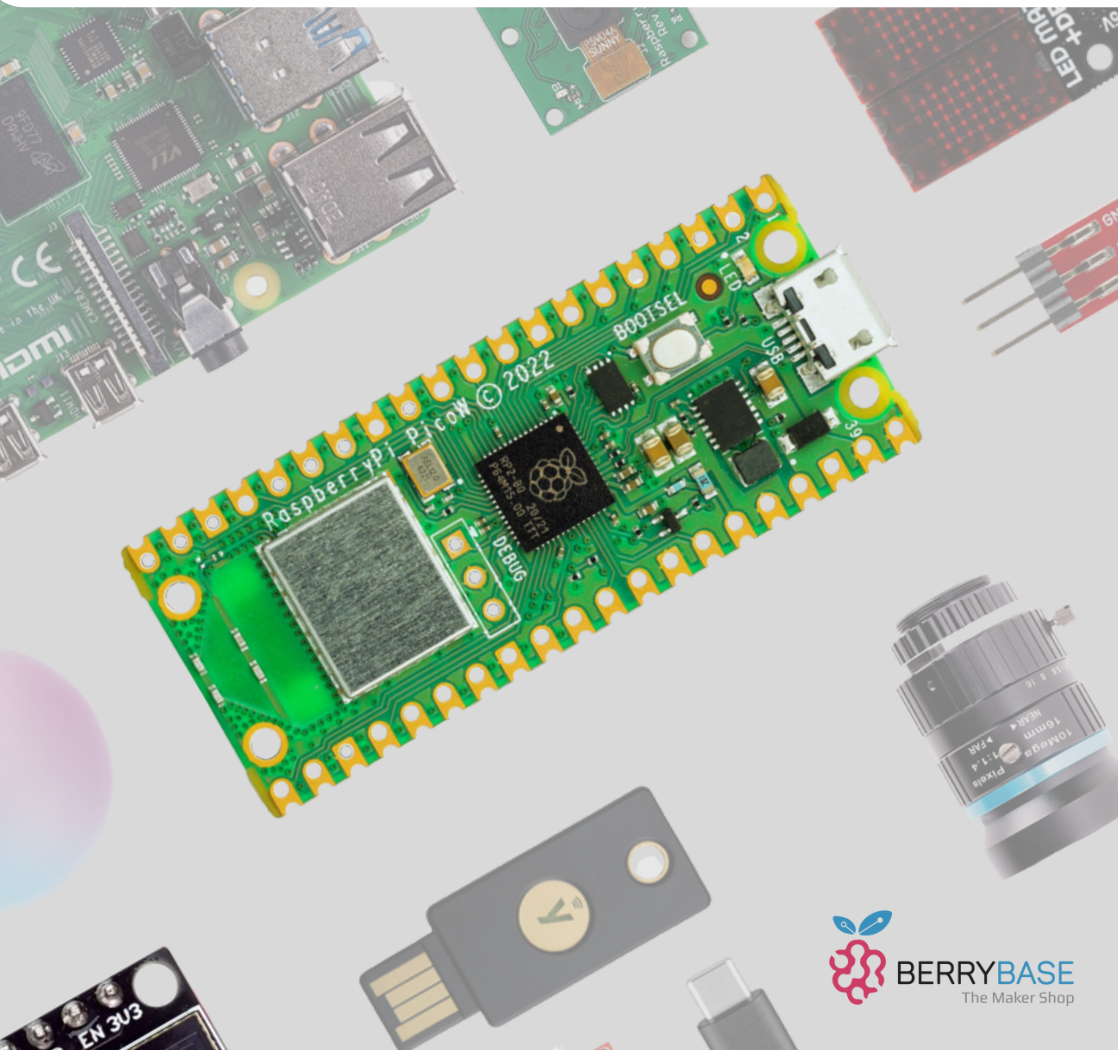


# GET YOUR STUFF RUNNING

THE  
JUMPSTART  
GUIDE FOR  
RASPBERRY  
PI PICO  
(W)





