



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!



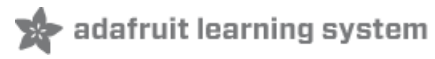
Contact us

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 nRF52 Pro Feather with Mynewt

Created by lady ada



Last updated on 2017-11-23 12:05:49 AM UTC

Guide Contents

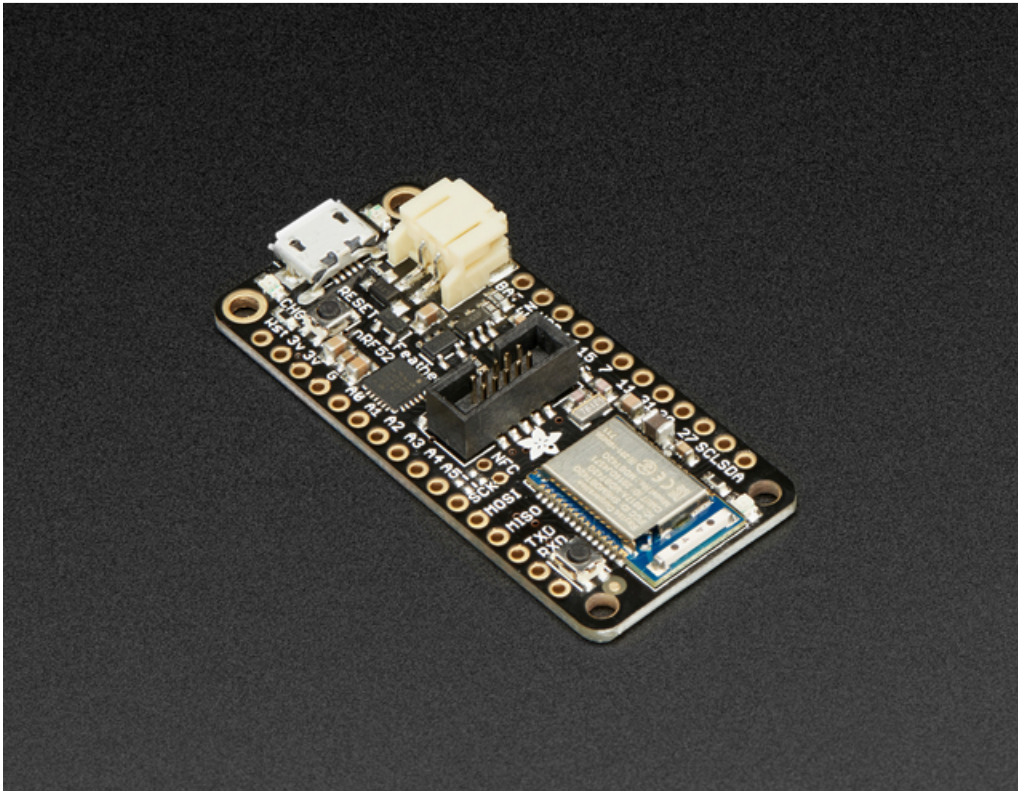
Guide Contents	2
Overview	6
Bluetooth Certification Warning	10
Device Pinout	12
Special Notes	13
Power Pins	13
Analog Inputs	14
PWM Outputs	14
I2C Pins	14
Assembly	15
Header Options!	15
Soldering in Plain Headers	18
Prepare the header strip:	18
Add the breakout board:	19
And Solder!	19
Soldering on Female Header	21
Tape In Place	21
Flip & Tack Solder	21
And Solder!	23
Native Installation (OS X)	25
Install newt and newtmgr Binaries	25
Install newt	25
Install newtmgr	25
Install an ARM Cross-Compiling Toolchain	26
Optional: Install the OpenOCD Debugger	27
Optional: Install Segger J-Link Drivers	27
Optional: Install minicom	27
Native Installation (Linux)	29
Install newt and newtmgr Binaries	29
Manual newt Installation	29
Manual newtmgr Installation	29
Install an ARM Cross-Compiling Toolchain	29
Optional: Install the OpenOCD Debugger (Segger J-Link)	29
Optional: Install Segger J-Link Drivers	30
Optional: Install minicom	31
Native Installation (Win)	32
Cut auto-reset trace	32
Newt	32
Setup Go Path	32
Install newtmgr Binaries	34
ARM Cross-Compiler	35
Apache Mynewt Tools	37

