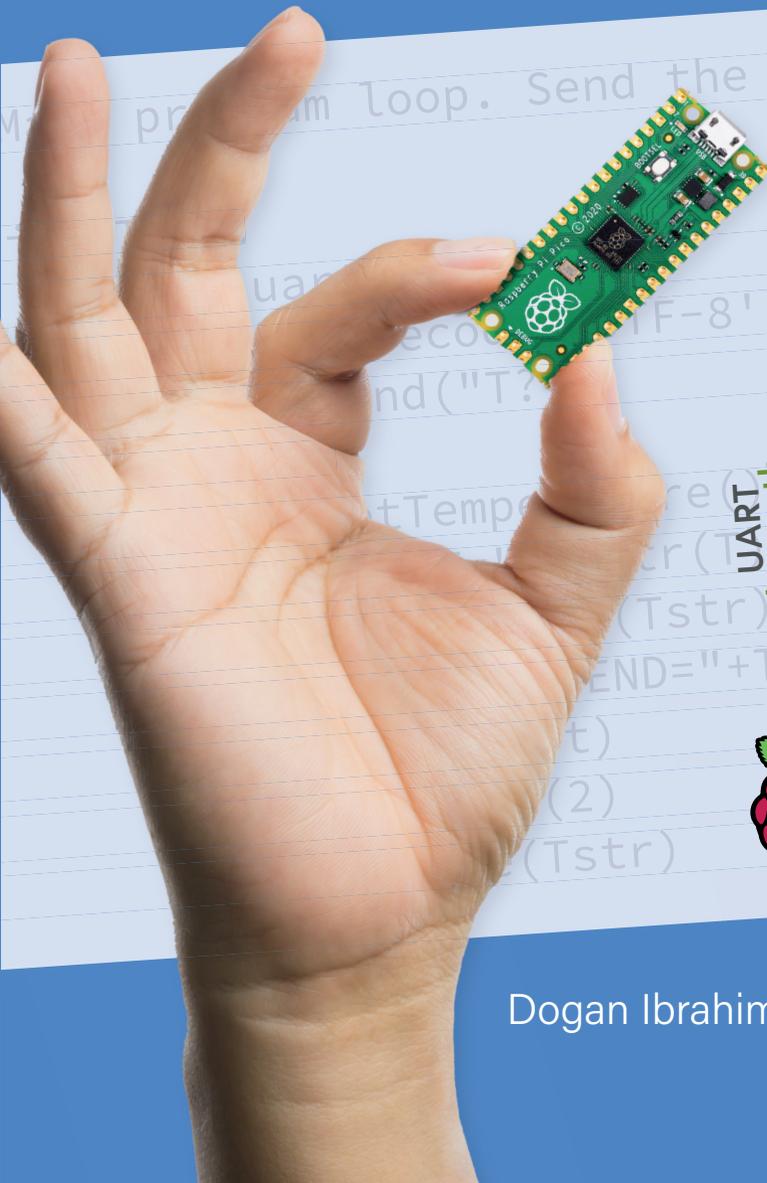


# Raspberry Pi Pico

Mehr als 50 Grundlagenprojekte mit  
MicroPython und dem RP2040-Mikrocontroller



GPIO  
Wi-Fi  
TMP102  
RS-232  
App  
Brushed-DC  
Hello World!  
Breadboard  
RP2040  
I<sup>2</sup>S  
EEPROM  
DAC & ADC  
7-segment  
BME-280  
Sensors  
LCD  
Autoren  
PWM  
LED



Dogan Ibrahim



---

# Raspberry Pi Pico

Mehr als 50 Grundlagenprojekte mit  
MicroPython und dem RP2040-Mikrocontroller



**Dogan Ibrahim**

---

● © 2021: Elektor Verlag GmbH, Aachen.

1. Auflage 2021

● Alle Rechte vorbehalten.

Die in diesem Buch veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen und Illustrationen sind urheberrechtlich geschützt. Ihre auch auszugsweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet.

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Die in diesem Buch erwähnten Soft- und Hardwarebezeichnungen können auch dann eingetragene Warenzeichen sein, wenn darauf nicht besonders hingewiesen wird. Sie gehören dem jeweiligen Warenzeicheninhaber und unterliegen gesetzlichen Bestimmungen.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autor können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für die Mitteilung eventueller Fehler sind Verlag und Autor dankbar.

● Erklärung

Autor, Übersetzer und Herausgeber haben sich nach besten Kräften bemüht, die Richtigkeit der in diesem Buch enthaltenen Informationen sicherzustellen. Sie übernehmen keine Haftung für Verluste oder Schäden, die durch Fehler oder Auslassungen in diesem Buch verursacht werden, unabhängig davon, ob diese Fehler oder Auslassungen auf Fahrlässigkeit, ein Versehen oder eine andere Ursache zurückzuführen sind und lehnen hiermit jegliche Haftung gegenüber Dritten ab.

Umschlaggestaltung: Elektor, Aachen

Übersetzung: Andreas Riedenauer

Satz und Aufmachung: D-Vision, Julian van den Berg | Oss (NL)

Druck: Ipskamp Printing, Enschede, Niederlande

● **ISBN 978-3-89576-456-1** Print  
**ISBN 978-3-89576-457-8** eBook

Elektor-Verlag GmbH, Aachen

[www.elektor.de](http://www.elektor.de)

Elektor ist Teil der Unternehmensgruppe Elektor International Media (EIM), der weltweit wichtigsten Quelle für technische Informationen und Elektronik-Produkte für Ingenieure und Elektronik-Entwickler und für Firmen, die diese Fachleute beschäftigen. Das internationale Team von Elektor entwickelt Tag für Tag hochwertige Inhalte für Entwickler und DIY-Elektroniker, die über verschiedene Medien (Magazine, Videos, digitale Medien sowie Social Media) in zahlreichen Sprachen verbreitet werden. [www.elektor.de](http://www.elektor.de)

---

<b>Vorwort</b> .....	<b>9</b>
<b>Kapitel 1 • Raspberry Pi Pico Hardware</b> .....	<b>11</b>
1.1 Überblick .....	11
1.2 Pico-Hardware-Modul .....	11
1.3 Vergleich mit dem Arduino UNO .....	13
1.4 Betriebsbedingungen und Spannungsversorgung des Pico .....	14
1.5 Pinbelegung des Mikrocontrollers RP2040 und des Pico-Moduls .....	15
1.6 Andere RP2040 Mikrocontroller-basierte Boards .....	17
1.6.1 Adafruit Feather RP2040 .....	17
1.6.2 Adafruit ItsyBitsy RP2040 .....	17
1.6.3 Pimoroni PicoSystem .....	18
1.6.4 Arduino Nano RP2040 Connect .....	18
1.6.5 SparkFun Thing Plus RP2040 .....	19
1.6.6 Pimoroni Pico Explorer Basis .....	19
1.6.7 SparkFun MicroMod RP2040-Prozessor .....	20
1.6.8 SparkFun Pro Micro RP2040 .....	20
1.6.9 Pico-RGB-Tastatursockel .....	21
1.6.10 Pico Omnibus .....	21
1.6.11 Pimoroni Pico VGA Demo Base .....	22
<b>Kapitel 2 • Raspberry Pi Pico Programmierung</b> .....	<b>23</b>
2.1 Überblick .....	23
2.2 Installation von MicroPython auf dem Pico .....	23
2.2.1 Raspberry Pi 4 zur Unterstützung der Installation von MicroPython auf dem Pico .....	23
2.2.2 Installation von MicroPython auf dem Pico unter Verwendung eines PCs (Windows 10) .....	29
<b>Kapitel 3 • Raspberry Pi Pico einfache Hardware-Projekte</b> .....	<b>48</b>
3.1 Überblick .....	48
3.2 Projekt 1 - Blinkende LED - Verwendung der on-board LED .....	48
3.3 Projekt 2 - Externe blinkende LED .....	51
3.4 Projekt 3 - SOS blinken .....	53
3.5 Projekt 4 - Blinkende LED - Verwendung eines Timers .....	55
3.6 Projekt 5 - Abwechselnd blinkende LEDs .....	56