	Seite
<b>Der Raspberry Pi Pico (W)</b>	4
<b>Warum solltest du dich für den Raspberry Pi Pico (W) entscheiden?</b>	5
<b>Die Spezifikationen</b>	7
<b>Erste Schritte mit dem Raspberry Pi Pico (W)</b>	9
<b>Den Raspberry Pi Pico (W) aktualisieren</b>	10
<b>Die Programmierumgebung (IDE)</b>	11
<b>Programmierungsumgebung (IDE): C++ vs. MicroPython – Welche ist für dich die Richtige?</b>	12
<b>Abschließende Worte</b>	13



## **Liebe Leserinnen und Leser,**

Raspberry Pi ist für viele das Synonym für den beliebten Einplatinenrechner Raspberry Pi 4. Neben dem Raspberry Pi Zero (W) / 2 W, als abgespeckte Version, war der nächste logische Schritt der Raspberry Pi Foundation, im MicroControllermarkt Fuß zu fassen. Mit dem Raspberry Pi Pico (W) ist dabei ein interessanter MicroController herausgekommen, der vieles nicht neu erfindet, aber trotzdem durch ein Feature für die Makerszene eine interessante Alternative zu Arduino und ESP32 / ESP8266 darstellt.

**Berrybase.de** ist der Onlinehändler deiner Wahl, wenn du den Einstieg in die Welt des Raspberry Pi Pico (W) beginnen oder spannende Projekte mit Hardware umsetzen willst. Mein Name ist Jörn, ich bin Dipl.-Ing- (FH) Mechatronik und Softwareentwickler, großer Fan von Raspberry Pi und MicroControllern. In meinen Blogs habe ich schon einige Themen zu MicroControllern und dem Raspberry Pi behandelt, um den Einstieg so einfach wie möglich für Interessierte zu gestalten.



## Der Raspberry Pi Pico (W)

Der Raspberry Pi Pico (W) ist ein sogenannter MicroController der, ähnlich wie der Arduino oder ESP, gerade in Projekten zum Einsatz kommt, wo der Programmcode zyklisch abgearbeitet werden soll, es aber gleichzeitig keine teure Computerhardware benötigt. Mit dem Raspberry Pi Pico W, wobei das W für Wireless steht, hat die Raspberry Pi Foundation die Kritik am Raspberry Pi Pico aufgenommen und dem Raspberry Pi Pico W WiFi und Bluetooth spendiert. Bluetooth war aber nicht zum Release des Raspberry Pi Pico W verfügbar, sondern wurde durch ein MicroPython-Update am 14 Juni 2023 erst freigeschaltet. Wer also einen alten Raspberry Pi Pico W besitzt, kann dies nachträglich freischalten, siehe [hier](#).

Auf beiden Raspberry Pi Pico – Varianten kommt der RP2040 – dual-core Arm Cortex-M0+ Prozessor mit 264kB internen Speicher zum Einsatz. Dieser Chip wird mittlerweile auch von anderen MicroControllerherstellern für eigene MicroControllervarianten eingesetzt. Wie man es durch den Arduino oder ESP gewohnt ist, kann man mit C++ entwickeln oder, zumindest bei der ESP-Familie, mit MicroPython. MicroPython ist eine angepasste Python-Version, um MicroController in dieser Programmiersprache zu programmieren. Gerade in vielen Foren und Blogbeiträgen wird diese Sprache verwendet, da Python mittlerweile eine der beliebtesten Programmiersprachen unter Programmierern ist, siehe [hier](#). Ein zweiter Faktor, der hinzukommt ist, dass Python sehr verständlich in der Syntax ist und selbst Anfänger schnell in die Programmiersprache hineinfinden.

Ein Vorteil vom Raspberry Pi Pico (W) ist, dass viele Pins für gängige Protokolle wie I2C oder SPI verwendet werden können. Dabei gibt es keine strenge Einschränkung bei der Pinbelegung, wie sie z.B. beim Arduino Uno zu finden ist. Dazu aber mehr in den Spezifikationen.



