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.

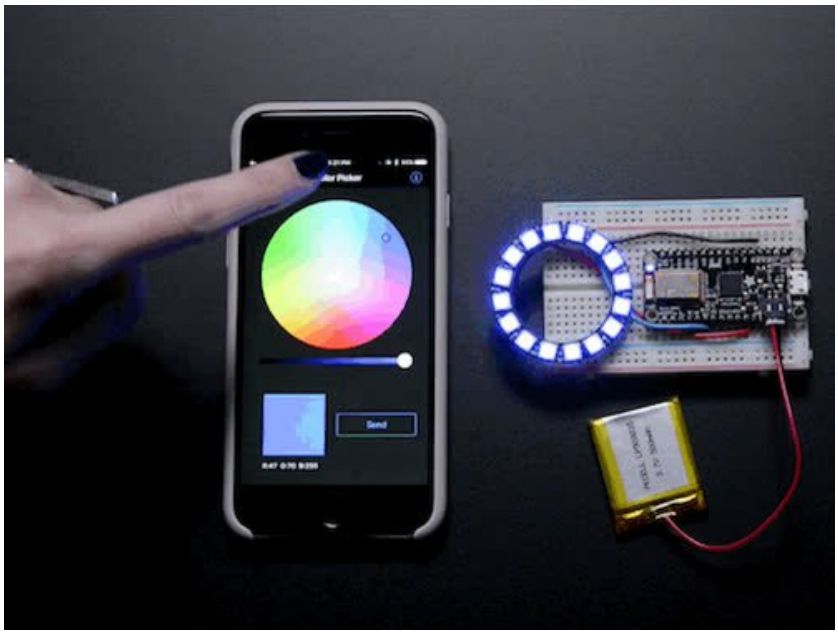
We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

