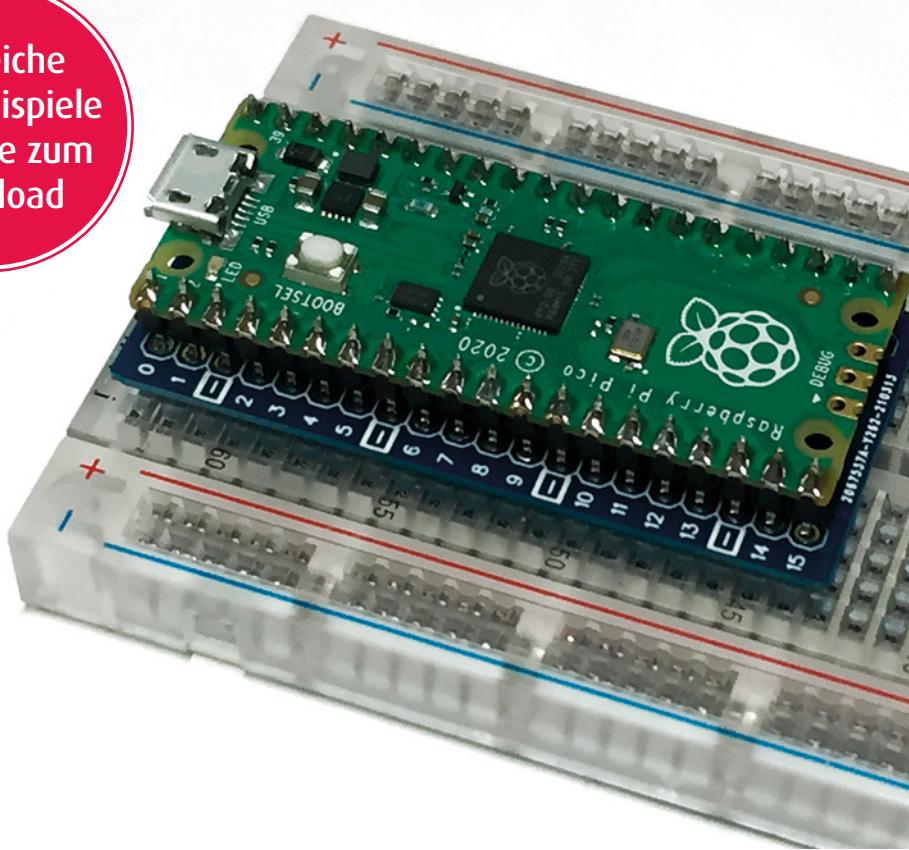


Thomas Brühlmann

# RASPBERRY PI PICO Schnelleinstieg

Kompakter Leitfaden für die Hardware  
Einfache Programmierung mit MicroPython

Zahlreiche  
Praxisbeispiele  
mit Code zum  
Download



# Inhalt

<b>Einführung .....</b>	<b>9</b>
-------------------------	----------

## 1 Das Board

1.1 Die Leiterplatte .....	13
1.2 Anschlüsse .....	16
1.3 Spannungsversorgung .....	20

## 2 Die Software

2.1 Thonny-Entwicklungsumgebung (IDE) .....	21
2.1.1 Installation von Thonny .....	21
2.1.2 Konfiguration von Thonny .....	22
2.2 MicroPython .....	24
2.2.1 MicroPython für Raspberry Pi Pico .....	25
2.2.2 Installation von MicroPython .....	26
2.2.3 MicroPython in Thonny IDE .....	28

## 3 Erste Schritte

3.1 Erstes Programm .....	31
3.2 Projekt »Blink LED« .....	33
3.3 Programmaufbau .....	35
3.4 Steckbrett und Elektronik .....	40
3.4.1 Steckbrett .....	40
3.4.2 Bauteile der Elektronik .....	44

## 4 Digitale Ein- und Ausgänge

4.1 Ein- und Ausgänge am Pico .....	51
4.2 Eingang einlesen .....	53
4.2.1 Pullup oder Pulldown .....	53
4.3 Praxisbeispiel: Taster einlesen und Status ausgeben .....	56
4.4 LED ansteuern .....	57
4.5 PWM .....	60
4.5.1 Praxisbeispiel: LED mit PWM .....	62

## Inhalt

---

4.6	Servo .....	63
4.6.1	Praxisbeispiel: Servo mit Potentiometer .....	64
4.7	Transistor, Relais .....	67
4.7.1	Transistor .....	67
4.7.2	Relais .....	70
4.8	Motor .....	73
4.8.1	Einfache Motor-Stufe (ein Motor) .....	73
4.8.2	Motor-Treiber für zwei Motoren .....	74
4.8.3	Praxisbeispiel: Motor-Ansteuerung (1 Motor) .....	75
4.8.4	Praxisbeispiel: Motor-Ansteuerung (2 Motoren) .....	78
4.8.5	Praxisbeispiel: Mini-Roboter .....	81

## 5 Analoge Welt

5.1	Spannung einlesen .....	85
5.1.1	Praxisbeispiel: Messung mit dem internen Temperatursensor ...	86
5.2	Praxisbeispiel: Poti mit LED-Ampel .....	87
5.3	Praxisbeispiel: Lichtmesser mit LDR .....	90
5.4	Praxisbeispiel: Temperaturmessung mit NTC .....	94

