

Der etwas andere Raspberry Pi

Raspberry Pi Pico mit Raspberry Silicon

Es war ein Paukenschlag für die Mikrocontroller-Community, als James Adams von der Raspberry Pi Foundation Ende Januar 2021 den Verkaufsstart des brandneuen Raspberry Pi Pico (RPi Pico) bekannt gab. Der RPi Pico sieht ganz anders aus als der gewohnte komplette Single-Board-Computer. Der ist ein vielseitiges Mikrocontroller-Board mit einem eigens entwickelten Prozessor, dem RP2040 mit Dual-Core-ARM-Prozessor, reichlich Anschlussmöglichkeiten und UF2-Unterstützung für die einfache Drag-and-drop-Programmierung. Wir stellen das neue Board kurz vor.



Raspberry-Silicon erobert die Boards

Bisher basierten die Raspberry Pi-SBCs auf Chipsätzen von Broadcom, mit dem RP2040 wurde ein im eigenen Hause entwickelter SoC mit Dual-Core-ARM-Cortex-M0+ Prozessor und 264-kB-Onchip-RAM, getaktet mit 133 MHz vorgestellt. Er unterstützt bis zu 16 MB externen Flash-Speicher via QSPI-Bus, der RPi Pico selbst hat 2 MB Flash an Bord. Die Foundation nennt ihn selbst Raspberry Si – naturgemäß ist man stolz auf die Eigenentwicklung dieses auf besonders sparsamen Betrieb ausgelegten Prozessordesigns.

Das Ziel der Entwicklung war ein besonders vielseitig einsetzbares Universal-ARM-Board (u. a. mit acht frei programmierbaren State-Machines), das sehr einfach zu programmieren und mit geringstem Hardwareaufwand herzustellen sein soll. Letzteres schlägt sich auch im geringen Preis des Boards nieder: Es kostet nicht einmal 5 Euro [1]. Damit ist der RPi Pico geradezu eine Ideallösung für kleine Mikrocontroller-Projekte und für den Einsatz in Schule und Lehre.

Welche Bedeutung der RP2040-Chip erlangen könnte, sieht man allein schon an den diversen, parallel mit dem Erscheinen des RPi Pico angekündigten RP2040-Boards weiterer Hersteller wie Adafruit, SparkFun und Arduino (Bild 1), die faktisch alle der

zeit weit verbreiteten Plattform-Formate und Footprints abdecken. Besonders gespannt darf man auf den Arduino Nano RP2040 Connect sein: Er hat eine komplette WLAN-/Bluetooth-Einheit von u-Blox, eine 9-Achsen-MEMS-IMU und ein MEMS-Mikrofon. Alle RP2040-Boards dieser Hersteller haben gleich die Maximalausstattung von 16 MB Flash-Speicher an Bord.

Der RPi Pico kommt als einfaches Board, direkt vom Automaten-Reel geschnitten, ins Haus. Die Pin-Header muss der Anwender selbst beisteuern, diese lassen sich am einfachsten mithilfe eines Breadboards (Bild 2) einlöten. Leider findet man die sehr praktische und gut lesbare

Spezifikation RP2040

- Dual-Core-ARM-Cortex-M0+ Prozessor, bis 133 MHz Taktfrequenz
- Betriebsspannung 1,8 bis 5,5 V, interner 3,3-V-Schaltregler
- 264 kB SRAM (unterteilt in 6 Speicherbänke)
- 2 MB Flash-Speicher, extern per QSPI auf 16 MB erweiterbar
- 30 Multifunktions-GPIO (3,3 V, Funktion programmierbar)
- 2x SPI, 2x I2C, 2x UART, 4x 12-Bit-ADC, 16x PWM
- 1x RTC, 4x Timer, 1x Temperatursensor
- 2x programmierbare I/O-Blocks, 8x State-Machines
- USB1.1-Interface mit PHY, Betrieb als Host und Device möglich
- Emulation als Massenspeicher-Interface möglich
- Programmierung via USB-Massenspeicherfunktion (UF2-Bootloader) oder Serial-Wire-Debug-Port (SWD-Port)



 <p>Adafruit Feather RP2040</p> <p>21 GPIO LiPo-Akkulader 200 mA 4 MB QSPI-Flash STEMMA I2C-Anschluss für Qwiic/QT/Grove-Sensoren WS2812 onboard</p>	 <p>Adafruit ItsyBitsy RP2040</p> <p>23 GPIO 4 MB QSPI-Flash WS2812 onboard</p>	 <p>SparkFun Thing Plus RP2040</p> <p>Feather Footprint 18 GPIO LiPo-Akkulader 500 mA 16 MB QSPI-Flash I2C-Qwiic-Anschluss SD-Karten-Slot WS2812 onboard</p>	 <p>SparkFun Pro Micro RP2040</p> <p>Pro Micro Footprint 20 GPIO 16 MB QSPI-Flash WS2812 onboard</p>	 <p>SparkFun MicroMod RP2040 Processor</p> <p>MicroMod M.2-Processorboard für MicroMod-Boards 29 GPIO 16 MB QSPI-Flash</p>	 <p>Arduino Nano RP2040 Connect</p> <p>Arduino Nano Footprint WLAN/Bluetooth (u-blox NINA) 16 MB QSPI-Flash 9-Achs-MEMS-IMU MEMS Mikrophon OTA-fähig ECC608-Crypto-Chip Per Arduino-IDE programmierbar inkl. Bibliotheken</p>
--	---	--	--	--	---

Bild 1: Der RP2040 bildet die Basis für viele Mikrocontroller-Boards in den gängigsten Formfaktoren. Bilder: Adafruit Industries (2), SparkFun (3), Arduino Foundation (1)

Pin-Beschriftung nur auf der Unterseite des Boards (Bild 2 rechts).

