Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

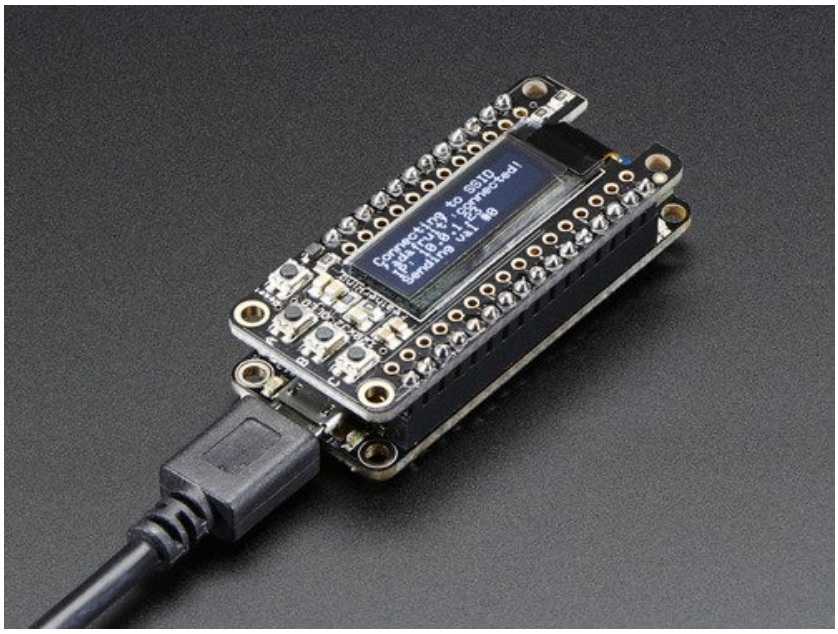
We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Adafruit Feather M0 WiFi with ATWINC1500

Created by lady ada



Last updated on 2017-03-31 12:02:55 AM UTC

Guide Contents

Guide Contents 2
Overview 4
Pinouts 9
Power Pins 9
Logic pins 10
WiFi Module & LEDs 10
Other Pins! 11
Assembly 13
Header Options! 13
Soldering in Plain Headers 15
- Prepare the header strip: 15
- Add the breakout board: 16
- And Solder! 16

Soldering on Female Header 18
- Tape In Place 18
- Flip & Tack Solder 19
- And Solder! 20

Power Management 22
Battery + USB Power 22
Power supplies 23
Measuring Battery 23
ENable pin 24
Power Usage & Saving with WiFi 24
Arduino IDE Setup 28
- https://adafruit.github.io/arduino-board-index/package_adafruit_index.json 29

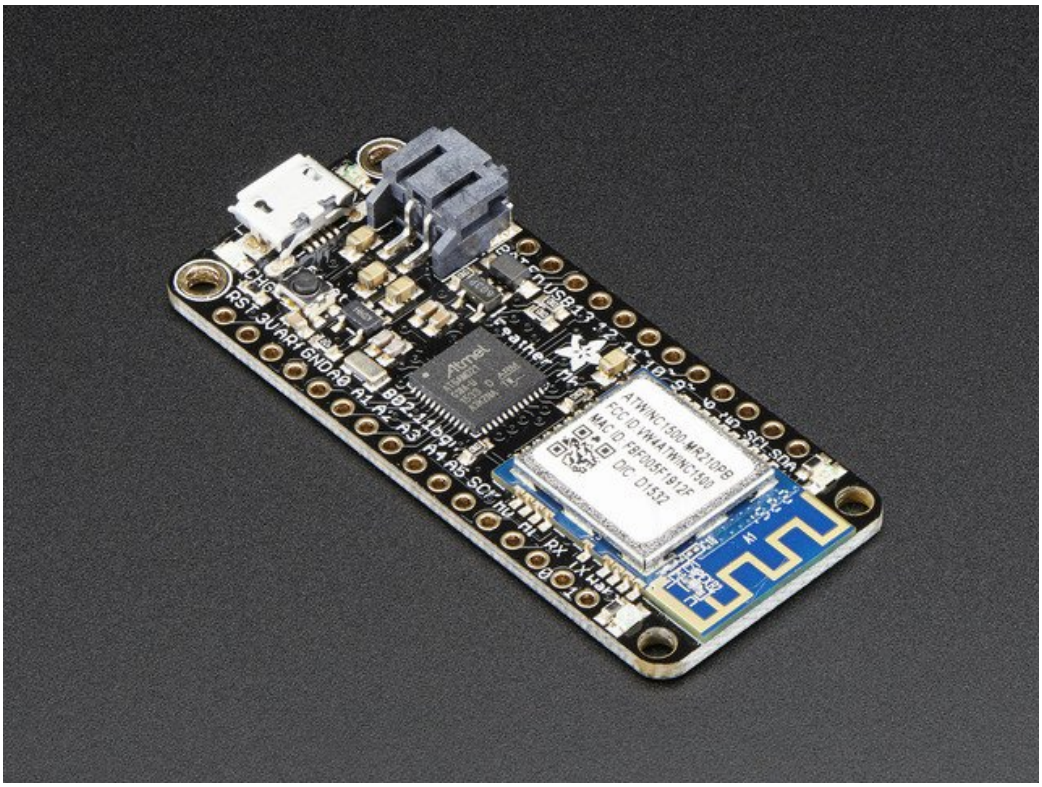
Using with Arduino IDE 31
Install SAMD Support 31
Install Adafruit SAMD 32
Install Drivers (Windows Only) 33
Blink 34
Sucessful Upload 35
Compilation Issues 35

Manually bootloading 36
Ubuntu & Linux Issue Fix 36
Using the WiFi Module 37
Install the Library 37
Check Connections & Version 38
Scanning WiFi 39
Connect & Read Webpage 40
Creating an Access Point + Webserver 41
Updating Firmware 46
Updating SSL Certificates 51
Adapting Sketches to M0 55
Analog References 55
Pin Outputs & Pullups 55
Serial vs SerialUSB 55
AnalogWrite / PWM 56
Missing header files 57
Bootloader Launching 57
Aligned Memory Access 57
Floating Point Conversion 57
How Much RAM Available? 58
Storing data in FLASH 58
Feather HELP! 59
Downloads 62
Datasheets & Files 62
Schematic 62
Fabrication Print 62

Overview

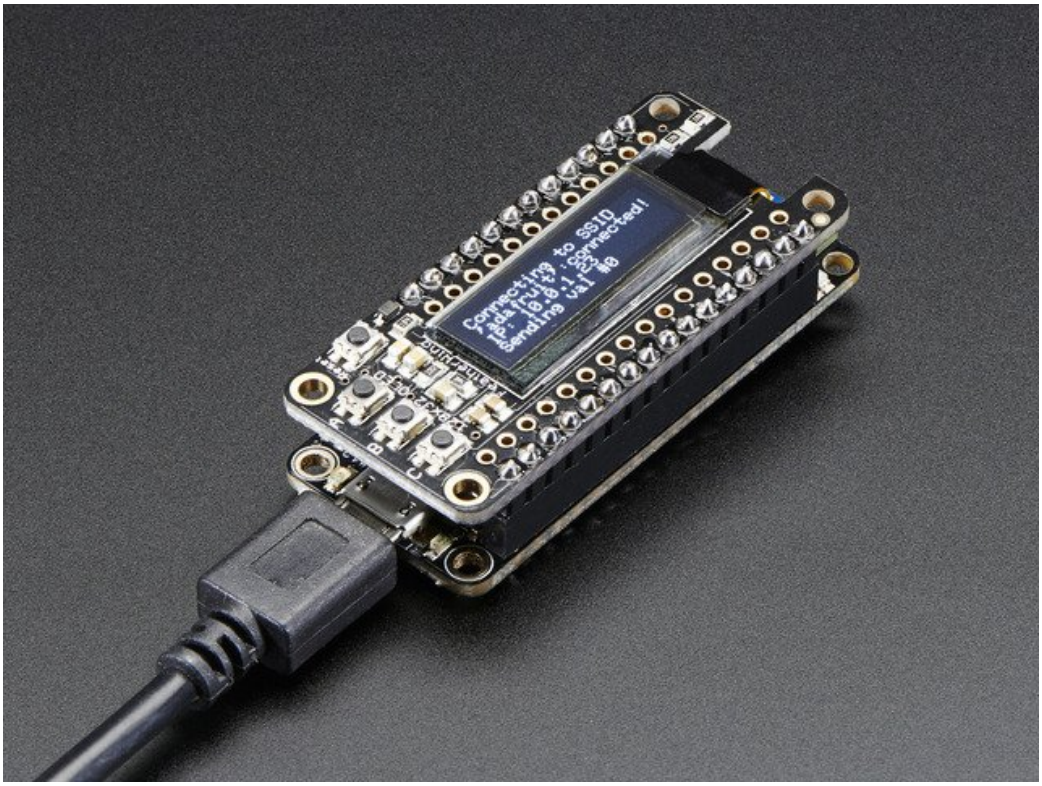
Feather is the new development board from Adafruit, and like its namesake it is thin, light, and lets you fly! We designed Feather to be a new standard for portable microcontroller cores.

This is the **Adafruit Feather M0 WiFi** w/ATWINC1500 - our take on an 'all-in-one' Arduino-compatible + high speed, reliable WiFi with built in USB and battery charging. Its an Adafruit Feather M0 with a WiFi module (http://adafru.it/2999), ready to rock! We have other boards in the Feather family, check'em out here (http://adafru.it/l7B).



Connect your Feather to the Internet with this fine new FCC-certified WiFi module from Atmel. This 802.11bgn-capable WiFi module is the best new thing for networking your devices, with built-in low-power management capabilites, Soft-AP, SSL support and rock solid performance. We were running our adafruit.io MQTT demo for a full weekend straight with no hiccups (it would have run longer but we had to go to work, so we unplugged it). This module is very fast & easy to use in comparison to other WiFi modules we've used in the past.

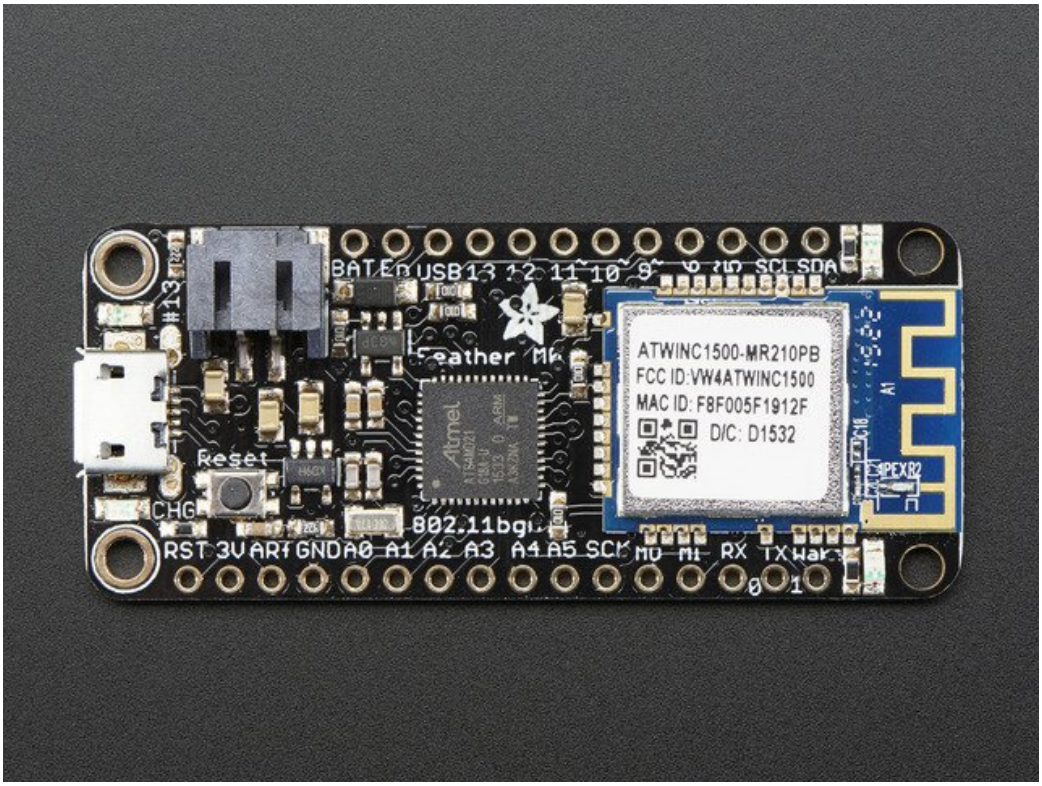
This module works with 802.11b, g, or n networks & supports WEP, WPA and WPA2 encryption. You can connect to your own WiFi networks or create your own with "Soft AP" mode, where it becomes its own access point (we have an example of it creating a webserver that you can then control the Arduino's pins). You can clock it as fast as 12MHz for speedy, reliable packet streaming. And scanning/connecting to networks is very fast, just a second or two.



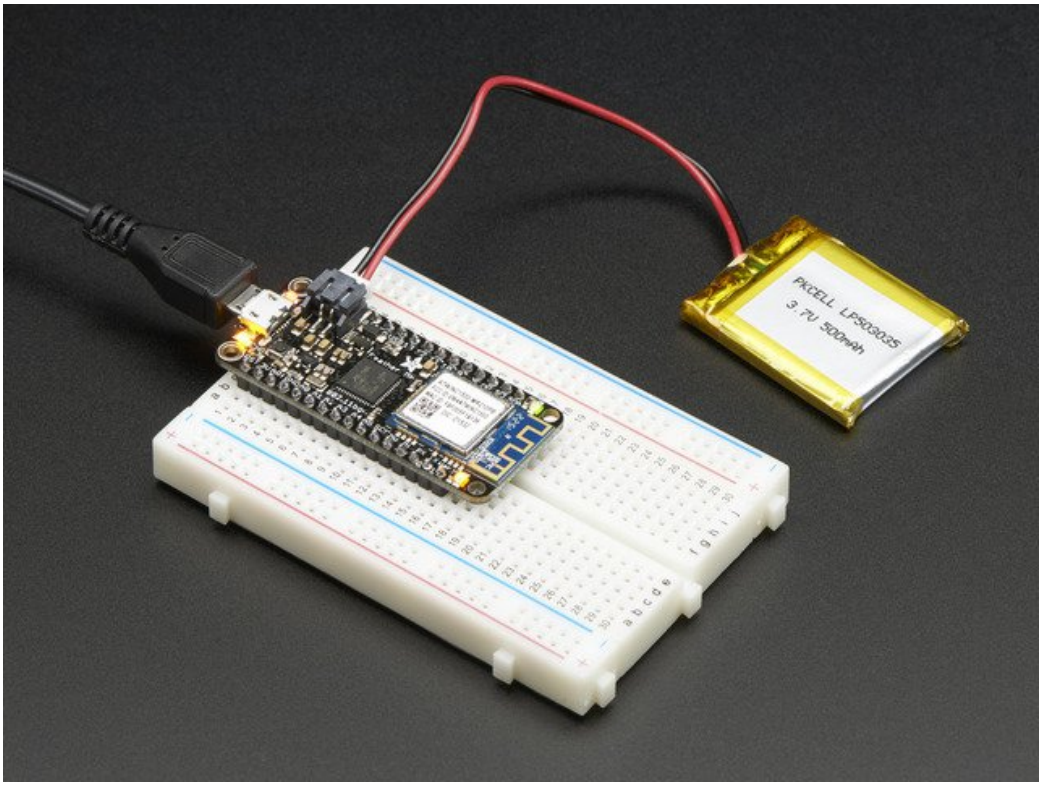
You might be wondering why use this [when you can get a HUZZAH Feather?](http://adafru.it/2821) Well, you get

- A highly-capable Cortex M0+ processor with ton more I/O pins, lots of 12-bit ADCs, a 10-bit DAC, 6 total SERCOMs that can each do SPI, I2C *or* UART (3 are used by the existing interfaces, leaving you 3), plenty of timers, PWMs, DMA, native USB, and more ([check out the Datasheet](http://adafru.it/l3e))
- The ATWINC has much lower power usage, about 12mA for the WINC & 10mA for the ATSAMD21 with auto-powermanagement on for the WiFi and no power management for the ARM. With manual power management, you can get the WiFi module to down to ~2mA by putting it to sleep. This is compared to the ESP's ~70mA average current draw, and whose deep sleep mode requires a WDT reset.
- We also found that we could stream more reliably (less 'bursty') with the ATWINC, although altogether the ESP has higher throughput.
- You also dont have to 'yield' all the time to the WiFi core, since its a separate chip. You get full reign of the processor and timing

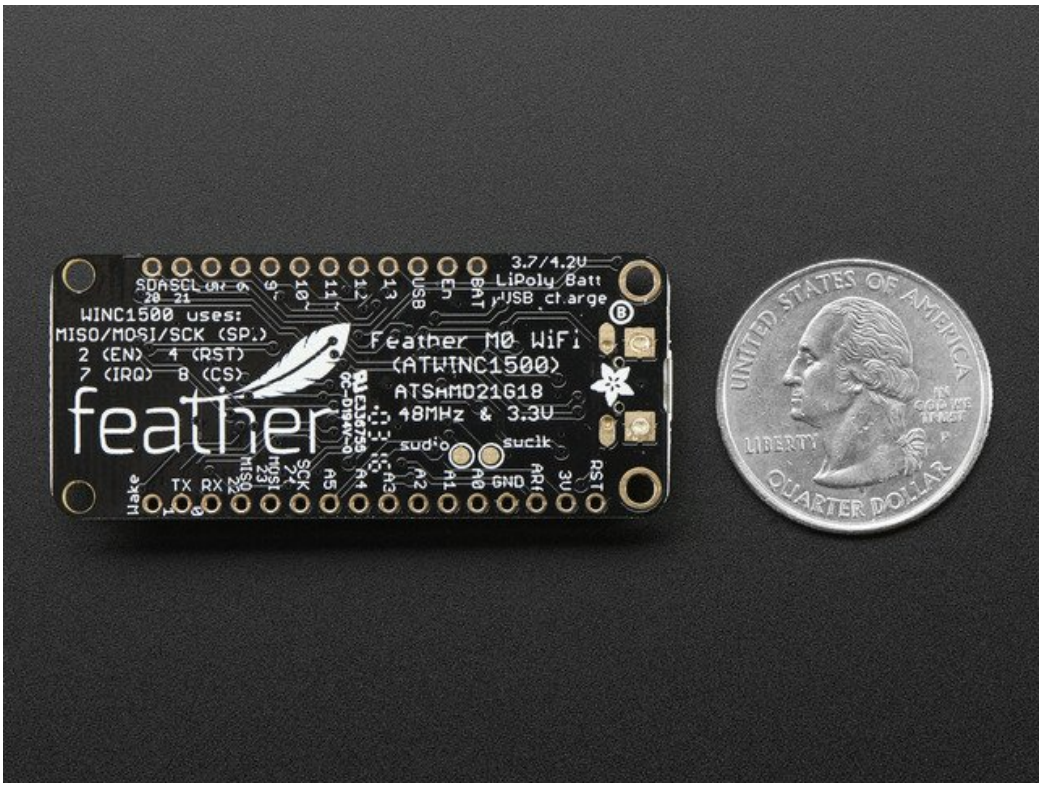
Of course, both WiFi-capable Feathers have their strengths and tradeoffs, & we love both equally!



At the Feather M0's heart is an ATSAMD21G18 ARM Cortex M0 processor, clocked at 48 MHz and at 3.3V logic, the same one used in the new Arduino Zero (http://adafru.it/2843). This chip has a whopping 256K of FLASH (8x more than the Atmega328 or 32u4) and 32K of RAM (16x as much)! This chip comes with built in USB so it has USB-to-Serial program & debug capability built in with no need for an FTDI-like chip. For advanced users who are comfortable with ASF, the SWDIO/SWCLK pins are available on the bottom, and when connected to a CMSIS-DAP debugger can be used to use Atmel Studio for debugging.

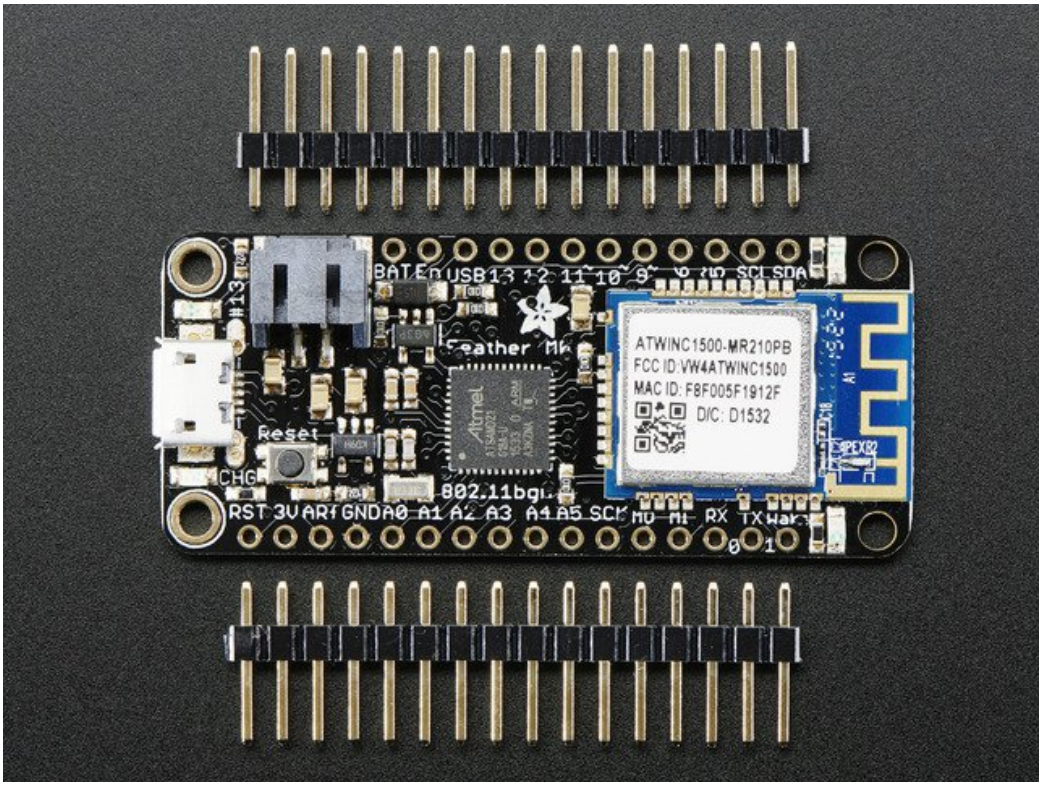


To make it easy to use for portable projects, we added a connector for any of our 3.7V Lithium polymer batteries and built in battery charging. You don't need to use a battery, it will run just fine straight from the micro USB connector. But, if you do have a battery, you can take it on the go, then plug in the USB to recharge. The Feather will automatically switch over to USB power when its available. We also tied the battery through a divider to an analog pin, so you can measure and monitor the battery voltage to detect when you need a recharge.



Here's some handy specs! Like all Feather M0's you get:

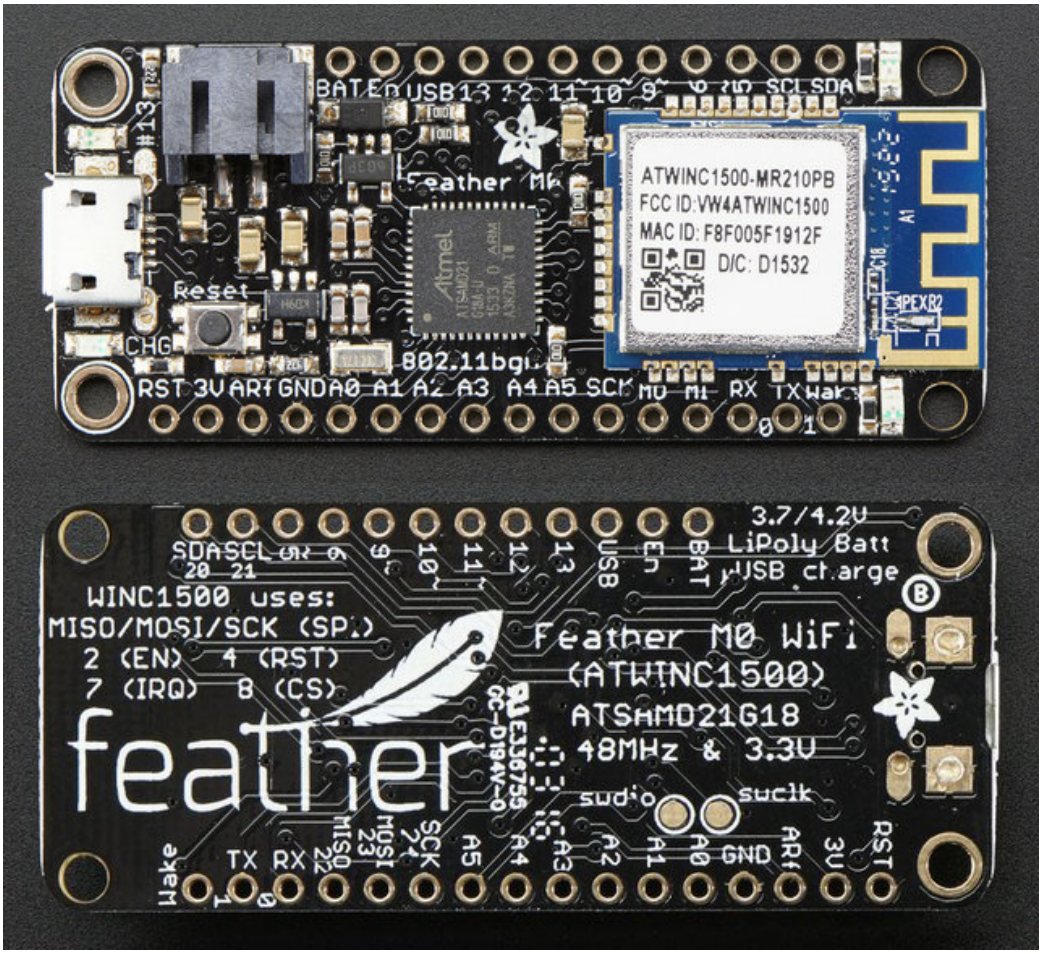
- Measures 2.1" x 0.9" x 0.3" (53.65mm x 23mm x 8mm) without headers soldered in. Note it is 0.1" longer than most Feathers
- Light as a (large?) feather - 6.1 grams
- ATSAMD21G18 @ 48MHz with 3.3V logic/power
- 256KB FLASH, 32KB SRAM, No EEPROM
- 3.3V regulator (AP2112K-3.3) with 600mA peak current output, WiFi can draw 300mA peak during xmit
- USB native support, comes with USB bootloader and serial port debugging
- You also get tons of pins - 20 GPIO pins
- Hardware Serial, hardware I2C, hardware SPI support
- 8 x PWM pins
- 10 x analog inputs
- 1 x analog output
- Built in 200mA lipoly charger with charging status indicator LED
- Pin #13 red LED for general purpose blinking
- Power/enable pin
- 4 mounting holes
- Reset button



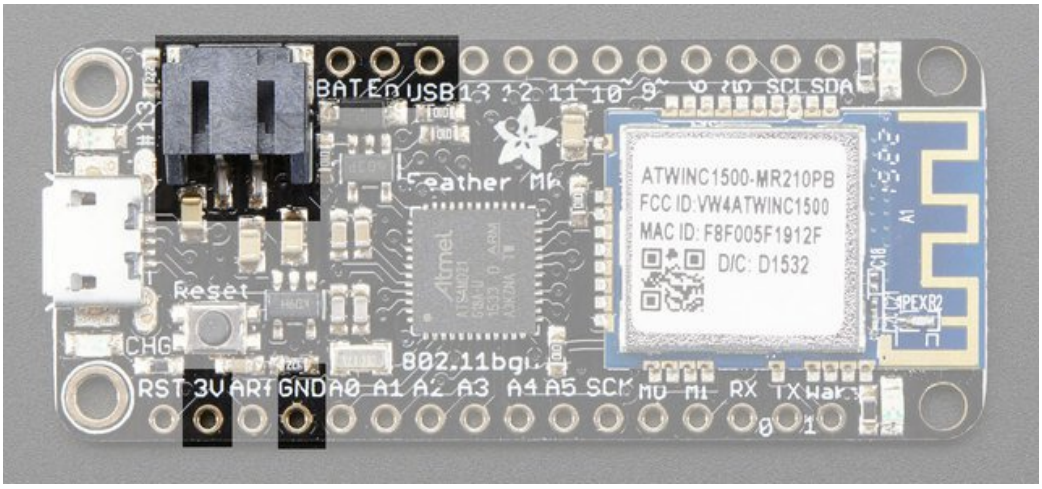
Comes fully assembled and tested, with a USB bootloader that lets you quickly use it with the Arduino IDE. We also toss in some header so you can solder it in and plug into a solderless breadboard. **Lipoly battery (http://adafru.it/e0v) and MicroUSB cable (http://adafru.it/aM5) not included** (but we do have lots of options in the shop if you'd like!)

Pinouts

The Feather M0 Adalogger 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 Pins



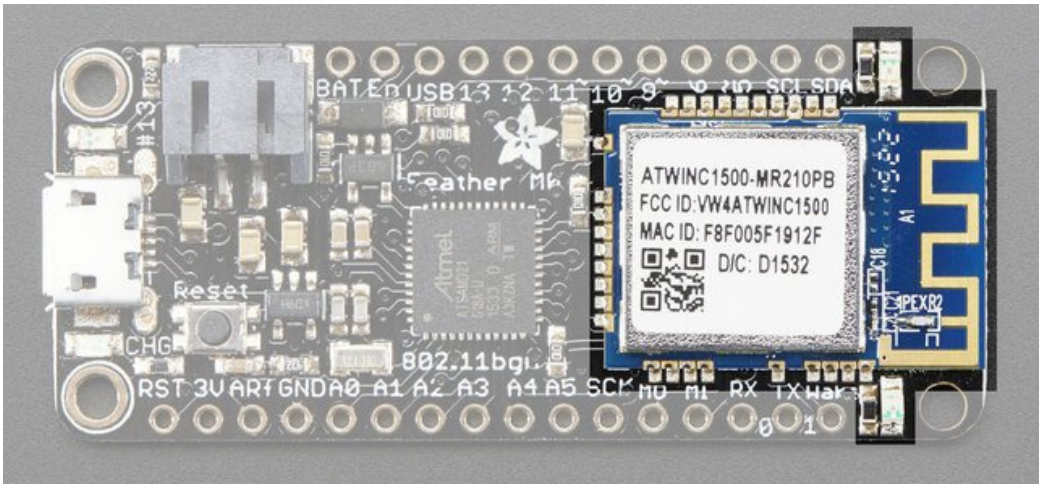
- **GND** - this is the common ground for all power and logic
- **BAT** - this is the positive voltage to/from the JST jack for the optional Lipoly battery
- **USB** - this is the positive voltage to/from the micro USB jack if connected
- **EN** - this is the 3.3V regulator's enable pin. It's pulled up, so connect to ground to disable the 3.3V regulator
- **3V** - this is the output from the 3.3V regulator, it can supply 600mA peak

Logic pins

This is the general purpose I/O pin set for the microcontroller.
All logic is 3.3V
All pins can do PWM output
All pins can be interrupt inputs

- **#0** / **RX** - GPIO #0, also receive (input) pin for **Serial1** (hardware UART), also can be analog input
- **#1** / **TX** - GPIO #1, also transmit (output) pin for **Serial1**, also can be analog input
- **#20** / **SDA** - GPIO #20, also 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.
- **#21** / **SCL** - GPIO #21, also 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.
- **#5** - GPIO #5
- **#6** - GPIO #6
- **#9** - GPIO #9, also analog input **A7**. This analog input is connected to a voltage divider for the lipoly battery so be aware that this pin naturally 'sits' at around 2VDC due to the resistor divider
- **#10** - GPIO #10
- **#11** - GPIO #11
- **#12** - GPIO #12
- **#13** - GPIO #13 and is connected to the **red LED** next to the USB jack
- **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.
- **SCK**/**MOSI**/**MISO** (GPIO **24**/**23**/**22**)- These are the hardware SPI pins, 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)

WiFi Module & LEDs

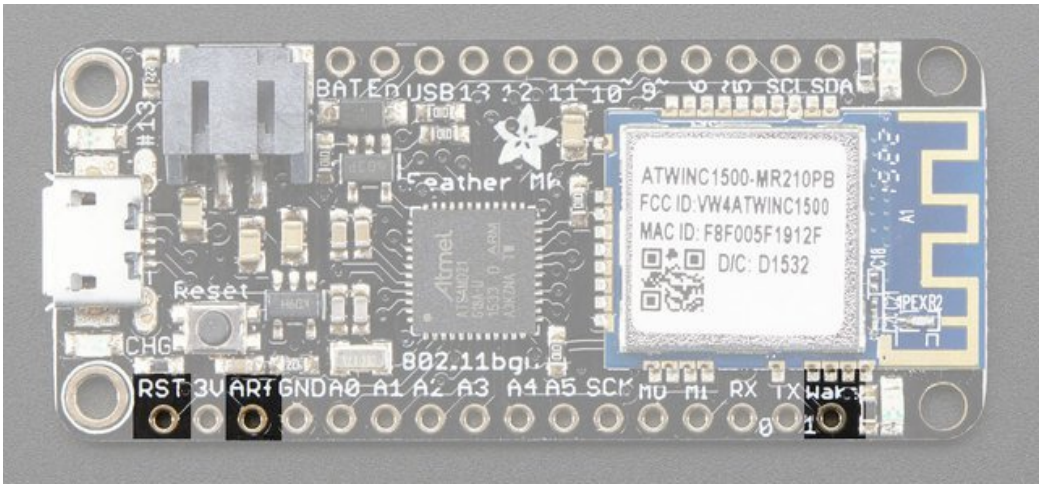


Since not all pins can be brought out to breakouts, due to the small size of the Feather, we use these to control the WiFi module

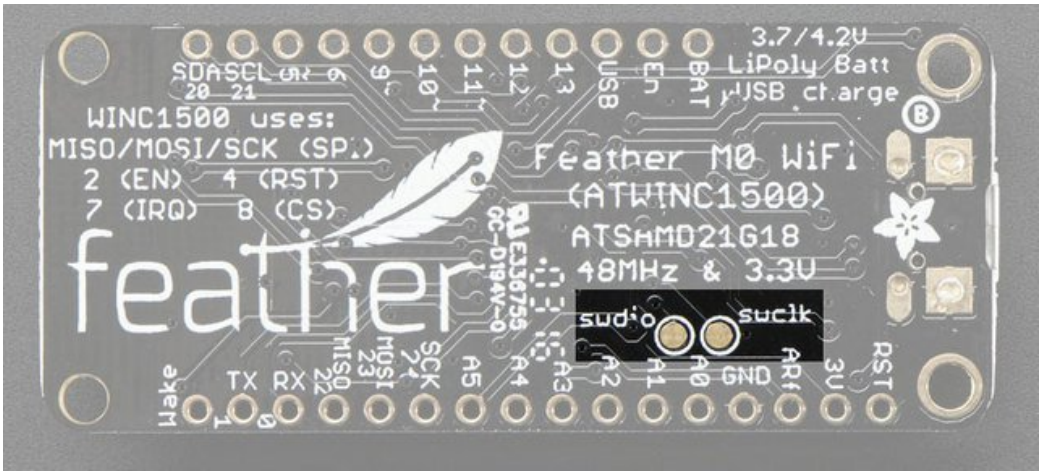
- **#2** - used as the **EN**able pin for the WiFi module, by default pulled down low, set HIGH to enable WiFi
- **#4** - used as the **Reset** pin for the WiFi module, controlled by the library
- **#7** - used as the **IRQ** interrupt request pin for the WiFi module, controlled by the library
- **#8** - used as the **C**hip **S**elect pin for the WiFi module, used to select it for SPI data transfer
- **MOSI** / **MISO** /**SCK** - the SPI pins are also used for WiFi module communication
- **Green LED** - the top LED, in green, will light when the module has connected to an SSID
- **Yellow LED** - the bottom LED, in yellow, will blink during data transfer

Other Pins!

- **RST** - this is the Reset pin, tie to ground to manually reset the AVR, 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!
- **Wake** - connected to the Wake pin on the WiFi module, not used at this time but it's there if you want it



SWCLK & SWDIO - These pads on the bottom are used to program the chip. They can also be connected to an SWD debugger.

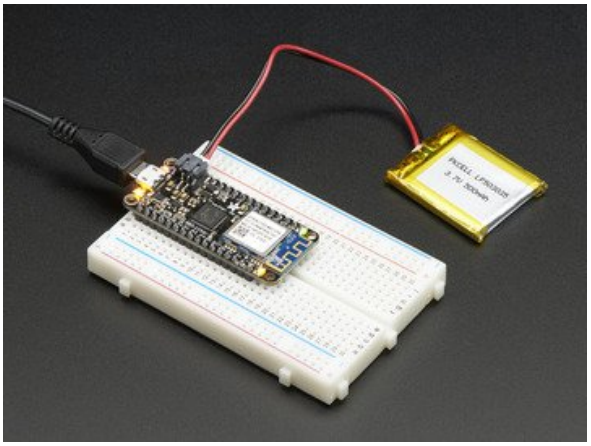
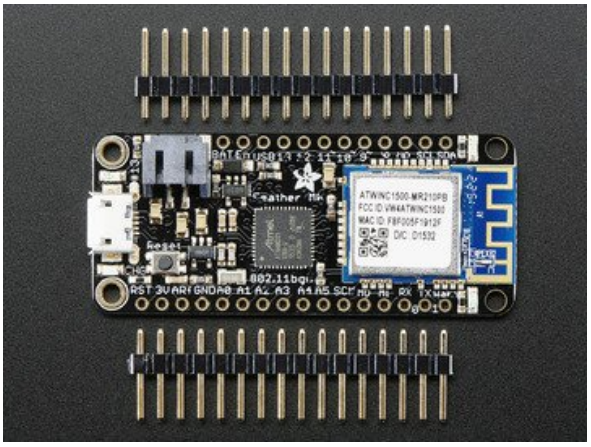

3.7/4.2V
LiPoly Batt
+USB charge
WINC1500 uses:
MISO/MOSI/SCK (SPI)
2 (EN) 4 (RST)
7 (IRQ) 8 (CS)
Feather M0 WiFi
(ATWINC1500)
ATSAMD21G18
48MHz & 3.3V
feather
swdio
swclk
SDA SCL
20 21
9 10 11 12 13 USB BAT EN
Wake TX RX MISO MOSI SCK A5 A4 A3 A2 A1 A0 GND AREF 3V RST
0 1 22 23 24

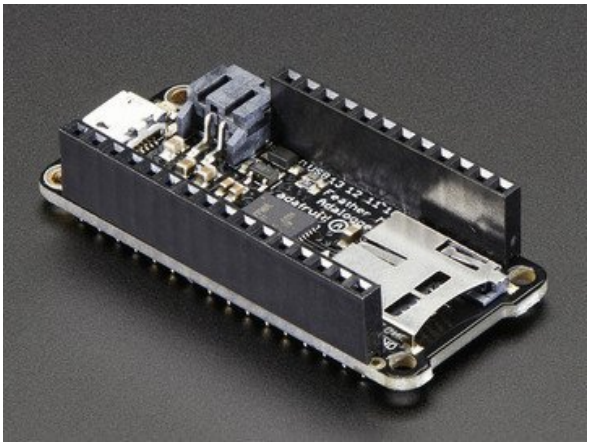
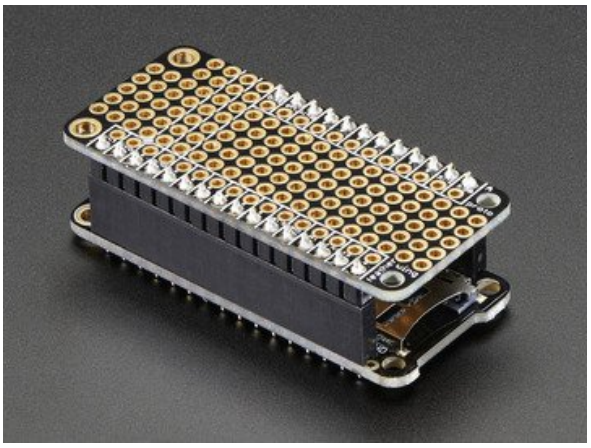
Assembly

We ship Feathers fully tested but without headers attached - this gives you the most flexibility on choosing how to use and configure your Feather

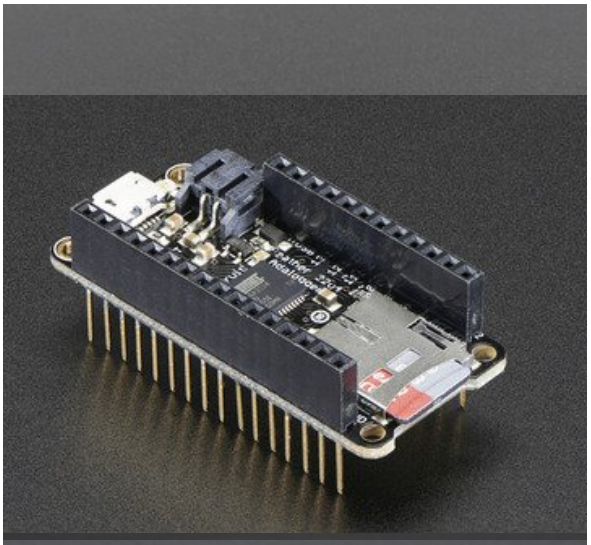
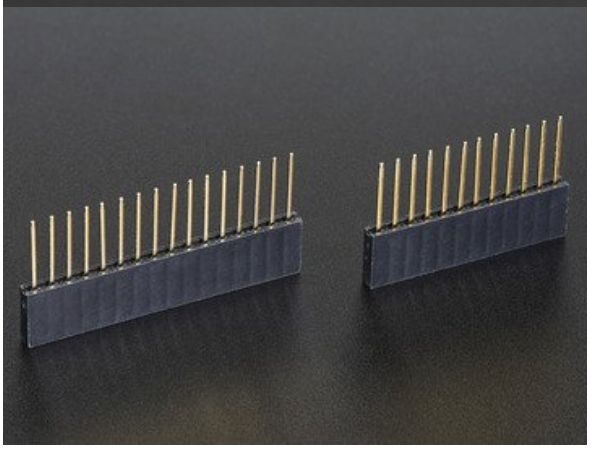
Header Options!

Before you go gung-ho on soldering, there's a few options to consider!

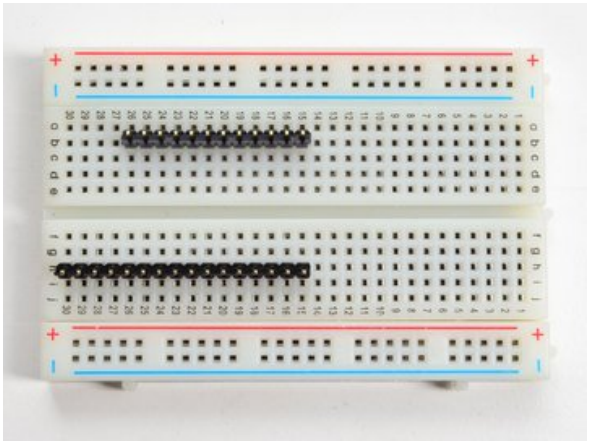
-  The first option is soldering in plain male headers, this lets you plug in the Feather into a solderless breadboard
- 

- 
  Another option is to go with socket female headers. This won't let you plug the Feather into a breadboard but it will let you attach featherwings very easily
- 

We also have 'slim' versions of the female headers, that are a little shorter and give a more compact shape

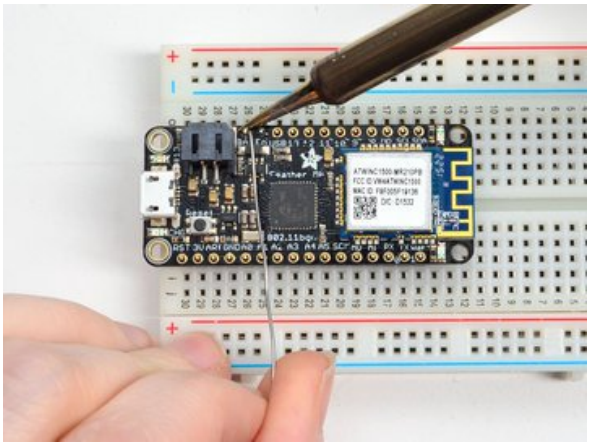
- 
- 
- Finally, there's the "Stacking Header" option. This one is sort of the best-of-both-worlds. You get the ability to plug into a solderless breadboard *and* plug a featherwing on top. But its a little bulky

Soldering in Plain Headers

- 

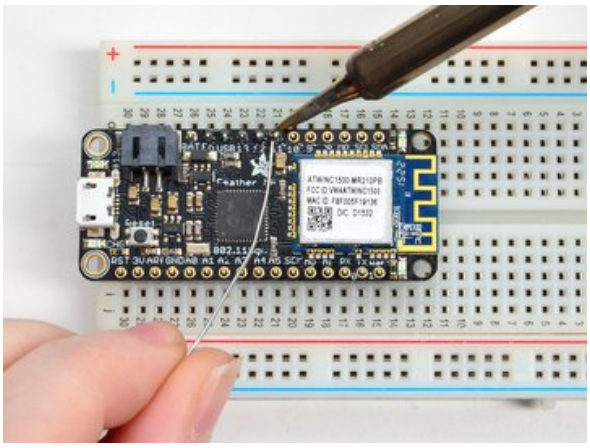
Prepare the header strip:

Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**

- 

Add the breakout board:

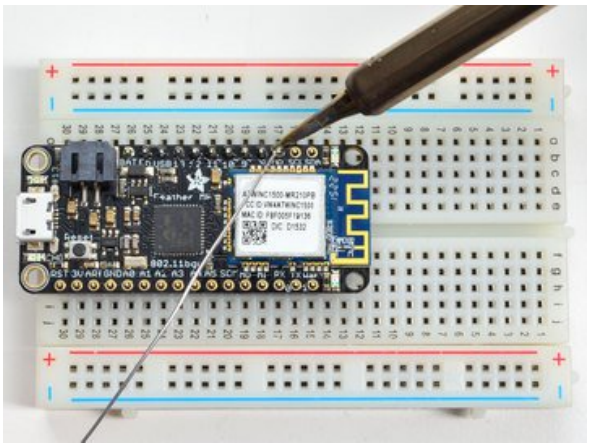
Place the breakout board over the pins so that the short pins poke through the breakout pads

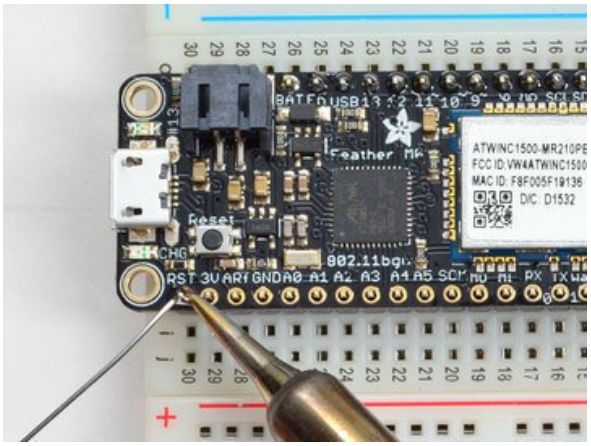
- 

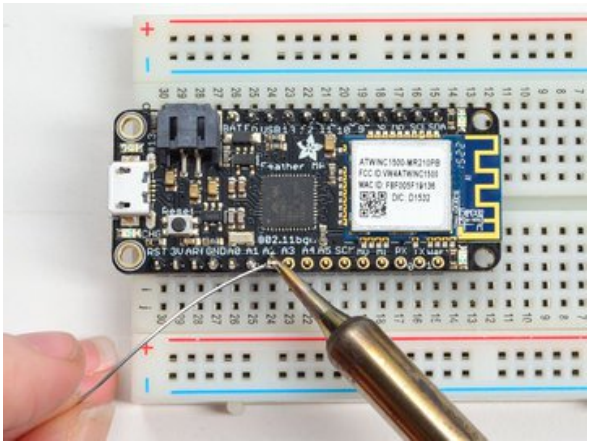
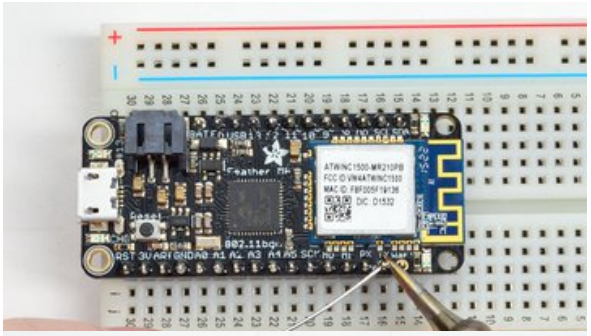
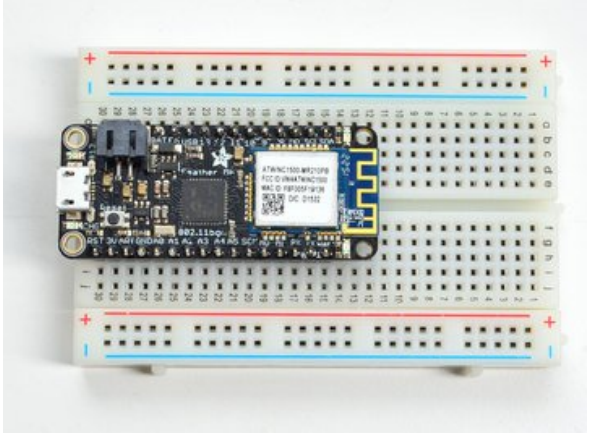
And Solder!

Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our[Guide to Excellent Soldering](http://adafru.it/aTk) (http://adafru.it/aTk)).

- 

- 

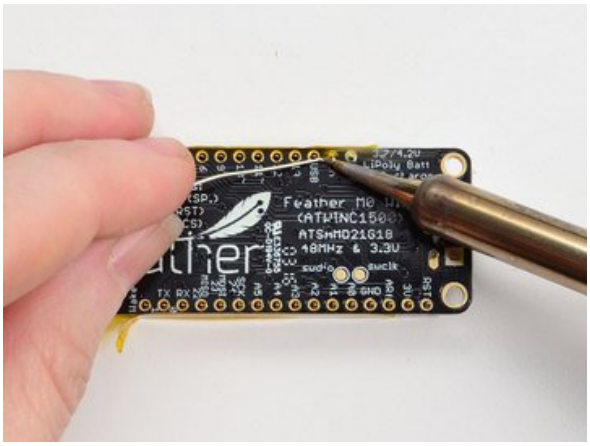
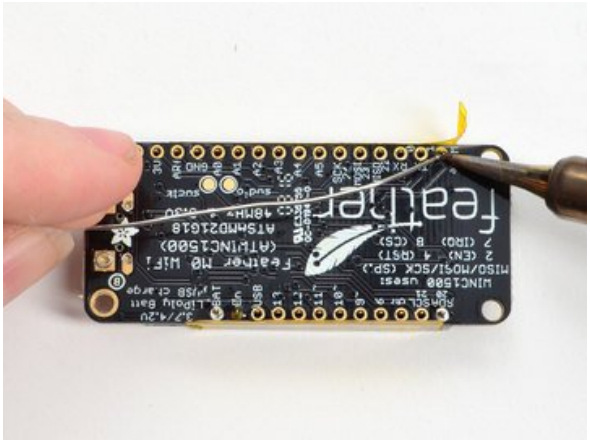
Solder the other strip as well.
- 
- 
- 
You're done! Check your solder joints visually and continue onto the next steps

Soldering on Female Header

- 

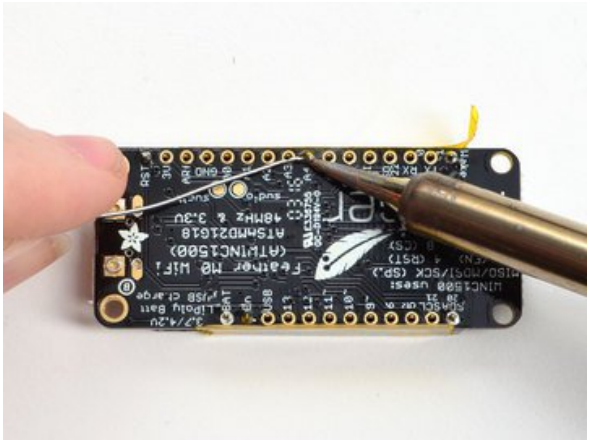
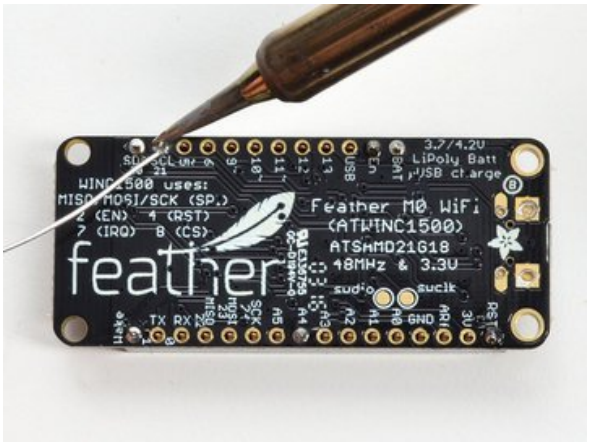
Tape In Place

For sockets you'll want to tape them in place so when you flip over the board they don't fall out

- 
- 
- 

Flip & Tack Solder

After flipping over, solder one or two points on each strip, to 'tack' the header in place

- 
- 

- 

And Solder!

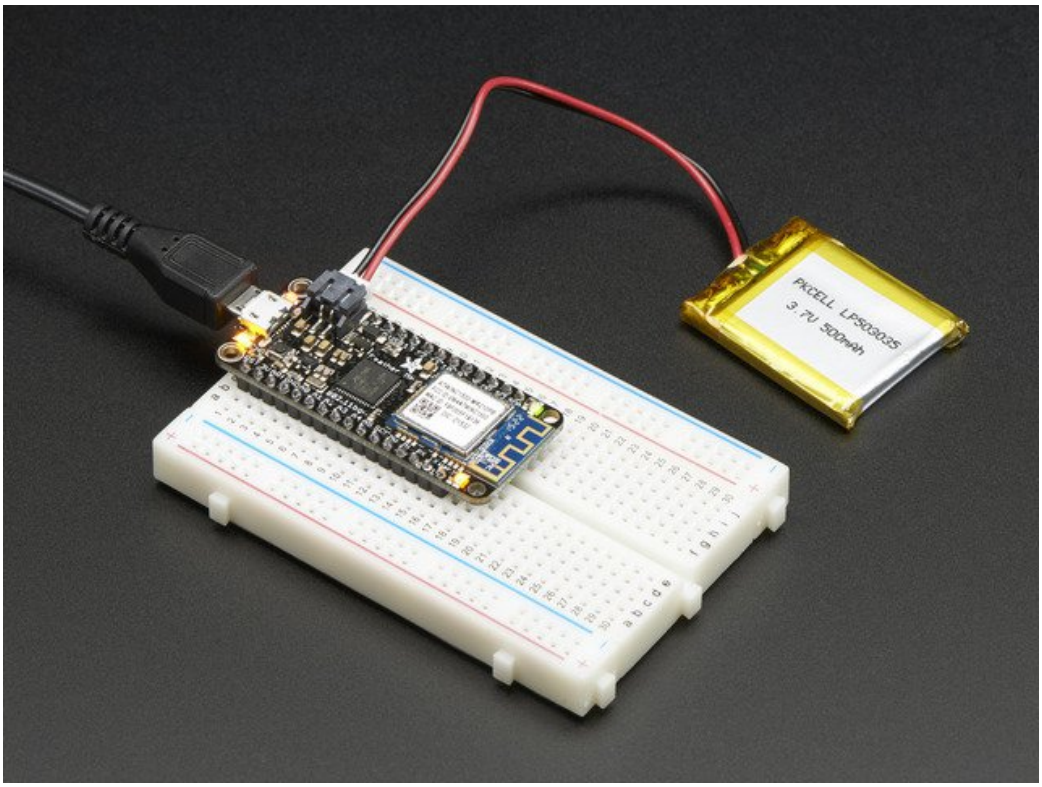
Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our[Guide to Excellent Soldering](http://adafru.it/aTk) (http://adafru.it/aTk)).

You're done! Check your solder joints visually and continue onto the next steps

- 
3.7/4.2V
LiPoly Batt
/USB charge
WINC1500 uses:
MISO/MOSI/SCK (SPI)
2 (EN) 4 (RST)
7 (IRQ) 8 (CS)
Feather M0 WiFi
(ATWINC1500)
ATSAMD21G18
48MHz & 3.3V
feather
TX RX
GND
RST
- 

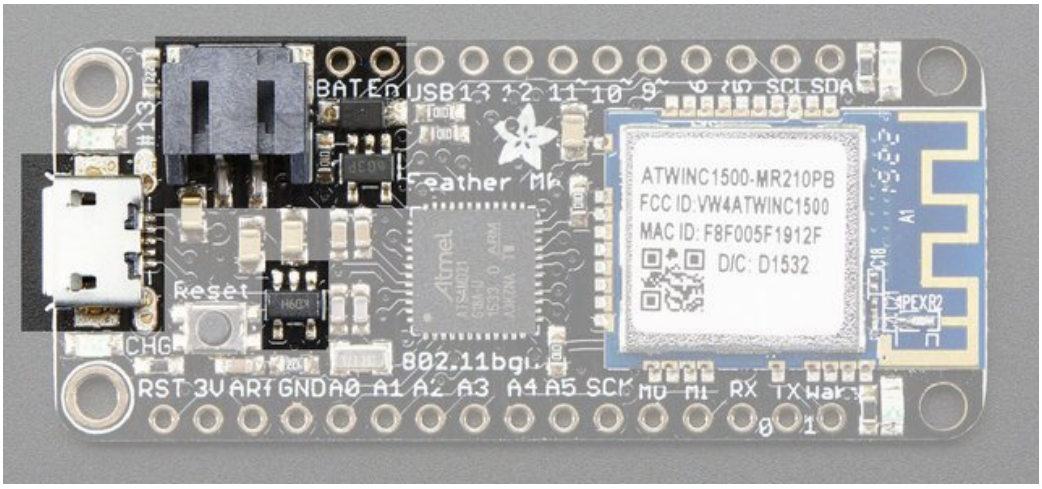
Power Management



Battery + USB Power

We wanted to make the Feather easy to power both when connected to a computer as well as via battery. There's **two ways to power** a Feather. You can connect with a MicroUSB cable (just plug into the jack) and the Feather will regulate the 5V USB down to 3.3V. You can also connect a 4.2/3.7V Lithium Polymer (Lipo/Lipoly) or Lithium Ion (LiIon) battery to the JST jack. This will let the Feather run on a rechargable battery. **When the USB power is powered, it will automatically switch over to USB for power, as well as start charging the battery (if attached) at 200mA.** This happens 'hotswap' style so you can always keep the Lipoly connected as a 'backup' power that will only get used when USB power is lost.

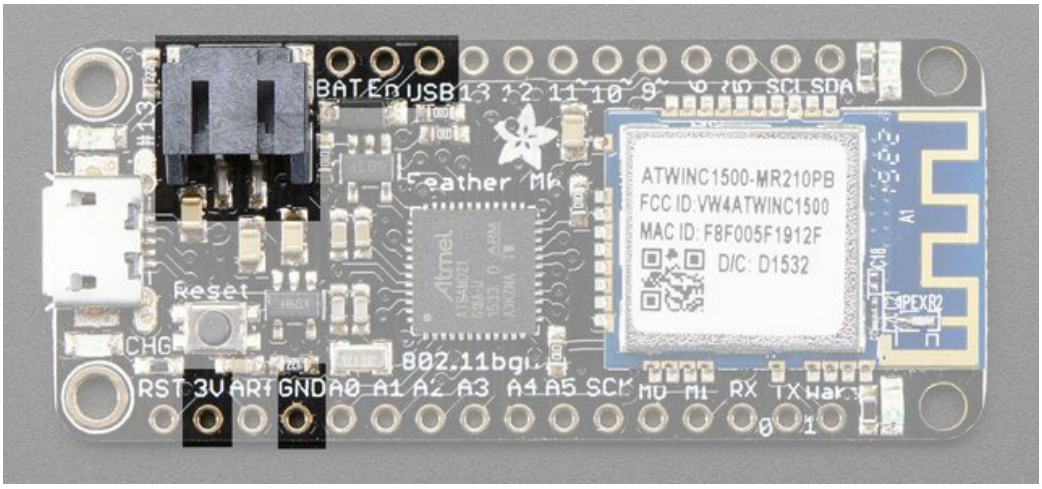
The JST connector polarity is matched to Adafruit LiPoly batteries. Using wrong polarity batteries can destroy your Feather



The above shows the Micro USB jack (left), Lipoly JST jack (top left), as well as the 3.3V regulator and changeover diode (just to the right of the JST jack) and the Lipoly charging circuitry (to the right of the Reset button). There's also a **CHG** LED, which will light up while the battery is charging. This LED might also flicker if the battery is not connected.

Power supplies

You have a lot of power supply options here! We bring out the **BAT** pin, which is tied to the lipoly JST connector, as well as **USB** which is the +5V from USB if connected. We also have the **3V** pin which has the output from the 3.3V regulator. We use a 600mA peak AP2112K-33. While you can get 600mA from it, you can't do it continuously from 5V as it will overheat the regulator. It's fine for, say, powering the attached WiFi chip or XBee radio though, since the current draw is 'spiky' & sporadic.



Measuring Battery

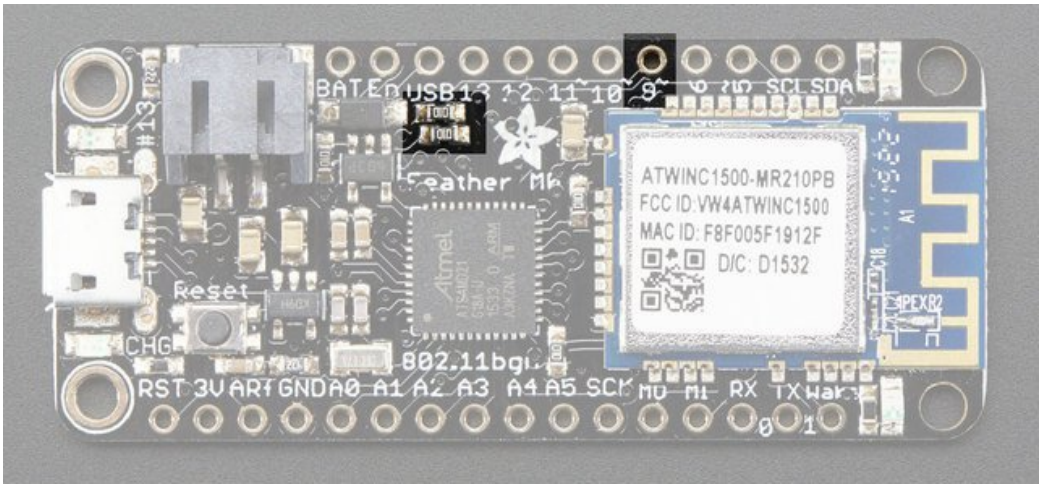
If you're running off of a battery, chances are you wanna know what the voltage is at! That way you can tell when the battery needs recharging. Lipoly batteries are 'maxed out' at 4.2V and stick around 3.7V for much of the battery life, then slowly sink down to 3.2V or so before the protection circuitry cuts it off. By measuring the voltage you can quickly tell when you're heading below 3.7V

To make this easy we stuck a double-100K resistor divider on the **BAT** pin, and connected it to **D9** (a.k.a analog #7

A7). You can read this pin's voltage, then double it, to get the battery voltage.

```
#define VBATPIN A7

float measuredvbat = analogRead(VBATPIN);
measuredvbat *= 2;    // we divided by 2, so multiply back
measuredvbat *= 3.3;  // Multiply by 3.3V, our reference voltage
measuredvbat /= 1024; // convert to voltage
Serial.print("VBat: " ); Serial.println(measuredvbat);
```

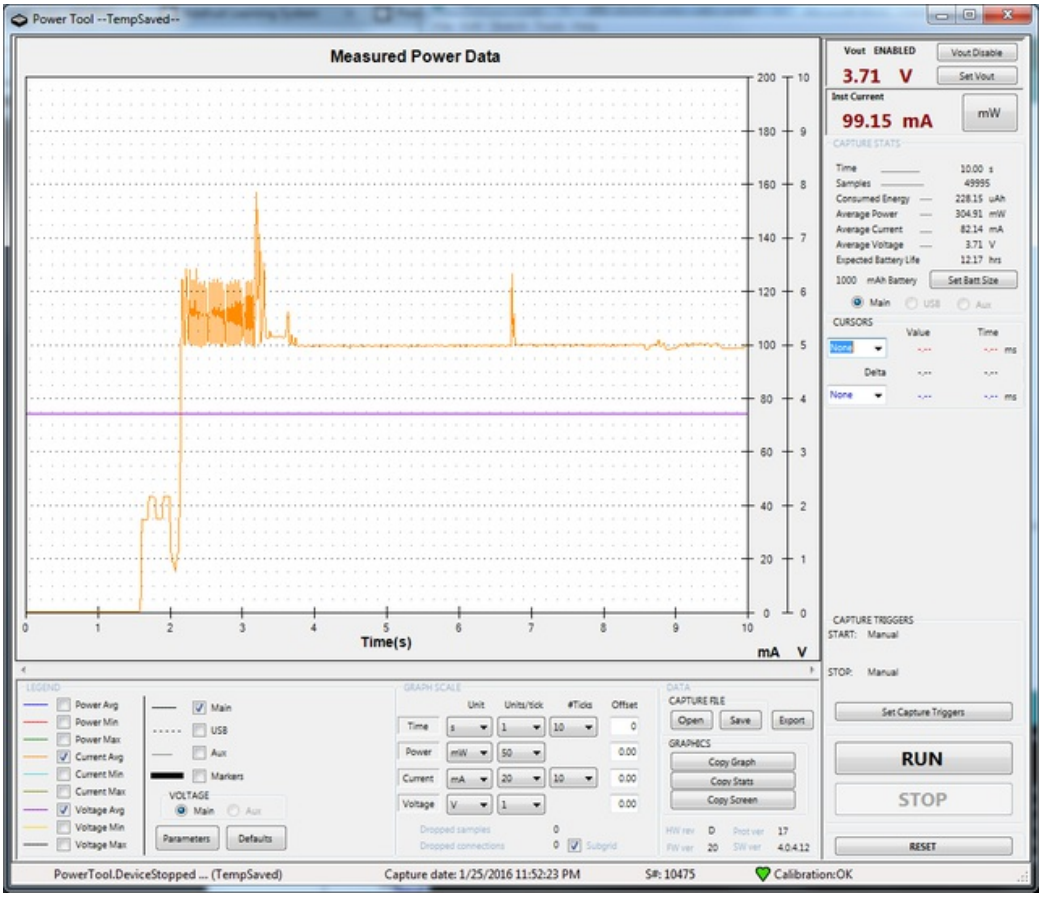


ENable pin

If you'd like to turn off the 3.3V regulator, you can do that with the**EN**(able) pin. Simply tie this pin to**Ground** and it will disable the 3V regulator. The **BAT** and **USB** pins will still be powered

Power Usage & Saving with WiFi

WiFi is a very power-hungry protocol. During transmit and SSID association, you'll see high power usages. For example, here is an MQTT demo running where it connects to the WPA SSID and then sents a packet every 5 seconds or so:



You can see the chip launch at about 1.5 seconds, then turn on the WiFi and at about 2s make the SSID connection and MQTT connection. The average current is about 100ms afterwards, and a packet spikes up to ~130mA at the 7 second mark.

100mA is still quite a bit, you can very easily reduce this by letting the WINC1500 manage its own power:

```
WiFi.setSleepMode(M2M_PS_H_AUTOMATIC, 1); // go into power save mode when possible!
```

When this line is added, it lets the WINC1500 know that when nothings going on, shut down unneeded parts. You dont have to manage the power modes, and the power will drop down nearly instantly to about 22mA average (there's still spikes during transmit of course)

If you're using the Arduino WiFi101 library, call this instead to enable automatic sleep:

```
WiFi.lowPowerMode();
```