3.7 Projekt 6 - Ändern der LED-Blinkrate mit Tasterinterrupts . . . . .	58
3.8 Projekt 7 - RGB: rote, grüne und blaue LED blinken im Wechsel . . . . .	63
3.9 Projekt 8 - Zufällig blinkende rote, grüne und blaue LEDs - RGB . . . . .	66
3.10 Projekt 9 - Rotierende LEDs . . . . .	67
3.11 Projekt 10 - Binärer Zähler mit LEDs . . . . .	70
3.12 Projekt 11 - X-MAS Lights (8 zufällig blinkende LEDs) . . . . .	74
3.13 Projekt 12 - Elektronischer Würfel . . . . .	75
3.14 Projekt 13 - Glückstag der Woche . . . . .	79
3.15 Projekt 14 - Tür Alarm mit 7-farbig blinkender LED . . . . .	82
3.16 Projekt 15 - 2-stellige 7-Segment-Anzeige . . . . .	85
3.17 Projekt 16 - 4-stelliger Sekundenzähler mit 7-Segment-Anzeige . . . . .	95
3.18 LCDs . . . . .	100
3.19 Projekt 17 - LCD-Funktionen - Anzeige von Text . . . . .	102
3.20 Projekt 18 - Sekundenzähler - LCD . . . . .	107
3.21 Projekt 19 - Reaktions-Timer mit LCD . . . . .	108
3.22 Projekt 20 - Ultraschall-Abstandsmessung . . . . .	111
3.23 Projekt 21 - Größe einer Person (Stadiometer) . . . . .	115
3.24 Projekt 22 - Ultraschall-Rückwärtseinkaufshilfe mit Summer . . . . .	117
<b>Kapitel 4 • Verwendung von Analog-Digital-Wandlern (ADC) . . . . .</b>	<b>120</b>
4.1 Überblick . . . . .	120
4.2 Projekt 1 - Spannungsmesser . . . . .	120
4.3 Projekt 2 - Temperaturmessung - Verwendung des internen Temperatursensors . . . . .	122
4.4 Projekt 3 - Temperaturmessung - mit einem externen Temperatursensor . . . . .	123
4.5 Projekt 4 - EIN/AUS-Temperaturregler . . . . .	125
4.6 Projekt 5 - EIN/AUS-Temperaturregler mit LCD . . . . .	128
4.7 Projekt 6 - Messung der Umgebungs-Helligkeit . . . . .	131
4.8 Projekt 7 - Ohmmeter . . . . .	133
4.9 Projekt 8 - Innen- und Außentemperatur . . . . .	136
4.10 Projekt 9 - Verwendung eines Thermistors zur Temperaturmessung . . . . .	138
<b>Kapitel 5 • Datenaufzeichnung . . . . .</b>	<b>143</b>
5.1 Überblick . . . . .	143
5.2 Projekt 1 - Protokollierung der Temperaturdaten . . . . .	143

---

5.3 Projekt 2 - Lesen der protokollierten Daten. . . . .	145
<b>Kapitel 6 • Impulsbreitenmodulation (PWM) . . . . .</b>	<b>147</b>
6.1 Überblick . . . . .	147
6.2 Grundlegende Theorie der Pulsweitenmodulation. . . . .	147
6.3 PWM-Kanäle des Raspberry Pi Pico . . . . .	149
6.4 Projekt 1: Erzeugen eines 1000-Hz-PWM-Signals mit 50 % Tastverhältnis . . . . .	150
6.5 Projekt 2: Ändern der Helligkeit einer LED . . . . .	151
6.6 Projekt 3 - Drehzahleinstellung eines Gleichstrom-Bürsten-Motors . . . . .	152
6.7 Projekt 4 - Frequenzgenerator mit LCD . . . . .	154
6.8 Projekt 5 - Messung der Frequenz und des Tastverhältnisses eines PWM-Signals. . . . .	156
6.9 Projekt 6 - Melodiegeber . . . . .	157
<b>Kapitel 7 • Serielle Kommunikation (UART). . . . .</b>	<b>161</b>
7.1 Überblick . . . . .	161
7.2 Raspberry Pi Pico UART serielle Schnittstellen . . . . .	163
7.3 Projekt 1 - Senden der internen Temperatur des Pico an den Arduino Uno . . . . .	164
7.4 Projekt 2 - Empfangen und Anzeigen von Zahlen vom Arduino Uno . . . . .	168
7.5 Projekt 3 - Kommunikation mit dem Raspberry Pi 4 über die serielle Schnittstelle . . . . .	170
<b>Kapitel 8 • Das I2C Bus Interface . . . . .</b>	<b>174</b>
8.1 Überblick . . . . .	174
8.2 Der I2C-Bus . . . . .	174
8.3 I2C-Pins des Raspberry Pi Pico . . . . .	175
8.4 Projekt 1 - I2C-Port-Erweiterung . . . . .	176
8.5 Projekt 2 - EEPROM-Speicher . . . . .	181
8.6 Projekt 3 - Temperatursensor TMP102 . . . . .	187
8.7 Projekt 4 - Temperatur- und Atmosphärendrucksensor BMP280 . . . . .	193
8.8 Projekt 5 - Anzeige von Temperatur und Luftdruck des BMP280 auf dem LCD . . . . .	200
<b>Kapitel 9 • Die SPI-Bus-Schnittstelle. . . . .</b>	<b>203</b>
9.1 Überblick . . . . .	203
9.2 Raspberry Pi Pico SPI-Anschlüsse . . . . .	204
9.3 Projekt 1 - SPI Port-Erweiterung . . . . .	205
<b>Kapitel 10 • Verwendung von Wi-Fi mit dem Raspberry Pi Pico . . . . .</b>	<b>211</b>
10.1 Überblick . . . . .	211