## 6 Anzeigen

6.1	RGB-LED .....	99
6.2	LC-Display (LCD) .....	102
6.3	OLED-Display .....	107
6.4	Projekt: Wetterstation mit Umweltsensor DHT22 .....	111

## 7 Schnittstellen

7.1	UART .....	117
7.1.1	Praxisbeispiel: Datenaustausch mit Arduino .....	118
7.2	I2C .....	122
7.2.1	I2C-Bus auf dem Pico .....	123
7.2.2	Definition I2C mit MicroPython .....	124
7.2.3	I2C-Scanner .....	125
7.3	Praxisbeispiel: Lichtmesser mit BH1750 .....	126

## 8 Programm-Erweiterungen

8.1	Bibliotheken .....	129
8.2	Programmable Input and Output (PIO) .....	131
8.2.1	Praxisbeispiel: Blink mit State Machine .....	132
8.2.2	Praxisbeispiel: Blinker als Alarmmelder .....	135

## 9 Pinout und Boards

9.1	Pico-Pinout und Beschreibung .....	139
9.2	Technische Daten .....	143
9.3	RP2040-Boards .....	144
9.3.1	Sparkfun .....	144
9.3.2	Arduino .....	145
9.3.3	Adafruit .....	146
9.3.4	PIMORONI .....	148
9.4	Hardware-Erweiterungen .....	149
9.4.1	Reset-Schalter .....	150
9.4.2	Pico-Pinout-Board für Steckbrett .....	151

## 10 Stücklisten .....

---

## Stichwortverzeichnis .....

153

159

# Einführung

Im Januar 2021 hat eine News-Meldung die Maker- und Bastlerszene überrascht. Die Raspberry Pi Foundation, also die Organisation, die den bisher bekannten Minicomputer Raspberry Pi entwickelt hat, präsentierte ein neues Mitglied in ihrer Produkte-Reihe. Willkommen Raspberry Pi Pico!

Die Vorstellung des neuen Raspberry Pi Pico, in diesem Buch in der Kurzform »Pico«, wurde unbemerkt im Hintergrund vorbereitet und nur wenige Hardware-Hersteller aus dem Maker-Umfeld waren informiert. Zu diesen Anbietern gehören Adafruit, Sparkfun, Arduino und Pimoroni. Diese Hardware-Firmen präsentieren mit der Vorstellung des Pico gleichzeitig eigene kompatible Boards und Lösungen. Diese Lösungen sind zum aktuellen Zeitpunkt (März 2021) noch in Entwicklung. Die bisher präsentierten Informationen zu den neuen Boards werden viele Maker, Bastler und Microcontroller-Board-Anwender erfreuen.

Dank der zeitlich geplanten Vorstellung des Pico ist das neue Microcontroller-Board bereits bei einzelnen Händlern und Online-Shops verfügbar. Abonnenten und Käufer der Ausgabe 39 der englischen Bastlerzeitschrift HackSpace (<http://hsmag.cc>) erhielten zusätzlich zur Zeitschrift ein Pico-Board. Leider war diese Aktion breit gefächert nur auf der englischen Insel verfügbar. Glückliche Besitzer haben sich vielleicht auch an einem Zeitschriftenshop auf einem Flughafen eine Ausgabe mit Board sichern können.

Da Sie dieses Buch gekauft haben, gehören Sie vermutlich auch zur Maker-Community und interessieren sich für das neue Board der Raspberry Pi Foundation und möchten gerne interaktive Anwendungen mit Schalter, Anzeigen, Motoren usw. realisieren. Möglicherweise haben Sie bereits erste Erfahrungen mit Microcontrollern wie Arduino gemacht.

Der neue Raspberry Pi Pico ist im Gegensatz zu seinen Geschwistern aus der Raspberry-Serie kein Minicomputer mit Betriebssystem, sondern ein kleines, kompaktes Microcontroller-Board, auf dem Programme in MicroPython oder C/C++ ausgeführt werden können.

Mit der Vorstellung des Pico sind die News aber noch nicht fertig erzählt. Neben dem neuen Microcontroller-Board gibt es auch einen neuen Microcontroller. Auf dem Pico wird ein eigener, von der Raspberry Pi Foundation entwickelter Chip eingesetzt – der Microcontroller hat die Bezeichnung RP2040.

Diese spannenden Neuigkeiten bringen wieder Schwung in die Maker-Szene. Ein neues Microcontroller-Board, das einen solch markanten Eindruck in der Szene und in den sozialen Medien hinterlässt, erscheint nicht täglich. Obwohl erst kurze Zeit seit der Präsentation des Pico vergangen ist, findet man auf den bekannten sozialen Plattformen wie Twitter, YouTube und Hackaday schon eine ganze Menge an nützlichen Informationen, Beispielen und Tutorials. Täglich kommen neue Projekte dazu und erweitern so die Möglichkeiten des kleinen Boards.

Dieser kompakte Guide soll Sie beim Einstieg ins Thema Raspberry Pi Pico und MicroPython unterstützen. Idealerweise haben Sie schon mit anderen Microcontroller-Boards gearbeitet und schon Grundkenntnisse in Python oder einer anderen Programmiersprache.

Das Buch ist so aufgebaut, dass Sie die Kapitel nacheinander durcharbeiten können.

In [Kapitel 1](#) wird der Raspberry Pi Pico vorgestellt, die Anschlüsse erklärt und die technischen Daten und Funktionen vorgestellt.

