

Soundausgabe mit PCM5100A Modul

Moin Keule,

ich habe mich mit diesem Audio-Converter und -Amplifier auseinandergesetzt und nach drei Tagen des Versuchens, Umsteckens und einigem Frust endlich einen Ton rausbekommen. 😊

Folgende Dateiformate kannst du abspielen: .wav, .pcm und Frequenzen wie bei der 07.2_Alertor.

Es war nicht einfach, ohne eine Anleitung und ohne die Modulnamen zu wissen, wie man das Ganze zum Laufen bringt, aber jetzt hast du die Lösung! :D

Die folgenden Import-Befehle sollten als Standardbibliothek bereits vorhanden sein:

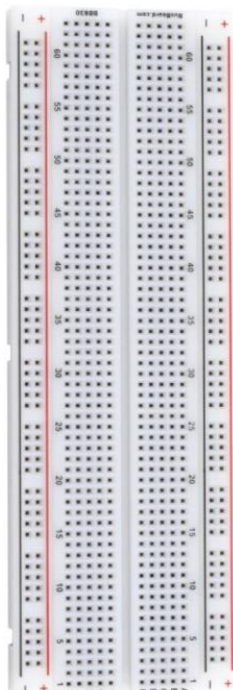
Falls du dennoch zusätzliche Bibliotheken installieren möchtest, kannst du dies in Thonny wie folgt tun:

1. Öffne Thonny und klicke in der Menüleiste auf „Tools“.
2. Wähle im Dropdown-Menü die Option „Manage packages“.
3. Im daraufhin geöffneten Fenster kannst du nach dem gewünschten Paket suchen und es installieren.

Am Ende habe ich dir meinen gesamten Code hier reingepackt, damit du ihn direkt nutzen kannst.

Nun lass uns das Modul mit dem Pico verbinden. Dafür brauchst du folgendes:

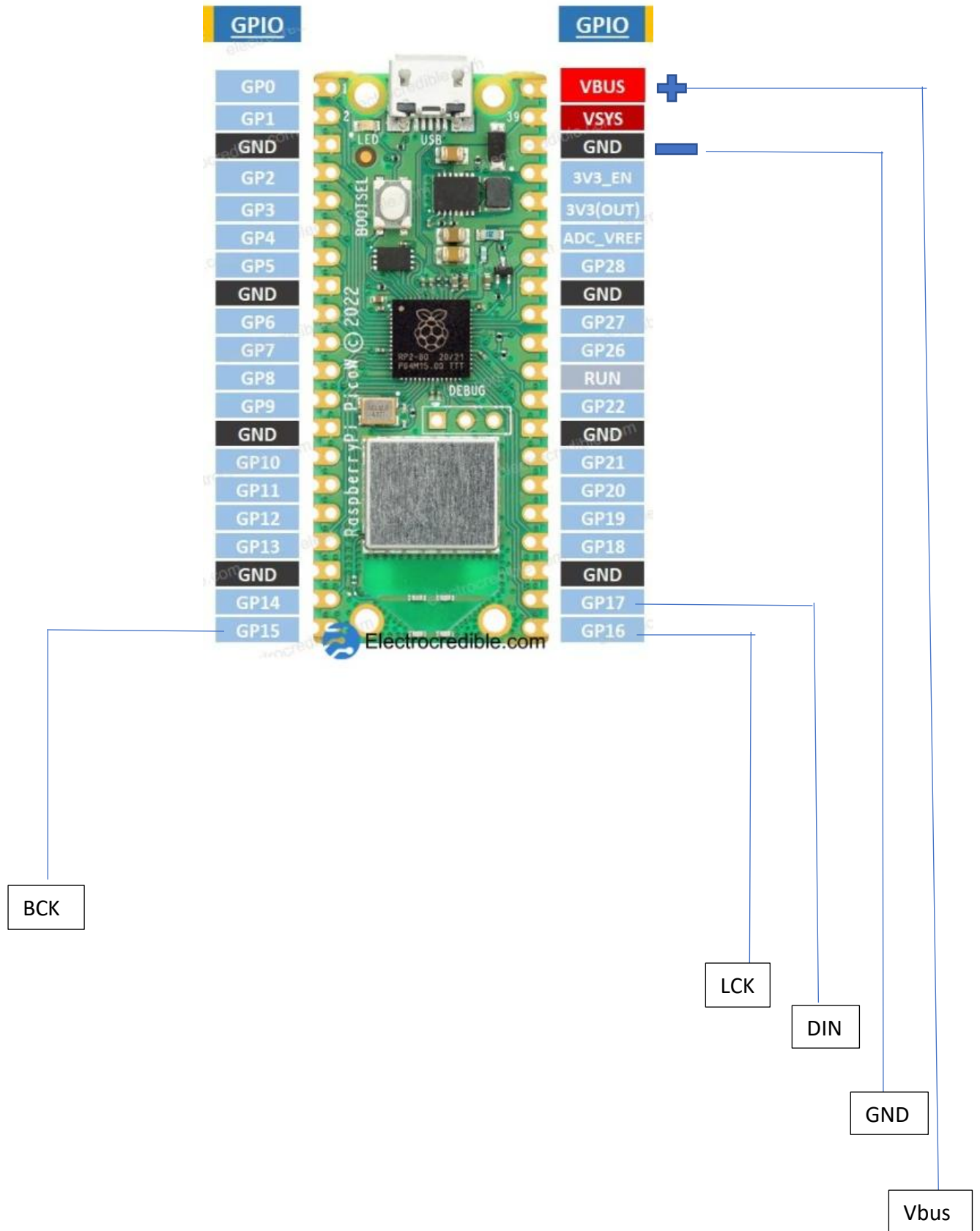
- Pico W
- Dein Modul
- 5 Männlich-Weiblich Kabel (Jumper)
- 2 x 8 Ohm, 2W Lautsprecher