Der Betrieb des RPi Pico kann mit Spannungen zwischen 1,8 und 5,5 V erfolgen, sie werden auf dem Board auf 3,3 V umgesetzt. Dies ist auch der System-Logikpegel. So kann man den Controller u. a. auch mit einer Lithiumzelle oder zwei 1,5-V-Zellen betreiben.

Einfach programmieren per Drag-and-drop

Wie wird ein solches Board programmiert? Ein Betriebssystem gibt es hier nicht, lediglich ein Bootloader bildet die Firmware ab Werk. Die Platine bietet ein neues Feature, einen Bootselect-Taster. Hält man diesen gedrückt und schließt dabei den RPi Pico per USB an einen Rechner an, bootet der RPi Pico als Massenspeicherlaufwerk (siehe Titelbild). Für das Schreiben und Übertragen der MicroPython-Programme wird die Thonny Python IDE [2], siehe Titelbild, eingesetzt.

Die Raspberry Pi Foundation hat von Anbeginn eine wirklich umfassende Unterstützung bereitgestellt. Man kann direkt vom gebooteten RPi Pico aus beginnen, denn hier ist der Link zu den Tutorials [3], [4] hinterlegt. Man lädt von dort die Python-Firmware, ein UF2-File, herunter. Dieses File wird per Drag-and-drop auf den RPi Pico kopiert, und dieser startet anschließend mit der nun neu geflashten Firmware wieder neu. Nun kann man erste Programme auf der IDE schreiben, auf dem RPi Pico abspeichern und sodann starten.

Der Einstieg, besonders für Programmieranfänger, an die sich ja der RPi Pico vor allem wendet, gelingt per MicroPython sehr einfach, zumal auch für andere Controller geschriebene, vorhandene Python-Pro-

gramme unmittelbar einsetzbar sind. Entsprechend rasant hat sich die Projektvielfalt von für den RPi Pico bereitgestellten Programmen bereits entwickelt, Bild 3 zeigt eine solche Anwendung. Ein Hackspace-Sonderheft [5] und das Buch „Get started with MicroPython on Raspberry Pi Pico“ [6] bieten weitere umfassende Hilfe zum Einstieg.

Zusätzlich bietet die Foundation eine umfassend dokumentierte C/C++ SDK, sodass auch erfahrene Programmierer hier schnell eigene Programme auf dem RPi Pico einsetzen können [7]. **ELV**



Weitere Infos:

- [1] RPi Pico bei ELV: Artikel-Nr. 251905
- [2] Thonny, Python-IDE for Beginners: <https://thonny.org/>
- [3] Dokumentation, Start mit MicroPython/C/C++, Projekte: <https://www.raspberrypi.org/documentation/pico/getting-started/>
- [4] Start mit dem RPi Pico: <https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico>
- [5] Sonderheft Hackspace „Introducing Pico“ zum Download: <https://hackspace.raspberrypi.org/>
- [6] Buch „Get started with MicroPython on Raspberry Pi Pico“ zum Download: <https://hackspace.raspberrypi.org/books/micropython-pico>
- [7] <https://github.com/raspberrypi/pico-sdk>

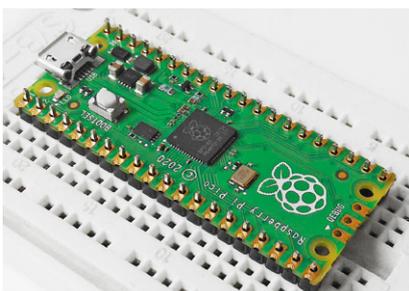


Bild 2: Auf einem Breadboard lassen sich die Headerleisten am besten einsetzen und verlöten. Rechts die Beschriftung der Anschlüsse auf der Unterseite des RPi Pico.

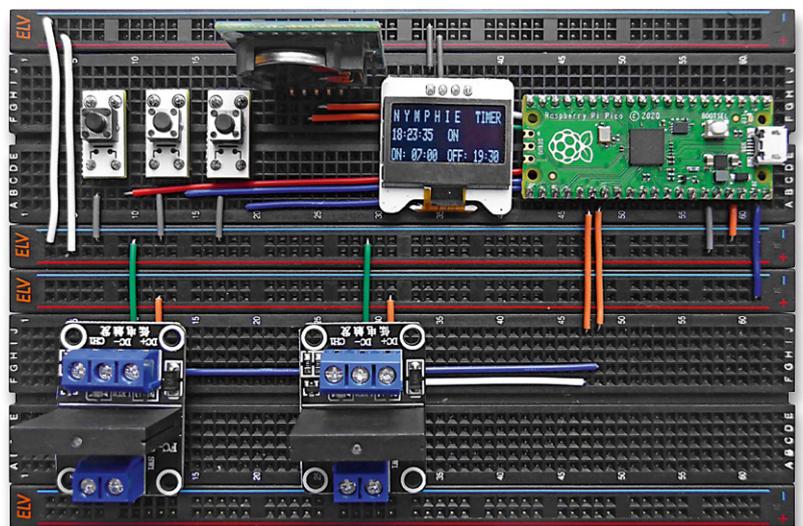


Bild 3: Per Python sind Anwendungen schnell realisiert. Hier ein Timer, der den Antrieb von Vorhängen an einer Voliere für Ziervögel steuert.