In [Kapitel 2](#) wird die Firmware der Programmiersprache MicroPython, eine kompakte und abgespeckte Version von Python, installiert. Anschließend wird die Entwicklungsumgebung Thonny installiert und konfiguriert. Nach dem Verbindungsauflauf zwischen der Entwicklungsumgebung und dem Pico steht die nötige Infrastruktur bereit.

Das erste Programm, im Hardware-Umfeld ein Blink-Programm, wird in [Kapitel 3](#) erstellt und auf den Pico geladen. Anschließend werden die Struktur der Programme und der Programmaufbau erklärt. Gleichzeitig werden die nötige Elektronik und Hardware, die für die nachfolgenden Beispiele verwendet werden, erläutert.

Das [Kapitel 4](#) beschreibt den Einsatz der digitalen Ein- und Ausgänge des Pico. In praktischen Beispielen werden die Zustände von Tastern eingelesen und Leuchtdioden, Servos und Relais angesteuert.

Die analoge Welt mit den integrierten Analog/Digital-Wandlern wird in [Kapitel 5](#) vorgestellt. Analoge Sensoren wie Fotowiderstand und Temperatursensor werden in Betrieb genommen.

In [Kapitel 6](#) werden Anzeige-Elemente eingesetzt. Zuerst wird eine Leuchtdiode als optisches Element über eine Dimm-Funktion angesteuert. Anschließend erklären Praxisbeispiele den Einsatz von LC- und OLED-Displays. Zum Schluss wird eine kleine Wetterstation realisiert.

Kapitel 7 beschreibt den Einsatz der seriellen Schnittstelle (UART) und des I2C-Bus. In einem Praxisprojekt wird ein Lichtmesser realisiert.

In Kapitel 8 werden Programmerweiterungen wie die State-Machine und Module und Bibliotheken beschrieben.

Kapitel 9 beinhaltet technische Themen wie die Beschreibung des Pinouts und der einzelnen Anschluss-Pins sowie technische Daten zum Pico und dem Microcontroller RP2040.

Alle Stücklisten zu den einzelnen Projekten werden in Kapitel 10 bereitgestellt.

## Weitere Informationen

Weitere Informationen zum Buch und zu den Projekten mit dem Raspberry Pi Pico sind auf meiner Website erhältlich:

<https://555circuitslab.com>

Die Beispielprogramme aus dem Buch können über meinen Github-Account bezogen werden:

<https://github.com/arduinopraxis>

Auf meiner Website findet man auch weitere Informationen zu meinen Microcontroller-Projekten wie auch Details zu meinen Büchern über die Themen Arduino und Sensoren.

Auf der Website des Verlages sind Informationen zu diesem Buch unter folgender Adresse verfügbar:

<https://mitp.de/0377>

## Kontakt zum Autor

Anregungen, Rückmeldungen und Fragen können Sie über Twitter oder per E-Mail an mich senden.

E-Mail: [maker@555circuitslab.com](mailto:maker@555circuitslab.com)

Twitter: <https://twitter.com/arduinopraxis>

## Danksagung

Ein großer Dank geht an meine Familie, meine Frau Aga und meine Jungs Tim und Nik. Auch bei diesem Buchprojekt haben sie mir wieder den nötigen Freiraum gegeben.

## **Einführung**

---

Herzlichen Dank an meine Lektorin Sabine Schulz für den unkomplizierten und schnellen Ablauf bei der Entstehung dieses Buchprojekts.

Im Mai 2021

Thomas Brühlmann

# Kapitel 1

## Das Board

In diesem Kapitel werden die Hardware des Raspberry Pi Pico, die Anschlüsse und die Spannungsversorgung beschrieben.

### 1.1 Die Leiterplatte

Der Raspberry Pi Pico ist das bisher kleinste Board der Raspberry-Pi-Familie und hat eine Größe von 50 x 20 mm. Der Pico wird als kleine Leiterplatte mit seitlichen Anschläßen geliefert. In Abbildung 1.1 ist der Raspberry Pi Pico zu sehen.

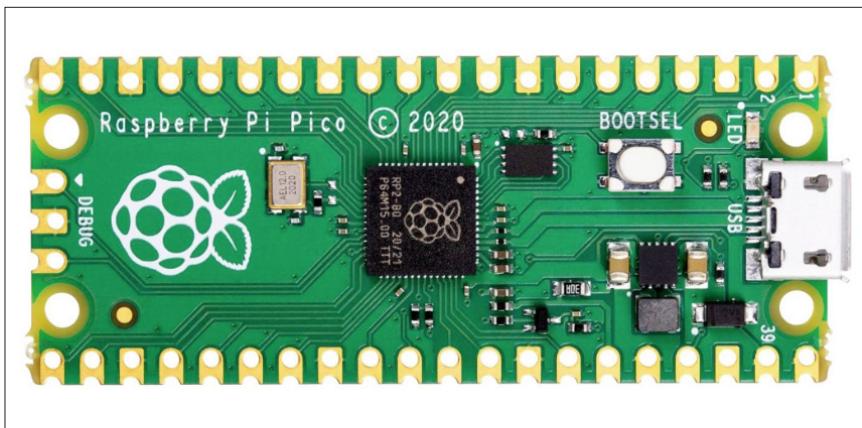


Abb. 1.1: Raspberry Pi Pico

Die wichtigsten Komponenten des Raspberry Pi Pico sind in Abbildung 1.2 farbig dargestellt.