Umut55



Datenblatt zum Anschluss des mitgelieferten Audio-Moduls an den Raspberry Pi Pico W



Nützliches zu wissen!

Die Abkürzungen BCK, LCK und DIN beziehen sich auf die Pins, die bei der I2S-Kommunikation (Inter-IC Sound) verwendet werden, welche häufig für die Audioübertragung genutzt wird. Hier eine kurze Erklärung der Begriffe:

1. Bit Clock Pin (BCK) – Pin 15
Dieser Pin sorgt für den Bit Clock, also den Takt, mit dem die einzelnen Bits der Audiodaten übertragen werden.
2. Word Select Pin (LCK) – Pin 16
Der Word Select Pin gibt an, ob die übertragenen Daten dem linken oder rechten Audiokanal zugeordnet sind. Er sorgt also dafür, dass der Wechsel zwischen den Kanälen (Stereo oder Mono) korrekt funktioniert.
3. Serial Data Pin (DIN) – Pin 17
Der Serial Data Pin ist für die Datenübertragung verantwortlich. Über ihn werden die Audiodaten vom Mikrocontroller an das Audio-Modul (z. B. einen DAC) gesendet.

Ich habe ein paar Codes für dich im Ordner: wavcode.py und pcmcode.py. Beachte aber, dass du WAV-Dateien erst komprimieren musst, damit der Speicher (2 MB) sie aufnehmen kann.

Hier sind ein paar nützliche Seiten, auf denen du Videos von YouTube herunterladen, schneiden und in das richtige Format umwandeln kannst:

- **Für das Umwandeln von YouTube-Videos in WAV:**
<https://notube.cc/de/youtube-app-84>
- **Zum Komprimieren von WAV-Dateien:**
<https://www.youcompress.com/de/wav/>
- **WAV-Datei in PCM umwandeln:**
<https://www.ezyzip.com/konvertiere-wav-nach-pcm.html>
- **Zum Schneiden der WAV-Datei:**
<https://www.veed.io/de-DE/werkzeuge/audio-schneider/wav-schneider>