10.2 Projekt 1 - Steuerung einer LED über Wi-Fi vom Smartphone aus . . . . .	211
10.3 Projekt 2 - Anzeige der Innentemperatur auf einem Smartphone über Wi-Fi . . . . .	217
<b>Kapitel 11 • Bluetooth mit dem Raspberry Pi Pico. . . . .</b>	<b>222</b>
11.1 Überblick . . . . .	222
11.2 Raspberry Pi Pico Bluetooth-Schnittstelle . . . . .	222
11.3 Projekt 1 - LED-Ansteuerung mittels Smartphone via Bluetooth . . . . .	222
11.4 Projekt 2 - Senden der Raspberry Pi Pico-Innentemperatur an das Smartphone . . . . .	227
<b>Kapitel 12 • Verwendung von Digital-Analog-Wandlern (DACs) . . . . .</b>	<b>230</b>
12.1 Überblick . . . . .	230
12.2 Der MCP4921 DAC . . . . .	230
12.3 Projekt 1 - Erzeugen eines Rechtecksignals mit einer Amplitude kleiner als +3,3V . . . . .	231
12.4 Projekt 2 - Erzeugen von Festspannungen . . . . .	236
12.5 Projekt 3 - Erzeugen eines Sägezahnsignals . . . . .	238
12.6 Projekt 4 - Erzeugen eines Dreieckssignals . . . . .	240
12.7 Projekt 5 - Arbiträre periodische Wellenform . . . . .	242
12.8 Projekt 6 - Sinusgenerator . . . . .	244
12.9 Projekt 7 - Erzeugen eines genauen Sinussignals mit Hilfe von Timer-Interrupts . . . . .	247
<b>Kapitel 13 • Automatischer Programmstart nach dem Booten des Pico . . . . .</b>	<b>250</b>
<b>Anhang A • Komponentenliste . . . . .</b>	<b>252</b>
<b>Stichwortverzeichnis . . . . .</b>	<b>254</b>

## Vorwort

Traditionell werden Computer mit einem Mikroprozessor und vielen externen Chips aufgebaut. Ein Mikroprozessor enthält die Zentraleinheit (CPU), die Arithmetik- und Logikeinheit (ALU) sowie die Zeitgeber- und Steuerschaltungen und ist als solcher allein nicht einsatzfähig. Er muss durch viele externe Bauteile wie Speicher, Eingabe-Ausgabe-Geräte, Timer, Interrupt-Schaltungen usw. unterstützt werden, bevor daraus ein brauchbarer Computer wird. Ein Nachteil dabei ist die große Anzahl an Bauteilen, was Design und Verdrahtung komplex macht und zu hohem Stromverbrauch führt.

Ein Mikrocontroller hingegen ist im Grunde ein Ein-Chip-Computer, der eine CPU, Speicher, Eingangs-/Ausgangsschaltungen, Timer, Interrupt-Schaltungen, Taktgeber und verschiedene andere Module enthält, die alle in einem einzigen Siliziumchip untergebracht sind. Frühe Mikrocontroller waren in ihrer Kapazität und Geschwindigkeit sehr begrenzt und verbrauchten relativ viel Strom. Die meisten der frühen Mikrocontroller hatten 8-Bit-CPU's mit Taktgeschwindigkeiten im Bereich von einigen MHz und mit nur einigen Hundert Bytes Programm- und Datenspeicher. Diese Mikrocontroller wurden traditionell in ihrer jeweiligen Assembler-Sprache programmiert. 8-Bit-Mikrocontroller sind immer noch weit verbreitet, besonders in kleinen Projekten, bei denen große Speichermengen oder hohe Geschwindigkeiten nicht die Hauptanforderungen sind. Die Weiterentwicklung der Chiptechnologie ermöglicht heute jedoch 32- und 64-Bit-Mikrocontroller mit Geschwindigkeiten im Bereich von mehreren GHz und mit mehreren GB Speicher. Programme für Mikrocontroller werden heutzutage in einer Hochsprache wie C, C#, BASIC, PASCAL, JAVA, etc. erstellt.

Der Raspberry Pi Pico ist ein leistungsstarkes Mikrocontroller-Board, das speziell für das Physical Computing - also hardwarenahen Einsatz - entwickelt wurde. Sie sollten sich darüber im Klaren sein, dass sich Mikrocontroller stark von Einplatinencomputern wie dem Raspberry Pi 4 (und anderen Familienmitgliedern des Raspberry Pi) unterscheiden. Auf dem Raspberry Pi Pico befindet sich kein Betriebssystem. Mikrocontroller wie der Raspberry Pi Pico können so programmiert werden, dass sie eine einzelne Aufgabe ausführen, und sie können in schnellen Anwendungen wie Echtzeit-Steuerung und Überwachung eingesetzt werden. Der Raspberry Pi Pico basiert auf dem mit bis zu 133 MHz sehr schnellen und effizienten Dual-Core Arm Cortex-M0+ RP2040 Mikrocontroller-Chip und verfügt mit 264KB SRAM und 2MB über große Speicher-Ressourcen. Was den Raspberry Pi Pico weiterhin sehr attraktiv macht, sind seine vielen GPIO-Pins, zahlreiche populäre Peripherie-Schnittstellen wie SPI, I2C, UART, PWM, schnelle und genaue Timing-Module sowie die geringen Kosten. Während dieses Buch entstand, lagen diese bei etwa € 4,-.

Programme für den Raspberry Pi Pico kann man leicht in einigen der gängigen Hochsprachen wie MicroPython oder C/C++ erstellen. Im Internet gibt es viele Anwendungshinweise, Tutorials und Datenblätter zur Verwendung des Raspberry Pi Pico.

Dieses Buch ist eine Einführung in die Verwendung des Raspberry Pi Pico unter Verwendung der Programmiersprache MicroPython. Die Thonny-Entwicklungsumgebung (IDE) wird in allen Projekten des Buchs verwendet, und auch Ihnen wird diese IDE empfohlen. Es gibt über 50 funktionierende und getestete Projekte im Buch, die fast alle Aspekte des Raspberry Pi Pico abdecken.

Die folgenden Unterüberschriften werden für jedes Projekt angegeben, um es leicht nachvollziehbar zu machen:

- Titel
- Kurze Beschreibung
- Ziel
- Blockschaltbild
- Stromlaufplan
- Programm-Listing mit vollständiger Beschreibung

Ich hoffe, dass Ihre nächsten Mikrocontroller-Projekte den Raspberry Pi Pico verwenden und dieses Buch bei der Entwicklung Ihrer Projekte nützlich wird.

***Dr. Dogan Ibrahim***

London, 2021

## Kapitel 1 • Raspberry Pi Pico Hardware

### 1.1 Überblick

Raspberry Pi Pico ist ein Mikrocontroller-Modul, das von der Raspberry Pi Ltd. entwickelt wurde. In diesem Kapitel werden wir uns mit den Hardware-Details des Raspberry Pi Pico Mikrocontroller-Moduls näher beschäftigen. Von nun an werden wir dieses Modul kurz „Pico“ nennen.

### 1.2 Pico-Hardware-Modul

Der Pico ist mit ca. € 4,- ein sehr preiswertes Mikrocontrollermodul, das auf dem RP2040-Mikrocontrollerchip mit Dual-Cortex-M0+ Prozessor basiert. Bild 1.1 zeigt die Vorderansicht des Pico. In der Mitte der Platine befindet sich der winzige 7 x 7 mm große RP2040-Mikrocontroller-Chip, der in einem QFN-56-Gehäuse untergebracht ist. An den beiden Rändern der Platine befinden sich 40 goldfarbene Metall-GPIO (General-Purpose-Input-Output) Pins mit Löchern. Die Löcher sind gekennzeichnet, beginnend mit der Nummer 1 in der linken oberen Ecke der Platine. Die Nummern nehmen - entgegen dem Uhrzeigersinn - links von oben nach unten und rechts von unten nach oben zu, bis zur Nummer 40 in der rechten oberen Ecke, wenn die Platine mit der USB-Buchse nach oben zeigt. Die Platine ist Breadboard-kompatibel (d.h. 0,1 Zoll Pin-Abstand) und kann, nach dem Einlöten von Stiftleisten, auf ein Breadboard gesteckt werden, um die GPIO-Pins mit Hilfe von Jumper-Drähten einfach anzuschließen. Neben diesen Löchern befinden sich kreisrunde Aussparungen, die auf andere Module aufgesteckt werden können, ohne dass diese mit physischen Pins versehen sind.



