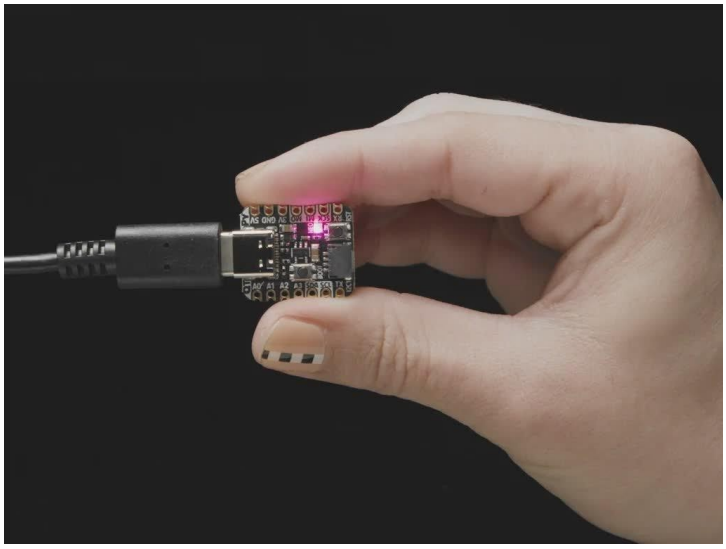
Adafruit QT Py RP2040



Artikel-Nr.:	ADA4900
Hersteller:	Adafruit
Herkunftsland:	USA
Zolltarifnummer:	85423119
Gewicht:	0.004 kg

Was für ein süßes Kerlchen! Oder ist es... ein QT Py? Dieses winzige Entwicklungsboard ist mit einem unserer neuen Lieblingschips ausgestattet, dem RP2040. Er wurde im neuen Raspberry Pi Pico und in unserem Feather RP2040 und ItsyBitsy RP2040 berühmt, aber was wäre, wenn wir etwas wirklich Kleines wollten?

Ein neuer Chip bedeutet einen neuen QT Py, und der Raspberry Pi RP2040 ist da keine Ausnahme. Als wir diesen Chip sahen, dachten wir: "Dieser Chip wird fantastisch sein, wenn wir ihm die QT Py-Behandlung verpassen", und das haben wir auch getan! Dieser QT Py verfügt über den RP2040 und alle Annehmlichkeiten, die Sie vom ursprünglichen QT Py kennen und lieben

PLUG-AND-PLAY STEMMA QT

Der Star des QT Py ist unser Lieblingsanschluss - der STEMMA QT, ein verkettbarer I2C-Anschluss, der mit allen unseren STEMMA QT-Sensoren und -Zubehörteilen verwendet werden kann. Dieser Anschluss bedeutet, dass Sie keine Lötarbeiten vornehmen müssen, um loszulegen.

Was kann man an den QT-Anschluss anschließen? Wie wäre es mit OLEDs! Trägheitsmessgeräte! Sensoren in Hülle und Fülle. Alles Plug-and-Play dank des innovativen kettbaren Designs: SparkFun Qwiic-kompatible STEMMA QT-Stecker für den I2C-Bus, so dass Sie nicht einmal löten müssen. Stecken Sie einfach ein kompatibles Kabel ein und verbinden Sie es mit der MCU Ihrer Wahl, und schon können Sie eine Software laden und Licht messen.

Verwenden Sie alle SparkFun Qwiic-Boards! Seeed Grove I2C Boards funktionieren ebenfalls mit diesem Adapterkabel.

SOFTWARE-UNTERSTÜTZUNG

Zum Zeitpunkt der Markteinführung gibt es keine Arduino-Core-Unterstützung für den Chip auf diesem Board. Es gibt eine großartige C/C++-Unterstützung, eine offizielle MicroPython-Portierung und eine CircuitPython-Portierung! Wir empfehlen natürlich CircuitPython, weil wir denken, dass es der einfachste Weg ist, um anzufangen, und weil es die meisten unserer Treiber, Displays, Sensoren und mehr unterstützt, und zwar von Anfang an, so dass Sie unseren CircuitPython-Projekten und -Tutorials folgen können.