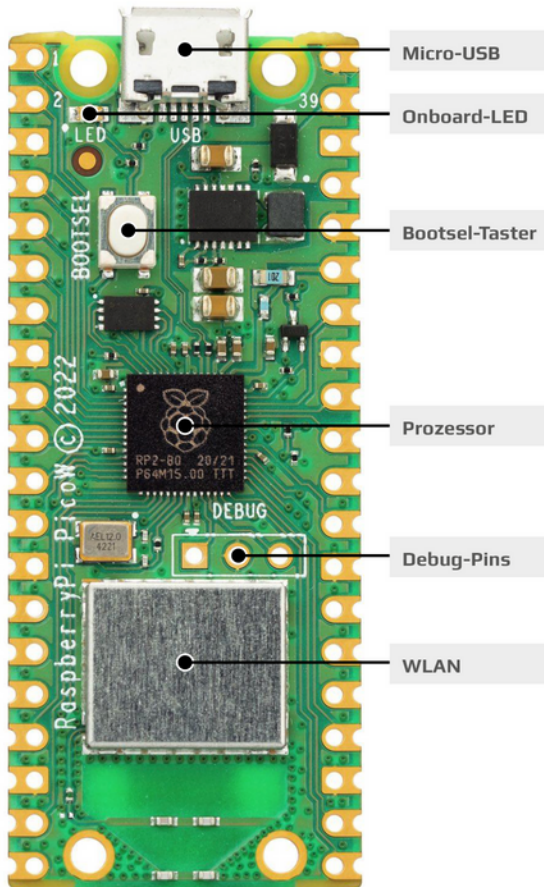
Bis zu diesem Punkt würde der Raspberry Pi Pico (W) erst einmal nicht groß in der Masse von MicroControllern herausstechen. Jedoch besitzt der Raspberry Pi Pico eine Besonderheit, die bisher einzigartig unter den MicroControllern ist, die programmierbaren I/Os, kurz PIO. Dabei haben die zwei PIO-Blöcke auf dem Raspberry Pi Pico jeweils 4 Zustandsmaschinen, um z.B. eine CAN-Schnittstelle nachzubauen. Das klingt erst einmal nicht großartig spannend, heißt aber im Klartext, dass man in der Theorie fast jedes Protokoll oder Standard nachprogrammieren könnte. Es gibt Blogbeiträge, in dem Programmierer mit einem Raspberry Pi Pico ein Bild auf einen VGA-Monitor gebracht haben, nur mit den PIOs. Mehr dazu kann man z.B. bei [heise.de](https://heise.de) nachlesen, siehe [hier](#). Wie es sich für viele gängige MicroController gehört, sind drei Analogwandler vorhanden. Bei allen Pins gilt, dass eine maximale Spannung von 3,3V zulässig ist!

## Warum solltest du dich für den Raspberry Pi Pico (W) entscheiden?

- **Top Preis-Leistungs-Verhältnis:** Leistungsstark und dennoch erschwinglich – typisch Raspberry Pi.
- **Programmierbare I/Os (PIO):** Damit kannst du fast jedes Protokoll selbst gestalten – ein Niveau der Anpassbarkeit, das selten ist.
- **Starke Community:** Egal ob Anfänger oder Profi, Hilfe und Inspiration sind nur einen Klick entfernt.
- **Drahtlose Möglichkeiten beim Pico W:** Integriertes WiFi und Bluetooth machen IoT-Projekte zum Kinderspiel.
- **Wahl der Programmiersprache:** Entscheide zwischen C++ für Systemnähe oder MicroPython für Schnelligkeit.



Kurz gesagt: Der Raspberry Pi Pico (W) ist vielseitig, leistungsstark und bietet dir einen hervorragenden Einstieg oder eine Weiterentwicklung in der Welt der MicroController.



## Die Spezifikationen

Im letzten Kapitel habe ich angesprochen, was der Raspberry Pi Pico (W) ist und wofür er gedacht ist. Interessant ist aber auch, wie die technischen Spezifikationen aussehen, siehe Tabelle 1.

