

Inhaltsverzeichnis

Pipimat 2000 - Die Teppichnässer- Alarmanlage	3
<i>Installation von Circuit Python</i>	3
<i>Das Programm</i>	3

Pipimat 2000 - Die Teppichnässer- Alarmanlage

Klingt komisch, is aber so: Diese Schaltung macht MP3- Radau, wenn die Sensormatte einen vierbeinigen Mitbewohner dabei erwischt, dass er mal wieder seine Reviermarke in den Teppich setzt.

Installation von Circuit Python

Wenn der Pico sich am Rechner nicht als USB-Drive zeigt: Beim Einstecken den BOOTSEL- Button so lange gedrückt halten, bis das Laufwerk erkannt wird.

Dann das aktuelle UF2 File von [circuitpython homepage](#) herunterladen und durch Drag & Drop auf dem Pico installieren.

Der Pico wird dann neu booten und sich als CIRCUITPI neu anmelden.

Dann wird der Audio-Anteil verkabelt wie auf <https://learn.adafruit.com/mp3-playback-rp2040> gezeigt:

Signal	Pico	Pin
PAM8302 A+	GP0	1
PAM8302 A-	GND	3
PAM8302 VIN	VBUS (USB 5V)	40
PAM8302 GND	GND	23

Für's Audio- Signal

Audio- Signal

brauchen wir MP3 Mono 16Bit PCM max. 24kHz Sample Rate. Wie man das mit Audacity hinbekommt, steht auf <https://learn.adafruit.com/microcontroller-compatible-audio-file-conversion>

Das Programm

```
import board
import audiomp3
import audiopwmio
import time
import analogio
import digitalio

led = digitalio.DigitalInOut(board.LED)
led.direction = digitalio.Direction.OUTPUT
led.value = True
# Set up an analog input on ADC0 (GP26), which is physically pin 31.
# E.g., this may be attached to photocell or photointerrupter with
associated pullup resistor.
sensor = analogio.AnalogIn(board.A0)
```

```
alarm = False
supress_alarm= False
lower_threshold = 50000
upper_threshold = 55000

button = digitalio.DigitalInOut(board.GP2)
button.switch_to_input(pull=digitalio.Pull.UP)

audio = audiopwmio.PWMAudioOut(board.GP0)

decoder = audiomp3.MP3Decoder(open("trash.mp3", "rb"))

while True:

    if alarm:
        led.value = not led.value
    else:
        led.value = True
    # Read the sensor once per cycle.
    sensor_level = sensor.value

    # uncomment the following to print tuples to plot with the mu editor
    print(sensor_level)
    if alarm:
        if sensor_level > upper_threshold:
            led.value = False
            alarm = False
            print("Off")
        else:
            supress_alarm=False
            if sensor_level < lower_threshold:
                led.value = True
                alarm = True
                print("On")

    if not button.value:
        audio.play(decoder)
        while audio.playing:
            pass
        print("Done playing!")
    if alarm and not supress_alarm:
        audio.play(decoder)
        while audio.playing:
            if not button.value:
                supress_alarm=True
                audio.stop()
        # comment the next line if the alarm shall run endless
        #supress_alarm=True
        print("Done alarm playing!")
```

```
time.sleep(1) # slow sampling to avoid flooding
```

From:

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

Permanent link:

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

Last update: **2023/08/21 16:28**

