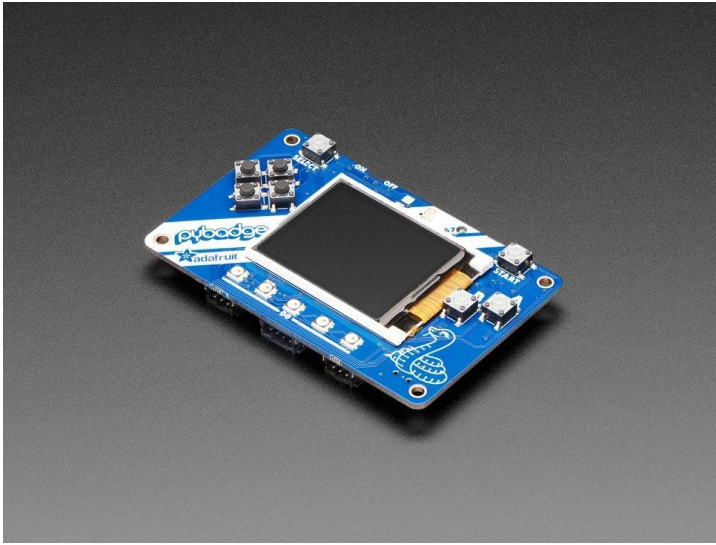




Adafruit PyBadge - MakeCode Arcade, CircuitPython, Arduino



Artikel-Nr.:	ADA4200
Hersteller:	Adafruit
Herkunftsland:	USA
Zolltarifnummer:	84715000
Gewicht:	0.032 kg

Was ist so groß wie eine Kreditkarte und kann CircuitPython, MakeCode Arcade oder Arduino ausführen? Das ist richtig, es ist das **Adafruit PyBadge!** Wir wollten sehen, wie viel wir in ein 8,5 x 2,1 cm großes, abgerundetes Rechteck packen können, um ein All-in-One-Dev-Board mit vielen Möglichkeiten zu schaffen, und das ist, was wir herausgefunden haben.

Das PyBadge ist ein kompaktes Board, wie wir schon sagten, es ist kreditkartengroß. Es wird von unserem Lieblingschip angetrieben, dem ATSAM51, mit 512KB Flash und 192KB RAM. Wir fügen 2 MB QSPI-Flash für Dateispeicher hinzu, praktisch für Bilder, Fonts, Sounds oder Spiel-Assets.

Auf der Vorderseite befindet sich ein 1,8" 160x128 Farb-TFT-Display mit dimmbarer Hintergrundbeleuchtung - wir haben schnelle DMA-Unterstützung für das Zeichnen, so dass Updates unglaublich schnell sind. Außerdem gibt es 8 Silikontasten, die zwar klickbar sind, aber eine weiche Tastenoberfläche haben, so dass sie schön griffig sind. Die Tasten sind so angeordnet, dass sie einem Gaming-Handheld nachempfunden sind, mit einem D-Pad, 2 Menü-Auswahltasten und 2 Feuer-Aktions-Tasten. Außerdem gibt es 5 NeoPixel-LEDs, um zu blenden oder die Aktivität zu verfolgen.

Auf der Rückseite haben wir ein komplettes Feather-kompatibles Header-Socket-Set, so dass Sie jeden FeatherWing anschließen können, um die Fähigkeiten des PyBadge zu erweitern. Es gibt auch 3 STEMMA-Anschlüsse - zwei 3-polige mit ADC/PWM-Fähigkeit und einen 4-poligen, der mit I2C verbunden ist - Sie können diesen auch für Grove-Sensoren verwenden.

An eingebauten Sensoren gibt es einen Lichtsensor, der nach vorne zeigt, und einen 3-Achsen-Beschleunigungsmesser, der Taps und freien Fall erkennen kann. Für Piep- und Blubbergeräusche gibt es einen eingebauten Buzzer-Lautsprecher. Für Projekte, bei denen Sie mehr Lautstärke benötigen, können Sie einen 8-Ohm-Lautsprecher anschließen.

Sie können den PyBadge mit einem LiPoly-Akku betreiben, aber wir mögen diesen 400mAh Akku. Ein Ein-Aus-Schalter spart Batteriestrom, wenn er nicht gebraucht wird. Oder Strom über den Micro-USB-Anschluss - der lädt auch den Akku auf, wenn einer angeschlossen ist.

Und wie programmiert man es nun? Nun, da haben Sie eine Menge Möglichkeiten!

- **MakeCode Arcade ist der einfachste Einstieg zum Erstellen von Spielen**, [man kann Blöcke per Drag-and-Drop ziehen und Spiele über den Bootloader des Laufwerks laden](#)
- **CircuitPython** lässt Sie Grafiken zeichnen, Wave-Dateien abspielen und Text in beliebigen Schriftarten ausgeben - alles in Python! Es gibt auch jede Menge Sensor-Unterstützung.
- Arduino ist Low-Level, leistungsfähig, aber ein wenig anspruchsvoller. [Sie können Adafruit Arcada](#) verwenden, um mit der Hardware zu interagieren, und es abstrahiert einige der kleinsten Details wie das Lesen von Tasten für Sie.

Hier ist eine Liste von allem, was Sie bekommen

- **ATSAMD51J19 @ 120MHz** mit 3,3V Logik/Strom - 512KB FLASH + 192KB RAM



- **2 MB SPI Flash** zum Speichern von Bildern, Sounds, Animationen, was auch immer!
- **1,8" 160x128 Farb-TFT-Display**, angeschlossen an einen eigenen SPI-Port
- **8 x Spiel-/Steuerungstasten** mit schönen Silikonastenaufsätzen (diese fühlen sich toll an)
- **5 x NeoPixels** für Abzeichenblendung oder Spielstandserfassung
- **Drei-Achsen-Beschleunigungsmesser** (Bewegungssensor)
- **Lichtsensord**, rückseitig montiert, so dass er nach vorne zeigt
- **Eingebauter Buzzer-Mini-Lautsprecher**
- **Mono Class-D Lautsprechertreiber** für 4-8 Ohm Lautsprecher, bis zu 2 Watt
- LiPoly-Akku-Anschluss mit integrierter Aufladefunktion
- USB-Anschluss für Batterieladung, Programmierung und Debugging
- Zwei Buchsenleisten mit Feather-kompatibler Pinbelegung, so dass Sie jeden FeatherWings anschließen können
- JST-Anschlüsse für NeoPixels, Sensoreingang und I2C (hier können Sie I2C Grove-Stecker einbauen)
- Reset-Taster
- Ein-Aus-Schalter

Weitere Bilder:

