

Un proyecto diferente: Encender y apagar led por wifi

En la lista de proyectos que propone PicoBricks sólo hay uno que usa la Wifi [SmartHome](#), pero **no utiliza la wifi de Raspberry Pi** sino que utiliza un módulo wifi ESP8266 auxiliar.

Proponemos uno que no use elementos auxiliares

Enunciado: Encender y apagar el led rojo conectado en GPI7 a través de una página web puesto en el servidor que se instala en la Raspberry

Solución

La explicación del programa está en <https://peppe8o.com/getting-started-with-wifi-on-raspberry-pi-pico-w-and-micropython/>

La fuente del programa en <https://github.com/raspberrypi/pico-micropython-examples/blob/master/wireless/webserver.py>

Recuerda que tienes que poner los datos de tu wifi en las líneas 35 y 36

```
import socket

#####33

import network, rp2
import time

def connectWiFi(ssid,password,country):
    rp2.country(country)
    wlan = network.WLAN(network.STA_IF)
    wlan.config(pm = 0xa11140)
    wlan.active(True)
    wlan.connect(ssid, password)
```

```
# Wait for connect or fail
max_wait = 10
while max_wait > 0:
    if wlan.status() < 0 or wlan.status() >= 3:
        break
    max_wait -= 1
    print('waiting for connection...')
    time.sleep(1)

# Handle connection error
if wlan.status() != 3:
    raise RuntimeError('network connection failed')
else:
    print('connected')
    status = wlan.ifconfig()
    print( 'ip = ' + status[0] )
    return status

#####333
from machine import Pin

led = Pin(7, Pin.OUT)

country = 'ES'
ssid = 'pon aqui el nombre de tu wifi'
password = 'pon aqui el password de tu wifi'

wifi_connection = connectWiFi(ssid,password,country)
#####33333
html = """<!DOCTYPE html>
<html>
<head> <title>Pico W</title> </head>
<body> <h1>Pico W</h1>
<p>Current status: %s</p>
<p><a href="http://""+wifi_connection[0]+""/light/on">Turn ON</a></p>
<p><a href="http://""+wifi_connection[0]+""/light/off">Turn OFF</a></p>
<p>by <a href="https://peppe8o.com">peppe8o.com</a></p>
```

```
</body>
```

```
</html>
```

```
"""
```

```
#####
```

```
# Open socket
```

```
addr = socket.getaddrinfo('0.0.0.0', 80)[0][-1]
```

```
s = socket.socket()
```

```
s.bind(addr)
```

```
s.listen(1)
```

```
print('listening on', addr)
```

```
# Initialize LED status
```

```
led.value(0)
```

```
stateis = "LED is OFF"
```

```
# Listen for connections
```

```
while True:
```

```
    try:
```

```
        cl, addr = s.accept()
```

```
        print('client connected from', addr)
```

```
        request = cl.recv(1024)
```

```
        print(request)
```

```
        request = str(request)[0:50] # The [0:50] avoids getting the url directory from referer
```

```
        led_status = request.find('GET / HTTP')
```

```
        led_on = request.find('/light/on')
```

```
        led_off = request.find('/light/off')
```

```
        print( 'led on = ' + str(led_on))
```

```
        print( 'led off = ' + str(led_off))
```

```
        if led_status >0:
```

```
            print("LED status request") # No LED action
```

```

if led_on >0:
    print("led on")
    led.value(1)
    stateis = "LED is ON"

if led_off >0:
    print("led off")
    led.value(0)
    stateis = "LED is OFF"

response = html % stateis

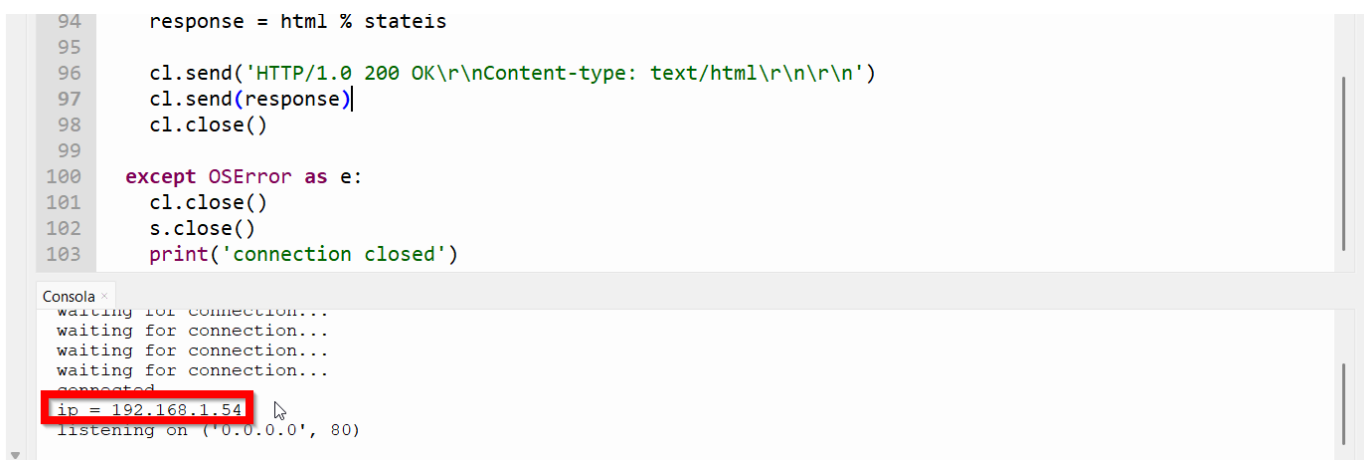
cl.send('HTTP/1.0 200 OK\r\nContent-type: text/html\r\n\r\n')
cl.send(response)
cl.close()

except OSError as e:
    cl.close()
    s.close()
    print('connection closed')

