Der Code

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

```
import time
import random
from machine import Pin, UART
from dfplayer import DFPlayer

# HC-SR501 Bewegungsmelder
pir_sensor = Pin(28, Pin.IN)  # Verbinde den Bewegungsmelder mit Pin 28

# Onboard-LED des Raspberry Pi Pico
led_pico = Pin(25, Pin.OUT)  # Onboard-LED an Pin 25

# Busy-Pin des DFPlayer Mini
busy_pin = Pin(22, Pin.IN)  # Verbinde den Busy-Pin des DFPlayer mit Pin 22

# UART-Konfiguration für DFPlayer
uart = UART(0, baudrate=9600, tx=Pin(16), rx=Pin(17))

# DFPlayer MP3 Player
dfplayer = DFPlayer(uart_id=0, tx_pin_id=16, rx_pin_id=17)  # Initialisiere DFPlayer mit UART Pins

time.sleep(2)  # Warte, bis der DFPlayer vollständig bereit ist

# Setze die Lautstärke einmalig
print("Setze Lautstärke...")
dfplayer.volume(10)  # Lautstärke auf 15 setzen (Bereich 0-30)
time.sleep(0.5)

# Funktion zum Abspielen einer zufälligen MP3-Datei
def play_random_mp3():
    track_number = random.randint(1, 4)  # Wähle eine zufällige Datei im Bereich 001-004
    time.sleep(0.2)  # Kurze Pause nach der Zufallsauswahl
    print(f"Spiele MP3-Datei {track_number} ab...")
    dfplayer.play(1, track_number)  # Spiele die Datei im Ordner ./01/
```

```
    time.sleep(0.2)  # Zusätzliche kurze Pause für Stabilität
    print("MP3-Wiedergabe gestartet.")

# Hauptprogramm
playing = False
waiting_for_reset = False
reset_timeout = 5  # Timeout für die Bewegungserkennung in Sekunden
reset_start_time = 0  # Startzeit für das Timeout

while True:
    # Prüfen, ob der Sensor keine Bewegung mehr meldet, bevor eine neue erkannt werden kann
    if not playing and not waiting_for_reset and pir_sensor.value() == 1:
        print("Bewegung erkannt!")
        led_pico.value(1)  # LED einschalten
        play_random_mp3()
        playing = True  # Setzt den Status auf "abspielend"

    if playing:
        # Überprüfe den Status des Busy-Pins und steuere die LED entsprechend
        if busy_pin.value() == 0:  # Busy-Pin ist LOW, wenn eine Datei abgespielt wird
            print("DFPlayer spielt MP3-Datei ab.")
        else:
            print("DFPlayer ist im Leerlauf.")
            led_pico.value(0)  # LED ausschalten
            playing = False  # Wiedergabe ist abgeschlossen
            waiting_for_reset = True  # Warte auf die Beendigung der Bewegungserkennung
            reset_start_time = time.time()  # Startzeit für das Timeout setzen

    # Überprüfen, ob das Timeout abgelaufen ist oder der Sensor keine Bewegung mehr meldet
    if waiting_for_reset:
        if pir_sensor.value() == 0:
            print("Bereit für neue Bewegungserkennung.")
            waiting_for_reset = False
        elif time.time() - reset_start_time > reset_timeout:
            print("Timeout erreicht. Bereit für neue Bewegungserkennung.")
            waiting_for_reset = False

    time.sleep(0.1)  # Kurze Pause, um Sensor nicht zu überlasten
```

Revision #1
Created 13 April 2025 18:48:23 by Shelly Pröhl
Updated 13 April 2025 18:48:55 by Shelly Pröhl