QT PY RP2040 SPEZIFIKATIONEN

Pinout und Form sind kompatibel mit Seeed Xiao, mit wabenförmigen Pads, so dass Sie es auf eine Platine mit einem Ausschnitt löten können, um den unteren Komponenten etwas Spielraum zu geben. Neben dem QT-Anschluss haben wir auch einen RGB-NeoPixel (mit einem steuerbaren Power-Pin, um einen Ultra-Low-Power-Betrieb zu ermöglichen) und sowohl Boot-Mode- als auch Reset-Tasten (ideal für den Neustart Ihres Programms oder den Zugriff auf den Bootloader) hinzugefügt. Dieser QT Py wird mit losen 0,1"-Leitungen geliefert, die Sie für die Verwendung auf einem Breadboard einlöten können.

Der RP2040 hat zwar viel Onboard-RAM (264KB), aber keinen eingebauten FLASH-Speicher. Dieser wird stattdessen von einem

Alle hier aufgeführten Namen und Zeichen sind Eigentum des jeweiligen Herstellers. Druckfehler, Änderungen & Irrtümer vorbehalten.

externen QSPI-Flash-Chip bereitgestellt. Auf diesem Board gibt es 8MB, die zwischen dem laufenden Programm und dem von MicroPython oder CircuitPython verwendeten Dateispeicher geteilt werden. Bei der Verwendung von C/C++ steht der gesamte Flash-Speicher zur Verfügung, bei der Verwendung von Python verbleiben etwa 7 MB für Code, Dateien, Bilder, Schriftarten usw.

- Gleiche Größe, Formfaktor und Anschlussbelegung wie unser SAMD-basierter QT Py
- **USB-Typ-C-Anschluss**
- **RP2040 32-Bit Cortex M0+** Dual-Core mit ~125 MHz bei 3,3 V Logik und Leistung
- 264 KB RAM
- **8 MB SPI FLASH**-Chip zum Speichern von Dateien und CircuitPython/MicroPython-Code. Kein EEPROM
- Native USB-Unterstützung durch jedes Betriebssystem - kann als serielle USB-Konsole, MIDI, Tastatur/Maus-HID und sogar als kleines Laufwerk zum Speichern von Python-Skripten verwendet werden.
- Kann mit **MicroPython** oder **CircuitPython** verwendet werden
- **Eingebaute RGB NeoPixel LED**
- **13 GPIO-Pins (11 Breakout-Pads und zwei QT-Pads):**
  - **Vier** 12-Bit-ADCs (einer mehr als beim Pico)
  - Zwei I2C-Ports (einer am QT-Anschluss, einer an den Breakout-Pads)
  - SPI- und UART-Peripheriegeräte an den standardmäßigen QT Py-Positionen,
  - PWM-Ausgänge an jedem IO-Pin - für Servos, LEDs, etc.
  - Es gibt 6 GPIO in fortlaufender Reihenfolge für PIO-Kompatibilität
- 3,3V-Regler mit 600mA Spitzenausgang
- 12 MHz Quarz
- **Sowohl Reset-Taster als auch Bootloader-Auswahltasten** für schnelle Neustarts (kein Abziehen und Wiederanstecken, um den Code neu zu starten)
- **Wirklich sehr klein**