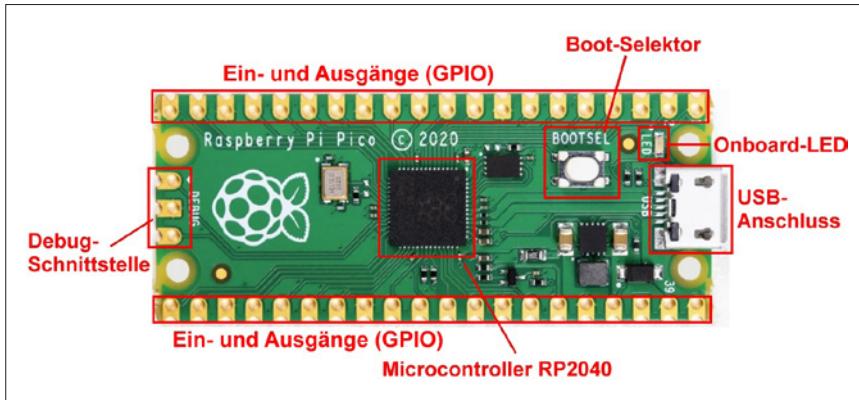


Abb. 1.2: Raspberry Pi Pico – Komponenten des Boards

### Microcontroller

Die Zentraleinheit, also das Gehirn des Pico, ist der Microcontroller vom Typ RP2040. Dieser Microcontroller wurde von der Raspberry Pi Foundation eigens für dieses neue Board entwickelt. Die technischen Daten zum Board und dem Microcontroller sind in [Kapitel 9](#) beschrieben.

Im Gegensatz zu den bisherigen Raspberry-Pi-Boards kann auf dem Pico kein Betriebssystem betrieben werden. Auf den Microcontroller wird eine Firmware geladen, die anschließend einzelne Programme ausführt. In diesem Buch wird der Einsatz von MicroPython beschrieben.

### Ein- und Ausgänge

Über die goldenen Anschlusspads oder Lötflächen auf den Längsseiten können die externen Bauteile angeschlossen werden. Dabei kann ein einzelner Anschlussdraht oder eine Stiftleiste angelötet werden.

In der Praxis lohnt sich das Anlöten einer 20-poligen Stiftleiste. Dabei werden die Anschluss-Pins von der Unterseite des Pico aufgesteckt und an der Oberseite der Leiterplatte angelötet. In [Abbildung 1.3](#) sind die Stiftleisten angelötet. Anschließend wurde der Pico auf ein Steckbrett aufgesteckt.

Das Anlöten von Stiftleisten für den Einsatz auf einem Steckbrett kennen Sie vielleicht schon von anderen Microcontroller-Boards wie dem Arduino Nano.

Die Anschlussbelegung des Pico ist im nachfolgenden Abschnitt beschrieben.

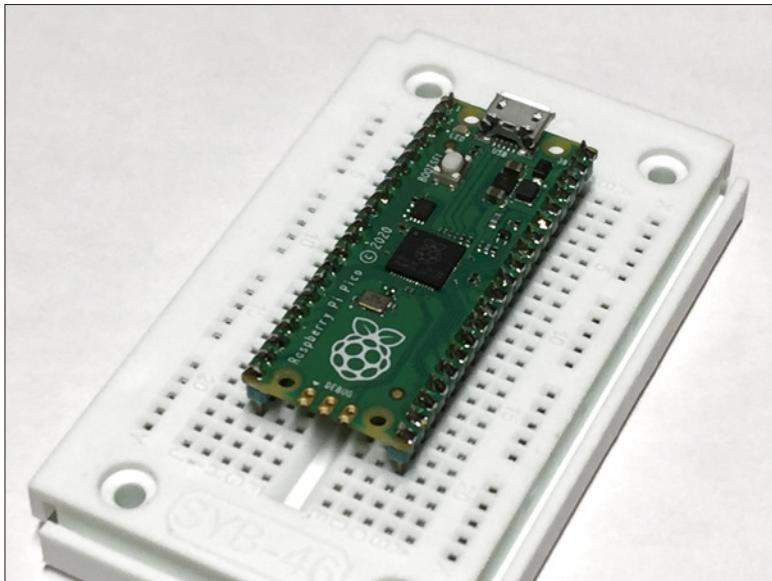


Abb. 1.3: Raspberry Pi Pico mit Stifteleisten auf Steckbrett

## USB-Anschluss

Mit dem Anschließen eines USB-Kabels an den USB-Anschluss (Micro-USB-Typ B) wird der Raspberry Pi Pico mit Spannung versorgt. Gleichzeitig erfolgt über diese USB-Verbindung der Datenaustausch vom Pico zum angeschlossenen Rechner. Dazu gehört auch das Hochladen von neuen Programmen auf den Pico.

## Onboard-LED

Die Onboard-LED ist intern am Ausgangspin GP25 des RP2040 angeschlossen und kann für Statusanzeigen verwendet werden.

## Boot-Selektor

Der Boot-Selektor ist ein Drucktaster zur Selektion des Startmodus. Der Boot-Selektor wird später noch verwendet und beschrieben.

