

Guía rápida de instalación de MicroPython ene ESP8266

1. Instalat esptool

pip3 install esptool

2. Ver información de la placa

esptool flash_id

3. Borrar firmware de la placa

esptool.py --port /dev/ttyUSB0 erase_flash

4. Flashear firimware de MicroPython (se requiere archivo bin del firmware)

**esptool.py --port /dev/ttyUSB0 --baud 460800 write_flash --flash_size=detect 0
esp8266-20170108-v1.8.7.bin**

5. Buscar la red MicropythonXXXX y conectarse con password "micropythoN"

6. Conectarse con el monitor serial del IDE Arduino para habilitar el WEBRepl (consola Web). NOTA: REPL significa Read, Evaluate, Print y Loop back.

import webrepl_setup

E (Para habilitar el servicio)

XXXXXX (dar password y confirmarlo)

7. Descargar WEBREPL

<https://github.com/micropython/webrepl>

8. Después ir al navegador para invocar la página "webrepl.html"

la cual se conectará al dispositivo vía web cuya ip por defecto y puerto

es 192.168.4.1:8266. Dar el password proporcionado anteriormente cuando se configuró el webrepl en el paso 6.

Algunos comandos útiles:

machine.reset()

os.listdir()

os.rename('nombre_anterior','nombre_nuevo')

os.remove('archivo')

Código de prueba

(Para manipular un display de 7 segmentos)

```
import machine
import time

#           D2
#           #####
#       D3 #   # D1
#           #   #
#           #####
#       D5 # D4 # D7
#           #   #
#           #####
#           D6
#
# Correspondencia en los pines:
# D1 pin 5, D2 pin 4, D3 pin 0, D4 pin 2, D5 pin 14, D6 pin 12, D7 pin 13

ninguno = {0:0, 14:0, 2:0, 13:0, 4:0, 5:0, 12:0}
cero = {0:1, 14:1, 2:0, 13:1, 4:1, 5:1, 12:1}
uno = {0:0, 14:0, 2:0, 13:1, 4:0, 5:1, 12:0}
dos = {0:0, 14:1, 2:1, 13:0, 4:1, 5:1, 12:1}
tres = {0:0, 14:0, 2:1, 13:1, 4:1, 5:1, 12:1}
cuatro = {0:1, 14:0, 2:1, 13:1, 4:0, 5:1, 12:0}
cinco = {0:1, 14:0, 2:1, 13:1, 4:1, 5:0, 12:1}
seis = {0:1, 14:1, 2:1, 13:1, 4:1, 5:0, 12:1}
siete = {0:0, 14:0, 2:0, 13:1, 4:1, 5:1, 12:0}
ocho = {0:1, 14:1, 2:1, 13:1, 4:1, 5:1, 12:1}
nueve = {0:1, 14:0, 2:1, 13:1, 4:1, 5:1, 12:1}

secuencia = [cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho, nueve, ninguno]
secuencia2 = [nueve, ocho, siete, seis, cinco, cuatro, tres, dos, uno, cero, ninguno]
```

```

def numero(lista):
    for i in lista:
        if lista[i]==1:
            pin=machine.Pin(i,machine.Pin.OUT)
            pin.on()
        else:
            pin=machine.Pin(i,machine.Pin.OUT)
            pin.off()

```

```

def cuenta(tiempo,sec):
    for i in sec:
        numero(i)
        time.sleep(tiempo)
        print (".")

```

```

numero(ninguno)
print ("Iniciando sharks...")
cuenta(1,secuencia)
print ("Termina cuenta 1 seg")
cuenta(0.5,secuencia)
print ("Termina cuenta 0.5 seg")
cuenta(0.25,secuencia)
print ("Termina cuenta 0.25 seg")
cuenta(0.750,secuencia2)
print ("Adios tiburones !!")

```

Código 2 (Para conectar a una red WIFI)

```

def conectar_wifi():
    import network
    sta_if = network.WLAN(network.STA_IF)
    if not sta_if.isconnected():
        print('Conectando a la red...')
        sta_if.active(True)
        sta_if.connect('SSID', '12341234') #Modificar por datos de WiFi
        while not sta_if.isconnected():
            print ("+" )
            print ("OK")
        red = sta_if.ifconfig()
        print('Configuracion de la red:', red[0:3])
conectar_wifi()

```

Código 3 (Servidor Web para controlar el ESP8266)

```
import socket
import gc
import machine

ninguno = {0:0, 14:0, 2:0, 13:0, 4:0, 5:0, 12:0}
cero = {0:1, 14:1, 2:0, 13:1, 4:1, 5:1, 12:1}
uno = {0:0, 14:0, 2:0, 13:1, 4:0, 5:1, 12:0}
dos = {0:0, 14:1, 2:1, 13:0, 4:1, 5:1, 12:1}
tres = {0:0, 14:0, 2:1, 13:1, 4:1, 5:1, 12:1}
cuatro = {0:1, 14:0, 2:1, 13:1, 4:0, 5:1, 12:0}
cinco = {0:1, 14:0, 2:1, 13:1, 4:1, 5:0, 12:1}
seis = {0:1, 14:1, 2:1, 13:1, 4:1, 5:0, 12:1}
siete = {0:0, 14:0, 2:0, 13:1, 4:1, 5:1, 12:0}
ocho = {0:1, 14:1, 2:1, 13:1, 4:1, 5:1, 12:1}
nueve = {0:1, 14:0, 2:1, 13:1, 4:1, 5:1, 12:1}

def numero(lista):
    for i in lista:
        if lista[i]==1:
            pin=machine.Pin(i,machine.Pin.OUT)
            pin.on()
        else:
            pin=machine.Pin(i,machine.Pin.OUT)
            pin.off()

# For more details and step by step guide visit: Microcontrollerslab.com
led_state = "OFF"

led = machine.Pin(2,machine.Pin.OUT)

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('', 80))
s.listen(5)
```

```

while True:
    try:
        if gc.mem_free() < 102000:
            gc.collect()
        conn, addr = s.accept()
        conn.settimeout(3.0)
        print('Received HTTP GET connection request from %s' % str(addr))
        request = conn.recv(1024)
        conn.settimeout(None)
        request = str(request)
        print('GET Rquest Content = %s' % request)
        led_on = request.find('/?led_2_on')
        led_off = request.find('/?led_2_off')

        display_0 = request.find('/?disp0')
        display_1 = request.find('/?disp1')
        display_2 = request.find('/?disp2')
        display_3 = request.find('/?disp3')
        display_4 = request.find('/?disp4')
        display_5 = request.find('/?disp5')
        display_6 = request.find('/?disp6')
        display_7 = request.find('/?disp7')
        display_8 = request.find('/?disp8')
        display_9 = request.find('/?disp9')
        display_n = request.find('/?dispN')

        if led_on == 6:
            print('LED ON -> GPIO2')
            led_state = "ON"
            led.on()
        if led_off == 6:
            print('LED OFF -> GPIO2')
            led_state = "OFF"
            led.off()
        if display_0 == 0:
            print ("CERO")
            numero(cero)
        if display_1 == 6:
            print ("UNO")
            numero(unos)

```

```
if display_2 == 6:
    print ("DOS")
    numero(dos)
if display_3 == 6:
    print ("TRES")
    numero(tres)
if display_4 == 6:
    print ("CUATRO")
    numero(cuatro)
if display_5 == 6:
    print ("CINCO")
    numero(cinco)
if display_6 == 6:
    print ("SEIS")
    numero(seis)
if display_7 == 6:
    print ("SIETE")
    numero(siete)
if display_8 == 6:
    print ("OCHO")
    numero(ocho)
if display_9 == 6:
    print ("NUEVE")
    numero(nueve)
if display_n == 6:
    print ("APAGADO")
    numero(ninguno)
response = "OK"
conn.send('HTTP/1.1 200 OK\n')
conn.send('Content-Type: text/html\n')
conn.send('Connection: close\n\n')
conn.sendall(response)
conn.close()
except OSError as e:
    conn.close()
    print('Connection closed')
```