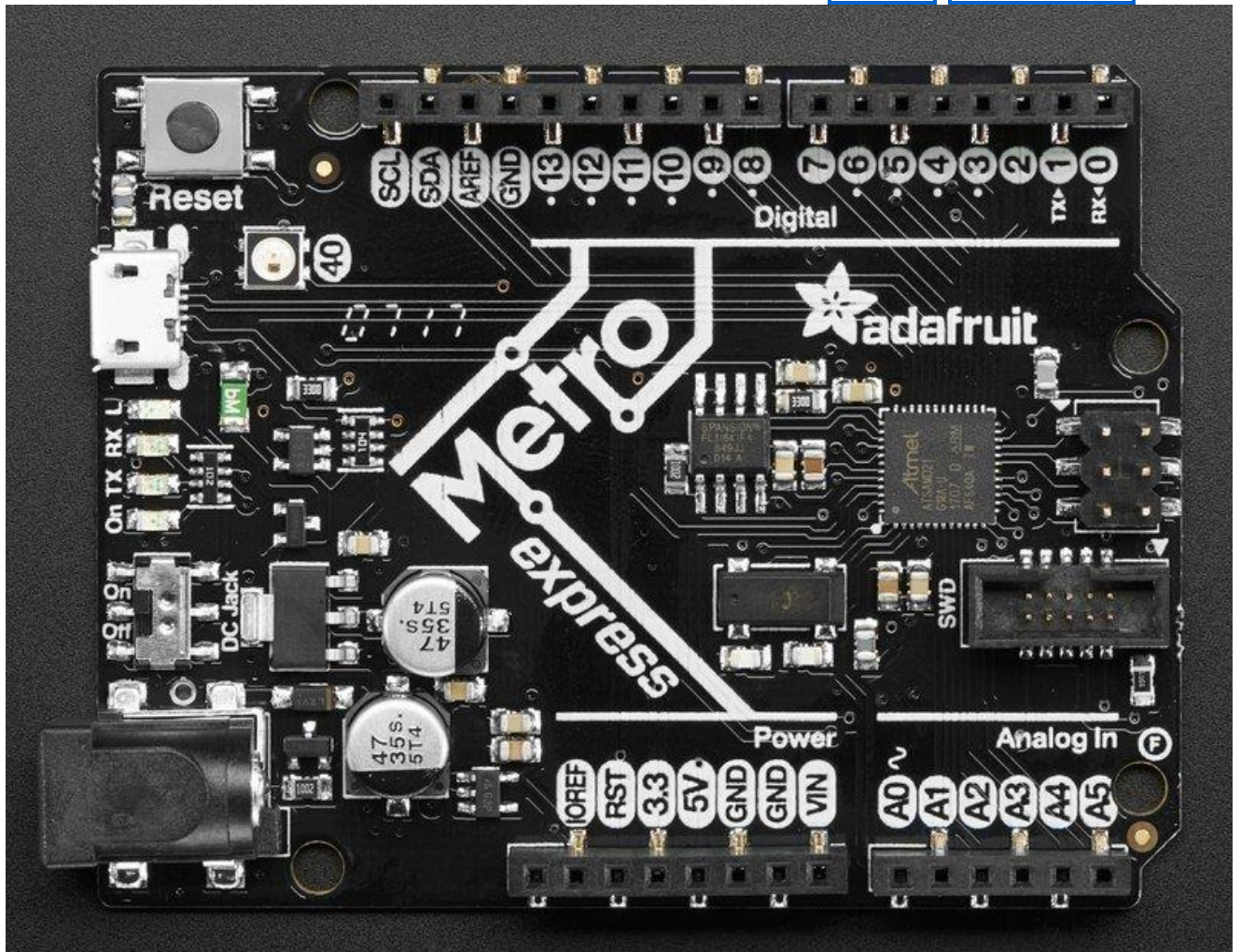


<https://learn.adafruit.com/adafruit-metro-m0-express-designed-for-circuitpython/pinouts>