	Raspberry Pi Pico	Raspberry Pi Pico W
MicroController	RP2040	
Architektur	Dual-core Arm Cortex M0+ Prozessor	
Taktrate	Bis zu 133MHz	
SRAM	264 kB	
Ob-Board Speicher	2 MB	
USB	1.1 mit Device und Host-Support	
I/Os	26 Pins für I/Os	
Schnittstellen	2 × SPI 2 × I2C 2 × UART 3 × 12-bit ADC 16 × controllable PWM channels	
WiFi	Nicht vorhanden	Wireless (802.11n) single-band (2.4 GHz)
Bluetooth	Nicht vorhanden	5.2 (seit dem 14.06.2023 verfügbar)
Dimensionen	21mm x 51,3 mm	

*Tabelle 1: Technische Spezifikation  
Raspberry Pi Pico (W)*

Als weiteren Anhaltspunkt für dein Projekt, sollte dir das Pinout vom Raspberry Pi Pico helfen, siehe Abbildung 1. Damit sieht man auch gleich, welcher Pin für welche Funktion gedacht ist.

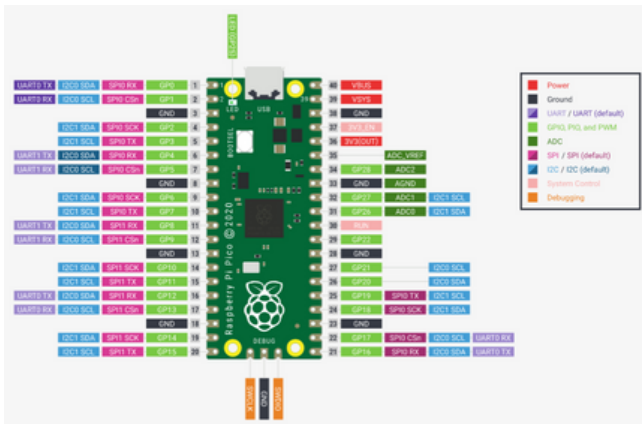


Abbildung 1:  
Pinout vom  
Raspberry Pi  
Pico, von  
raspberrypi.com

Im Prinzip hat sich an dem Pinout beim Raspberry Pi Pico W wenig geändert, siehe Abbildung 2. Teilweise hat sich nur die Position einiger Pins geändert, wie die DEBUG-Pins.

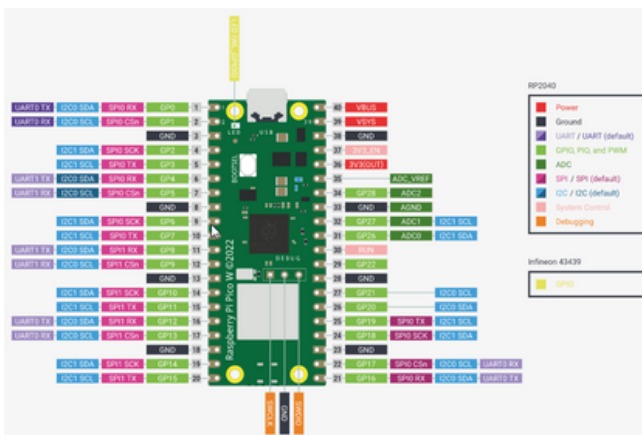


Abbildung 2:  
Pinout vom  
Raspberry Pi  
Pico W, von  
raspberrypi.com



Schaut man sich die Layouts genauer an, ist der Unterschied auf den ersten Blick kaum zu erkennen. Einzig die WiFi-Antenne, welche silbern im Layout dargestellt ist, gibt einen Hinweis darauf.

Wichtig bei beiden Raspberry Pi Pico – Versionen, die Pins können alle nur 3,3V, bis auf VSYS, der eine Eingangsspannung von 5V verträgt.

Wie in der Spezifikation zu sehen, unterstützt der Pico 2 mal i2c, SPI und UART, wobei die Pinbelegung für die Kommunikation fast frei wählbar sind.

## Erste Schritte mit dem Raspberry Pi Pico (W)

Nachdem du nun eine Übersicht über den Raspberry Pi Pico (W) erhalten hast, lass uns mit einem einfachen Projekt beginnen.

