



## Pimoroni Servo 2040 - 18 Channel Servo Controller



**PIMORONI**

<b>Order number:</b>	PIM613
<b>Hersteller:</b>	Pimoroni
<b>EAN:</b>	0769894021439
<b>Herkunftsland:</b>	Großbritannien
<b>Zolltarifnummer:</b>	84733080
<b>Gewicht:</b>	0.02 kg

Baue den Hexapod/Roboterarm/eine andere gelenkige Konstruktion deiner Träume mit diesem All-in-One RP2040 Servoregler mit Strommessung, Sensorleisten und RGB-LEDs.

Der Servo 2040 ist ein **Standalone-Servoregler** für den Bau von Dingen mit vielen beweglichen Teilen. Er hat vorgelötete Stiftleisten für den Anschluss von **bis zu 18 Servos** - genug für den langbeinigsten Hexapod oder jede Menge Freiheitsgrade für deine Roboterarme, Beine oder Tentakel. Servos können ziemlich stromhungrig sein, vor allem die klobigen, deshalb haben wir ein paar nette **Stromüberwachungsfunktionen** eingebaut, damit du den Stromverbrauch im Auge behalten kannst.

Es gibt **sechs adressierbare RGB-LEDs** (auch bekannt als Neopixels) für visuelles Feedback und Statusmeldungen sowie Stiftleisten für den Anschluss von **bis zu sechs analogen Sensoren** - nützlich, um zu erkennen, wo der Boden ist, ob du gegen eine Wand prallst oder wie viel Druck *die Klaue* auf deine unglückliche Testperson ausübt. Wir haben auch einen **QW/ST-Anschluss** eingebaut, damit du ganz einfach ein Qwiic oder STEMMA QT anschließen kannst.

Der Servo 2040 wird von einer gut dokumentierten C++/MicroPython Servo-Bibliothek mit vielen Beispielen unterstützt, die dir zeigen, wie du die einzelnen Funktionen (und alles zusammen) nutzen kannst.

### RP2040 x Servos

Wir haben das RP2040 als Herzstück dieses Boards verwendet, weil es so flexibel ist und über programmierbare IOs (PIOs) verfügt. Normalerweise muss jedes Servo an einen eigenen PWM-fähigen Kanal des Mikrocontrollers angeschlossen werden. Das RP2040 hat nur 16 PWM-Kanäle, aber es ist möglich, bis zu 30 Servos mit Hilfe der PIOs anzusteuern (wenn du dich mit der Verkabelung auskennst). Die PIOs des RP2040 sind außerdem superschnell und können Servos mit einer Auflösung von unter einer Mikrosekunde ansteuern.

Wir haben den RP2040-Mikrocontroller direkt in den Servo 2040 eingebaut, sodass du keine separaten Mikrocontroller- und Servotreiberplatinen benötigst. Das ermöglicht eine kompakte Bauweise - perfekt für kleine Roboter!

### Merkmale

- Powered by RP2040 (Dual Arm Cortex M0+ mit bis zu 133Mhz und 264kB SRAM)
- 2MB QSPI-Flash mit Unterstützung für XiP
- 18 Sätze von Header-Pins für den Anschluss von 3-poligen Hobbyservos
- Unterstützt Servos mit höherer Spannung (bis zu 11V) \*
- 6 adressierbare RGB LEDs/Neopixels
- 6 Sätze Header-Pins für den Anschluss analoger Sensoren



- Eigene Spannungs- und Strommessung
- Reset- und BOOT-Taste (die BOOT-Taste kann auch als Benutzertaste verwendet werden)
- USB-C-Anschluss für Programmierung und Stromversorgung (max. 3A)
- Schraubklemmen für die externe Stromversorgung (mit Verpolungsschutz) (10A max. Dauerstrom)
- Qw/ST (Qwiic/STEMMA QT) Anschluss für Breakouts
- Fertig montiert (kein Löten erforderlich)
- [C++/MicroPython Bibliotheken](#)
- [Schematic](#)
- [Mechanische Zeichnung](#)

## Software

Da es sich um ein RP2040 Board handelt, ist der Servo 2040 unabhängig von der Firmware! Du kannst es mit C/C++, MicroPython oder CircuitPython programmieren.

Unsere **C++/MicroPython** Bibliotheken helfen dir, das Beste aus dem Servo 2040 herauszuholen, denn sie sind vollgepackt mit leistungsstarken Funktionen für die Arbeit mit Servos. Die beste Leistung erzielst du mit C++, aber wenn du ein Anfänger bist, empfehlen wir dir, unseren MicroPython-Baukasten zu verwenden, um dir den Einstieg zu erleichtern.

- [Piratenmarke MicroPython herunterladen](#)
- [MicroPython API Dokumentation](#)
- [MicroPython Beispiele](#)
- [C++ Beispiele](#)

Du kannst auch **CircuitPython** auf deinem Servo 2040 verwenden, wenn du Zugang zu allen Annehmlichkeiten des Adafruit-Ökosystems haben möchtest (beachte, dass du mit CircuitPython nur bis zu 16 Servos steuern kannst).

- [Download CircuitPython für Servo 2040](#)
- [Einstieg in CircuitPython](#)
- [CircuitPython Beispiele](#)

## Verbinden von Breakouts

Wenn dein Breakout einen QW/ST-Anschluss JST-SH auf JST-SH Kabel hat, oder du kannst jedes unserer I2C Breakout Garden Breakouts mit einem JST-SH auf JST-SH Kabel verbinden; gekoppelt mit einem Qw/ST auf Breakout Garden-Adapter.

- [Liste der Breakouts](#), die derzeit mit unserem C++/MicroPython Build kompatibel sind.

## Hinweise

- Abmessungen: 62mm x 42mm x 12mm (L x B x H, einschließlich Steckern). Die Befestigungslöcher sind M2,5 und 2,7 mm von jeder Kante entfernt.
- \* Wenn du Servos mit einer höheren Spannung als 5V betreiben willst, musst du die Leiterbahn "Separate USB and Ext. Power" auf der Rückseite der Platine durchtrennen, um zu verhindern, dass das RP2040 durch die höhere Spannung beschädigt wird.
- Wenn du diese Leiterbahn durchschneidest, musst du die Logik auf der Platine separat mit Strom versorgen (über USB oder die 5 V auf dem ausgebrochenen Header).
- Wenn du einen batteriebetriebenen Roboter über USB programmierst, empfehlen wir dir, einen Nur-Daten-USB-Adapter zu verwenden, um eine Rückspeisung deines Computers oder der Batterie zu vermeiden.

## Über den RP2040

Der RP2040 Mikrocontroller des Raspberry Pi ist ein Dual-Core ARM Cortex M0+, der mit bis zu 133Mhz läuft. Er verfügt über 264kB SRAM, 30 Multifunktions-GPIO-Pins (einschließlich eines vierkanaligen 12-Bit-ADC), einen Haufen Standard-Peripheriegeräte (I2C, SPI, UART, PWM, Clocks usw.) und USB-Unterstützung.

Ein sehr interessantes Merkmal des RP2040 sind die programmierbaren IOs, die es dir ermöglichen, eigene Programme auszuführen, die GPIO-Pins manipulieren und Daten zwischen Peripheriegeräten übertragen können - sie können Aufgaben auslagern, die hohe Datenübertragungsraten oder ein präzises Timing erfordern, für die normalerweise eine Menge Arbeit von der CPU verlangt worden wäre.



Weitere Bilder:

