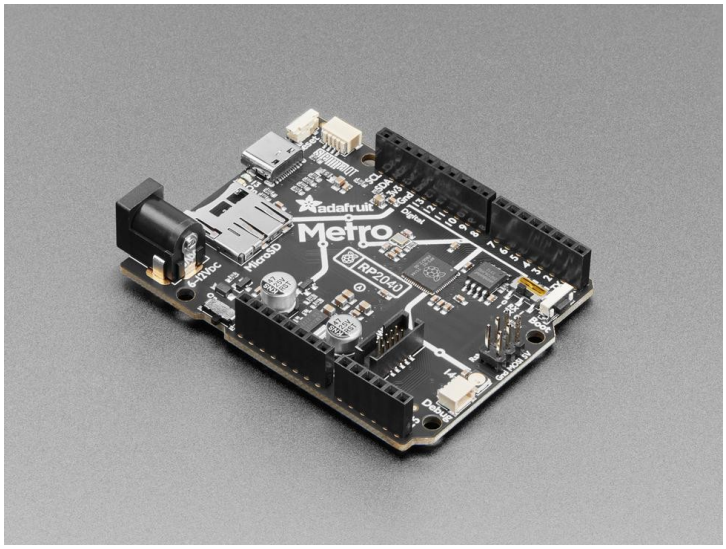
Adafruit Metro RP2040



Artikel-Nr.:	ADA5786
Hersteller:	Adafruit
Herkunftsland:	China
Zolltarifnummer:	85423111
Gewicht:	0.022 kg

Choo! Choo! Willkommen in der RP2040 Metro Line, die an allen Haltestellen "Dual Cortex M0+ Mountain", "264K RAM Round-About" und "16 Megabytes of Flash Town" hält. Dieser Zug ist mit Hardware beladen, die den Raspberry Pi RP2040-Chip ergänzt und ihn zu einem ausgezeichneten Entwicklungsboard für Projekte macht, die Arduino-kompatible Formfaktoren erfordern oder einfach mehr Platz und Debugging-Ports benötigen.

- **RP2040 Hauptchip**, 133 MHz Taktfrequenz, 3,3V Logik
- **16 MB QSPI Flash** für Programmspeicher
- **24 GPIO**, 4 davon sind auch analoge Eingänge
- **Micro-SD-Kartensteckplatz** für SPI-Interfacing, hat auch zusätzliche Pins für erweitertes SDIO-Interfacing (beachte, dass es keinen veröffentlichten Verwendungscode für SDIO in Arduino/Python gibt, daher handelt es sich um eine super moderne Einrichtung)
- Integrierter **RGB NeoPixel**
- Integrierte **LED #13**
- **Stemma QT**-Port für I2C-Peripheriegeräte und Sensoren
- **Reset- und Boot-Tasten** am PCB-Rand
- **Pico Probe** Debug-Port - 3-polig JST SH-kompatibel
- **SWD Debug** Port - 2x5 0,05" Standard-Anschluss
- **USB Type C** für Strom und Daten
- 5,5mm / 2,1mm **DC-Buchse** für 6-12V DC-Stromversorgung
- **Ein-/Ausschalter** für DC-Buchse
- GPIO-Pin-Nummern entsprechen klassischen Arduino-Pins, außer A4/A5, die D24 und D25 sind (es gibt nur 4 ADC-Pins auf dem RP2040)
- RX / TX-Schalter zum Umschalten der Positionen von D0 und D1

Du fragst dich vielleicht nach dem RX-TX-Schalter: Wir haben ihn hinzugefügt, weil traditionelle Arduino-Boards die GPIOs für die digitalen Pins mit 0-7 und dann 8-13 zählen. Die D0/D1-Pins sind jedoch auch traditionell die Hardware-UART Serial1, wobei D0 Rx und D1 Tx ist. Auf dem RP2040 sind die UART-Pins jedoch vertauscht: D0 ist Tx und D1 ist Rx. Daher ein DPDT-Schalter: In einer Richtung, um die GPIOs in der Reihenfolge von 0-7 zu haben, in die andere Richtung, um die logischen Positionen der Hardware-UART korrekt zu haben, aber jetzt ist die Pin-Reihenfolge 1, 0, 2, 3...7. Natürlich ist es auch praktisch, wenn du, wie wir, die Pins oft tauschst - jetzt musst du keine Leiterbahnen mehr schneiden/löten!