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 32u4 Bluefruit LE

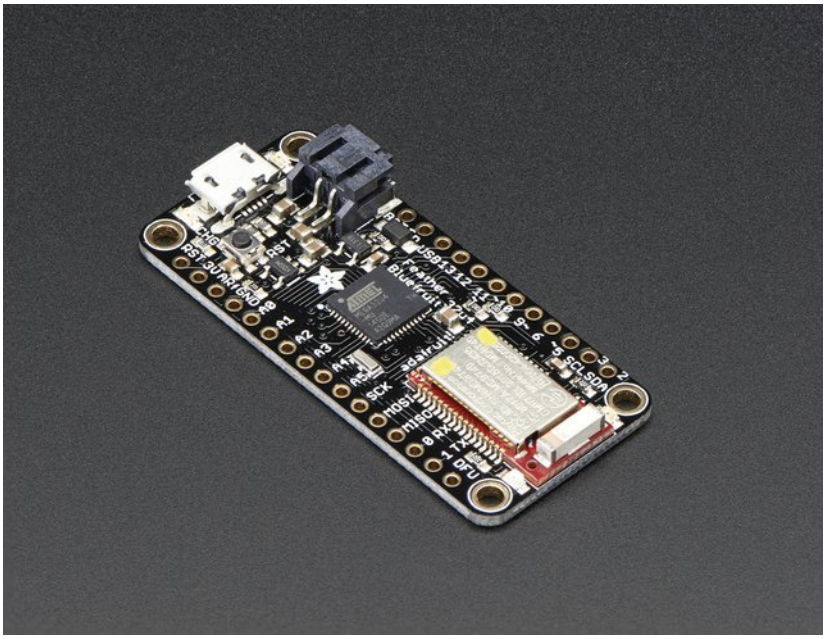
Created by lady ada



Last updated on 2017-01-26 06:44:21 PM UTC

Guide Contents

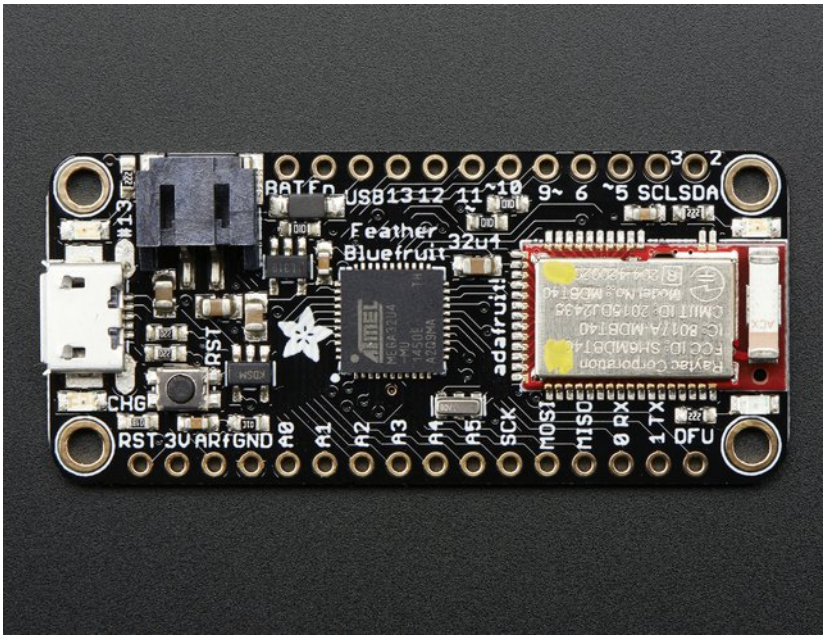
Overview



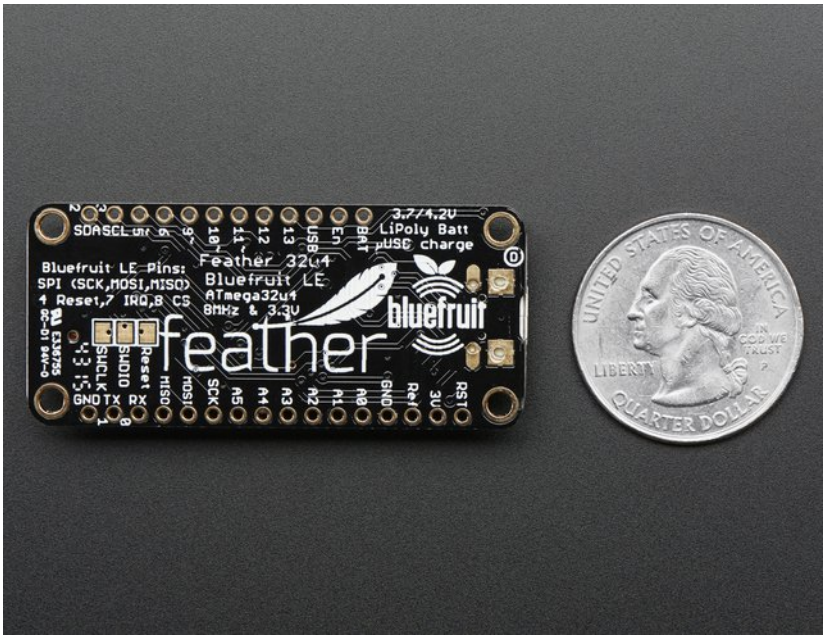
Feather is the new development board from Adafruit, and like it's namesake it is thin, light, and lets you fly! We designed Feather to be a new standard for portable microcontroller cores.

This is the **Adafruit Feather 32u4 Bluefruit** - our take on an 'all-in-one' Arduino-compatible + Bluetooth Low Energy with built in USB and battery charging. Its an Adafruit Feather 32u4 with a BTLE module, ready to rock! We have other boards in the Feather family, check'em out here (http://adafru.it/jAQ)

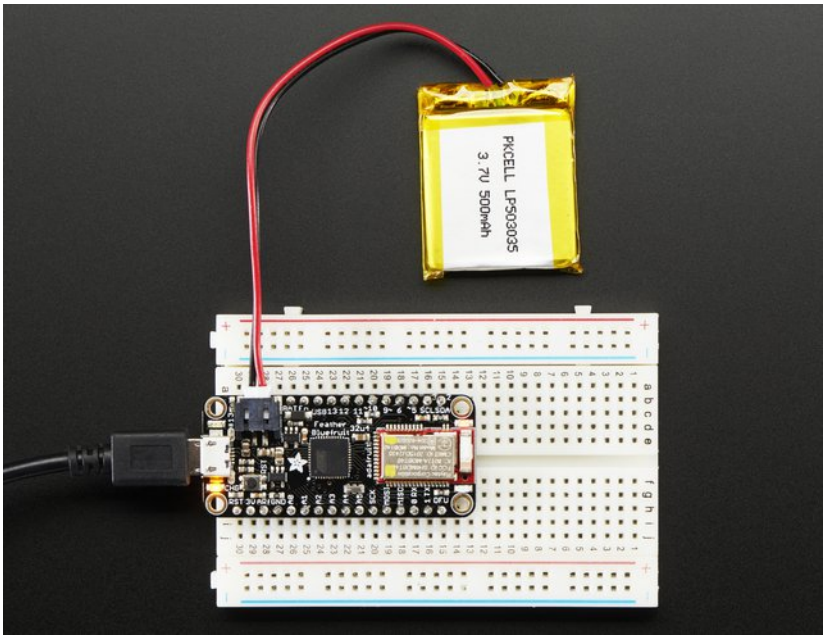
Bluetooth Low Energy is the hottest new low-power, 2.4GHz spectrum wireless protocol. In particular, its the only wireless protocol that you can use with iOS without needing special certification and it's supported by all modern smart phones. This makes it excellent for use in portable projects that will make use of an iOS or Android phone or tablet. It also is supported in Mac OS X and Windows 8+



At the Feather 32u4's heart is at ATmega32u4 clocked at 8 MHz and at 3.3V logic, a chip setup we've had tons of experience with and it's the same as the Flora (http://adafru.it/dVl). This chip has 32K of flash and 2K of RAM, with built in USB so not only does it have a USB-to-Serial program & debug capability built in with no need for an FTDI-like chip, it can also act like a mouse, keyboard, USB MIDI device, etc.



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 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 thru a divider to an analog pin, so you can measure and monitor the battery voltage to detect when you need a recharge.

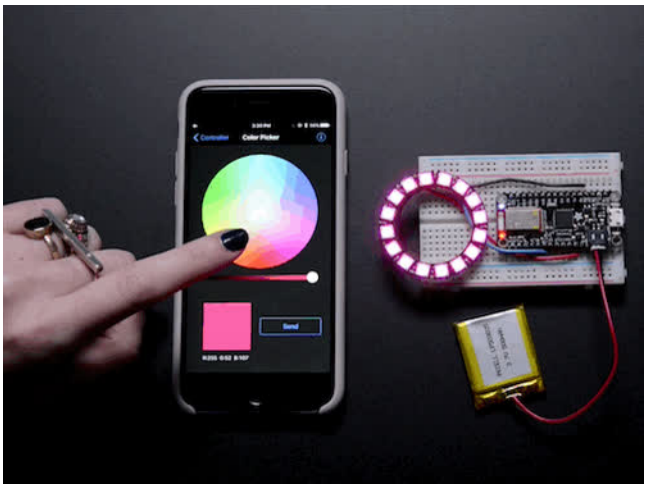


Here's some handy specs! Like all Feather 32u4's you get:

- Measures 2.0" x 0.9" x 0.28" (51mm x 23mm x 8mm) without headers soldered in
- Light as a (large?) feather - 5.7 grams
- ATmega32u4 @ 8MHz with 3.3V logic/power
- 3.3V regulator with 500mA peak current output
- 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
- Built in 100mA lipoly charger with charging status indicator LED
- Pin #13 red LED for general purpose blinking
- Power/enable pin

- 4 mounting holes
- Reset button

The **Feather 32u4 Bluefruit LE** uses the extra space left over to add our excellent Bluefruit BTLE module + two status indicator LEDs



The Power of Bluefruit LE

The Bluefruit LE module is an nRF51822 chipset from Nordic, programmed with multi-function code that can do quite a lot! For most people, they'll be very happy to use the standard Nordic UART RX/TX connection profile. In this profile, the Bluefruit acts as a data pipe, that can 'transparently' transmit back and forth from your iOS or Android device. You can use our iOS App (http://adafru.it/iCi) or Android App (http://adafru.it/f4G), or write your own to communicate with the UART service (http://adafru.it/iCF).

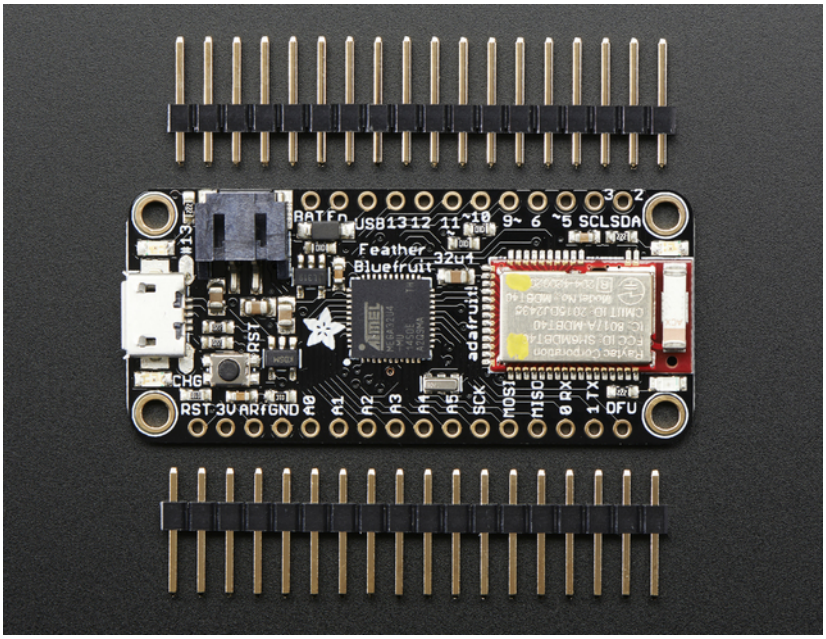
The board is capable of much more than just sending strings over the air! Thanks to an easy to learn AT command set (http://adafru.it/iCG), you have full control over how the device behaves, including the ability to define and manipulate your own GATT Services and Characteristics (http://adafru.it/iCH), or change the way that the device advertises itself for other Bluetooth Low Energy devices to see. You can also use the AT commands to query the die temperature, check the battery voltage, and more, check the connection RSSI or MAC address, and tons more. Really, way too long to list here!

Use the Bluefruit App to get your project started

Using our Bluefruit iOS App (http://adafru.it/iCi) or Android App (http://adafru.it/f4G), you can quickly get your project prototyped by using your iOS or Android phone/tablet as a controller. We have a color picker (http://adafru.it/iCI), quaternion/accelerometer/gyro/magnetometer or location (GPS) (http://adafru.it/iCI), and an 8-button control game pad (http://adafru.it/iCI). This data can be read over BLE and piped into the ATmega32u4 chip for processing & control

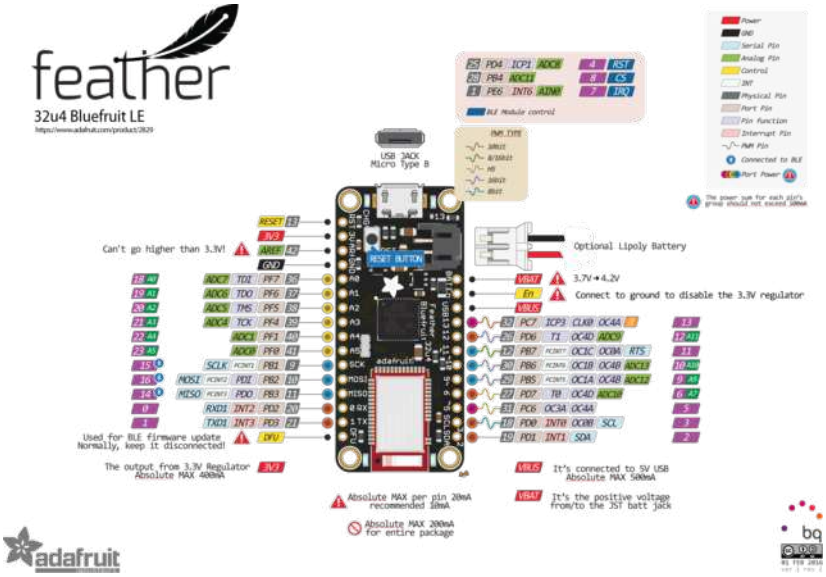
You can do a lot more too!