Pinouts

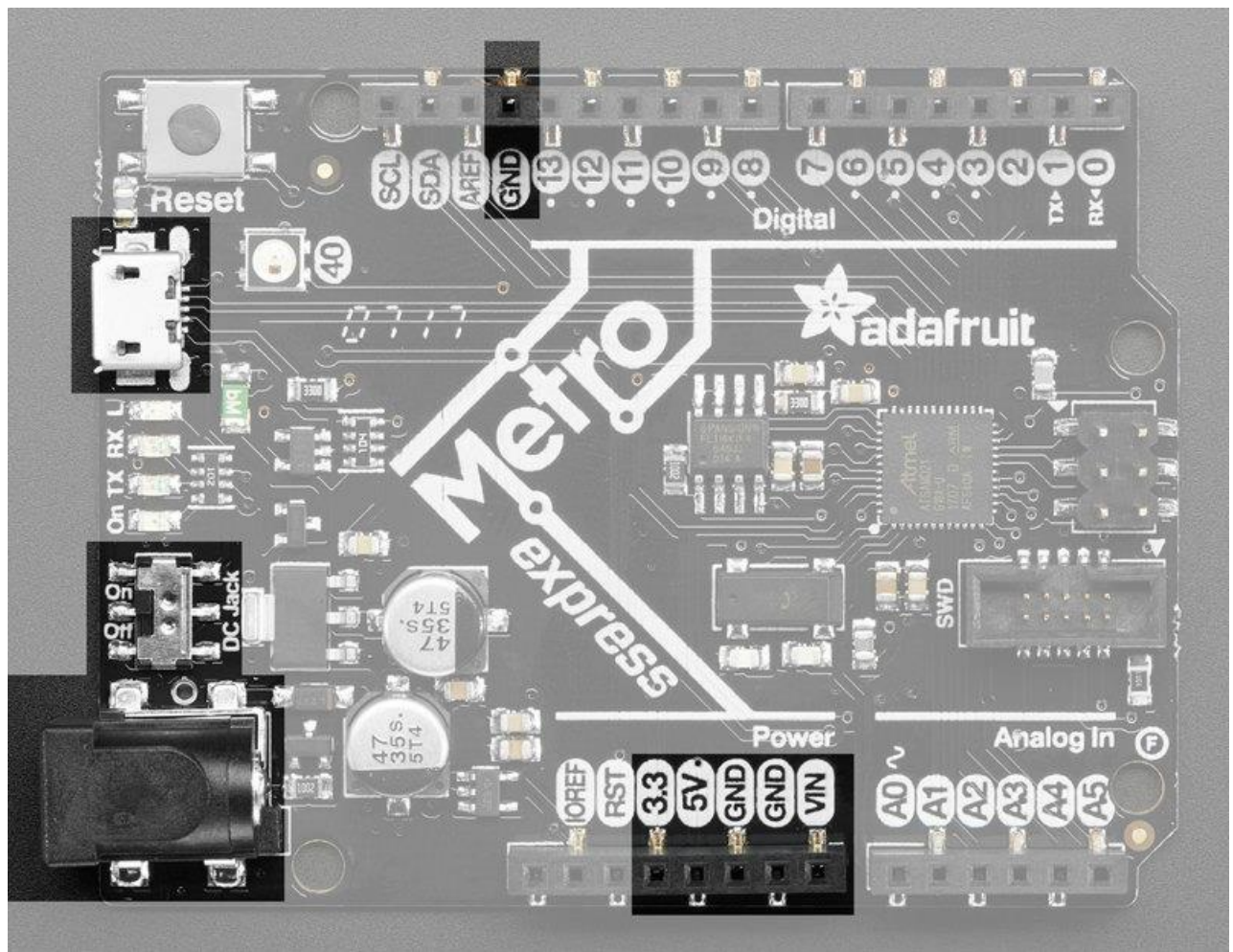
[Save](#)

[Subscribe](#)



The Metro M0 is chock-full of microcontroller goodness. There's also a lot of pins and ports. We'll take you a tour of them now!

Power Connections



There's a lot of ways to power the Metro M0 Express, and a lot of ways to *get* power out as well.

There are two primary ways to power the Metro:

- Through the Micro USB port up at the top left
- Through the DC jack at the bottom left

The MicroUSB jack provides 5V at 500mA or so, there is a fuse that will shut off temporarily when more than 1000mA is drawn, this is to protect a computer USB port. You can plug this into any computer or USB charger with a USB cable. You can draw up to 500mA between the Vin, 5V and 3.3V supplies (combined).

The DC Jack is a 5.5mm/2.1mm center-positive DC connector, which is the most common available. Provide about 6V–12V here

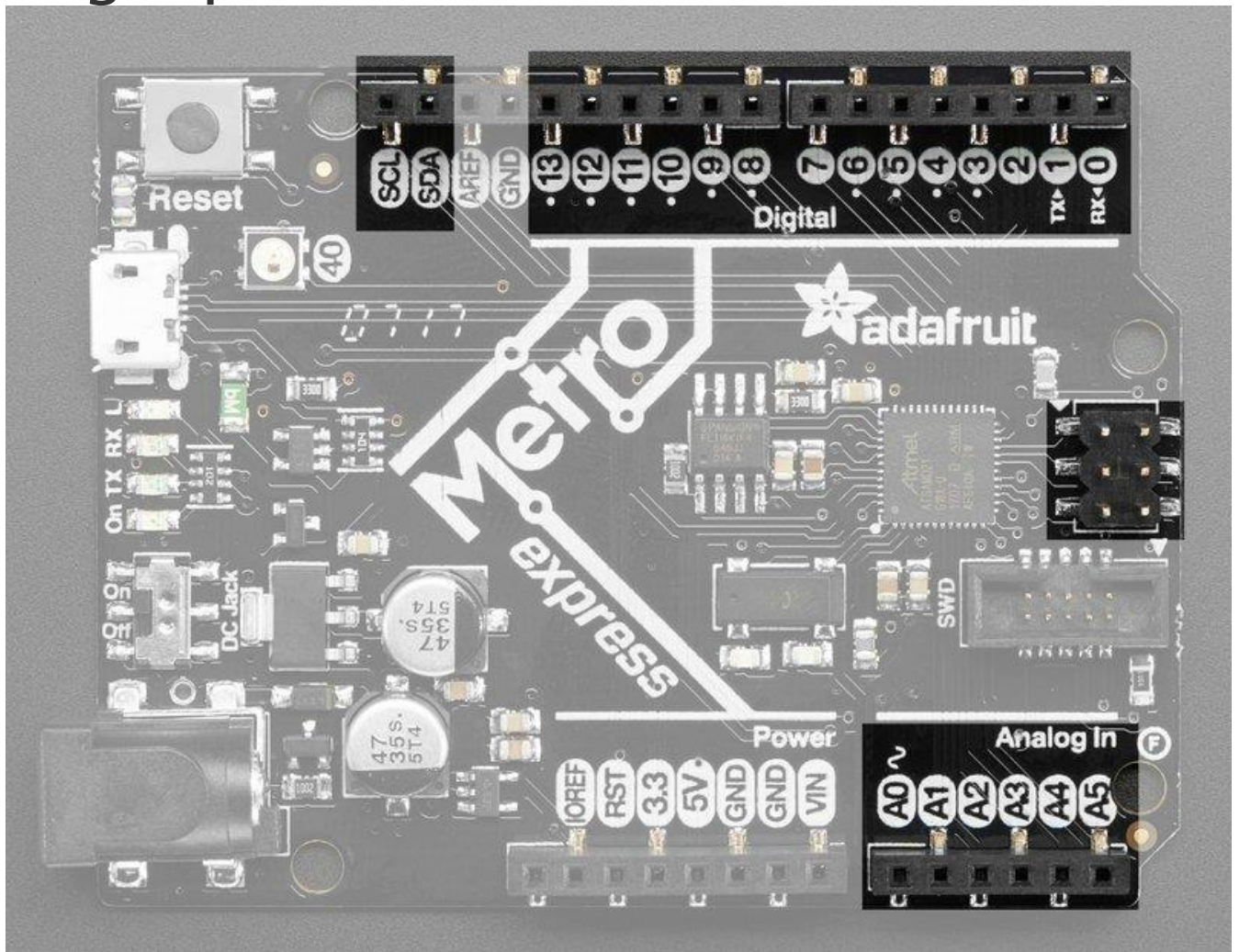
to power the Metro. There is no fuse on this connection so you can draw more current, up to 800mA between the 5V and 3.3V supplies, and 2A from **Vin**.

Onboard regulators take the USB or DC power and linearly convert it to **3.3V** and **5V**:

- **3V** – this is the output from the 3.3V regulator, it can supply 500mA peak
- **5V** – this is the output from the 5V regulator (when DC jack is used), or from USB. It can supply ~500mA peak from USB and ~800mA peak from DC
- **GND** – this is the common ground for all power and logic
- **Vin** – this is the *higher* of the DC jack or USB voltage. So if the DC jack is plugged in and 9V, Vin is 9V. If only USB connected, this will be 5V.

There is also an on/off switch. This switch is only for the DC jack and does not affect powering via USB

Logic pins



This is the general purpose I/O pin set for the microcontroller.

All logic is 3.3V

Most pins can do PWM output

All pins except D4 can be interrupt inputs

Top Row

- #0 / RX – GPIO #0, also receive (input) pin for Serial1 (hardware UART)
- #1 / TX – GPIO #1, also transmit (output) pin for Serial1
- #2 through #12 – These are general purpose GPIO. If there's a dot next to the pad it can act as a PWM output.
- #13 – GPIO #13 and is connected to the red LED marked L next to the USB jack

- **SDA** – the I2C (Wire) data pin. There's no pull up on this pin by default so when using with I2C, you may need a 2.2K–10K pullup.
- **SCL** – the I2C (Wire) clock pin. There's no pull up on this pin by default so when using with I2C, you may need a 2.2K–10K pullup.

Bottom Row

- **A0** – This pin is analog *input* **A0** but is also an analog *output* due to having a DAC (digital-to-analog converter). You can set the raw voltage to anything from 0 to 3.3V, unlike PWM outputs this is a true analog output
- **A1 thru A5** – These are each analog input as well as digital I/O pins.

Right side



- **SCK/MOSI/MISO** - These are the hardware SPI pins, are connected to the 2x3 header on the right hand side. you can use them as everyday GPIO pins (but recommend keeping them free as they are best used for hardware SPI connections for high speed.)

