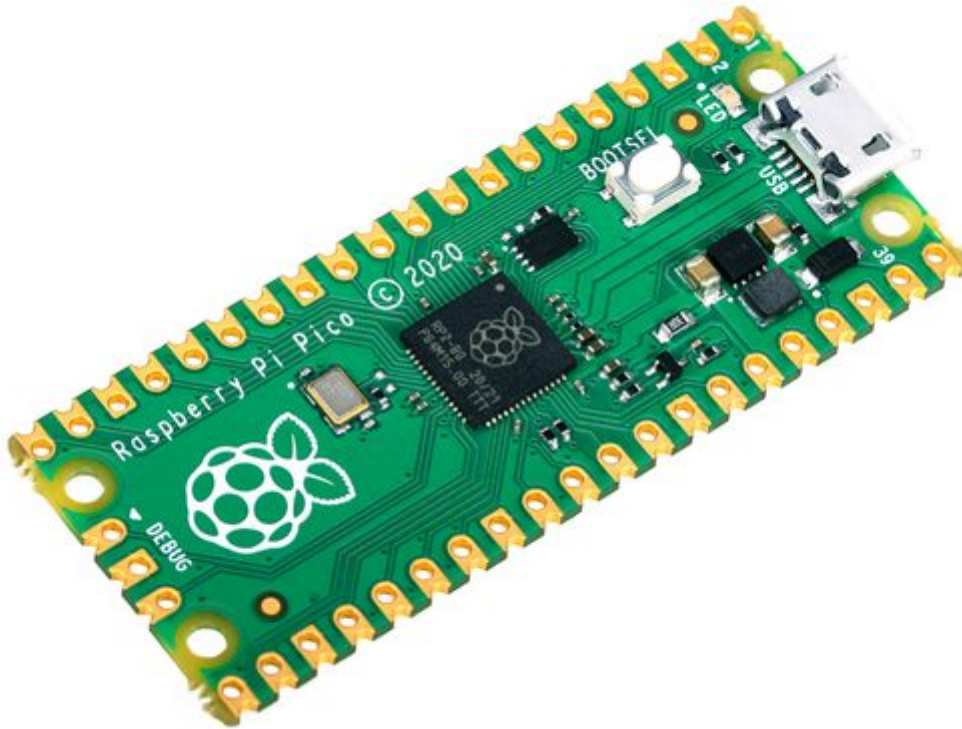


Raspberry Pi Pico: MP3-Audioausgabe mit DFPlayer Mini



Mit ein paar Tricks kann man einem Raspberry Pi Pico an seinen digitalen Ausgängen Töne entlocken. Das funktioniert natürlich nicht, wenn die Audio-Daten als MP3-, WAV- oder WMA-Dateien vorliegen. Die bekommt man natürlich ohne zusätzliche Hardware nicht abgespielt.

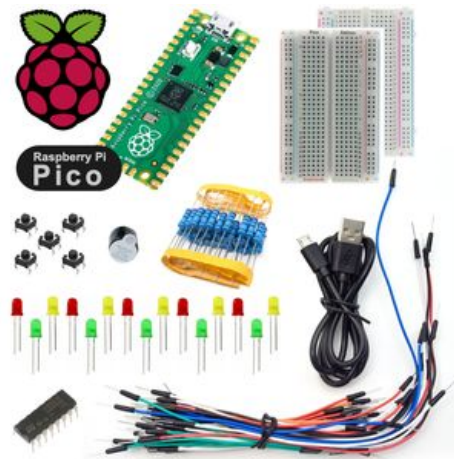
Hierfür eignen sich externe MP3-Player als Erweiterung, die durch den Raspberry Pi Pico gesteuert werden können. Besonders beliebt und am leichtesten steuern lässt sich der DFPlayer Mini, den man bedarfsweise einfach auch nur mit Tastern bedienen kann. Für den DFPlayer Mini braucht mal also nicht zwangsläufig einen Mikrocontroller.

Hier wollen wir einen DFPlayer Mini mit dem Raspberry Pi Pico verbinden und alle Audio-Dateien auf einer MicroSD-Karte nacheinander abspielen. Zur Audio-Wiedergabe dient ein 1-Watt/8-Ohm-Lautsprecher.

Wenn Du mit dem DFPlayer Mini noch keine Erfahrung gemacht hast, dann solltest Du Dich umfassend informieren. Beispielsweise, welche Dateien und welche Verzeichnisstruktur er auf der MicroSD-Karte erwartet und welche Audio-Dateien geeignet sind. Und auch, wie Du ihn ohne Mikrocontroller steuern kannst.

- [Mehr Informationen über DFPlayer Mini](#)

Hardware-nahes Programmieren mit dem Raspberry Pi Pico und MicroPython

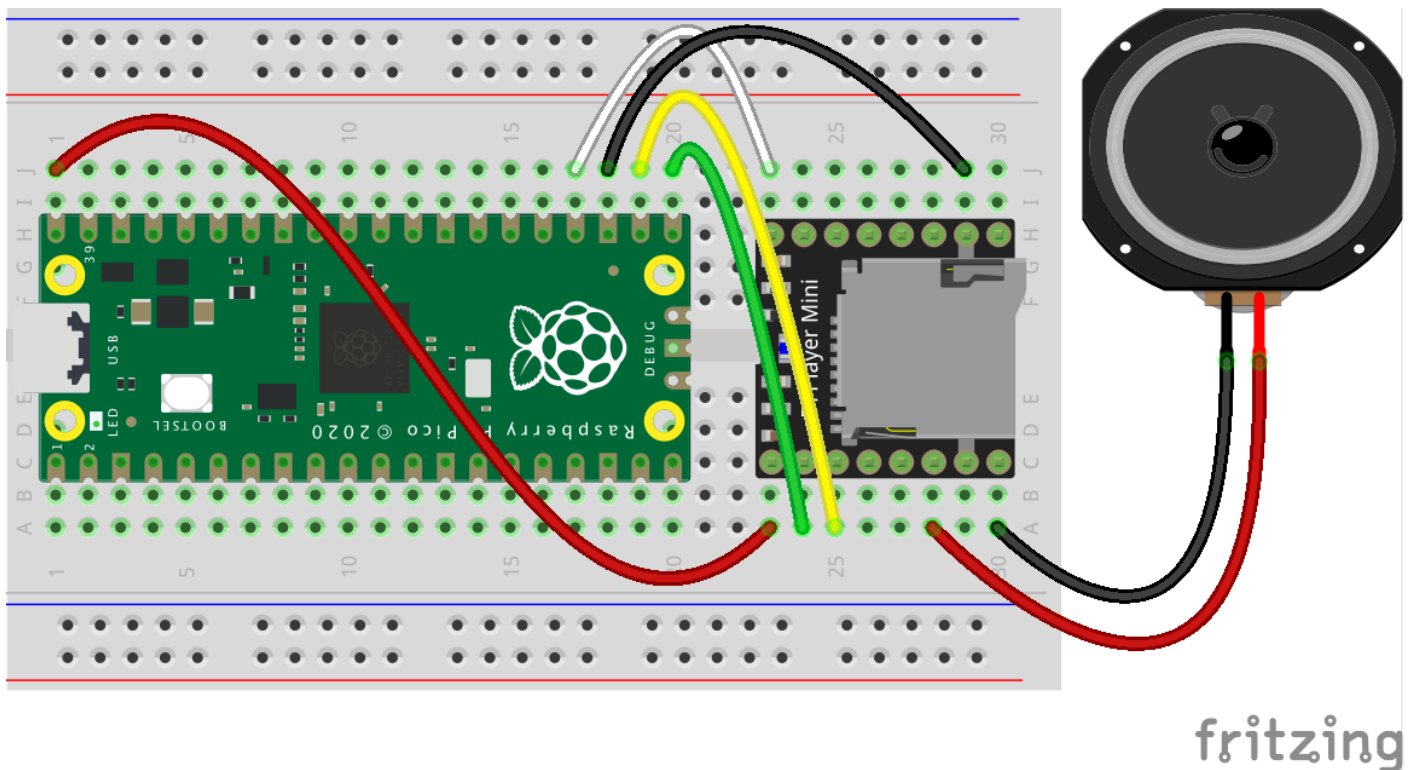


Das Elektronik-Set Pico Edition ist ein Bauteile-Sortiment mit Anleitung zum Experimentieren und Programmieren mit MicroPython.

- LED: Einschalten, ausschalten, blinken und Helligkeit steuern
- Taster: Entprellen und Zustände anzeigen
- LED mit Taster einschalten und ausschalten
- Ampel- und Lauflicht-Steuerung
- Elektronischer Würfel
- Eigene Steuerungen programmieren

[Elektronik-Set jetzt bestellen](#) [Online-Workshop buchen](#)

Aufbau und Bauteile



- [DFPlayer Mini](#)
- Speicherkarte

- Lautsprecher
- [GPIO-Belegung \(Pinout\)](#)

Raspberry Pi Pico		DFPlayer Mini		Lautsprecher
Pin 40	VCC / +5V	Pin 1	VCC	
Pin 24	GPIO18	Pin 16	Busy	
Pin 23	GND / 0 V	Pin 10	GND	
Pin 22	GPIO17 / UART0 / RX	Pin 3	TX	
Pin 21	GPIO16 / UART0 / RX	Pin 2	RX	
		Pin 6	SPK +	+
		Pin 8	SPK -	-

Schnell und einfach alle Bauteile zusammen bestellen

- [Jetzt alle Bauteile bestellen](#)
- [Eigenes Elektronik-Set konfigurieren](#)

MicroPython-Bibliothek für den DFPlayer

Die Software-seitige Ansteuerung erfolgt über die serielle Schnittstelle. Damit der DFPlayer Mini etwas tut, zum Beispiel einen Titel abspielen, muss er einen Code über die serielle Schnittstelle übermittelt bekommen. Hilfreich ist es, man verwendet eine Software-Bibliothek, die bestimmte Steuerungs-codes bereits mit einem Programmierbefehl kombiniert haben. So Dinge, wie Play, Pause, Laut, Leise, usw.

Die folgende MicroPython-Bibliothek muss heruntergeladen und mit dem Dateinamen „picodfplayer.py“ auf dem Raspberry Pi Pico gespeichert werden.

- [Dokumentation](#)
- [Download](#)

Programmcode

Der Programmcode initialisiert den DFPlayer Mini über eine UART-Schnittstelle und die Busy-Leitung. Über den Zustand der Busy-Leitung ermitteln wir im Programmcode, ob gerade ein Titel gespielt wird, oder nicht. Über die UART-Schnittstelle bekommt der DFPlayer Mini seine Steuerungs-codes gesendet.

Das DFPlayer Mini braucht nach der Verbindung mit der Stromversorgung etwa 500 bis 1.500 ms, bis er seine Speicherkarte vollständig indiziert hat. Wie lange das dauert, ist abhängig von der Anzahl der Dateien auf der Speicherkarte. Bei mehreren hundert Titeln kann es etwas dauern, bis der DFPlayer Mini Kommandos entgegennehmen kann. Aus diesem Grund ist nach der Initialisierung eine kleine Pause eingebaut.

Weil am Anfang die Lautstärke auf dem höchsten Wert eingestellt ist, wird die Lautstärke im

Programmcode zuerst auf einen mittleren Wert eingestellt. Anschließend bekommt der DFPlayer das Kommando zum Abspielen des Titels „001“ im Verzeichnis „01“. Die führenden Nullen werden im Programmcode weggelassen.

In der anschließenden Endlos-Schleife wird die Busy-Leitung abgefragt. Erst wenn der laufende Titel beendet ist, wird der nächste Titel abgespielt.

```
# Bibliotheken laden
from time import sleep
from picodfplayer import DFPlayer

# Initialisierung DFPlayer (UART, TX-Pin, RX-Pin, Busy-Pin)
player = DFPlayer(0, 16, 17, 18)
sleep(1)
player.setVolume(15) # Lautstärke einstellen: 0 bis 30

# Titel-Zähler
count = 1

print('Abspielen:', count)
# Verzeichnis: 01 / Datei: 001 (/01/001)
player.playTrack(1,1)

# Wiederholung: Endlos-Schleife
while True:
    sleep(5)
    # Wenn der Titel beendet ist
    if player.queryBusy() == False:
        count += 1
        print('Nächster Titel:', count)
        player.nextTrack()
```

Das Kommando „nextTrack“ führt in der Endlos-Schleife dazu, dass alle Titel nacheinander abgespielt werden.

Hinweis: Mit diesem Aufbau und Programmcode ist es nicht vorgesehen, das Abspielen der Audio-Dateien zu beenden oder zu unterbrechen. Das Beenden des Programmcodes hat keinen Einfluss auf den DFPlayer Mini. Der lässt den aktuellen Titel unabhängig vom Mikrocontroller weiterlaufen. Zum Beenden der laufenden Audioausgabe musst Du den Lautsprecher oder gleich die Stromversorgung (VCC) zum DFPlayer Mini entfernen/abziehen/abklemmen.