Bild 1.1 Vorderansicht des Pico-Hardwaremoduls.

An einem Ende der Platine befindet sich der Micro-USB-B-Anschluss zur Stromversorgung und zur Programmierung. Neben dem USB-Anschluss befindet sich eine LED, die während der Programmentwicklung frei genutzt werden kann, daneben wiederum der Taster BOOTSEL, der während der Programmierung des Mikrocontrollers benötigt wird, wie wir in den nächsten Kapiteln sehen werden. Am anderen Ende des Boards, neben dem Raspberry Pi-Logo, befinden sich 3 Anschlüsse, die zum Debuggen Ihrer Programme dienen.

Bild 1.2 zeigt die Rückansicht des Pico-Hardwaremoduls. Hier sind alle GPIO-Pins mit Buchstaben und Zahlen gekennzeichnet:

GND	-	Masse der Spannungsversorgung (digitale Masse)
AGND	-	Masse der Spannungsversorgung (analoge Masse)
3V3	-	+3.3 V Spannungsversorgung (Ausgang)
GP0 – GP22	-	Digitale GPIOs
GP26_A0 – GP28_A2	-	Analoge Eingänge
ADC_VREF	-	ADC Referenz-Spannung
TP1 – TP6	-	Testpunkte
SWDIO, GND, SWCLK	-	Debug-Schnittstelle
RUN	-	Standard-RUN-Pin. LOW anschließen, um den RP2040 zurückzusetzen
3V3_EN	-	dieser Pin schaltet die +3,3V-Versorgung ein; kann durch Verbinden dieses Pins mit LOW deaktiviert werden.
VSYS	-	System-Eingangsspannung (1,8V bis 5,5V), die vom On-Board-SMPS verwendet wird, um die +3,3V-Versorgung für die Karte zu erzeugen
VBUS	-	Micro-USB-Eingangsspannung (+5V)

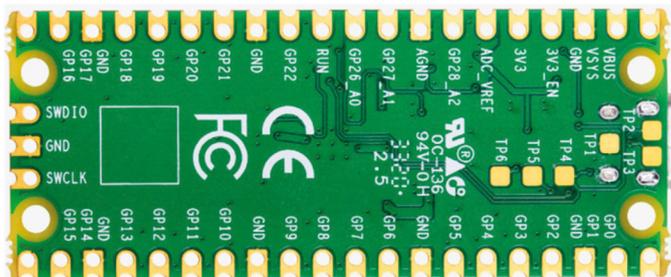


Bild 1.2 Rückansicht des Pico-Hardwaremoduls.

Einige der GPIO-Pins werden für interne Board-Funktionen verwendet:

GP29 (Eingang)	-	dient im ADC-Modus (ADC3) zur Messung von VSYS/3
GP25 (Ausgang)	-	Verbunden mit der Anwender-LED
GP24 (Eingang)	-	VBUS Sense – HIGH, falls VBUS aktiv, sonst LOW
GP23 (Ausgang)	-	Steuert den SMPS Power Save Pin

Die Spezifikationen des Pico-Hardwaremoduls:

- 32-Bit RP2040 Cortex-M0+ Dual-Core-Prozessor, 133 MHz Taktfrequenz
- 2 MB Q-SPI-Flash-Speicher
- 264 KB SRAM-Speicher
- 26 GPIO (+3,3V kompatibel)
- 3 x 12-Bit ADC-Pins
- Serial Wire Debug (SWD)-Anschluss
- Micro-USB-Anschluss (USB 1.1) für Stromversorgung (+5V) und Daten (Programmierung)
- 2 x UART, 2 x I2C, 2 x SPI-Bus-Schnittstelle
- 16 x PWM-Kanäle
- 1 x Timer (mit 4 Alarm-Optionen), 1 x Echtzeituhr (RTC)
- Temperatursensor
- LED (an Anschluss GP25)
- MicroPython, C, C++ Programmierung
- Drag & Drop-Programmierung mit Speicherstick über USB

Die GPIO-Hardware des Pico ist +3,3V-kompatibel. Daher muss beachtet werden, dass diese Spannung nicht überschritten wird, wenn externe Geräte an die GPIO-Pins angeschlossen werden. Wenn Komponenten mit +5V-Ausgängen an (als Eingänge konfigurierte) Pico-GPIO-Pins angeschlossen werden sollen, müssen Pegelwandler von +5V auf +3,3V oder ohmsche Spannungsteiler verwendet werden.

Bild 1.3 zeigt einen ohmschen Spannungsteiler zur Absenkung von +5V auf +3,3V.

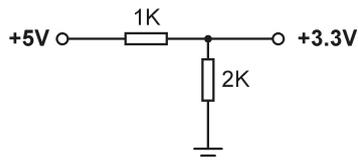


Bild 1.3 Widerstands-Spannungsteiler.

### 1.3 Vergleich mit dem Arduino UNO

Arduino UNO ist bei Studenten, Ingenieuren und Hobbyisten eines der beliebtesten Mikrocontroller-Entwicklungsboards. Tabelle 1.1 zeigt einen Vergleich zwischen dem Pico und dem Arduino UNO. Die Tabelle zeigt, dass der Pico viel schneller ist als der Arduino UNO, einen größeren Flash- und Datenspeicher hat sowie mehr digitale Eingangs-/Ausgangs-Pins und einen On-Board-Temperatursensor. Der Arduino UNO wiederum arbeitet mit +5V und seine GPIO-Pins sind +5V-kompatibel. Weitere Vorteile des Arduino UNO sind sein interner EEPROM-Speicher und sein ADC mit 6 Kanälen anstelle von 3 beim Pico.

Merkmal	Pico	Arduino UNO
Mikrocontroller	RP2040	Atmega328P
CPU	Dual core, 32-bit, Cortex-M0+	Single-Core, 8-Bit, AVR
RAM	264KB	2 KB
Flash	2MB	32KB
CPU-Takt	48MHZ bis 133MHZ	16MHZ
EEPROM	-	1K
Spannungseingang (Vcc)	+5V über USB-Port	+5V über USB-Port
Alternative Vcc	2 - 5V über VSYS-Pin	7 - 12V
MCU-Betriebsspannung	+3,3V	+5V
GPIO-Anzahl	26	20
ADC-Anzahl (Kanäle)	3	6
Hardware UART	2	1
Hardware I2C	2	1
Hardware SPI	2	1
Hardware PWM	16	6
Programmiersprachen	MicroPython, C, C++	C (Arduino IDE)
Eingebaute LED	1	1
Kosten	\$4	\$20

Tabelle 1.1 Vergleich von Pico und Arduino UNO.

## 1.4 Betriebsbedingungen und Spannungsversorgung des Pico

Die empfohlenen Betriebsbedingungen für den Pico sind:

- Betriebstemperatur: -20°C bis +85°C
- VBUS-Spannung: +5V ±10%
- VSYS-Spannung: +1,8V bis +5,5V

Ein on-board SMPS wird verwendet, um die +3,3V zur Versorgung des RP2040 aus einer Eingangsspannungen von 1,8V bis +5,5V zu erzeugen. Zum Beispiel können 3 AA-Batterien +4,5V für die Stromversorgung des Pico bereitstellen.

