

# Der Code

Hier ist ein Beispielcode für eure Programmierung in Micropython in der IDE Thonny auf dem Raspberry Pi Pico.

```
# Bibliotheken laden
#-----

from machine import Pin, Timer
from neopixel import NeoPixel
from utime import sleep_ms
import time
from random import randint


# Onboard LED vom Raspberry Pi Pico (Blinkt zum testen einer Verbindung)
#-----

led = Pin(25, Pin.OUT)
timer = Timer()
def blink(timer):
    led.toggle()
timer.init(freq=2.5, mode=Timer.PERIODIC, callback=blink)


# Bewegungssensor
#-----

# Definiert eine Variable für den PIR-Sensor
# PIR ist eine Abkürzung und bedeutet Pulse Repetition Interval (Deut.: Intervall der Impulsfolge) und bezieht
sich auf die Frequenz, mit der ein Bewegungsmelder ein Signal auslöst, wenn dieser eine bestimmte Bewegung
erkennt.
pirsensorpin = machine.Pin(28, machine.Pin.IN)


# LEDs
#-----

# Festlegung der Farben im RGB-Code
# RGB-Farben könnt ihr unter >> https://html-color.codes/ << erstellen
colors = [
    (24, 0, 0),
```

```

(0, 24, 0),
(0, 0, 24),
(12, 12, 0),
(0, 12, 12),
(12, 0, 12),
]

# Farben aus colors in eine Liste (Array) schreiben
array = len(colors)

# Pin für WS2812B LED-Streifen festlegen
ledpin = 22

# Anzahl der LEDs des LED-Streifens angeben
ledanzahl = 15

# Geschwindigkeit definieren! Je höher der Wert, des so langsamer die Animation
speed = 120

# Initialisierung des WS2812B LED-Streifens durch die Variable 'ledstreifen'
ledstreifen = NeoPixel(Pin(ledpin, Pin.OUT), ledanzahl)


# Eine eigene Funktion
#-----
# Wir bauen (definieren) unsere eigene Funktion, um eine Animation zu simulieren. die neue Funktion heißt bei
uns rainbow (Deut.: Regenbogen)
def rainbow():
    # In der Schleife wird eine bestimmte Anzahl an Wiederholungen durchlaufen. Dabei ist die Anzahl der
    Durchläufe gleich die Anzahl der LED's. Die Variable i erhöht sich mit jedem Durchlauf, bis die Anzahl der LED's
    (bei uns 15) erreicht ist.
    # Hinweis: Eine Iteration ist ein wiederholtes Durchlaufen eines bestimmten Prozesses.
    for i in range (ledanzahl):
        # Es wird eine zufällige Farbe aus dem array 'colors' für jedes LED-Element (an der Stelle i) des LED-Streifens
        erzeugt.
        ledstreifen[i] = colors[randint(0, array-1)]
    # sendet die angegebenen LED-Streifen-Daten zu dem LED-Streifen. Dies schaltet die LEDs auf den
    angegebenen Einstellungen auf dem LED-Streifen ein.
    ledstreifen.write()
    # Angabe der Millisekunden, wo der Code eine Pause machen soll. In Micropython gibt die sleep_ms()-Funktion
    dem Benutzer die Möglichkeit, Wartezeiten von einigen Millisekunden bis zu 255 Sekunden einzustellen.
    sleep_ms(speed)


# Wiederholung (Endlos-Schleife)
#-----
while True:

```

```
# Wenn der Bewegungssensor einen Wert erhält, wie bspw. durch eine Bewegung, soll die  
if prisensorpin.value():  
    print("motion detected")  
    rainbow()  
else:  
    ledstreifen.fill((0,0,0))  
    ledstreifen.write()
```

---

Revision #1

Created 13 April 2025 19:47:45 by Michelle Pröhl

Updated 13 April 2025 19:48:04 by Michelle Pröhl