Programmes en Micropython SmartHome

Test Led pin 12

exemple001.py

```python
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

```python
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):
```

```python
            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

```python
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

```python
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()
```

```python
        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

```python
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

```python
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)
```

```python
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

```python
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)
```

Test Capteur Humidité 0= Fenetre Ouverte, 0> Fenetre fermée

exemple008.py

```python
# Import Pin, ADC and DAC modules.
from machine import ADC,Pin,DAC,PWM
import time
pwm = PWM(Pin(5))
pwm.freq(50)

# Turn on and configure the ADC with the range of 0-3.3V
adc=ADC(Pin(34))
adc.atten(ADC.ATTN_11DB)
adc.width(ADC.WIDTH_12BIT)

# Read ADC value once every 0.1seconds, convert ADC value to DAC value
and output it,
# and print these data to "Shell".
try:
    while True:
        adcVal=adc.read()
        dacVal=adcVal//16
        voltage = adcVal / 4095.0 * 3.3
        print("ADC
Val:",adcVal,"DACVal:",dacVal,"Voltage:",voltage,"V")
        if(voltage > 0.6):
            pwm.duty(46)
        else:
            pwm.duty(100)
```

```
        time.sleep(0.1)
except:
    pass
```

Test Neopixels : Blanc, Rouge, Vert, Bleu

exemple009.py

```python
#Import Pin, neopiexl and time modules.
from machine import Pin
import neopixel
import time

#Define the number of pin and LEDs connected to neopixel.
pin = Pin(26, Pin.OUT)
np = neopixel.NeoPixel(pin, 4)

#brightness :0-255
brightness=100
colors=[[brightness,0,0],                   #red
        [0,brightness,0],                   #green
        [0,0,brightness],                   #blue
        [brightness,brightness,brightness], #white
        [0,0,0]]                            #close

#Nest two for loops to make the module repeatedly display five states
of red, green, blue, white and OFF.
while True:
    for i in range(0,5):
        for j in range(0,4):
            np[j]=colors[i]
            np.write()
            time.sleep_ms(50)
        time.sleep_ms(500)
    time.sleep_ms(500)
```

Test Bouton et Neopixels Bp1 - Bp2 + incremente les couleurs

exemple010.py

```python
#Import Pin, neopiexl and time modules.
from machine import Pin
import neopixel
import time
```

```python
button1 = Pin(16, Pin.IN, Pin.PULL_UP)
button2 = Pin(27, Pin.IN, Pin.PULL_UP)
count = 0

#Define the number of pin and LEDs connected to neopixel.
pin = Pin(26, Pin.OUT)
np = neopixel.NeoPixel(pin, 4)

#brightness :0-255
brightness=100
colors=[[0,0,0],
        [brightness,0,0],                       #red
        [0,brightness,0],                       #green
        [0,0,brightness],                       #blue
        [brightness,brightness,brightness]  #white
        ]                                       #close

def func_color(val):
    for j in range(0,4):
        np[j]=colors[val]
        np.write()
        time.sleep_ms(50)

#Nest two for loops to make the module repeatedly display five states
of red, green, blue, white and OFF.
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)
                if(count <= 0):
                    count = 0

    btnVal2 = button2.value()
    if(btnVal2 == 0):
        time.sleep(0.01)
        while(btnVal2 == 0):
            btnVal2 = button2.value()
            if(btnVal2 == 1):
                count = count + 1
                print(count)
                if(count >= 4):
                    count = 4

    if(count == 0):
```

```
        func_color(0)
    elif(count == 1):
        func_color(1)
    elif(count == 2):
        func_color(2)
    elif(count == 3):
        func_color(3)
    elif(count == 4):
        func_color(4)
```

From:
https://chanterie37.fr/fablab37110/ - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:
https://chanterie37.fr/fablab37110/doku.php?id=start:arduino:esp32:smart:micropython&rev=1740575314

Last update: **2025/02/26 14:08**