```

Ejecución del programa

Para encender y apagar el led tienes que entrar en la IP de la Raspberry Pi, puedes verlo en la ventana del puerto serie (cónsola) que puedes ver en el programa Thonny:



```

94     response = html % stateis
95
96     cl.send('HTTP/1.0 200 OK\r\nContent-type: text/html\r\n\r\n')
97     cl.send(response)
98     cl.close()
99
100 except OSError as e:
101     cl.close()
102     s.close()
103     print('connection closed')

```

Console

```

waiting for connection...
waiting for connection...
waiting for connection...
waiting for connection...
connected
ip = 192.168.1.54
listening on ('0.0.0.0', 80)

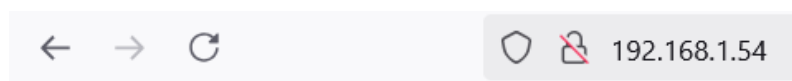
```



Otro truco es ejecutar un programa de rastreo de IPs como el [Wireless Network Watcher](#) y ver la IP de la Raspberry. O poner una IP estática ver [aquí](#)

IP Address	Device Name	MAC Address	Network A
192.168.1.1	WiFi_pico.domain.name	88-7E-28-03-00-00	192.168.1.1
192.168.1.2	192-168-1-2	88-7E-28-03-00-00	192.168.1.2
192.168.1.3	192-168-1-3	88-7E-28-03-00-00	192.168.1.3
192.168.1.4	192-168-1-4	88-7E-28-03-00-00	192.168.1.4
192.168.1.5	192-168-1-5	88-7E-28-03-00-00	192.168.1.5
192.168.1.54	PicoW.domain.name	28-CD-C1-06-03-7C	192.168.1.54
192.168.1.6	192-168-1-6	88-7E-28-03-00-00	192.168.1.6
192.168.1.7	192-168-1-7	88-7E-28-03-00-00	192.168.1.7

Abrimos un navegador y ponemos la IP de la Raspberry en mi caso 192.168.1.54



Pico W

Current status: LED is OFF

[Turn ON](#)

[Turn OFF](#)

by [peppe8o.com](#)

https://www.youtube.com/embed/pRtXEg_cCCI

Si os sale el error OSError: [Errno 98] EADDRINUSE es porque no se ha cerrado bien la conexión, desconectar PicoBrikcs y volverlo a conectar y solucionado

Revision #5

Created 28 December 2024 22:28:08 by Javier Quintana

Updated 30 December 2024 20:18:42 by Javier Quintana