

eBook zum

Raspberry Pi Pico

RP2040





Inhaltsverzeichnis

Einführung	3
Spezifikationen	4
Pinout	5
Installation der Arduino IDE	6
Zusätzliche Einstellungen	10
Blink Beispiel	12
Installation der MicroPython Firmware	13
Installation der Python IDE Thonny	14
MicroPython Blink Beispiel	17
Pico mit Raspberry Pi OS	18
MicroPython Blink Beispiel	21



Einführung

Vielen Dank, dass Sie sich für den Raspberry Pi Pico entschieden haben. In den nachfolgenden Seiten werden wir Ihnen erklären wie Sie das Gerät einrichten und nutzen können.

Der Raspberry Pi Pico ist der neue flexible Mikrocontroller von Raspberry Pi. Er ist kleiner als der Raspberry Pi Zero und vielseitig einsetzbar. Anders als die anderen Raspberry Pi Produkte wird auf dem Pico hardwarenah programmiert. Der Pico ist mit dem RP2040 Prozessor von Raspberry Pi ausgestattet. Die 26 GPIOs sind multifunktional, das heißt I2C, SPI und PWM kann auf jedem Pin betrieben werden.

Der Pi Pico kann mit C/C++ oder Micropython programmiert werden. In diesem eBook erfahren Sie, wie Sie den Pico mit der Arduino IDE programmieren können, und mit MicroPython in Thonny auf Windows oder einem Raspberry Pi.



Spezifikationen

Processor	Dual core cortexM0+ (up to 133MHz)
Flash size	2MB
SRAM	264kB
40 pins(3.3V logic)	26 GPIOs 23 digital, 3 analog
ADC	12bit 500ksps
Pripherals	2x UART 2x I2C 2x SPI 16 PWM channels



Pinout









Installation der Arduino IDE

Die kostenlose Arduino IDE Entwicklungsumgebung können Sie unter folgendem Link herunterladen: <u>https://www.arduino.cc/en/Main/Software</u>

Windows Benutzer sollten unbedingt eine der ersten beiden Download-Optionen für die Arduino IDE verwenden. Die "Windows App" Version aus dem Windows Store führt insbesondere bei der Verwendung von Board-Definitionen von Drittanbietern zu Verbindungsproblemen.

Download the Arduino IDE



Nach dem Starten der Arduino IDE Installationsdatei

"arduino-1.X.X-windows.exe" müssen die Lizenzbedingungen der Software gelesen und akzeptiert werden:



💿 Arduino Setup: License Agreement	_		×
Please review the license agreement before i accept all terms of the agreement, click I Agr	installing Ardu ee.	uino. If you	
SNU LESSER GENERAL PUBLIC LICENSE			^
Version 3, 29 June 2007			
Copyright (C) 2007 Free Software Foundation, Inc.	< <u>http://fsf.o</u>	<u>ra/</u> >	
Everyone is permitted to copy and distribute verbati document, but changing it is not allowed.	im copies of t	his license	
This version of the GNU Lesser General Public Licens and conditions of version 3 of the GNU General Publi by the additional permissions listed below.	e incorporate ic License, su	s the terms pplemented	v
1			_
Cancel Nullsoft Install System v3.0		I Agre	e

Im nächsten Schritt können unterschiedliche Optionen zur Installation ausgewählt werden.

💿 Arduino Setup: Installation	Options	_		×
Check the components y you don't want to install	you want to instal . Click Next to cor	and uncheck the htinue.	compone	ints
Select components to install:	 ✓ Install Ard ✓ Install US ✓ Create St ✓ Create De ✓ Associate 	duino software B driver art Menu shortcu esktop shortcut .ino files	t	
Space required: 482.4MB				
Cancel Nullsoft Inst	all System v3.0	< Back	Next	t >

Es folgt eine kurze Übersicht der verschiedenen Optionen, mit einer kurzen Erklärung zu jeder Option:



Option	Erklärung
Install Arduino Software	Installiert die Arduino IDE - Diese Option kann nicht abgewählt werden
Install USB Driver	Installiert USB Treiber für verschiedene andere Microcontroller. Für die Verwendung der Software mit dem D1 mini sind diese nicht erforderlich, wir empfehlen jedoch unbedingt die Installation, falls Sie auch andere Mikrocontroller verwenden
Create Start Menu shortcut	Erstellt eine Verknüpfung im Windows Startmenü (Optional)
Create Desktop shortcut	Erstellt eine Verknüpfung auf dem Arbeitsplatz (Optional)
Associate .ino files	Erstellt eine Dateinamenerweiterung für Dateien mit der Endung .ino und verknüpft diese mit der Arduino IDE

Zuletzt muss noch der Zielordner angegeben werden. Die Installation benötigt ca. 500MB freien Speicherplatz.



Klicken Sie auf "Install", um die Installation zu starten.



💿 Arduino Setu	p: Installing	_		\times
Extract: a	vr-c++filt.exe			
Show details]			
Cancel	Nullsoft Install System v3,0	< Back	Clos	e

Nach erfolgreicher Installation kann das Installationsprogramm über die Schaltfläche "Close" beendet werden:

💿 Arduino Setu	p: Completed	_		\times
	1			
Show details				
Cancel	Nullsoft Install System v3,0	< Back	Clos	se



Zusätzliche Einstellungen

Um den Pico mit der Arduino IDE zu verwenden, müssen folgende Einstellungen durchgeführt werden:

Installieren Sie den Support für die RP2040-Plattform, öffnen Sie die Arduino IDE und gehen Sie zu: Datei > Voreinstellungen, und finden Sie das Feld "Zusätzliche Boardverwalter-URLs".

Kopieren Sie folgenden Link:

https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp 2040_index.json

Anschließend mit OK bestätigen.

Voreinstellungen				×
Einstellungen Netzwerk				
Sketchbook-Speicherort:				
C:\Users\ligma\Documents\Ard	uino			Durchsuchen
Editor-Sprache:	System Default	~	(erfordert Neustart von A	Arduino)
Editor-Textgröße:	12			
Oberflächen-Zoomstufe:	Automatisch 100 🔷 % (er	fordert Neustart von Ar	duino)	
Thema:	Standardthema \vee (erfordert Ne	eustart von Arduino)		
Ausführliche Ausgabe während:	Kompilierung Hochladen			
Compiler-Warnungen:	Keine 🖂			
🗹 Zeilennummern anzeigen		Code-Faltung aktiv	ieren	
🗹 Code nach dem Hochladen ü	iberprüfen	Externen Editor ver	rwenden	
Beim Start nach Updates su	chen	Speichern beim Üb	erprüfen oder Hochladen	
Use accessibility features				
Zusätzliche Boardverwalter-URLs	m/earlephilhower/arduino-pico	/releases/download/glo	bal/package_rp2040_inde	x.json
Mehr Voreinstellungen können d	irekt in der Datei bearbeitet werde	n		
C:\Users\ligma\AppData\Local\A	rduino15\preferences.txt			
(nur bearbeiten, wenn Arduino r	nicht läuft)			
			0	K Abbruch



Als Nächstes gehen Sie auf Werkzeuge > Board > Boardverwalter und geben Sie im Suchfeld "Pico" ein und installieren "Raspberry Pi Pico/RP2040"

Boardverwalter	\times
Typ Alle v pico	
Arduino Mbed OS RP2040 Boards	^
by Arduino In diesem Paket enthaltene Boards: Raspberry Pi Pico. <u>Online Help</u> <u>More Info</u>	
Raspberry Pi Pico/RP2040	
by Earle F. Philhower, III Version 1.9.5 INSTALLED In diesem Paket enthaltene Boards: Raspberry Pi Pico, Adafruit Feather RP2040, Adafruit ItsyBitsy RP2040, Adafruit Macropad RP2040, Adafruit QTPy RP2040, Adafruit STEMMA Friend RP2040, Adafruit Trinkey RP2040 QT, Arduino Nano RP2040 Connect, SparkFun ProMicro RP2040, Generic RP2040 Module. More Info	
Version auswählen V Installieren	n
[DEPRECATED - Please install standalone packages] Arduino Mbed OS Boards	
by Arduino DEPRECATED In diesem Paket enthaltene Boards: Arduino Nano 33 BLE, Arduino Nano 33 BLE Sense, Arduino Nano RP2040 Connect, Arduino Portenta H7, Arduino Edge Control, Raspberry Pi Pico, Nicla Sense ME. Online Help	i
	Schließen

Jetzt sollten Sie unter Werkzeuge > Board > Raspberry Pi RP2040 Boards den Raspberry Pi Pico auswählen können.

Um den Pico flashen zu können, müssen Sie die "BOOTSEL" Taste gedrückt halten bevor Sie ihn mit USB verbinden, Sie können die Taste nach dem verbinden loslassen.

Hinweis: Für den Pico muss man beim Flashen in der Arduino IDE keinen Port angeben, dieser bleibt ausgegraut.



Blink Beispiel

Öffnen Sie das Beispielprogramm "blink" unter Datei > Beispiele > Basics > Blink. Der Sketch kann ohne Anpassungen hochgeladen werden.

```
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on
    (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(LED_BUILTIN, LOW); // turn the LED off
    by making the voltage LOW
    delay(1000); // wait for a second
}
```

Nach dem Hochladen wird der Pico automatisch neu gestartet und die Onboard-LED fängt an zu blinken.





Installation der MicroPython Firmware

Um die MicroPython Firmware herunterzuladen klicken Sie auf den folgenden Link:

https://www.raspberrypi.com/documentation/microcontrollers/micropython.h tml

und klicken Sie auf:



Halten Sie die BOOTSEL Taste gedrückt während Sie den Pico per USB verbinden, der Pico erscheint jetzt wie ein Speichermedium.

Kopieren Sie die heruntergeladene .uf2 Datei auf den Pico, dieser startet anschließend neu und damit läuft die MicroPython Firmware.



Installation der Python IDE Thonny

Gehen Sie auf <u>https://thonny.org/</u> und laden Sie die Thonny IDE herunter und installieren diese.

Thonny Bython IDE for beginners			Download <u>Windows</u>	version <u>3.3.13</u> • <u>Mac</u> • Lin	for ux
Image: Thomas File Edit View Run Tools Help Image:	•	1	-	- 0	×
<pre>factorial.py × def fact(n): if n == 0: return 1 else:</pre>		^	Variables Name fact	Value <function a<="" fact="" th=""><th>^</th></function>	^
<pre>return fact(n-1) * n n = int(input("Enter a natural number print("Its factorial is", fact(3))</pre>	<pre>fact(3) fact def fact(n):</pre>	fact(2)	n	3 X	
<	if n == 0 retur else: retur	def fact if n else	(n): == 0: return 1 : return fact(2-1) * n	
Shell >>> %Debug factorial.py Enter a natural number: 3	Local variables Name Value n 3	Local vari Name V n 2	ables /alue	>	~



Starten Sie anschließend Thonny, Sie müssen jetzt die von Ihnen bevorzugte Sprache auswählen.

1			_		×
Th	Language: Initial settings:	English Standard	Let's	✓✓Go!	

Bei "initial settings" haben Sie die Wahl zwischen Standard und Raspberry Pi. Diese Einstellung ändert das Erscheinungsbild von Thonny, stellen Sie diese auf Raspberry Pi sieht das Thonny Fenster genauso wie auf Raspberry Pi OS aus.



Um mit Thonny den Raspberry Pi Pico programmieren zu können, müssen noch Einstellungen getroffen werden.

Gehen Sie auf Tools > Options und klicken Sie auf den Reiter "Interpreter". Wählen Sie als Interpreter den "MicroPython (Raspberry Pi Pico)"

🕼 Thonn	y options								×
General	Interpreter	Editor	Theme & Font	Run & Debug	Terminal	Shell	Assistant		
Which	interpreter o	r device	should Thonny	use for running	your code?	?			
The sa	me interprete	er which	runs Thonny (de	efault)	-				\sim
The sai Alterna Remot MicroP MicroP MicroP MicroP MicroP MicroP Circuitl	me interprete ative Python 3 e Python 3 (S ython (SSH) ython (BBC r ython (Raspt ython (ESP32 ython (ESP82 ython (gener Python (gener	er which 3 interpr iSH) nicro:bit perry Pi 2) 266) ric) ric)	runs Thonny (de reter or virtual er t) Pico)	fault) wironment					
A spec	ial virtual en	<i>/ironme</i>	nt (deprecated)					Ж	Cancel

Schließen Sie jetzt den Pico per USB an und klicken Sie anschließend auf OK. Die Shell sollte in etwa wie folgt aussehen:

Shell × MicroPython v1.17 on 2021-09-02; Raspberry Pi Pico with RP2040 Type "help()" for more information. >>> MicroPython (Raspberry Pi Pico)



MicroPython Blink Beispiel

Kopieren Sie folgenden Testcode in die Thonny IDE.

```
from machine import Pin, Timer
led = Pin(25, Pin.OUT)
timer = Timer()
def blink(timer):
   led.toggle()
timer.init(freq=1, mode=Timer.PERIODIC, callback=blink)
```

und klicken Sie auf "Run" (oder F5). Es erscheint folgendes Fenster, Thonny fragt nach dem Speicherort.



Wählen Sie hier "Raspberry Pi Pico aus", anschließend müssen Sie dem Programm einen Namen vergeben. "Blink.py" eignet sich dafür gut.

Die Onboard-LED auf dem Pico blinkt jetzt im Sekundentakt.



Pico mit Raspberry Pi OS

Der Raspberry Pi Pico kann auch mit einem Raspberry Pi programmiert werden. Auf dem Raspberry Pi OS ist Python und Thonny bereits vorinstalliert und man kann direkt loslegen.

Starten Sie über Thonny mit "programming > Thonny IDE", beim ersten Start sieht das Fenster so aus:

			Tho	nny - <untitl< th=""><th>ed> @ 1:1</th><th></th><th></th><th></th><th>~ ^ X</th></untitl<>	ed> @ 1:1				~ ^ X
-	Ê	Ľ		ţ <u></u>	ΙΞ	4	r i i	0	Þ
New	Load	Save	Run	Debug	Over	Into	Out	Stop	Zoor
<untitled> ∞</untitled>									
Shell Python 3	7.3 (/us	r/bin/nvi	thon3)						
>>>		.,,							
								Pytl	non 3.7.3

Wir müssen Thonny noch sagen das wir in MicroPython programmieren wollen. Klicken Sie dazu auf "Python" unten links im Fenster.



Es klappt ein kleines Menü auf, gehen Sie auf "Configure interpreter..."



Als nächstes gehen Sie auf den Reiter "interpreter" und wählen "MicroPython (Raspberry Pi Pico)" aus, bevor Sie auf OK klicken, schließen Sie jetzt Ihren Pi Pico an. Halten Sie dabei die BOOTSEL Taste gedrückt, und lassen Sie die Taste erst los nachdem Sie den Pico per USB verbunden haben.

				Thonny	options				~	^	×
Genera	I Interpreter	Editor	Theme & Font	Run & Debug	Terminal	Shell	Assistant				
14/1-1-							I				
The	n interpreter of same interpre	r device ter whi	ch runs Thonny u	(default)	our code?					-	
The	same interpre	ter whic	ch runs Thonny	(default)							
Alter	native Python	n 3 inter	preter or virtual	environment							
Rem	ote Python 3	(SSH)									
Micr	oPython (loca	al)									
Micr	oPython (SSF	H)									
Micr	oPython (BBC) micro:	bit)								
Micr	oPython (Ras	pberry I	Pi Pico)								
Micr	oPython (ESP	232)									
Micr	oPython (ESP	8266)									
Circu	oPython (gen uitDuthon (gen	eric)									
A sn	acial virtual e	nvironm	ent (deprecate	4)							
Азр		TWITCHIT	lent (deprecate	u)							
								OK	Ca	ncel	



Jetzt fordert Thonny Sie dazu auf, die MicroPython Firmware zu installieren oder zu updaten.

:

klicken Sie auf "Install"

Zum Schluss sieht die Shell wie folgt aus:

Shell	
MicroPython v1.17 on 2021-09-02; Raspberry Pi Pico with RP2040 Type "help()" for more information. >>>	
MicroPython v1.17 on 2021-09-02; Raspberry Pi Pico with RP2040 Type "help()" for more information. >>>	Į
MicroPython (I	Raspberry Pi Pico)

Jetzt ist alles eingerichtet und Sie können direkt loslegen mit der Programmierung.



MicroPython Blink Beispiel

Kopieren Sie folgenden Testcode in die Thonny IDE.

```
from machine import Pin, Timer
led = Pin(25, Pin.OUT)
timer = Timer()
def blink(timer):
    led.toggle()
timer.init(freq=1, mode=Timer.PERIODIC, callback=blink)
```

und klicken Sie auf "Run". Es erscheint folgendes Fenster, Thonny fragt nach dem Speicherort.



Wählen Sie hier "Raspberry Pi Pico aus", anschließend müssen Sie dem Programm einen Namen vergeben. "Blink.py" eignet sich dafür gut.

Die Onboard-LED auf dem Pico blinkt jetzt im Sekundentakt.



Jetzt ist es an der Zeit, zu lernen und eigene Projekte zu erstellen. Das können Sie mit Hilfe von vielen Beispielskripten und anderen Tutorials tun, die Sie im Internet finden können.

Wenn Sie auf der Suche nach den qualitativ hochwertigen Produkten für Arduino und Raspberry Pi sind, sind Sie bei der AZ-Delivery Vertriebs GmbH genau richtig. Sie erhalten zahlreiche Anwendungsbeispiele, vollständige Installationsanleitungen, eBooks, Bibliotheken und Unterstützung durch unsere technischen Experten.

https://az-delivery.de

Viel Spaß!

Impressum

https://az-delivery.de/pages/about-us