### Schritt 1: Hardware-Vorbereitung

- Raspberry Pi Pico (W) Board
- USB-Kabel (Typ C oder Micro USB, abhängig von deinem Board)
- Computer mit Internetzugang

### Schritt 2: Software-Installation

1. Verbinde das Pico (W) Board mit deinem Computer.
2. Besuche die offizielle [Raspberry Pi Webseite](#)
3. Lade die für dein Betriebssystem geeignete Software herunter und installiere sie.

```
python Copy code

from machine import Pin, Timer

led = Pin(14, Pin.OUT) # Die LED ist an Pin 14 angeschlossen
timer = Timer()       # Erstellen eines Timer-Objekts

# Diese Funktion wird jedes Mal aufgerufen, wenn der Timer abläuft
def blink(timer):
    led.toggle()      # Wechseln des LED-Zustands (AN/AUS)

# Richte den Timer so ein, dass er die Funktion blink alle 500ms aufruft
timer.init(freq=2, mode=Timer.PERIODIC, callback=blink)
```

Abbildung3: Pinout vom Raspberry Pi Pico W, von [raspberrypi.com](http://raspberrypi.com)

### Schritt 3: Firmware-Update

Folge den Anweisungen, die du bereits in deinem Text hast, um die Firmware zu aktualisieren und Bluetooth zu aktivieren.

### Schritt 4: Erstes Projekt - Blinkende LED

1. Schließe eine LED an den GPIO-Pin 14 und GND des Pico Boards an.
2. Öffne die Thonny IDE (oder eine andere, die du bevorzugst).
3. Erstelle ein neues Python-Skript mit folgendem Code:

### Schritt 5: Weitere Experimente

Von hier aus kannst du mit verschiedenen Projekten und Experimenten fortfahren. Probiere verschiedene Sensoren und Aktoren aus, verbinde dich mit dem Internet, und erkunde, was mit deinem Raspberry Pi Pico (W) alles möglich ist.

## Den Raspberry Pi Pico (W) aktualisieren

Die erste Frage, die du dir wahrscheinlich schon gestellt hast ist, wie du MicroPython-Updates auf den Raspberry Pi Pico (W) übertragen kannst. Zunächst musst du dazu den Raspberry Pi Pico (W) in den Bootloader-Modus bringen. Das klingt komplizierter als es wirklich ist. Drücke und halte dazu die BOOTSEL-Taste auf dem Raspberry Pi Pico (W), siehe Abbildung 3, und stecke dann erst das USB-Kabel an.

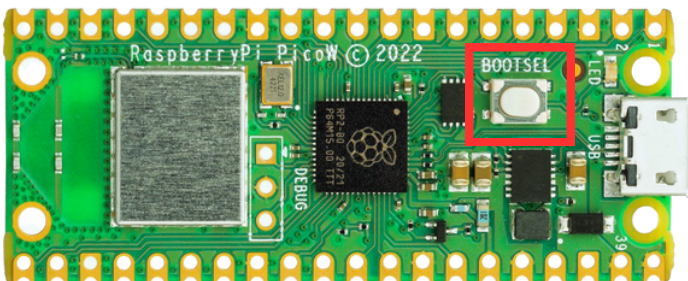


Abbildung 4: Die BOOTSEL-Taste vom Raspberry Pi Pico (W)

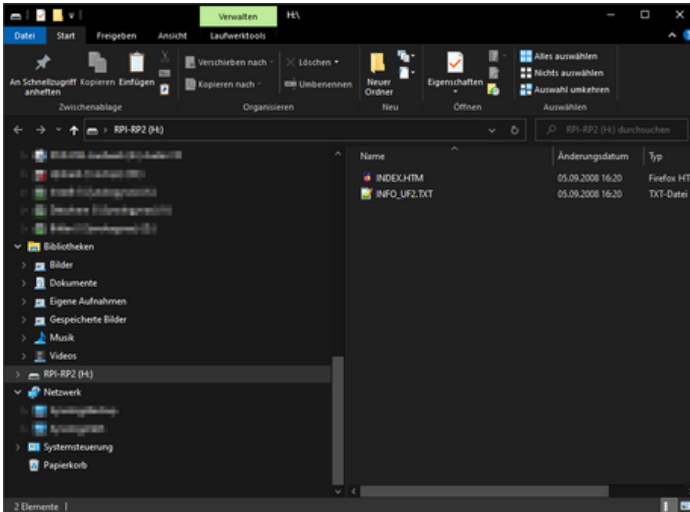


Abbildung 5:  
Raspberry Pi  
Pico (W) im  
Bootloader-  
Modus

Willst du die aktuelle Version von MicroPython installieren, die auch gleichzeitig die Bluetooth-Funktion freischaltet, musst du die folgende Datei herunterladen, siehe [hier](#). Lade diese Datei herunter und kopiere dann die \*.uf2-Datei auf das angezeigte USB-Device.