Über den RP2040
Der RP2040 ist ein leistungsstarker Chip, der über die Taktfrequenz unseres M4 (SAMD51) und zwei Kerne verfügt, die unserem M0 (SAMD21) entsprechen. Da es sich um einen M0-Chip handelt, verfügt er nicht über eine Gleitkommaeinheit oder DSP-Hardware-Unterstützung - daher wird, wenn du etwas mit aufwändigen Gleitkommaberechnungen durchführst, dies in Software erledigt und somit nicht so schnell wie bei einem M4. Für viele andere Berechnungsaufgaben erhältst du jedoch Geschwindigkeiten, die nahe an M4 heranreichen!

Für Peripheriegeräte gibt es zwei I2C-Controller, zwei SPI-Controller und zwei UARTs, die multiplex über die GPIOs verlaufen - überprüfe die Pinbelegung, um herauszufinden, welche Pins auf welche Weise eingestellt werden können. Es gibt 16 PWM-Kanäle, jedem Pin kann ein Kanal zugewiesen werden (Gleiches gilt für die Pinbelegung).

Du wirst feststellen, dass kein I2S-Peripheriegerät, keine SDIO oder keine Kamera vorhanden sind. Was ist damit los? Nun, anstelle einer spezifischen Hardware-Unterstützung für serielle Daten-ähnliche Peripheriegeräte wie diese, wird der RP2040 mit dem PIO-State-Machine-System geliefert, das eine einzigartige und leistungsstarke Möglichkeit bietet, *benutzerdefinierte Hardware-Logik und Datenverarbeitungsblöcke* zu erstellen, die eigenständig arbeiten, ohne die CPU zu belegen. Zum Beispiel NeoPixels - oft bitbangen wir das zeitkritische Protokoll für diese LEDs. Für den RP2040 verwenden wir stattdessen ein PIO-Objekt, das die Datenpuffer einliest und den richtigen Bitstrom mit perfekter Genauigkeit ausgibt. Das Gleiche gilt für I2S-Audio Ein- oder Ausgabe, LED-Matrix-Anzeigen, 8-Bit- oder SPI-basierte TFTs, sogar VGA! In MicroPython und CircuitPython kannst du PIO-Steuerbefehle erstellen, um das Peripheriegerät zu steuern und es zur Laufzeit zu laden. Es gibt 2 PIO-Peripheriegeräte mit jeweils 4 Zustandsmaschinen.

Es gibt großartige C/C++-Unterstützung, inoffizielle (aber wirklich gute) Arduino-Unterstützung, eine offizielle MicroPython-Portierung und eine CircuitPython-Portierung! Wir empfehlen natürlich CircuitPython, weil wir glauben, dass es der einfachste Weg ist, um loszulegen, und es unterstützt die meisten unserer Treiber, Displays, Sensoren und mehr, die sofort einsatzbereit sind, damit du unseren CircuitPython-Projekten und -Tutorials folgen kannst.

Der RP2040 verfügt zwar über viel integrierten RAM (264KB), hat jedoch keinen eingebauten FLASH-Speicher. Stattdessen wird dies durch den externen QSPI-Flash-Chip bereitgestellt. Auf diesem Board befinden sich 16 MB, die zwischen dem laufenden Programm und dem von MicroPython oder CircuitPython verwendeten Dateispeicher geteilt werden. Bei der Verwendung von C/C++ steht dir der gesamte Flash-Speicher zur Verfügung. Wenn du Python verwendest, bleiben etwa 7 MB für Code, Dateien, Bilder, Schriftarten usw. übrig. **RP2040 Chip-Funktionen:**

- Dual ARM Cortex-M0+ @ 133 MHz
- 264KB integrierter SRAM in sechs unabhängigen Banken
- Unterstützung von bis zu 16 MB externem Flash-Speicher über einen dedizierten QSPI-Bus
- DMA-Controller
- Vollständig verbundenes AHB-Kreuz
- Interpolator- und Integer-Divider-Peripheriegeräte
- Integrierter programmierbarer LDO zur Erzeugung der Kernspannung
- 2 integrierte PLLs zur Erzeugung von USB- und Kern-Takten
- 30 GPIO-Pins, 4 davon können als analoge Eingänge verwendet werden
- Peripheriegeräte
    - 2 UARTs
    - 2 SPI-Controller
    - 2 I2C-Controller
    - 16 PWM-Kanäle
    - USB 1.1-Controller und PHY mit Host- und Geräteunterstützung
    - 8 PIO-Zustandsmaschinen

Weitere Bilder: