.sleep(0.01)
        while(btnVal1 == 0):
            btnVal1 = button1.value()
```

```
        if(btnVal1 == 1):
            count = count + 1
            print(count)
    val = count % 2
    if(val == 1):
        led.value(1)
    else:
        led.value(0)
    time.sleep(0.1) #delay 0.1s
```

## Test PIR : detection de personnes

[exemple005.py](#)

```
from machine import Pin
import time

PIR = Pin(14, Pin.IN)
while True:
    value = PIR.value()
    print(value, end = " ")
    if value == 1:
        print("Des personnes sont dans la zone ...!")
    else:
        print("il n'y a personne ...!")
    time.sleep(0.1)
```

## Test Buzzer Music

[exemple006.py](#)

```
from machine import Pin, PWM
from time import sleep
buzzer = PWM(Pin(25))

buzzer.duty(1000)

# Happy birthday
buzzer.freq(294)
sleep(0.25)
buzzer.freq(440)
sleep(0.25)
buzzer.freq(392)
sleep(0.25)
buzzer.freq(532)
```

```
sleep(0.25)
buzzer.freq(494)
sleep(0.25)
buzzer.freq(392)
sleep(0.25)
buzzer.freq(440)
sleep(0.25)
buzzer.freq(392)
sleep(0.25)
buzzer.freq(587)
sleep(0.25)
buzzer.freq(532)
sleep(0.25)
buzzer.freq(392)
sleep(0.25)
buzzer.freq(784)
sleep(0.25)
buzzer.freq(659)
sleep(0.25)
buzzer.freq(532)
sleep(0.25)
buzzer.freq(494)
sleep(0.25)
buzzer.freq(440)
sleep(0.25)
buzzer.freq(698)
sleep(0.25)
buzzer.freq(659)
sleep(0.25)
buzzer.freq(532)
sleep(0.25)
buzzer.freq(587)
sleep(0.25)
buzzer.freq(532)
sleep(0.5)
buzzer.duty(0)
```

## Test Servo Porte entrée : angle 25° 77° 180°

[exemple007.py](#)

```
from machine import Pin, PWM
import time
pwm = PWM(Pin(13))
pwm.freq(50)

...

```

### Duty cycle corresponding to the Angle

```
0°----2.5%----25
45°----5%----51.2
90°----7.5%----77
135°----10%----102.4
180°----12.5%----128
...
```

```
angle_0 = 25
angle_90 = 77
angle_180 = 128
```

```
while True:
    pwm.duty(angle_0)
    time.sleep(1)
    pwm.duty(angle_90)
    time.sleep(1)
    pwm.duty(angle_180)
    time.sleep(1)
```

From:

<https://www.fablab37110.chanterie37.fr/> - Castel'Lab le Fablab MJC de Château-Renault

Permanent link:

<https://www.fablab37110.chanterie37.fr/doku.php?id=start:arduino:esp32:smart:micropython&rev=1740571198>

Last update: 2025/02/26 12:59

