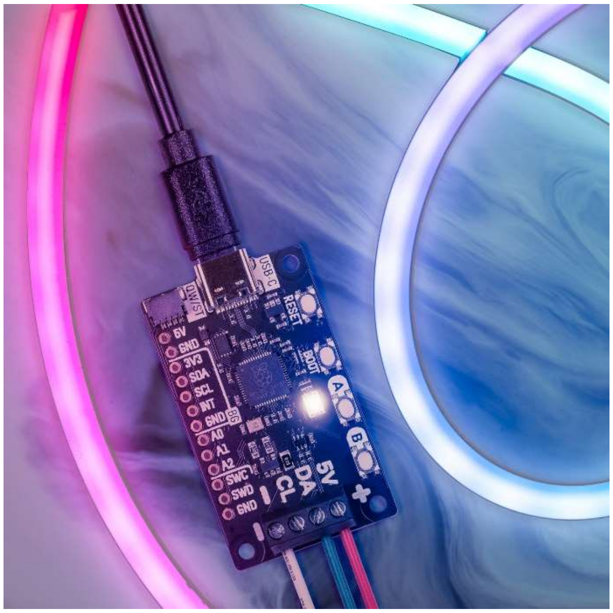
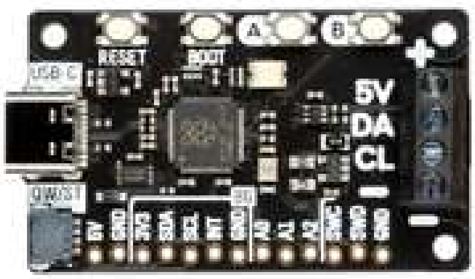
PIMORONI

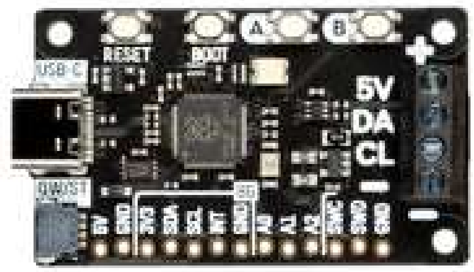
Purveyors of Maker Goods

Plasma 2040

PIM582







Swathe everything in rainbows with this all-in-one, USB-C powered controller for WS2812/Neopixel and APA102/Dotstar addressable LED strip.

Plasma 2040 is a RP2040-based driver board for addressable LED strip (also known as magical rainbows by the meter). It's designed to make rigging up bits of **custom, programmable lighting** as straightforward as possible - perfect for whipping up some quick under-cupboard illumination, dramatically underlighting your sofa or providing some atmospheric mood-lighting for your workspace, PC or vivarium.

Plasma 2040 is **powered and programmable by USB-C** and, because USB-C can draw up to 3A of power, that's enough to power a healthy chunk of LEDs. There are three **useful buttons** that you could use for turning the lights on and off or to switch between colors or effects, plus a reset button and an eternally popular **onboard RGB LED**.

We've also popped a **QW/ST connector** on there, to make it super easy to plug in Qwiic or STEMMA QT breakouts - how about hooking up an RGB potentiometer to adjust the speed or hue of your lights, an air quality sensor to make your desk lighting into a giant thermometer or tell you when you should crack a window or a light sensor to turn them on automatically when it gets dark?

Features

- Powered by RP2040 (Dual Arm Cortex M0+ running at up to 133Mhz with 264kB of SRAM)
- 2MB of QSPI flash supporting XiP
- Compatible with 5V WS2812/Neopixel/SK6812 and APA102/Dotstar/SK9822 LEDs
- Screw terminals for attaching your LED strip.
- USB-C connector for power and programming (3A max)
- Qw/ST (Qwiic/STEMMA QT) connector
- Low side current sensing (accessible via ADC3)
- Reset, BOOT and two user buttons (the BOOT button can also be used as a user button)
- RGB LED
- Fully assembled (no soldering required)
- Measurements: approx. 50 x 28 x 12mm (L x W x H, including connectors)
- C++/MicroPython libraries
- Schematic
- Dimensions

LED strip and connectors are sold separately, check out the extras tab for some options!

Software

You can program Plasma 2040 with **C/C++** or **MicroPython** in the same way as you would a Raspberry Pi Pico. You'll get best performance using C++, but if you're a beginner we'd recommend using our batteries included MicroPython build for ease of getting started.

- Download Pirate brand MicroPython (you'll need the standard `pimoroni-pico` .uf2)
- MicroPython examples
- MicroPython function reference
- C++ examples

You can also use **CircuitPython** on your Plasma 2040! CircuitPython is an easy to use, well-established ecosystem with lots of example code and drivers for interfacing with different kinds of hardware.

- Download CircuitPython for Plasma 2040
- Welcome to CircuitPython

Check out our **beginner friendly tutorial** that covers how to hook up different kinds of LEDs to Plasma 2040 and how to use it to build a simple busy light. It includes both MicroPython and CircuitPython code!

- Getting Started with Plasma 2040

Connecting Breakouts

If your breakout has a QW/ST connector on board, you can plug it straight in with a JST-SH to JST-SH cable, or you can easily connect any of our I2C breakouts with a JST-SH to JST-SH cable coupled with a Qw/ST to Breakout Garden adaptor.

- Breakouts currently compatible with our C++/MicroPython build

We've also broken out a set of I2C pins, analog pins and debug pins so you can solder things like breakouts or analog potentiometers directly to them (or solder on a strip of header and plug the whole shebang into a breadboard).

Notes

- Our C++/MicroPython software uses the RP2040's PIO state machines to drive each strip separately - this board only has one set of LED strip connectors, but if you're up for some inventive wiring it's possible to drive multiple strips simultaneously, even if they're of different types!
- If you're curious about how much current your LEDs are consuming, we've incorporated some current sensing circuitry onto this board, which you can measure by reading ADC3. You could use this in your code to do things like adjusting the brightness of your LEDs based on available power!

About RP2040

Raspberry Pi's RP2040 microcontroller is a dual core ARM Cortex M0+ running at up to 133Mhz. It bundles in 264kB of SRAM, 30 multifunction GPIO pins (including a four channel 12-bit ADC), a heap of standard peripherals (I2C, SPI, UART, PWM, clocks, etc.), and USB support.

One very exciting feature of RP2040 is the programmable IOs which allow you to execute custom programs that can manipulate GPIO pins and transfer data between peripherals - they can offload tasks that require high data transfer rates or precise timing that traditionally would have required a lot of heavy lifting from the CPU.