Der Pico kann auf mehrere Arten mit Strom versorgt werden. Die einfachste Methode ist, den Micro-USB-Anschluss mit einer +5V-Stromquelle zu verbinden, wie z.B. dem USB-Anschluss eines Computers oder einem +5V-Adapter. Dadurch wird der VSYS-Eingang (siehe Bild 1.4) über eine Schottky-Diode mit Strom versorgt. Die Spannung am VSYS-Eingang ist also die VBUS-Spannung abzüglich des Spannungsabfalls der Schottky-Diode (ca. +0,4V). Die VBUS- und VSYS-Pins können kurzgeschlossen werden, wenn die Platine über einen externen +5V-USB-Anschluss versorgt wird. Dadurch wird die Eingangsspannung leicht erhöht und somit die Restwelligkeit an VSYS reduziert. Die VSYS-Spannung wird durch den RT6150 an den SMPS geleitet, der konstante +3,3 Volt für die MCU und andere Teile des

Boards erzeugt. VSYS wird durch 3 geteilt und steht am analogen Eingangs-Port GPIO29 (ADC3) zur Verfügung, wo sie leicht zu messen ist. GPIO24 zeigt das Vorhandensein der VBUS-Spannung durch logischen HIGH-Pegel an.

Eine andere Methode, den Pico mit Strom zu versorgen, besteht im direkten Anschluss einer externen Spannung (+1,8V bis +5,5V) an den VSYS-Eingang (z.B. Batterien oder ein externes Netzteil). Wir können auch den USB-Eingang und die VSYS-Eingänge zusammen verwenden, um den Pico mit Strom zu versorgen, z. B. um sowohl mit Batterien als auch mit dem USB-Anschluss zu arbeiten. Wenn diese Methode verwendet wird, muss eine Schottky-Diode am VSYS-Eingang eingesetzt werden, um zu verhindern, dass sich die Versorgungen gegenseitig stören. Die höhere der beiden Spannungen wird VSYS versorgen.

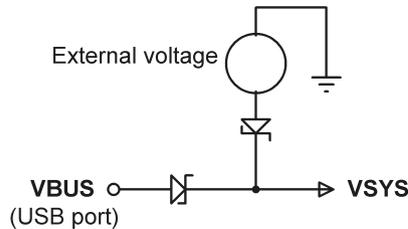


Bild 1.4 Stromversorgung des Pico.

### 1.5 Pinbelegung des Mikrocontrollers RP2040 und des Pico-Moduls

Bild 1.5 zeigt die RP2041 des RP2040 Mikrocontrollers, der in einem 56-Pin-Gehäuse untergebracht ist. Die Pinbelegung des Pico-Moduls ist in Bild 1.6 im Detail dargestellt. Wie man sieht, haben die meisten Pins mehrere Funktionen. Zum Beispiel ist GPIO0 (Pin 1) auch der UART0 TX, der I2C0 SDA und der SPI0 RX Pin.

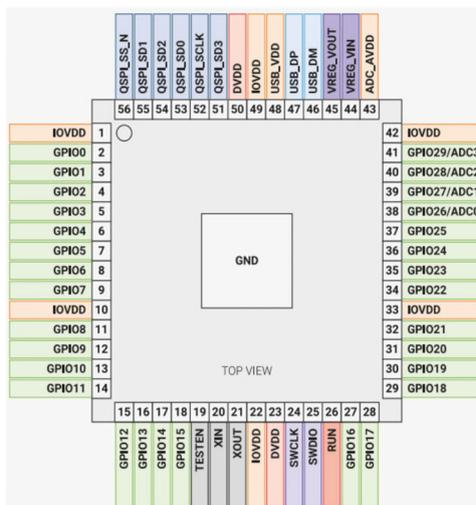


Bild 1.5 Pinbelegung des RP2040 Mikrocontrollers.

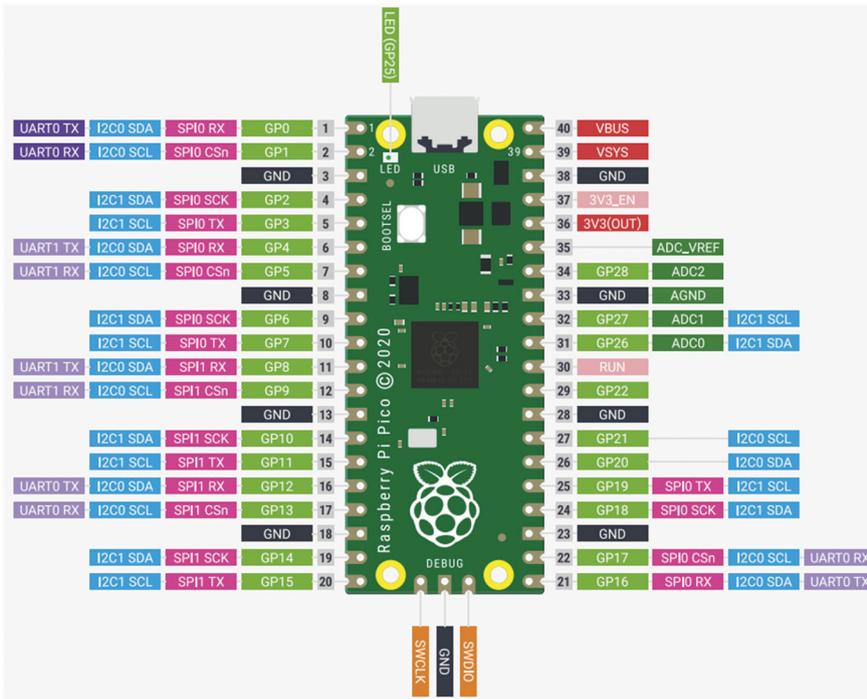


Bild 1.6 Pico-Pinbelegung.

Bild 1.7 zeigt ein vereinfachtes Blockdiagramm des Pico-Hardwaremoduls. Beachten Sie, dass die GPIO-Pins direkt mit dem Mikrocontroller-Chip verbunden sind. GPIO 26-28 können entweder als digitale GPIOs oder als ADC-Eingänge verwendet werden. Die ADC-Eingänge GPIO26-29 haben Sperrdioden auf 3V und daher darf die Eingangsspannung 3V3 + 300mV nicht überschreiten. Wenn der RP2040 nicht mit Strom versorgt wird, ist zu beachten, dass an die GPIO-Pins 26 bis 29 angelegte Spannungen durch die Diode zur Stromversorgung hinfließen (es gibt kein Problem mit den anderen GPIO Pins und Spannung kann dort auch angelegt werden, wenn der RP2040 nicht mit Strom versorgt wird).

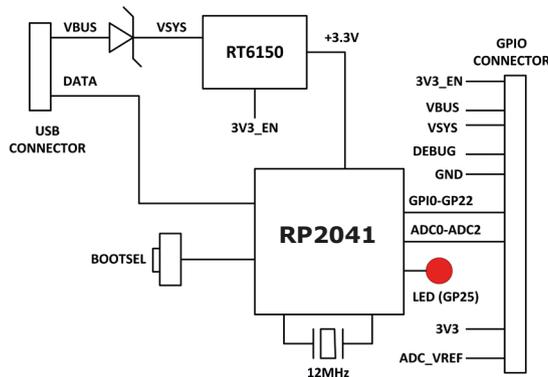


Bild 1.7 Vereinfachtes Blockschaltbild.

## 1.6 Andere RP2040 Mikrocontroller-basierte Boards

Während der Erstellung dieses Buches haben einige Dritthersteller Mikrocontroller-Platinen entwickelt, die ebenfalls auf dem RP2040 Chip basieren. Hier einige Beispiele:

### 1.6.1 Adafruit Feather RP2040

Dieses Mikrocontroller-Board (Bild 1.8) hat folgende Eigenschaften:

- RP2040 32-bit Cortex-M0+ mit 125MHz
- 4MB Flash-Speicher
- 264KB RAM
- 4 x 12-Bit ADC
- 2 x I2C, 2 x SPI, 2 x UART
- 16 x PWM
- 200mA Lipoly-Ladegerät
- Reset- und Bootloader-Tasten
- 24MHz-Quarz
- +3,3V-Regler mit 500mA Stromausgang
- USB-Typ-C-Anschluss
- Rote LED auf der Platine
- RGB-NeoPixel
- On-Board STEMMA QT-Anschluss mit optionalem SWD-Debug-Port

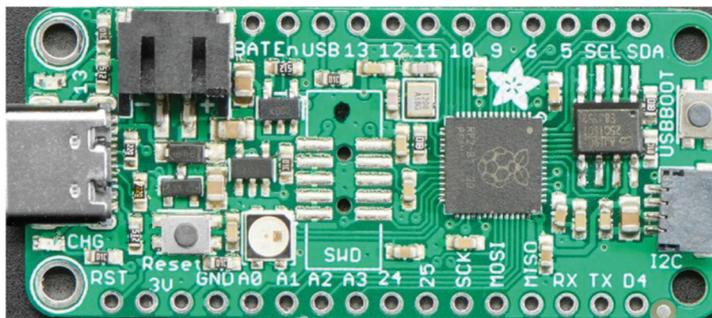


Bild 1.8 Adafruit Feather RP2040.

### 1.6.2 Adafruit ItsyBitsy RP2040

ItsyBitsy RP2040 (Bild 1.9) ist ein weiteres RP2040-basiertes Mikrocontroller-Board von Adafruit. Seine Eigenschaften sind denen des Feather RP2040 sehr ähnlich. Es hat einen USB-Micro-B-Anschluss und einen +5V-Ausgang.

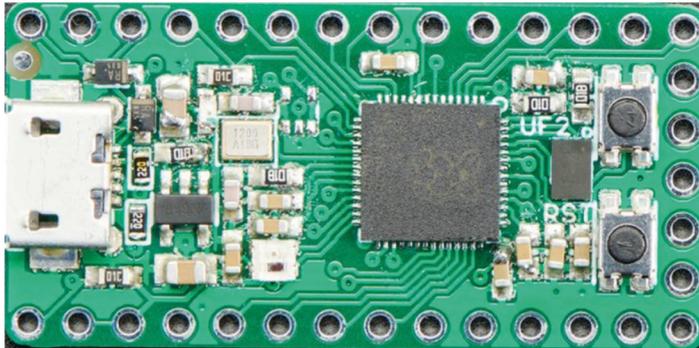


Bild 1.9 Adafruit ItsyBitsy RP2040.

### 1.6.3 Pimoroni PicoSystem

Dies ist ein Mini-Gaming-Board (Bild 1.10), das um den RP2040-Mikrocontroller herum entwickelt wurde. Seine grundlegenden Eigenschaften sind:

- 133MHz Taktfrequenz
- 264KB SRAM
- LCD-Bildschirm
- Joypad
- Taster
- Lipo-Akku
- USB-C-Stromanschluss



Bild 1.10 Pimoroni PicoSystem.

### 1.6.4 Arduino Nano RP2040 Connect

Dieses Board (Bild 1.11) mit 16 MB Flash, 9-Achsen-IMU und Mikrofon sowie einem sehr effizienten Leistungsteil und ist mit Wi-Fi/Bluetooth ausgestattet.

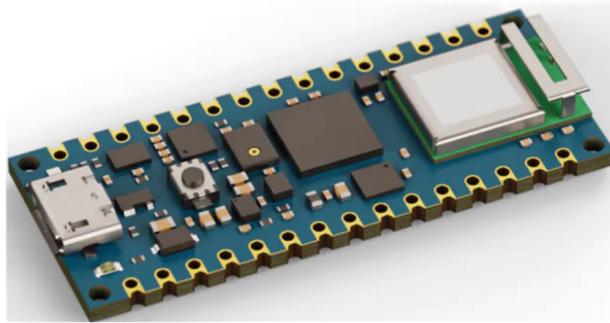


Bild 1.11 Arduino Nano RP2040 Connect.

### 1.6.5 SparkFun Thing Plus RP2040

Diese Entwicklungsplattform (Bild 1.12) enthält einen SD-Kartensteckplatz, 16 MB Flash-Speicher, einen JST-Einzelzellen-Batterieanschluss, eine WS2812-RGB-LED, JTAG-Pins und einen Qwiic-Anschluss. Weitere Eigenschaften:

- 133MHz Taktfrequenz
- 264KB SRAM
- 4 x 12-Bit ADC
- 2 x UART, 2 x I2C, 2 x SPI
- 16 x PWM
- 1 Timer mit 4 Alarm-Funktionen

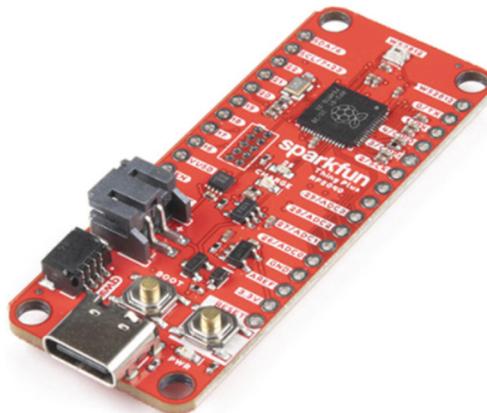


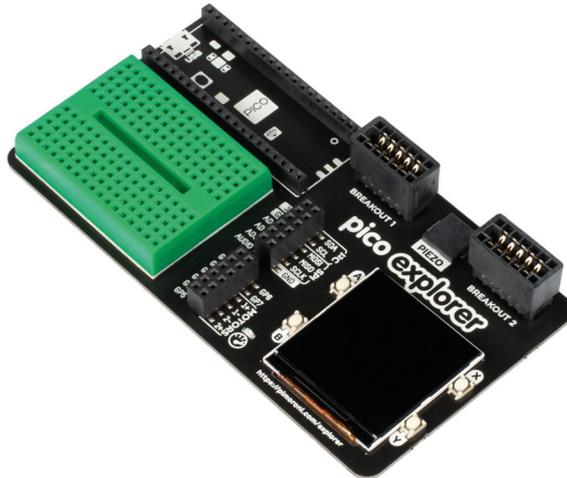
Bild 1.12 SparkFun Thing Plus RP2040.

### 1.6.6 Pimoroni Pico Explorer Basis

Dieses Entwicklungsboard (Bild 1.13) enthält ein kleines Breadboard und ein 240x240 IPS-LCD-Display mit 4 berührungsempfindlichen Tasten. Auf der Platine befindet sich ein Sockel zum Einstecken eines Pico. Die Grundfunktionen dieses Entwicklungsboards sind:

- Piezo-Lautsprecher
- 1,54 Zoll IPS-LCD

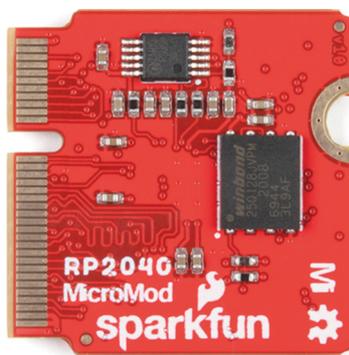
- 4 Taster
- 2 Halbbrücken-Motortreiber
- Zwei Breakout-I2C-Buchsen
- Einfacher Zugang zu GPIO- und ADC-Pins
- Mini-Breadboard
- Kein Löten erforderlich
- Pico wird nicht mitgeliefert



*Bild 1.13 Pimoroni Pico Explorer Basis.*

### 1.6.7 SparkFun MicroMod RP2040-Prozessor

Dieses Board (Bild 1.14) enthält einen MicroMod M.2-Anschluss für den Zugriff auf die GPIO-Pins.



*Bild 1.14 SparkFun MicroMod RP 2040 Prozessor.*

### 1.6.8 SparkFun Pro Micro RP2040

Dieses Board (Bild 1.15) enthält eine adressierbare ES2812B-LED, einen Boot-Taster, einen Reset-Taster, einen Qwiic-Anschluss, einen USB-C-Stromversorgungsanschluss, eine PTC-Sicherung und wabenförmige GPIO-Pads.

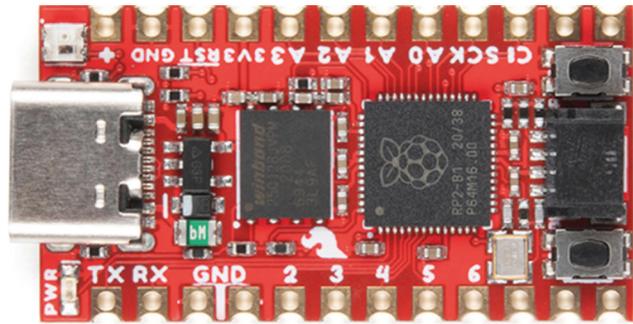


Bild 1.15 SparkFun Pro Micro RP2040.

### 1.6.9 Pico-RGB-Tastatursocket

Diese Platine ist mit einem farbig beleuchteten 4 x 4 Tastenfeld (Bild 1.16) mit APA102-LEDs ausgestattet. Die Grundfunktionen sind:

- 4 x 4 Tastenfeld
- 16 APA102-RGB-LEDs
- Tastatur angeschlossen über I2C I/O-Expander
- GPIO-Pins beschriftet



Bild 1.16 Pico RGB-Tastatur Basis.

### 1.6.10 Pico Omnibus

Dies ist eine Erweiterungsplatine (Bild 1.17) für den Pico. Grundlegende Merkmale dieser Platine sind:

- GPIO-Pins beschriftet
- Zwei Steckplätze mit beschrifteten (gespiegelten) Stiftleisten zum Anbringen von Add-Ons
- 4 GummifüÙe
- Kompatibel mit Pico
- Vollständig montiert
- Abmessungen: ca. 94 mm x 52 mm x 12 mm



Bild 1.17 Pico Omnibus.

### 1.6.11 Pimoroni Pico VGA Demo Base

Diese Platine (Bild 1.18) hat einen VGA-Ausgang und einen SD-Kartenslot. Die grundlegenden Eigenschaften sind:

- Stromversorgung durch Pico
- 15-poliger VGA-Anschluss
- I<sup>2</sup>S-DAC für Line-Out-Audio
- PWM-Audio-Ausgang
- SD-Kartensteckplatz
- Reset-Taste
- Header für die Installation Ihres Pico
- Drei frei verfügbare Schalter
- Kein Löten erforderlich

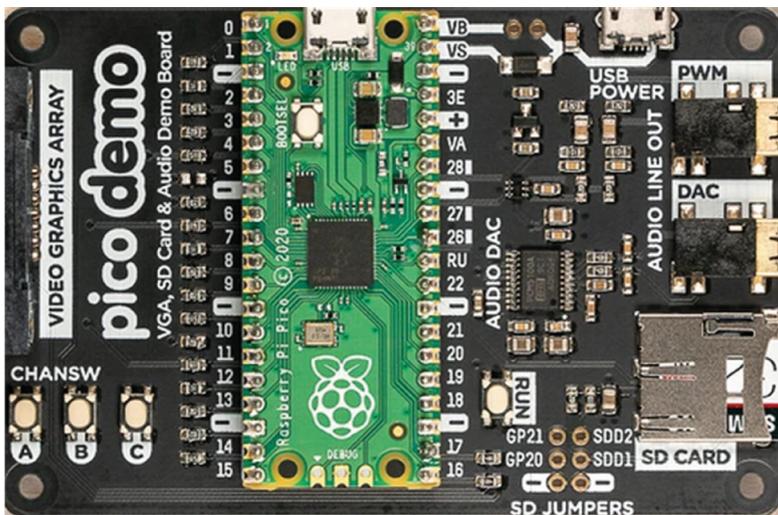


Bild 1.18 Pimoroni Pico VGA-Demobasis.

## Kapitel 2 • Raspberry Pi Pico Programmierung

### 2.1 Überblick

Zum Zeitpunkt der Erstellung dieses Buches standen für den Pico folgende Programmiersprachen zur Verfügung:

- C/C++
- MicroPython
- Assembler

Obwohl der Pico standardmäßig für die Verwendung der leistungsstarken und beliebten Sprache C/C++ eingerichtet ist, finden viele Anfänger es einfacher, MicroPython zu verwenden, eine Version der Programmiersprache Python, die speziell für Mikrocontroller entwickelt wurde.

In diesem Kapitel werden wir lernen, wie man die Programmiersprache MicroPython installiert und verwendet. Wir werden den Texteditor Thonny verwenden, der speziell für Python-Programme entwickelt wurde.

In den nächsten Kapiteln werden wir viele funktionierende und vollständig getestete Projekte mit MicroPython und unserem Pico vorstellen.

### 2.2 Installation von MicroPython auf dem Pico

MicroPython muss auf dem Pico installiert werden, bevor das Board verwendet werden kann. Einmal installiert, bleibt MicroPython auf dem Pico, sofern es nicht mit etwas anderem überschrieben wird. Die Installation von MicroPython erfordert eine Internetverbindung, aber diese ist nur einmal nötig. Da der Pico keine Wi-Fi-Verbindung hat, müssen wir einen Computer mit Internetzugang verwenden. Dies kann entweder mit einem Raspberry Pi (z. B. Raspberry Pi 4) oder mit einem PC geschehen. In diesem Abschnitt sehen wir uns beide Installations-Methoden an.

#### 2.2.1 Raspberry Pi 4 zur Unterstützung der Installation von MicroPython auf dem Pico

Die Schritte sind wie folgt:

- Booten Sie Ihren Raspberry Pi 4 und melden Sie sich am Desktop an.
- Stellen Sie sicher, dass Ihr Raspberry Pi mit dem Internet verbunden ist.
- Halten Sie die **BOOTSEL**-Taste an Ihrem Pico gedrückt.
- Verbinden Sie Ihren Pico über ein Micro-USB-Kabel mit einem der USB-Ports des Raspberry Pi 4, während Sie die Taste gedrückt halten.
- Warten Sie ein paar Sekunden und lassen Sie die **BOOTSEL**-Taste los.
- Der Pico sollte nun als Wechseldatenträger erscheinen. Klicken Sie im Fenster **Removable medium is inserted** (Bild 2.1) auf **OK**.

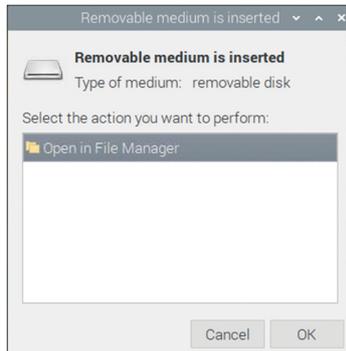


Bild 2.1: Klick auf OK.

- Im Fenster **File Manager** sehen Sie zwei Dateien mit den Namen **INDEX.HTM** und **INFO\_UF2.TXT** (Bild 2.2).

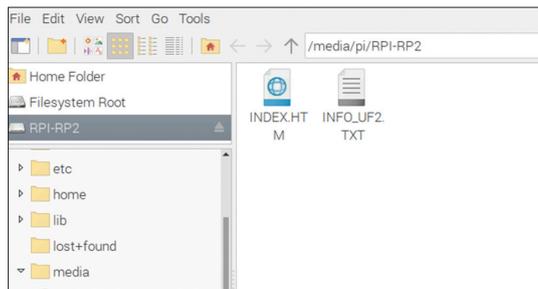


Bild 2.2: Beide Dateien beachten!

- Doppelklicken Sie auf die Datei **INDEX.HTM** und scrollen Sie nach unten.
- Sie sollten die Meldung **Welcome to your Raspberry Pi Pico** in einer Webseite angezeigt bekommen (Bild 2.3).

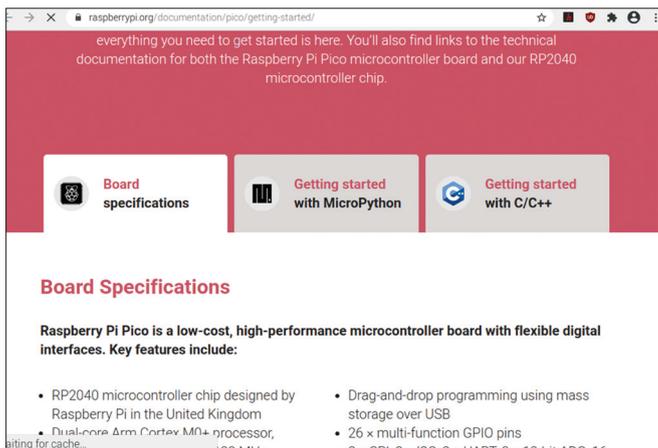


Bild 2.3: Angezeigte Meldung.

- Klicken Sie auf die Registerkarte **Getting started with MicroPython** und klicken Sie auf **Download UF2 file**, um die **MicroPython**-Firmware herunterzuladen. Sie sollten die heruntergeladene Datei am unteren Rand des Bildschirms sehen. Dies dauert nur ein paar Sekunden (Bild 2.4).

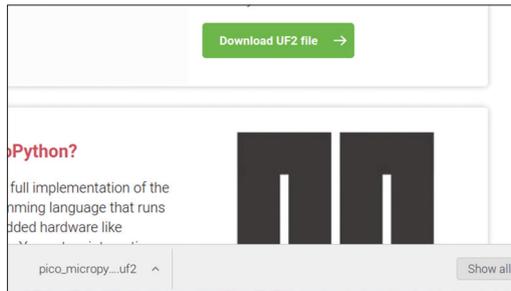


Bild 2.4: Herunterladen der UF2-Datei.

- Schließen Sie Ihr Browser-Fenster durch Anklicken des Kreuz-Symbols in der oberen rechten Ecke.
- Öffnen Sie den **Dateimanager**, indem Sie auf Menü und dann auf **Zubehör** klicken.
- Öffnen Sie den Ordner **Downloads** (unter **/home/pi**) und suchen Sie die Datei mit der Endung: **.uf2**. Diese Datei hat einen ähnlichen Namen wie: **micropython-20-Jan-2021.uf2** (Bild 2.5).

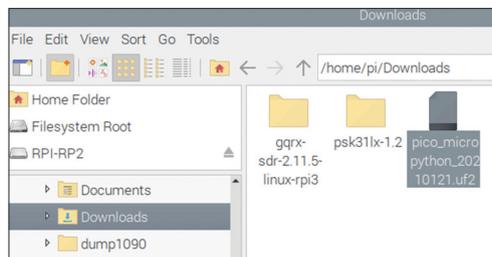


Bild 2.5: Suchen der Datei mit der Erweiterung: **.uf2**.

- Ziehen Sie diese Datei auf den Wechseldatenträger **RPI-RP2** des Pico (oben links auf dem Bildschirm - siehe Bild 2.5).
- Nach einer Weile wird die **MicroPython**-Firmware auf dem internen Speicher des Pico installiert und das Laufwerk verschwindet.
- Auf Ihrem Pico läuft nun **MicroPython**.
- Auch wenn Sie den Pico ausschalten, bleibt MicroPython im Speicher erhalten.

### Verwendung des Thonny-Texteditors vom Raspberry Pi aus

Thonny ist eine kostenlose integrierte Entwicklungsumgebung (IDE), die speziell für Python entwickelt wurde. Sie verfügt über einen eingebauten Texteditor und Debugger sowie eine Reihe anderer Dienstprogramme, die während der Programmentwicklung nützlich sein können.

In diesem Abschnitt werden wir lernen, wie man Thonny benutzt, indem man es vom Raspberry Pi aus aufruft. Sie sollten Ihren Pico mit dem Raspberry Pi verbunden lassen. Wir werden ein einzeliliges Programm erstellen, um die Meldung **Hello vom Raspberry Pi Pico** anzuzeigen:

Die Schritte sind unten angegeben.

- Klicken Sie auf Ihrem Raspberry Pi Desktop auf **Menu**, gefolgt von **Programming** und dann auf **Thonny Python IDE** (siehe Bild 2.6). Der Autor hatte die Version 3.3.3 von Thonny auf seinem Raspberry Pi 4 installiert.

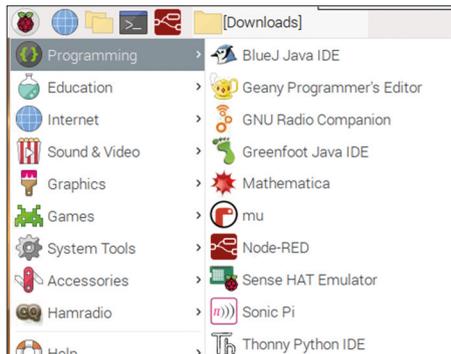


Bild 2.6: Thonny auf Ihrem Raspberry Pi starten.

- Klicken Sie auf die Bezeichnung **Python** in der rechten unteren Ecke von Thonny (Bild 2.7).



Bild 2.7: Auf Python in der rechten unteren Ecke klicken.

- Um **MicroPython (Raspberry Pi Pico)** auszuwählen, klicken Sie wie in Bild 2.8 gezeigt.

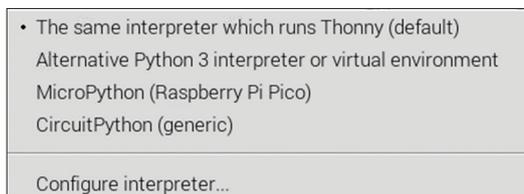


Bild 2.8: Auswahl des Raspberry Pi Pico.