Additional analog inputs

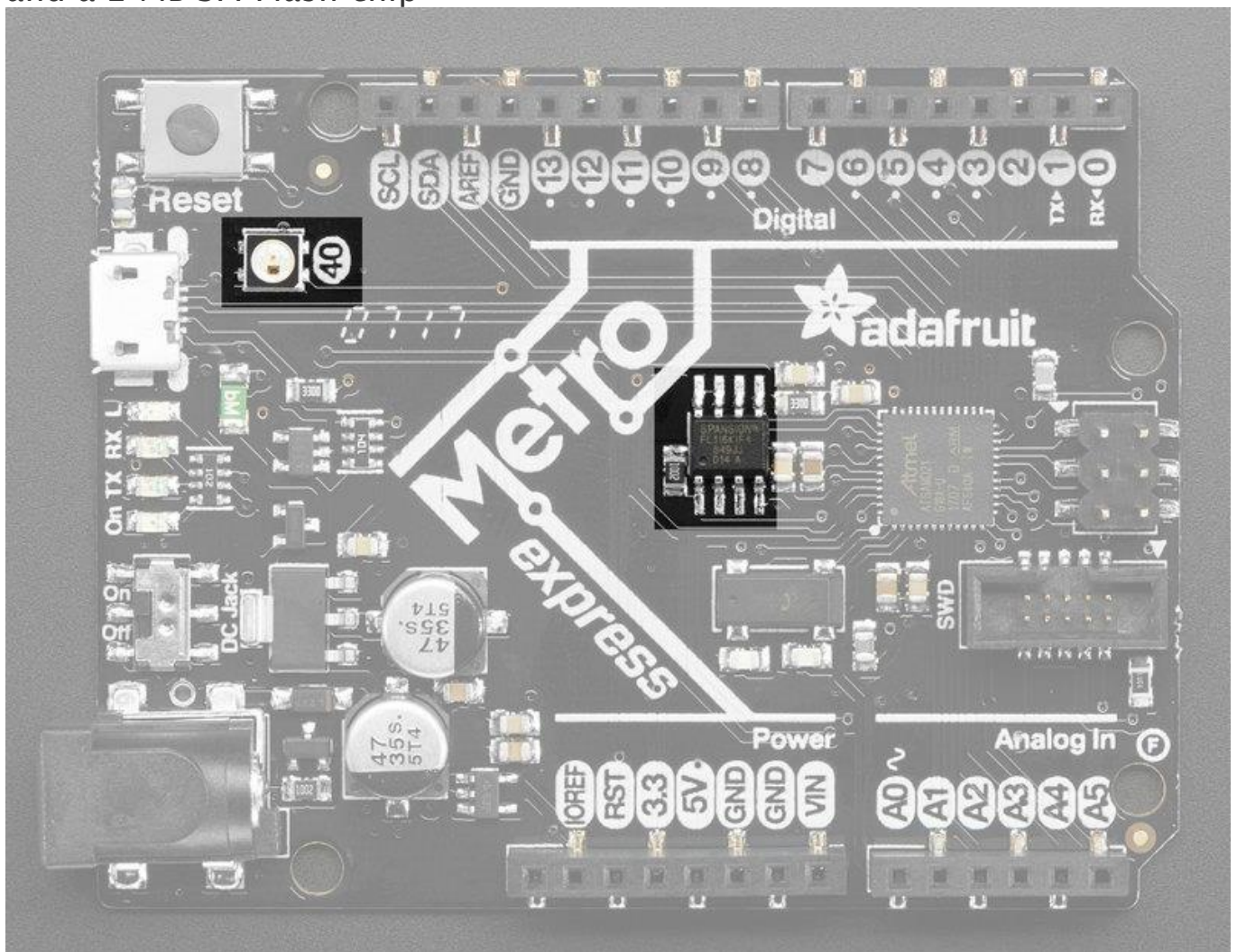
In addition to the A0–A5 pins, there are extra analog inputs available

- Digital #0 is also A6
- Digital #1 is also A7
- Digital #4 is also A8
- Digital #5 is also A9
- Digital #8 is also A10
- Digital #9 is also A11

These pins are available in CircuitPython under the `board` module. Names that start with # are prefixed with D and other names are as is. So #0 / RX above is available as `board.D0` and `board.RX` for example.

SPI Flash and NeoPixel

As part of the 'Express' series of boards, the Metro M0 Express is designed for use with CircuitPython. To make that easy, we have added two extra parts to this Metro M0: a mini NeoPixel (RGB LED) and a 2 MB SPI Flash chip

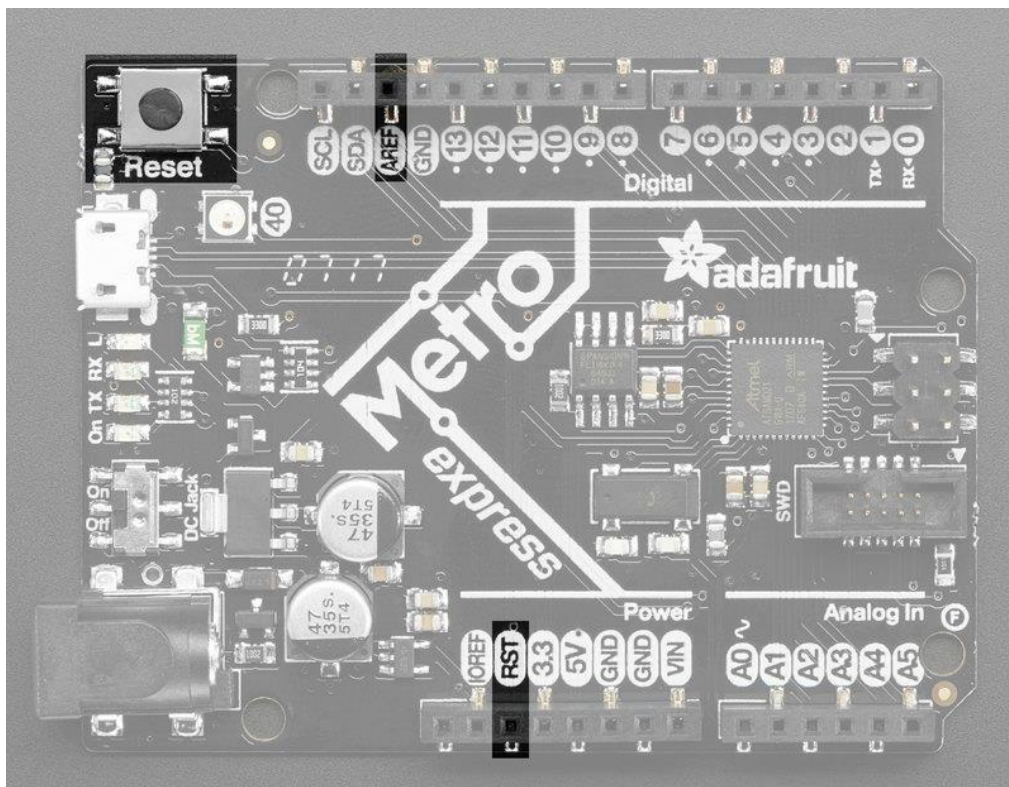


The NeoPixel is connected to pin #40 in Arduino, so [just use our NeoPixel library](#) and set it up as a single-LED strand on pin 40. In CircuitPython, the NeoPixel is `board.NEOPIXEL` and the library for it is [here](#) and in [the bundle](#). The NeoPixel is powered by the 3.3V

power supply but that hasn't shown to make a big difference in brightness or color. The NeoPixel is also used by the bootloader to let you know if the device has enumerated correctly (green) or USB failure (red). In CircuitPython, the LED is used to indicate the runtime status.

The SPI Flash is connected to 4 pins that are not brought out on the GPIO pads. This way you don't have to worry about the SPI flash colliding with other devices on the main SPI connection. Under Arduino, the FLASH SCK pin is #38, MISO is #36, MOSI is #37, and CS is #39. If you use **Metro M0 Express** as your board type, you'll be able to access the Flash SPI port under **SPI1** – this is a fully new hardware SPI device separate from the GPIO pins on the outside edge of the Feather. In CircuitPython, the SPI flash is used natively by the interpreter and is read-only to user code, instead the Flash just shows up as the writeable disk drive!

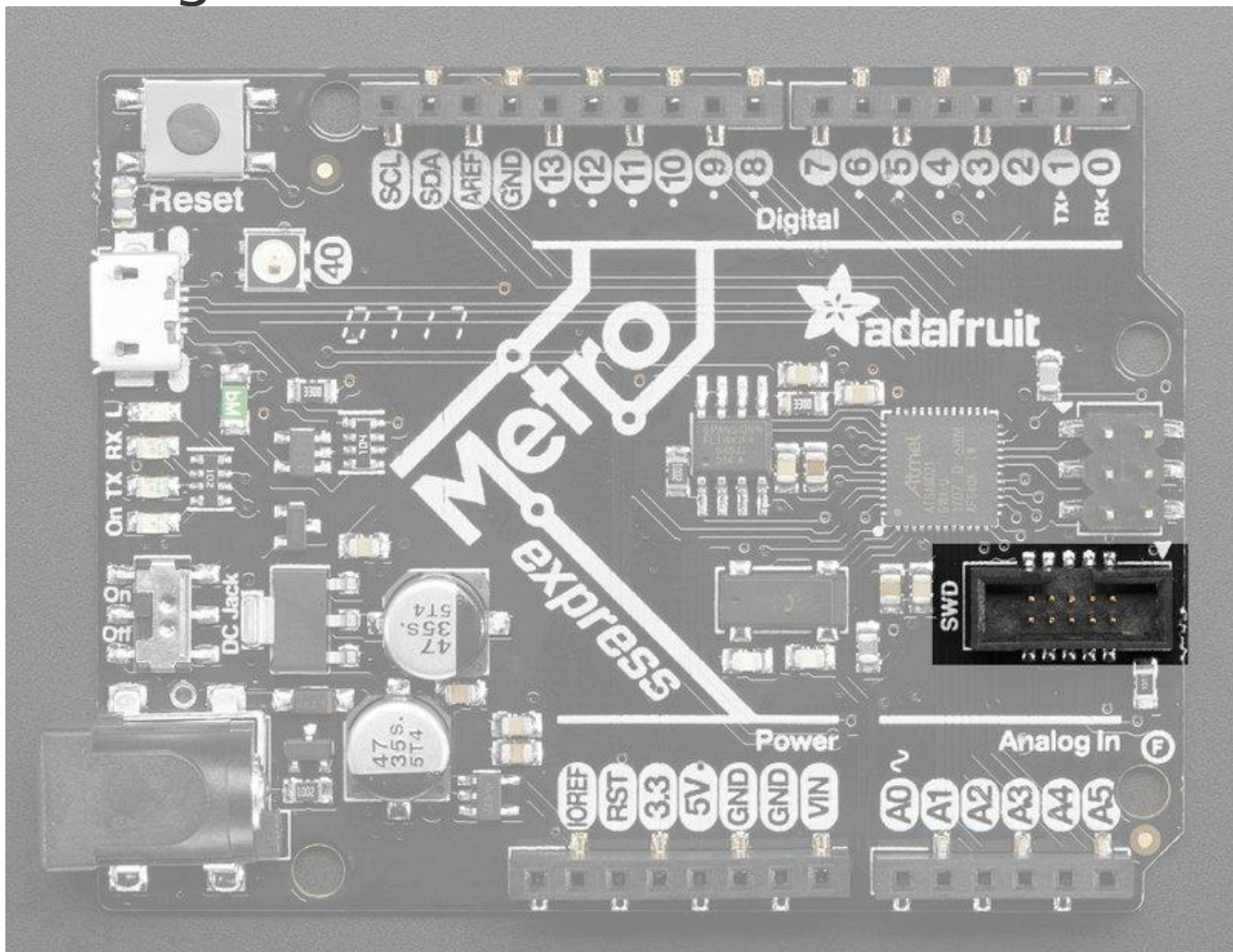
Other Pins!



- **RST** – this is the Reset pin, tie to ground to manually reset the ATSAM21, as well as launch the bootloader manually

- **AREf** – the analog reference pin. Normally the reference voltage is the same as the chip logic voltage (3.3V) but if you need an alternative analog reference, connect it to this pin and select the external AREF in your firmware. Can't go higher than 3.3V!
- **IORef** – the digital reference voltage pin. This reference voltage is the same as the chip logic voltage (3.3V) and cannot be changed.

Debug Interface



If you'd like to do more advanced development, trace-debugging, or not use the bootloader, we have the SWD interface exposed.

You can use any 2x5 0.05" pitch SWD interface to connect. We suggest a J-Link. Since the SWCLK pin is shared between the

NeoPixel, and the bootloader takes control of the pin, you need to reset the board *right before* beginning debug. OpenOCD and some other debug interfaces may not be able to do this. That's why we really really suggest a JLink!



[SEGGER J-Link EDU – JTAG/SWD Debugger](#)

The SEGGER J-Link EDU is identical to the more expensive J-Link BASE model except for the terms of use.If...

Out of Stock

[Out of Stock](#)



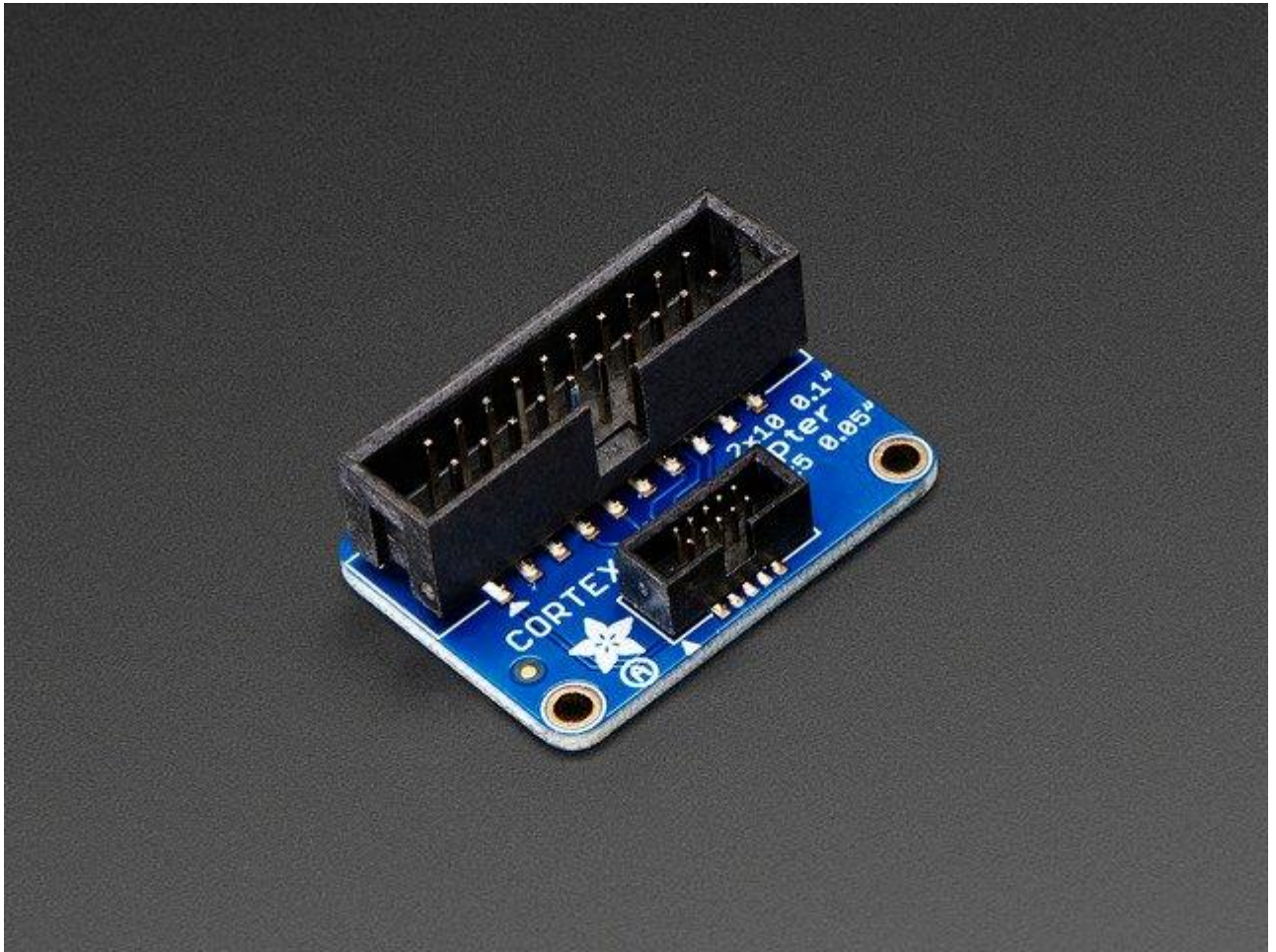
[SEGGER J-Link BASE – JTAG/SWD Debugger](#)

The SEGGER J-Link BASE is identical to the cheaper J-Link EDU model except for the terms of...

Out of Stock

Out of Stock

You'll need an adapter and cable to convert the 2x10 JTAG cable to SWD



[JTAG \(2x10 2.54mm\) to SWD \(2x5 1.27mm\) Cable Adapter Board](#)

This adapter board is designed for adapting a 'classic' 2x10 (0.1"/2.54mm pitch) JTAG cable to a slimmer 2x5 (0.05"/1.27mm pitch) SWD Cable. It's helpful...

\$4.95

In Stock

[Add to Cart](#)



[10-pin 2x5 Socket-Socket 1.27mm IDC \(SWD\) Cable – 150mm long](#)

These little cables are handy when programming or debugging a tiny board that uses 10-pin 1.27mm (0.05") pitch SWD programming connectors. We see these connectors often on ARM...

\$2.95

In Stock

[Add to Cart](#)

[Overview](#)[UF2 Bootloader Details](#)