ÜBER DEN RP2040

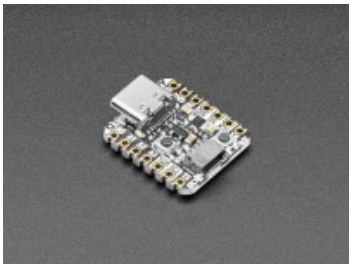
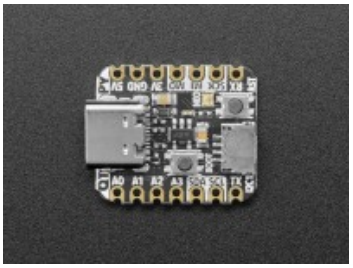
Im Inneren des RP2040 befindet sich ein **'permanenter ROM' USB UF2 Bootloader**. Das bedeutet, dass man, wenn man eine neue Firmware programmieren will, den BOOT-Knopf gedrückt halten kann, während man das Gerät an USB anschließt (oder den RUN/Reset-Pin auf Masse zieht), und es erscheint als USB-Laufwerk, auf das man die Firmware ziehen kann. Leute, die Adafruit-Produkte verwenden, werden dies sehr vertraut finden - wir verwenden diese Technik auf allen unseren nativen USB-Boards. Beachten Sie nur, dass Sie nicht auf Reset doppelklicken, sondern BOOTSEL während des Bootens gedrückt halten, um den Bootloader zu starten!

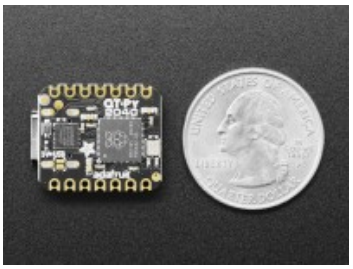
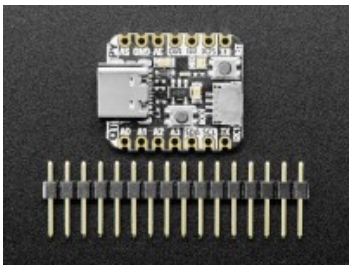
Der RP2040 ist ein leistungsfähiger Chip, der die Taktrate unseres M4 (SAMD51) hat und zwei Kerne, die unserem M0 (SAMD21) entsprechen. Da es sich um einen M0-Chip handelt, verfügt er nicht über eine Fließkommaeinheit oder DSP-Hardwareunterstützung - wenn Sie also etwas mit umfangreichen Fließkommaberechnungen machen, wird dies in Software erledigt und ist daher nicht so schnell wie ein M4. Bei vielen anderen Rechenaufgaben erreichen Sie annähernd M4-Geschwindigkeiten!

Für die Peripherie gibt es zwei I2C-Controller, zwei SPI-Controller und zwei UARTs, die über GPIO gemultiplext sind - welche Pins auf welche Weise belegt werden können, ist in der Pinbelegung angegeben. Es gibt 16 PWM-Kanäle, jeder Pin hat einen Kanal, auf den er eingestellt werden kann (dito in der Pinbelegung).

Sie werden feststellen, dass es kein I2S-Peripheriegerät, kein SDIO und keine Kamera gibt, was soll das? Nun, statt spezifischer Hardwareunterstützung für serielle datenähnliche Peripheriegeräte wie diese zu haben, kommt das RP2040 mit dem PIO-State-Machine-System, das eine einzigartige und leistungsstarke Möglichkeit bietet, benutzerdefinierte Hardwarelogik und Datenverarbeitungsblöcke zu erstellen, die eigenständig laufen, ohne eine CPU zu beanspruchen. Zum Beispiel NeoPixels - oft bitbangen wir das Timing-spezifische Protokoll für diese LEDs. Für den RP2040 verwenden wir stattdessen ein PIO-Objekt, das den Datenpuffer einliest und den richtigen Bitstream mit perfekter Genauigkeit ausgibt. Dasselbe gilt für I2S-Audioeingänge und -ausgänge, LED-Matrix-Displays, 8-Bit- oder SPI-basierte TFTs und sogar VGA! In MicroPython und CircuitPython können Sie PIO-Steuerbefehle erstellen, um das Peripheriegerät zu skripten und es zur Laufzeit zu laden. Es gibt 2 PIO-Peripheriegeräte mit jeweils 4 Zustandsautomaten.

Weitere Bilder:

Alle hier aufgeführten Namen und Zeichen sind Eigentum des jeweiligen Herstellers. Druckfehler, Änderungen & Irrtümer vorbehalten.