Direkt nach erfolgreichem kopieren der \*.uf2-Datei, startet der Raspberry Pi Pico (W) neu. Das ist ein automatischer Prozess, der das Update direkt durchführt und den Raspberry Pi Pico (W) im Anschluss normal startet.

## Die Programmierumgebung (IDE)

Wie bereits erwähnt unterstützt der Raspberry Pi Pico (W) als Programmiersprache C++ oder MicroPython. Je nach Programmiersprache können dafür verschiedene IDE's verwendet werden. Nutzt du z.B. den Raspberry Pi 4 Mod. B zum Programmieren vom Raspberry Pi Pico (W), so ist beim Raspberry Pi OS schon Thonny vorinstalliert.



Hingegen wird für C++ gerne Visual Studio Code als Programmierumgebung verwendet, was auch mit dem Windows Pico Installer mitinstalliert wird, siehe [hier](#).

Im Prinzip kann fast jede Entwicklerumgebung für MicroPython oder C++ verwendet werden, sofern diese direkt oder indirekt auf den Raspberry Pi Pico (W) zugreifen kann. Es gibt viele Blogs und Foreneinträge, die über „die perfekte“ IDE schreiben. Es kommt an der Stelle tatsächlich darauf an, was du für Vorlieben hast und was du von einer Programmierumgebung erwartest. Persönlich kann ich aber Thonny und/oder Visual Studio Code für den Anfang nur empfehlen.

## Programmierungsumgebung (IDE): C++ vs. MicroPython – Welche ist für dich die Richtige?

### C++:

- **Tiefe Kontrolle:** Wenn du jedes Detail deines Codes kontrollieren möchtest und Optimierungen auf Systemebene suchst, ist C++ deine Wahl.
- **Performance:** Ideal für Anwendungen, die schnelle Reaktionen und Effizienz erfordern.
- **Werkzeuge:** Arbeite am besten mit Visual Studio Code oder anderen gängigen C++ IDEs.
- **Gut für:** Fortgeschrittene Programmierer und komplexe Projekte.

### MicroPython:

- **Einfacher Einstieg:** Wenn du neu in der Programmierung oder Hardware-Welt bist, bietet MicroPython eine leichter verständliche Syntax.
- **Schnelle Entwicklung:** Mit MicroPython kannst du Prototypen und Projekte in kürzerer Zeit erstellen.
- **Werkzeuge:** Thonny ist eine hervorragende, vorinstallierte IDE auf dem Raspberry Pi OS für MicroPython.
- **Gut für:** Einsteiger und jene, die eine Balance zwischen Einfachheit und Funktionalität suchen.



Überlege, welches Ziel du verfolgst: Schnelle Prototypen oder tiefgreifende Systemoptimierung? Deine Antwort darauf wird dir helfen, die richtige Sprache für deine Projekte zu wählen.

### **Abschließende Worte**

Der Raspberry Pi Pico (W) ist ein interessanter MicroController für noch interessantere Projekte. In der Makerszene hat der Raspberry Pi Pico (W), neben dem Raspberry Pi, seinen Platz gefunden. Gerade mit dem PIO hat die Raspberry Pi Foundation ein Feature in ihren MicroController zur Verfügung gestellt, der viele Interessante Projekte schon zum Vorschein gebracht hat.

Es gibt Projekte über Rollroboter, kleinere Drucker aber auch über einen Commodore 64 – Emulator. Es gibt sowohl für Anfänger als auch für Fortgeschrittene diverse interessante Projekte. Den Raspberry Pi Pico (W) kann ich jedem empfehlen, der mittels C++ oder MicroPython Projekte entwickeln oder nachbauen möchte.