- The Bluefruit can also act like an HID Keyboard (http://adafru.it/iOA) (for devices that support BLE HID)
- Can become a BLE Heart Rate Monitor (http://adafru.it/iOB) (a standard profile for BLE) - you just need to add the pulse-detection circuitry
- Turn it into a UriBeacon (http://adafru.it/iOC), the Google standard for Bluetooth LE beacons. Just power it and the 'Friend will bleep out a URL to any nearby devices with the UriBeacon app installed.
- Built in over-the-air bootloading capability so we can keep you updated with the hottest new firmware (http://adafru.it/iOD). Use any Android or iOS device to get updates and install them. This will update the native code on the BLE module, to add new wireless capabilities, not program the ATmega chip.

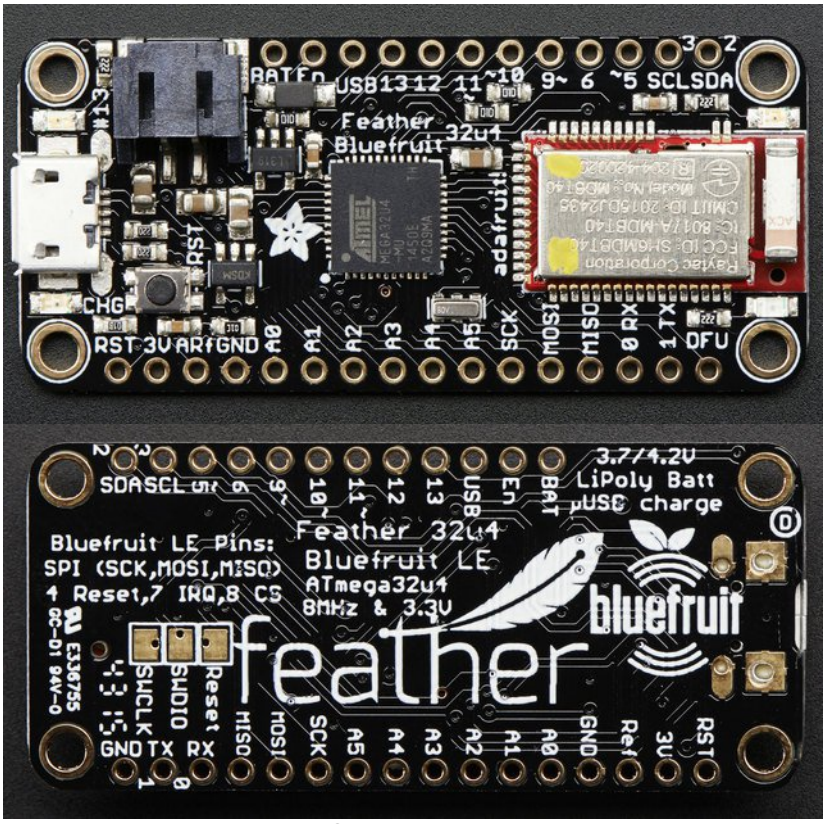


Comes fully assembled and tested, with a USB bootloader that lets you quickly use it with the Arduino IDE. We also toss in some header so you can solder it in and plug into a solderless breadboard. **Lipoly battery, MicroSD card and USB cable not included** (but we do have lots of options in the shop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

Pinouts

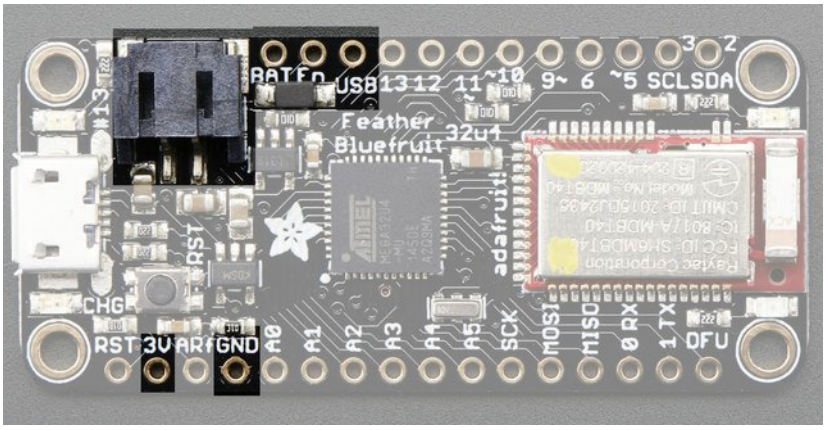


The Feather 32u4 Bluefruit LE is chock-full of microcontroller goodness. There's also a lot of pins and ports. We'll take you a tour of them now!



The DFU pin is accidentally labeled GND on the bottom, sorry about that! it should be labeled DFU, dont use it as a GND

Power Pins



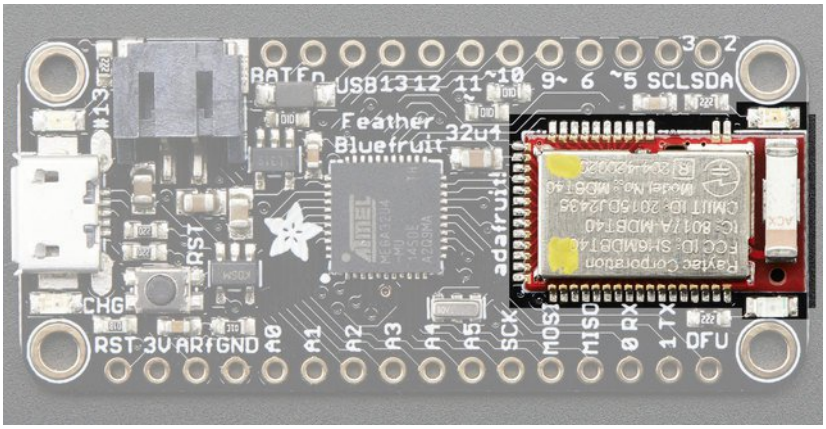
- **GND** - this is the common ground for all power and logic
- **BAT** - this is the positive voltage to/from the JST jack for the optional Lipoly battery
- **USB** - this is the positive voltage to/from the micro USB jack if connected
- **EN** - this is the 3.3V regulator's enable pin. It's pulled up, so connect to ground to disable the 3.3V regulator
- **3V** - this is the output from the 3.3V regulator, it can supply 500mA peak

Logic pins

This is the general purpose I/O pin set for the microcontroller. All logic is 3.3V

- **#0** / **RX** - GPIO #0, also receive (input) pin for **Serial1** and Interrupt #2
- **#1** / **TX** - GPIO #1, also transmit (output) pin for **Serial1** and Interrupt #3
- **#2** / **SDA** - GPIO #2, 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. Also Interrupt #1
- **#3** / **SCL** - GPIO #3, 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. Can also do PWM output and act as Interrupt #0.
- **#5** - GPIO #5, can also do PWM output
- **#6** - GPIO #6, can also do PWM output and analog input **A7**
- **#9** - GPIO #9, also analog input **A9** and can do PWM output. 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, also analog input **A10** and can do PWM output.
- **#11** - GPIO #11, can do PWM output.
- **#12** - GPIO #12, also analog input **A11**
- **#13** - GPIO #13, can do PWM output and is connected to the **red LED** next to the USB jack
- **A0 thru A5** - These are each analog input as well as digital I/O pins.
- **SCK**/**MOSI**/**MISO** - These are the hardware SPI pins, **used by the Bluefruit LE module too!** You can use them as everyday GPIO pins if you don't activate the Bluefruit and keep the BLE CS pin high. However, we really recommend keeping them free as they should be kept available for the Bluefruit. If they are used, make sure its with a device that will kindly share the SPI bus! Also used to reprogram the chip with an AVR programmer if you need.

Bluefruit LE Module + Indicator LEDs



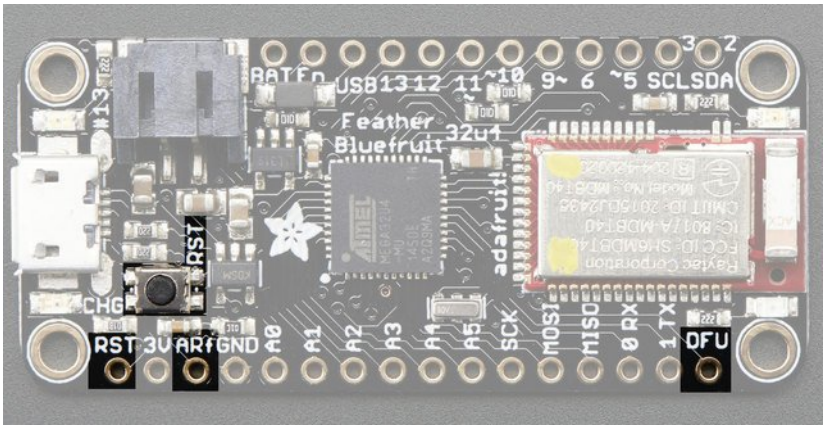
Since not all pins can be brought out to breakouts, due to the small size of the Feather, we use these to control the BLE module

- **#8** - used as the Bluefruit **CS** (chip select) pin
- **#7** - used as the Bluefruit **IRQ** (interrupt request) pin.
- **#4** - used as the Bluefruit **Reset** pin

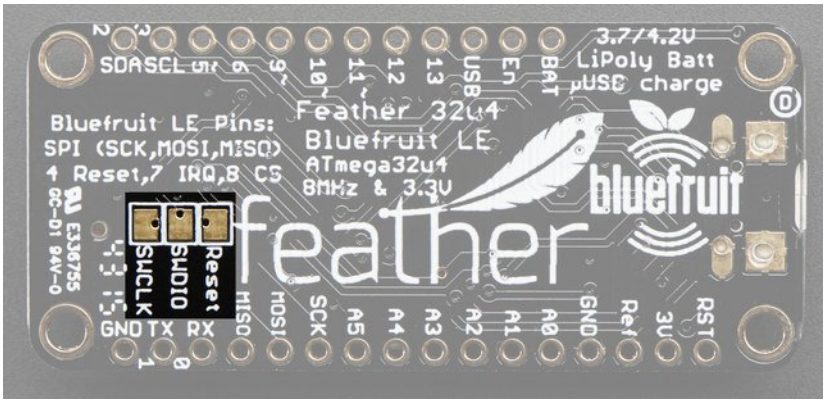
Since these are not brought out there should be no risk of using them by accident!

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!
- **DFU** - this is the force-DFU (device firmware upgrade) pin for over-the-air updates to the Bluefruit module. You probably don't need to use this but its available if you need to upgrade! Check out the **DFU Bluefruit Upgrades** page for how to use it. Otherwise, keep it disconnected.



On the back we also have **SWDIO/SWCLK/RST** pins, these are used for programming the Bluefruit LE module itself. You should not connect to these, unless you want to wipe out the Bluefruit LE module firmware for some reason. The RST pin is the factory reset pin, which is also rarely used, but you can use it to set the module back to the factory default settings if it gets really messed up.

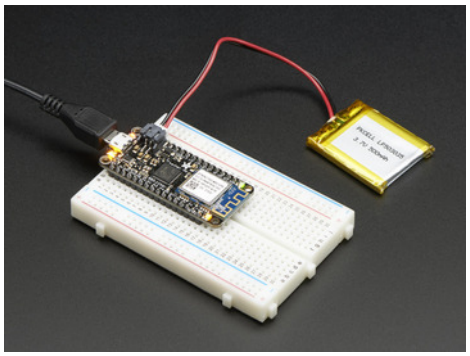
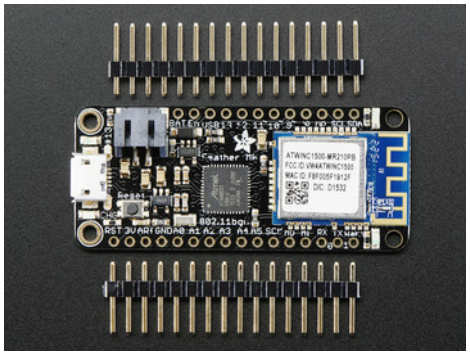
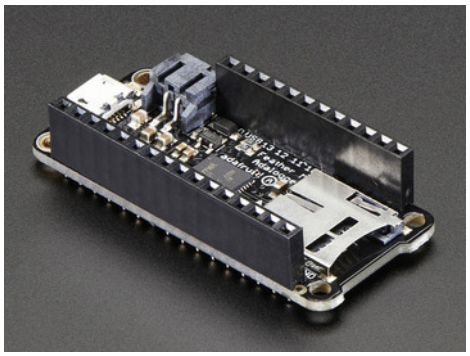
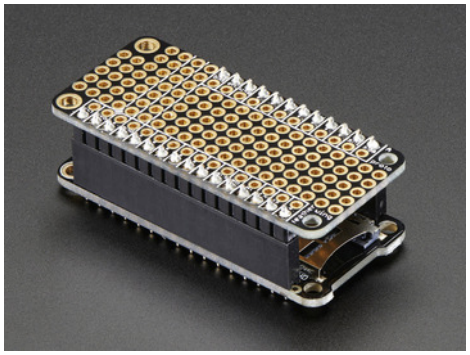


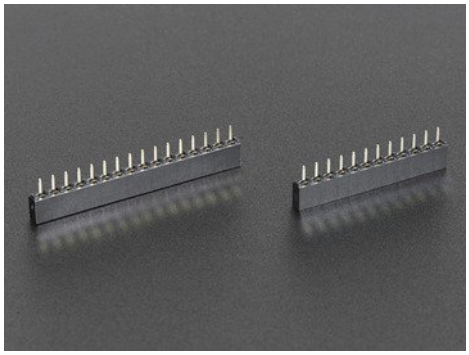
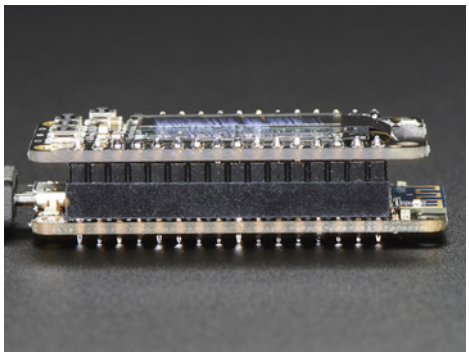
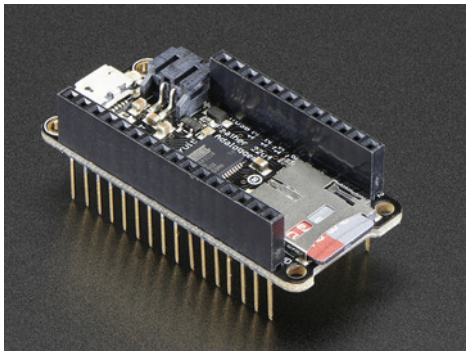
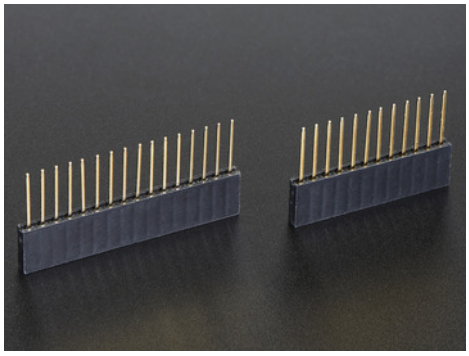
Assembly

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

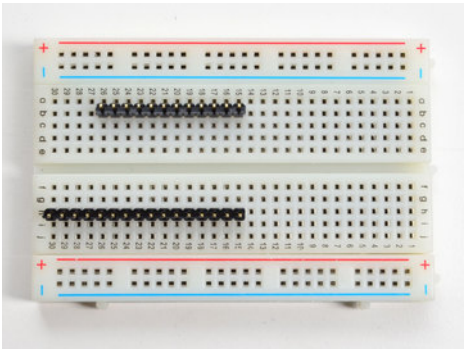
Header Options!

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

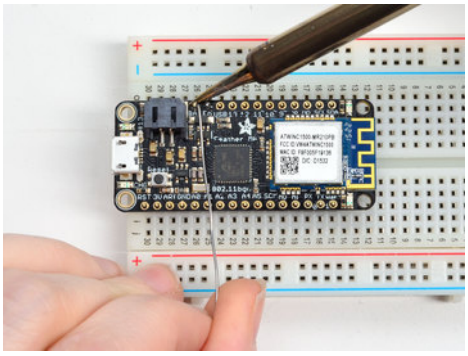
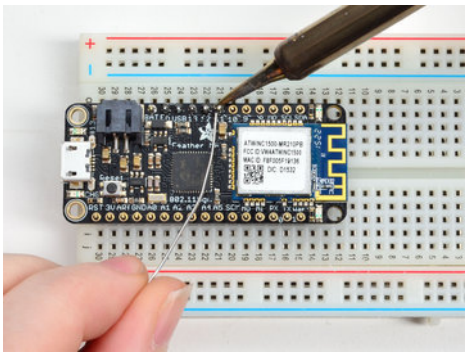
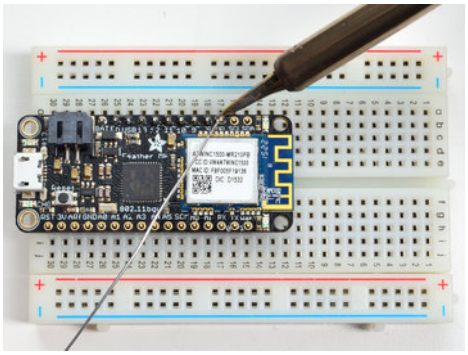
- 

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

- 
- 
- 

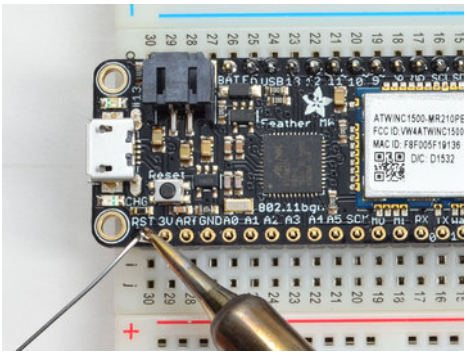
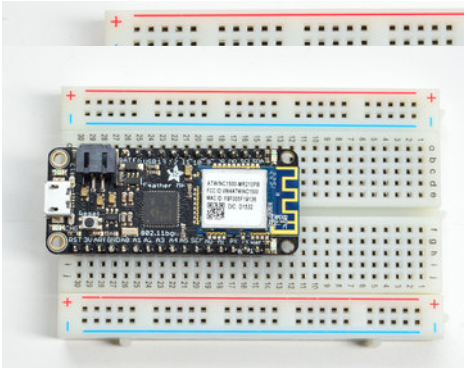
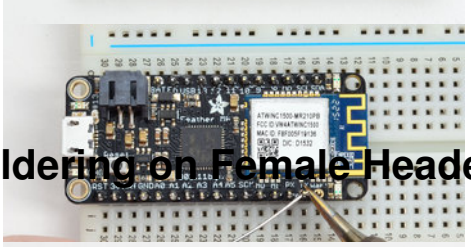
pads

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)).

Solder the other strip as well.

- 
- 
- 

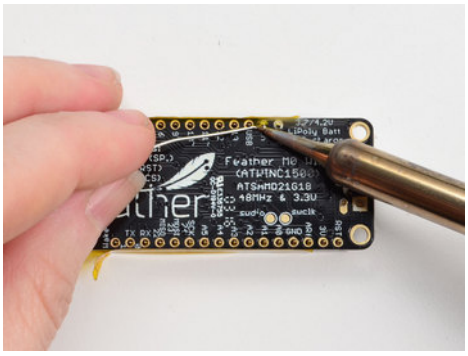
You're done! Check your solder joints visually and continue onto the next steps

Soldering on Female Header

- 
-

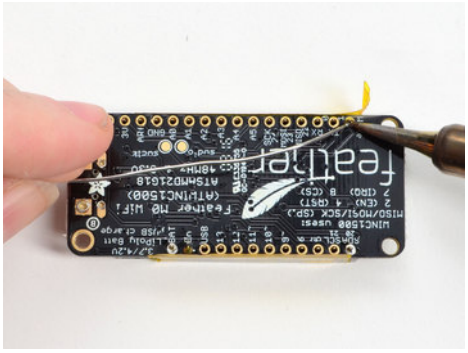
Tape In Place

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

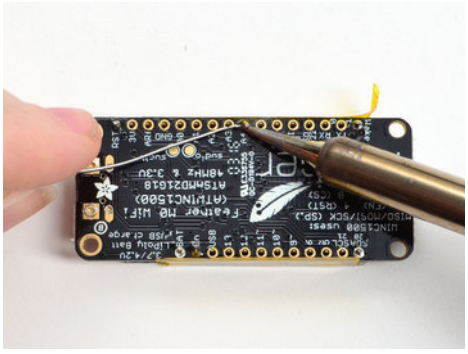
After flipping over, solder one or two points on each strip, to 'tack' the header in place

- 
- 

And Solder!

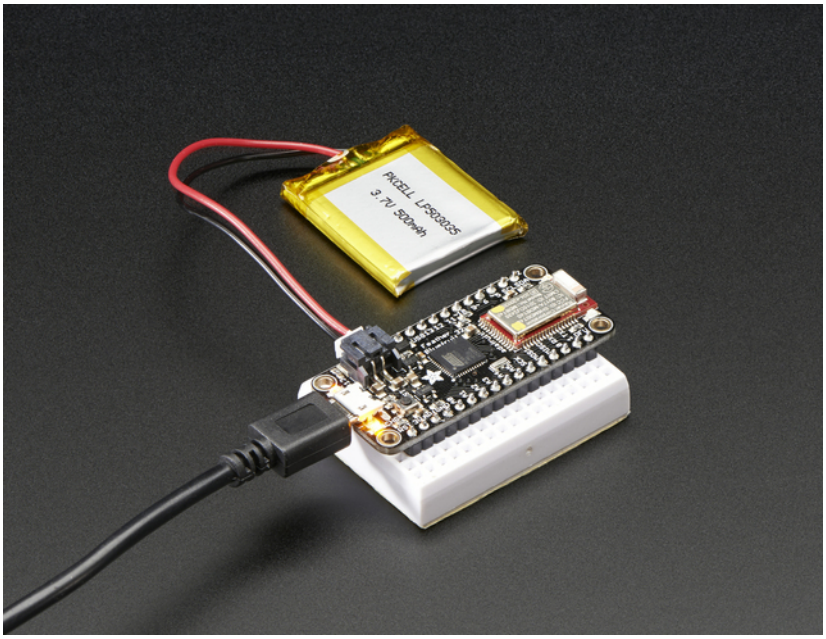
Be sure to solder all pins for reliable electrical contact.

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

- 
- 
- 

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

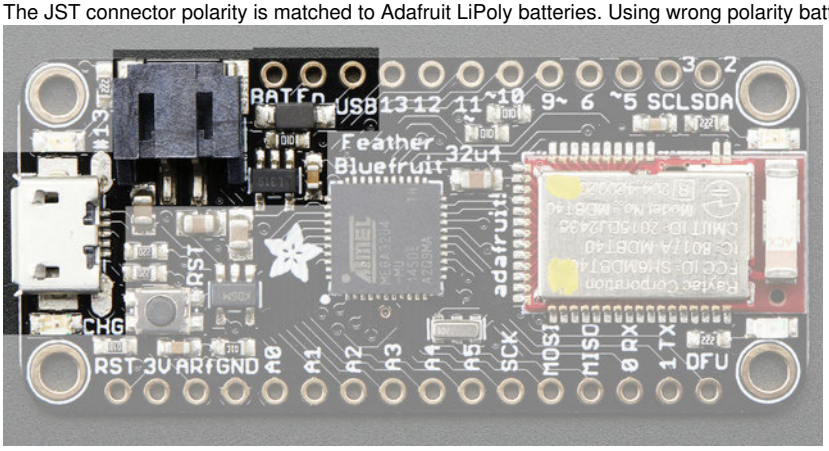
Power Management



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 100mA.** 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

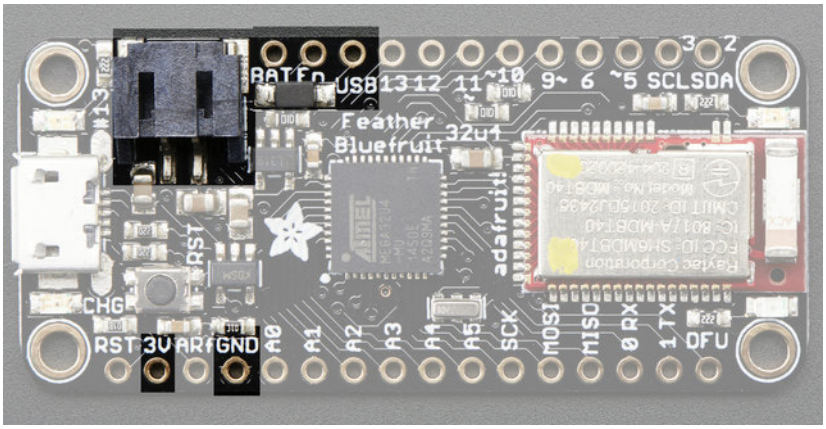


The above shows the Micro USB jack (left), Lipoly JST jack (top left), as well as the 3.3V regulator and changeover diode (just to the right of the JST jack) and the Lipoly charging circuitry (to the right of the Reset button). There's also a **CHG** LED, which will light up while the battery is charging. 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 500mA peak SPX3819. While you can get 500mA from it, you can't do it continuously from 5V as it will overheat the regulator. It's fine for, say, powering an ESP8266 WiFi chip or XBee radio

though, since the current draw is 'spiky' & sporadic.



Measuring Battery

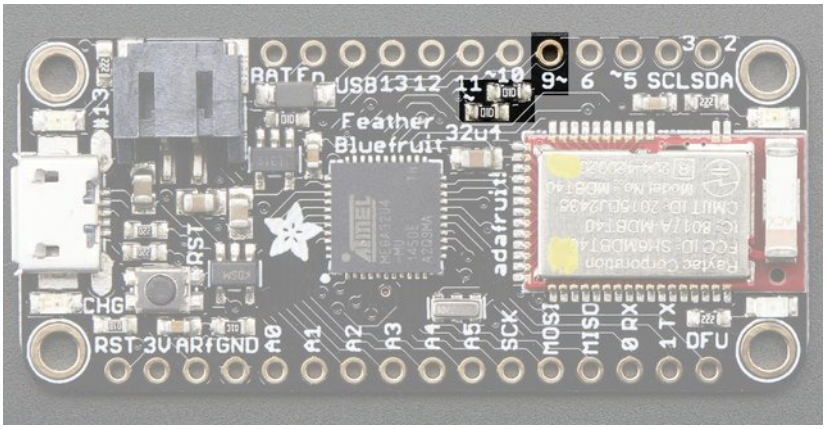
If you're running off of a battery, chances are you wanna know what the voltage is at! That way you can tell when the battery needs recharging. Lipoly batteries are 'maxed out' at 4.2V and stick around 3.7V for much of the battery life, then slowly sink down to 3.2V or so before the protection circuitry 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 #9 **A9**). You can read this pin's voltage, then double it, to get the battery voltage.

```
#define VBATPIN A9

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);
```

This voltage will 'float' at 4.2V when no battery is plugged in, due to the lipoly charger output, so its not a good way to detect if a battery is plugged in or not (there is no simple way to detect if a battery is plugged in)



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