

Inhaltsverzeichnis

Power Neopixel mit ESP8266 ESP-12S 3

Power Neopixel mit ESP8266 ESP-12S

Der [ESP8266 ESP-12S](#) hat ne Macke, was den Bootmodus angeht.

Erste Hinweise finden sich hier:

https://www.reddit.com/r/embedded/comments/16yo4uf/esp8266_esp12s_wifi_microcontroller_iotmcu/

Daraus wurde dann:

- GPIO0 (=IO0) auf GND
- GPIO15 auf GND (pin wie auf dem Bild gezeigt)
- RST auf GND beim Booten



Dann ist der ESP im Bootmode

[Firmware runterladen](#)

esptool und thony installieren

```
sudo apt install esptool
sudo snap install thony
```

Flash löschen

```
esptool --port /dev/ttyUSB0 erase_flash
```

Firmware flashen

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

mit der Console verbinden

```
screen /dev/ttyUSB0 115200
```

Das Webinterface aktivieren: (siehe

<https://learn.adafruit.com/micropython-basics-esp8266-webrepl/access-webrepl>)

```
import webrepl_setup
```

Achtung: Das gewählte Passwort gilt nur für WebREPL selber, der ESP eigene Hotspot hat das Passwort micropythonN (das große N ist kein Tippfehler)

WebREPL runterladen und local starten (<https://github.com/micropython/webrepl/archive/master.zip>)

Mit dem Computer- WLAN mit dem ESP- Hotspot verbinden. Das WebREPL liegt dann auf `ws://192.168.4.1:8266/`

Dem Neopixel- Tutorial folgen: <https://docs.micropython.org/en/latest/esp8266/tutorial/neopixel.html>

Ein paar fette Power- RGB- LEDs gibts bei [Zedfy](#)

```
Welcome to MicroPython!
```

```
Password:
```

```
WebREPL connected
```

```
import machine, neopixel
np = neopixel.NeoPixel(machine.Pin(2),1)
np[0] = (1,1,1)
np.write()
```

[source](#)

[main.py](#)

```
import urandom
import time
import machine, neopixel

#https://www.rapidtables.com/web/color/RGB\_Color.html

fackel = [
    (128,0,0) , #800000 maroon
    (139,0,0) , #8B0000 dark red
    (165,42,42) , #A52A2A brown
    (178,34,34) , #B22222 firebrick
    (220,20,60) , #DC143C crimson
    (255,0,0) , #FF0000 red
    (255,99,71) , #FF6347 tomato
    (255,127,80) , #FF7F50 coral
    (205,92,92) , #CD5C5C indian red
    (240,128,128) , #F08080 light coral
```

```
(233,150,122) , #E9967A dark salmon
(250,128,114) , #FA8072 salmon
(255,160,122) , #FFA07A light salmon
(255,69,0) , #FF4500 orange red
(255,140,0) , #FF8C00 dark orange
(255,165,0) , #FFA500 orange
]

campfire = [
(128,0,0) , #800000 maroon
(139,0,0) , #8B0000 dark red
(165,42,42) , #A52A2A brown
(178,34,34) , #B22222 firebrick
(220,20,60) , #DC143C crimson
(255,0,0) , #FF0000 red
(255,69,0) , #FF4500 orange red
(255,140,0) , #FF8C00 dark orange
]

brunnen = [
#campfire = [
(32,178,170) , #20B2AA light sea green
(47,79,79) , #2F4F4F dark slate gray
(0,128,128) , #008080 teal
(0,139,139) , #008B8B dark cyan
(0,255,255) , #00FFFF aqua
(0,255,255) , #00FFFF cyan
(224,255,255) , #E0FFFF light cyan
(0,206,209) , #00CED1 dark turquoise
(64,224,208) , #40E0D0 turquoise
(72,209,204) , #48D1CC medium turquoise
(175,238,238) , #AFEEEE pale turquoise
(127,255,212) , #7FFFD4 aqua marine
(176,224,230) , #B0E0E6 powder blue
(95,158,160) , #5F9EA0 cadet blue
(70,130,180) , #4682B4 steel blue
(100,149,237) , #6495ED corn flower blue
(0,191,255) , #00BFFF deep sky blue
(30,144,255) , #1E90FF dodger blue
(173,216,230) , #ADD8E6 light blue
(135,206,235) , #87CEEB sky blue
(135,206,250) , #87CEFA light sky blue
(25,25,112) , #191970 midnight blue
(0,0,128) , #000080 navy
(0,0,139) , #00008B dark blue
(0,0,205) , #0000CD medium blue
(0,0,255) , #0000FF blue
]
]
```

```
class LightTheme:
    """
    contains all to control one lamp
    """

    def __init__(self, neopixels: object, index:int, theme: list,
brightness: int, ticks: int = 1):
        self.neopixels = neopixels
        self.index=index
        self.theme = theme
        self.brightness = brightness
        self.ticks = ticks
        self.tick_count = 0
        self.old_red = 0
        self.old_green = 0
        self.old_blue = 0
        self.new_red = 0
        self.new_green = 0
        self.new_blue = 0
        self.actual_color_index = 0

    def randint(self, max):
        div = 0x3fffffff // max
        return urandom.getrandbits(30) // div

    def rgb(self) -> tuple:
        """
        this does it all:
        whenever the number of ticks are gone, it selects a new random
        color out of the array,
        corrects the brightness and returns the new color
        """
        self.tick_count += 1
        if self.tick_count >= self.ticks:
            self.tick_count = 0
            color_size = len(self.theme)
            self.old_red = self.new_red
            self.old_green = self.new_green
            self.old_blue = self.new_blue
            self.actual_color_index = (
                self.actual_color_index + self.randint(color_size - 1)
            ) % color_size
            self.new_red = self.theme[self.actual_color_index][0]
            self.new_green = self.theme[self.actual_color_index][1]
            self.new_blue = self.theme[self.actual_color_index][2]
            #print(self.color_index)
        new_color= (
            ## caution: for whatever reason Red and Green are swapped
            in the RGB strip!?

```

```
        (self.old_green + ((self.new_green - self.old_green)*
self.tick_count // self.ticks)) * self.brightness // 100,
        (self.old_red + ((self.new_red - self.old_red)*
self.tick_count // self.ticks)) * self.brightness // 100,
        (self.old_blue + ((self.new_blue - self.old_blue)*
self.tick_count // self.ticks)) * self.brightness // 100,
    )
    self.neopixels[self.index]=new_color
    return new_color

lamps = []
np = neopixel.NeoPixel(machine.Pin(2),5)
lamps.append(LightTheme(np, 0, campfire, 5, 4))
lamps.append(LightTheme(np, 1, campfire, 5, 4))
lamps.append(LightTheme(np, 2, campfire, 5, 4))
lamps.append(LightTheme(np, 3, brunnen, 2, 4))
lamps.append(LightTheme(np, 4, fackel, 5, 4))

while True:
    for lamp in lamps:
        lamp.rgb()
        #print("-" * 20)
        np.write()
        time.sleep(0.015)
```

From:

<http://koehlers.de/wiki/> - **Steffen Köhlers Online- Bastelbuch**

Permanent link:

<http://koehlers.de/wiki/doku.php?id=smarthome:esp12sneopixel>

Last update: **2025/10/18 06:59**