## Debug-Schnittstelle

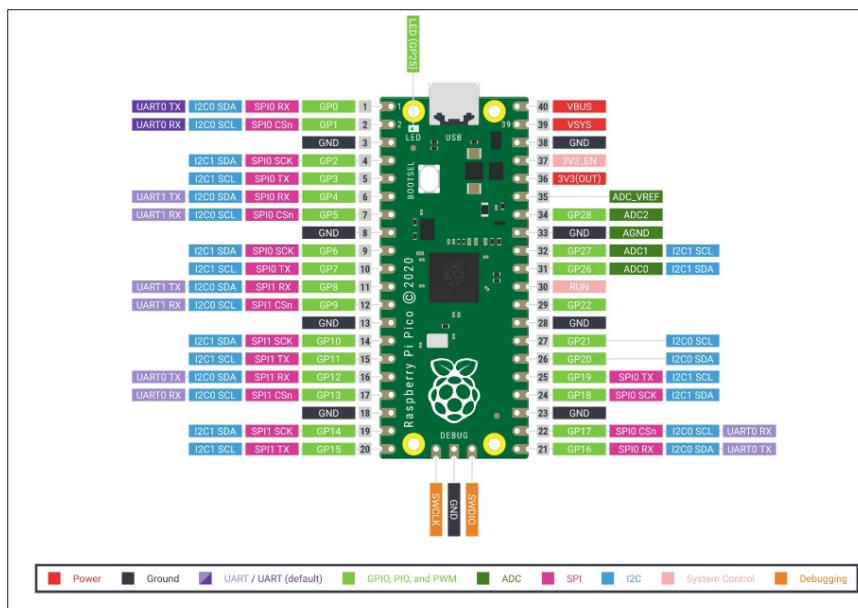
Die drei Anschluss-Pads mit der Bezeichnung DEBUG sind für die fortgeschrittenen Fehlersuche, das sogenannte Debugging, herausgeführt. Die Schnittstelle

wird als ARM-Serial-Wire-Debug-(SWD-)Schnittstelle bezeichnet. Die Verwendung dieser Schnittstelle ist in diesem Buch nicht beschrieben.

## 1.2 Anschlüsse

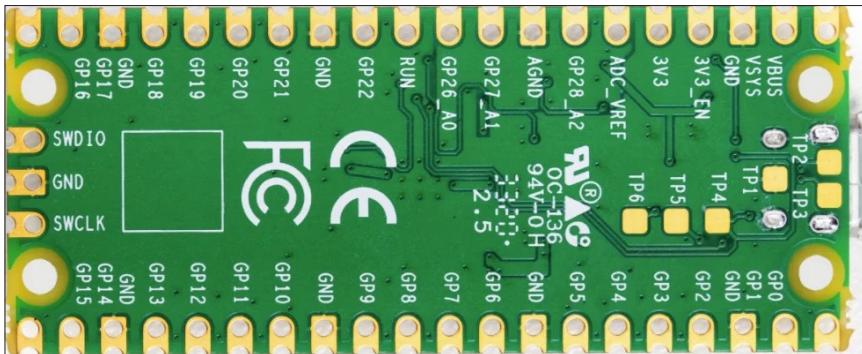
Die Anschluss-Pins des Pico sind an allen vier Seiten der Leiterplatte angeordnet. An den Längsseiten findet man alle Ein- und Ausgänge, um externen Komponenten wie Schalter, Taster, Leuchtdioden, Sensoren oder Anzeigen anzuschließen.

In Abbildung 1.4 ist die Anschlussbelegung (Pinout) mit den verschiedenen Funktionen der einzelnen Pins dargestellt. Die Funktionen werden zusätzlich mit einzelnen Farben gezeigt. Die Ein- und Ausgänge für übliche GPIO-Funktionen, also allgemeine Ein- und Ausgänge, sind hellgrün. Die vier analogen Eingänge sowie die interne Referenzspannung sind dunkelgrün markiert.



**Abb. 1.4:** Pico – Anschlussbelegung (Quelle: Website Raspberry Pi Foundation)

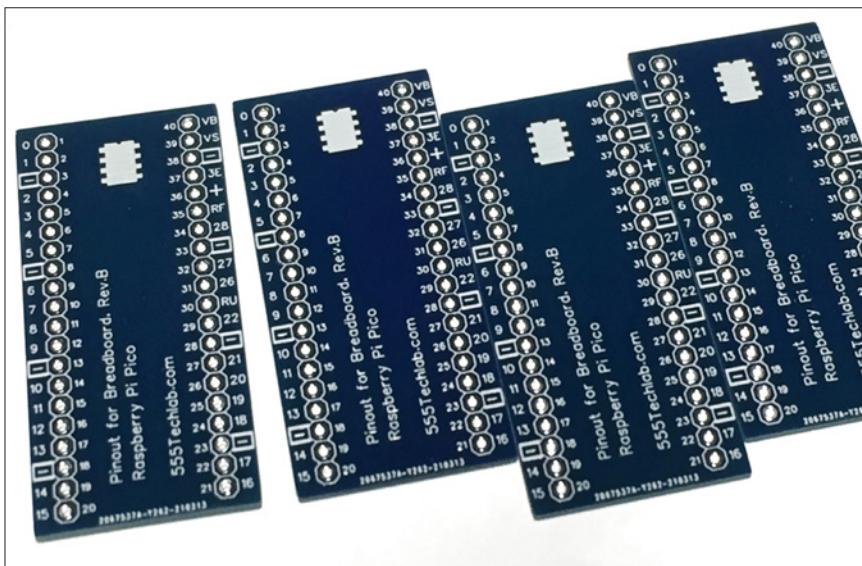
Durch die vielen Anschluss-Pins und die verschiedenen Funktionen muss beim Anstecken einer Drahtverbindung immer genau geprüft werden, ob man den richtigen Pin verwendet. Leider ist die Pinbelegung nur auf der Rückseite des Raspberry Pi Pico aufgedruckt ([Abbildung 1.5](#)).



**Abb. 1.5:** Anschlussbelegung auf Rückseite

Wie bereits im vorherigen Abschnitt erwähnt, empfiehlt es sich, zwei 20-polige Stifteleisten an den seitlichen Anschlusspads anzulöten.

Ich habe mir für die sichtbare Anzeige der Anschlussbelegung eine kleine Leiterplatte entwickelt. Diese kann von unten an die Stifteleiste angeschlossen werden ([Abbildung 1.6](#)).



**Abb. 1.6:** Leiterplatte mit Anschlussbelegung

### Stückliste (Anschluss-Pins)

- 1 Raspberry Pi Pico
- 2 Stifteleisten 1 x 20 Pin
- Lötzinn
- Lötkolben (30 bis 50 Watt)

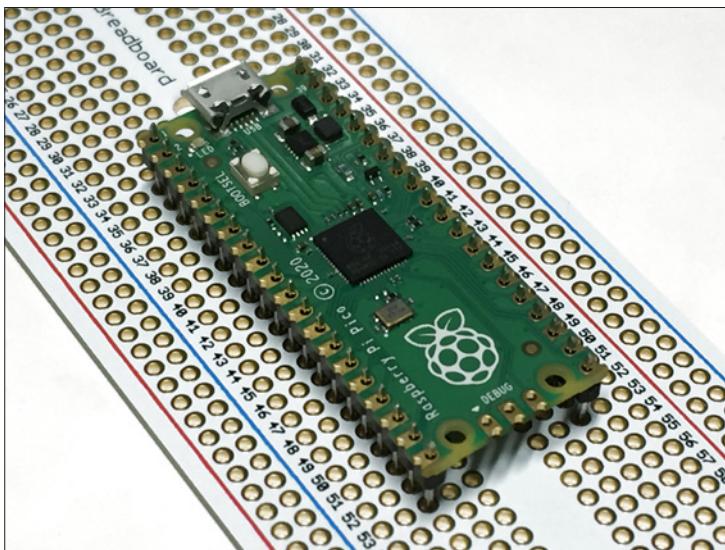
Im ersten Schritt wird das Board von oben auf die Stifteleisten gesteckt.

Als Hilfe für die Positionierung der Stifteleisten kann auch eine unbestückte Lochrasterplatine verwendet werden ([Abbildung 1.7](#)).

Anschließend lötet man je einen Pin der beiden Stifteleisten von oben an. Um die Stifteleiste gerade zu zentrieren, muss man allenfalls den Pin nochmals erhitzen und mit der Hand die Position korrigieren.

Bevor man nun alle Pins der Stifteleiste anlötet, kann man das Board auf ein Steckbrett platzieren und so prüfen, ob die Stifteleisten geradestehen.

Beim Anlöten der Stifteleiste dürfen die Lötpads nicht zu lange erhitzt werden, da andernfalls aufgelöste Bauteile auf der Leiterplatte des Pico durch die Hitze verschoben werden.



**Abb. 1.7:** Stifteleisten anlöten

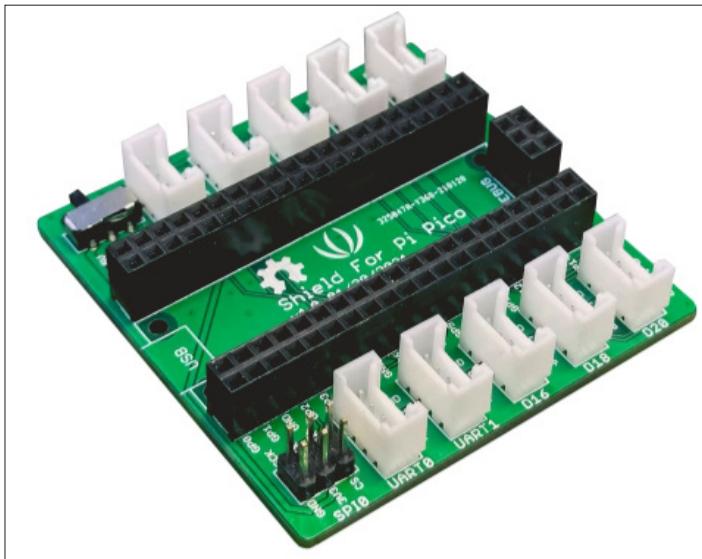


In vielen Tutorials und Anleitungen wird beschrieben, dass man die Stifteleisten am Pico anlöten soll, wenn diese im Steckbrett eingesteckt sind. Ich empfehle diese Variante nicht, da beim Löten Hitze entsteht, die den Kunststoff des Steckbretts angreifen kann. Falls man ein Steckbrett zur Verfügung hat, das nur für Lötarbeiten verwendet wird, kann diese Lösung trotzdem verwendet werden.

Für den praktischen Einsatz und für die Entwicklung von Anwendungen kann der Pico auch auf eine stabile Grundplatine gesteckt werden.

Zum aktuellen Zeitpunkt (März 2021) stehen aber erst wenige solcher Lösungen zur Verfügung. Als Beispiel kann das Grove Shield für Pi Pico von Seeed-studio erwähnt werden ([Abbildung 1.8](#)):

<https://www.seedstudio.com/Grove-Shield-for-Pi-Pico-v1-0-p-4846.html>



**Abb. 1.8:** Grove Shield für Pi Pico (Bild Seeedstudio)

Für meine Anwendungen habe ich ein kleines Protoboard mit Stiftleisten für den Pico sowie vielen freien Lötpads erstellt ([Abbildung 1.9](#)).



Abb. 1.9: Protoboard für Pico

Die Leiterplatte ist in meinem Shop verfügbar:

<https://www.tindie.com/products/23201/>

### 1.3 Spannungsversorgung

Auf dem Raspberry Pi Pico ist eine Spannungsversorgungsschaltung aufgebaut, die die internen 3,3 V generiert. Die Eingangsspannung für diese Versorgung kann dabei im Bereich von 1,8 V bis 5,5 V liegen.

Durch diesen großen Spannungsbereich kann der Raspberry Pi Pico zukünftig auch mit einem einzelnen Lipo-Akku oder über mehrere AA-Batterien versorgt werden.

Falls der Pico über eine externe Spannungsquelle versorgt wird, muss die Versorgungsspannung über den Anschluss VSYS zugeführt werden.

Für fortgeschrittene Anwender sind dazu im Datenblatt des Raspberry Pi Pico etliche Anwendungs-Beispiele beschrieben.

# Stichwortverzeichnis

## Symbolle

---

3,3-V-Logik .....	52
3V3_EN .....	143
3V3_OUT .....	142
380 V .....	70

## A

---

AA-Batterie .....	20
Adafruit .....	146
ADC_VREF .....	142
Adresse	
I2C-Bus .....	125
AGND .....	142
Aktor .....	51
Alarmmelder .....	136
Analog/Digital-Wandler .....	85
Analoges Signal .....	85
Anker .....	70
Anschlussbelegung .....	14
Anschlussdraht .....	43
Anschlüsse .....	13, 16
Anschluss-Pin .....	14
Ansteuerung	
Motor .....	73
Anzeige .....	99
Arduino .....	145
Arduino Nano RP2040 Connect ....	145
Ausgang .....	51

## B

---

Basis .....	67
Basis-Strom .....	68
Baud .....	119
BC546 .....	67
Bedieneinheit .....	84

Betriebssystem .....	14
BH1750 .....	126
Bibliothek .....	32, 129
Installation .....	129
Bitbanging .....	131
blink.py .....	35
Blockschaltbild .....	46
BOOTSEL .....	26
Boot-Selektor .....	15
Breadboard .....	40
Busadresse .....	123
I2C-Bus .....	125

## C

---

CircuitPython .....	147
Common-Anode .....	99

## D

---

Datenaustausch .....	119
Debug-Schnittstelle .....	15
DHT22 .....	111
DHT-Bibliothek .....	113
Drahtverbindung .....	43
Drehrichtung .....	73
Drehrichtungsänderung .....	81
Drehwinkel .....	64
DRV8871 .....	74
Dunkelheit .....	90
Durchlassrichtung .....	47
Duty Cycle .....	64

## E

---

Eingang .....	51, 53
Ein- und Ausgänge .....	14, 51
Elektronik .....	40
Bauteile .....	44

## Stichwortverzeichnis

---

Emitter ..... 67  
Endschalter ..... 84  
Entwicklungsumgebung ..... 10, 21  
Erstes Programm ..... 31

### F

---

Farad ..... 49  
Fast Mode ..... 123  
Fast Mode Plus ..... 123  
Feather RP2040 ..... 146  
Filesystem ..... 27  
Firmware ..... 10, 14, 21  
Flüssigkristall-Anzeige ..... 99  
    siehe LCD  
Fotowiderstand ..... 90  
Freilaufdiode ..... 71  
Frequenz ..... 61  
Funktion ..... 38, 129  
Funktionsbezeichnung ..... 139

### G

---

Geschwindigkeit ..... 75, 80  
GP25 ..... 15  
GPIO ..... 142  
Grundplatine ..... 19

### H

---

HackSpace ..... 9  
Hardware-Erweiterung ..... 149  
Hauptprogramm ..... 36  
HD44780 ..... 102

### I

---

I2C ..... 103, 122  
I2C-Bus ..... 106, 117, 122, 125  
I2C-Methode ..... 125  
I2C-Scanner ..... 125  
IC ..... 45  
IDE ..... 21  
Impuls ..... 60  
Impulszeit ..... 60

Input/Output-Controller ..... 131  
Integrierte Schaltung ..... 45  
    siehe IC  
Interactive Mode ..... 32  
Interpreter-Einstellungen ..... 29

### K

---

Kilohm ..... 46  
Kollektor ..... 67  
Kommentar ..... 36  
Kommunikationsschnittstelle ..... 117  
Kondensator ..... 49  
    Schaltzeichen ..... 49  
Kontakt ..... 70  
Koordinatensystem  
    OLED ..... 109

### L

---

Laststrom ..... 67  
LC-Display ..... 102  
    siehe LCD  
LCD ..... 99, 102  
LDR ..... 90  
    siehe Fotowiderstand  
LED ..... 47  
LED-Ampel ..... 87  
Leiterplatte ..... 13  
Leuchtdiode ..... 44, 47, 57, 99  
    Schaltzeichen ..... 48  
    Strom ..... 57  
Level-Shifter ..... 104, 119  
Lichtmessung ..... 90, 126  
Lichtsensor  
    BH1750 ..... 126  
Light Emitting Diode ..... 47  
    siehe LED  
Lipo-Akku ..... 20  
Lochrasterplatine ..... 18  
Logikpegel ..... 104

**M**

machine .....	32
machine.Pin() .....	53
main.py .....	35
Massenspeicher .....	27
Master/Slave-System .....	122
Megaohm .....	46
Messbereich	
NTC .....	98
Messsignal .....	87
Messwert .....	87
Microcontroller-Board .....	40
MicroPython .....	9, 14, 21, 24, 25
Grundstruktur .....	35
Installation .....	26
Interpreter .....	23
Mini-Steckbrett .....	41
Modellbau .....	63
Modus PULL_DOWN .....	54
Modus PULL_UP .....	53
Motor .....	73
Stromversorgung .....	75
Motor-Breakout-Board .....	73
Motorspannung .....	73
Motor-Treiber .....	74

**N**

Negativer Temperaturkoeffizient .....	94
NPN-Transistor .....	67
NTC .....	94
Messbereich .....	98
NTC-Sensor-Anwendung .....	96

**O**

Objekt .....	38
OLED .....	107
Bauform .....	107
Koordinatensystem .....	109
Onboard-LED .....	15, 51
Onboard-Leuchtdiode .....	32

**P**

Pegelwandler-Modul .....	104
siehe Level-Shifter	
Pico .....	9
siehe Raspberry Pi Pico	
PIMORONI .....	148
Pinout .....	16, 139
PIO .....	51, 129, 131
PNP .....	67
Port .....	24
Positionswert .....	66
Potentialtrennung .....	70
Potentiometer .....	62
Print-Relais .....	70
Programmable Input and	
Output .....	129, 131
siehe PIO	
Programmaufbau .....	35
Protoboard .....	19
Pulldown .....	53
Pull-Mode .....	55
Pullup .....	53
Pulsweitenmodulation .....	51
PWM .....	51, 60
Blöcke .....	61
Impuls .....	60
Pin .....	62
Servo .....	65
Signal .....	61, 64
Python .....	24

**R**

Raspberry Pi Foundation .....	9
Raspberry Pi Pico .....	9
Raspberry Pi .....	9
Relais .....	70
Relais-Schaltung .....	71
repl .....	32
Reset-Schalter .....	149
RGB-LED .....	99

## Stichwortverzeichnis

---

Richtung .....	80	Spannungsversorgung .....	20
Roboteranwendung .....	74	Sparkfun .....	74
Roboterfahrzeug .....	84	Spule .....	70
Roboter-Programm .....	81	State Machine .....	132
RP2040 .....	9, 14, 143, 144	Statusanzeige .....	57
RS-232 .....	117	Steckbrett .....	40
siehe UART		Steckbrett-Aufbau .....	45
RUN .....	142	Steinhart-Hart-Gleichung .....	96
RX .....	118	Stiftleiste .....	14
<b>S</b>		Stromlaufplan .....	44
Schalter .....	51	Raspberry Pi Pico .....	45
Schaltplan .....	44	Stromversorgung	
Schaltzeichen		Motor .....	75
Kondensator .....	49	Stückliste .....	153
Leuchtdiode .....	48	SWD .....	144
Transistor .....	67		
Widerstand .....	46	<b>T</b>	
Schnittstelle .....	40, 117	Taster .....	56
Schraubklemme .....	73	TB6612FNG .....	74
Sensor .....	51	Technische Daten .....	143
Serial Wire Debug .....	16	Temperaturkoeffizient .....	94
Serielle Schnittstelle .....	117	negativer .....	94
siehe UART		Temperaturmessung .....	86, 94
Servomotor .....	63	NTC .....	95
servo.py .....	65	Temperatur-Sensor	
SG90 .....	63	intern .....	86
Shell .....	22	Temperaturwert .....	39
Programm ausführen .....	32	Thermistor .....	94
Signal		siehe NTC	
analoges .....	85	Thonny .....	10
Signaltrennung .....	104	Installation .....	21
Single Wire Debug .....	144	Konfiguration .....	22
Software .....	21	Thonny IDE .....	21
Spannung		Tiny2040 .....	148
einlesen .....	85	Transistor .....	67
Spannungsquelle .....	20	Grundschaltung .....	68
Motor .....	75	Schaltzeichen .....	67
Spannungsteiler .....	91	TX .....	118

### U

---

UART .....	103, 117, 118
Überspannung .....	71
UF2-Datei .....	25
Umrechnungsfunktion .....	39
Umweltsensor .....	111
USB .....	15
USB-Anschluss .....	15
siehe USB	
USB-Kabel .....	26

### V

---

Variable .....	36
VBUS .....	143
Verbraucher .....	70

Vorwiderstand .....	45
---------------------	----

VSYS .....	143
------------	-----

### W

---

Wechselspannung .....	70
Wetterstation .....	111
Widerstand .....	46
Schaltzeichen .....	46

### Z

---

Zentraleinheit .....	14
siehe RP2040	
Zustand .....	51
Zustandsmaschine .....	132