Standard Tools	37
Adafruit Tools	37
newt	38
Installing newt	38
Common newt Commands	38
newt build <target_name>	38
newt create-image <target_name> <version>	38
newt load <target_name>	38
newt size <target_name>	39
newt target show	39
newtmgr	41
Installing newtmgr	41
Connection Profiles	41
Adding a Serial Connection Profile	41
Listing Existing Profiles	41
Test the Serial Connection Profile	41
Common newtmgr Commands	42
Display Scheduler Task Activity via 'taskstat'	42
Uploading Application Images with newtmgr	42
1. Build the Target Application (newt build)	42
2. Sign and Version the Image (newt create-image)	42
3. Upload the Image (newtmgr image upload)	43
4. Test the Image Upload (newtmgr image test [hash])	43
5. Reset to Perform the Test and Image Bank Switch (newtmgr reset)	44
6. Confirm the Image Switch (newtmgr image confirm)	44
Display Internal Statistics	45
List all Statistic Groups	45
List all Values in a Statistic Group	46
Adafruit Mynewt Manager	47
Getting the Application	47
Source Code	48
Apache Mynewt Applications	49
Create a New Project	50
Create a Project Skeleton	50
Download Project Dependencies	50
Create a New Application	51
Create a New Target	53
1. Set the Target's app Field	53
2. Set the Target's bsp Field	53
3. Set the build_profile Field	54
4. Test the Target Settings	54
Final Project Structure	54
Build and Flash the Target	54
Sign the Build with a Version Number	55
Flash the Image via a Segger J-Link	55
Flash the Image via the Serial Port and newtmgr	55
Adding Tasks	56

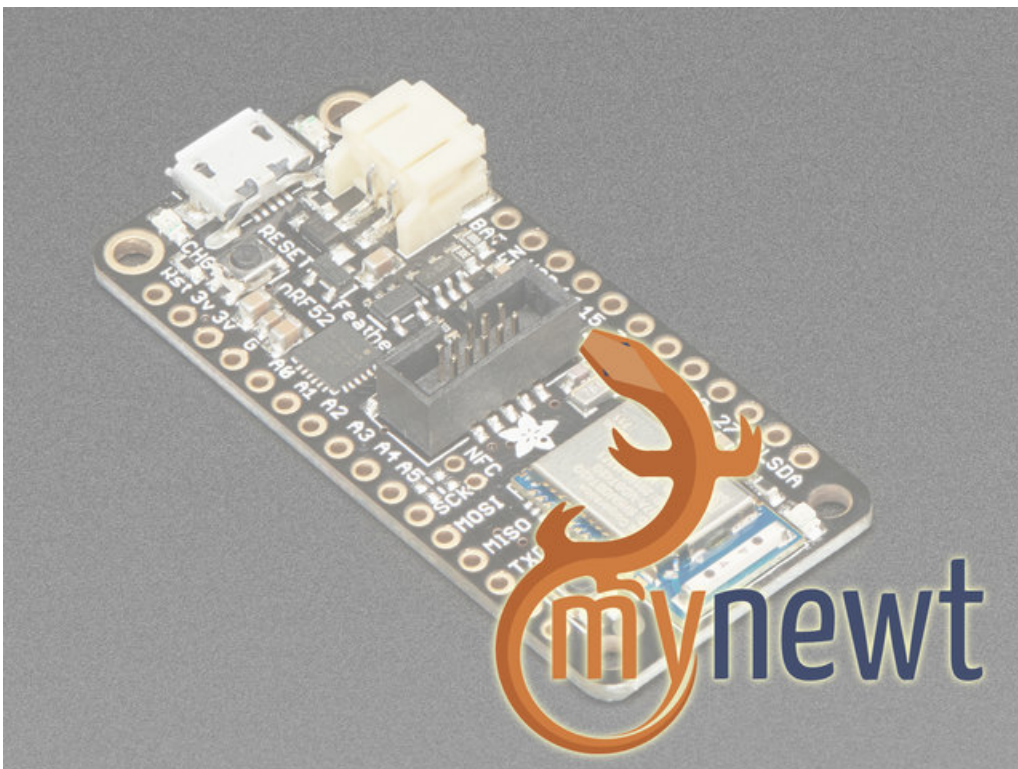
Declaring a task, priority and stack size	56
Initializing a task	56
Implementing the task handler	57
Task Delays	58
os_time_delay	58
Example: Updating apps/first/src/main.c	58
Checking Task Status	59
Adding Shell Commands	61
Adding shell support	61
Adding a custom command handler	61
Example: Updating apps/first/src/main.c	62
Testing the Shell Command	63
Adding Statistics	65
Configuring Your App for Statistics	65
Adding Stats to your main.c File:	65
Accessing the Stats in Your Code	66
Initializing the Stats	66
Updating the Stats Values	67
Incrementing	67
Accessing Stats with the Console or newtmgr	67
Console Access	67
newtmgr Access	68
Example: Adding Stats to apps/first/src/main.c	68
Monitoring via netwmgr	70
Monitoring via shell/console	70
Adding BLE UART Support	71
Mynewt Nimble (BLE Stack) Documentation	71
Advanced Debugging	72
GDB Debugging	73
Starting the Debugger	73
Displaying Values	73
Basic Example	73
Formatting Display Values	74
Displaying an Array of Values	74
Useful Mynewt/Nimble Structs and Fields	75
Memory Manipulation	75
Stack Manipulation	76
Display the Stack Trace	76
Display Stack Frame Details	77
Displaying ARM Registers	77
Working with Source Code	78
Displaying a function	78
Displaying code at a specific address	79
Running an arbitrary function when halted at a breakpoint	79
Command Line Debugging	80

Grep'ing Source Code	80
Grep recursively for a partial string	80
Grep recursively for an exact string	80
Grep recursively for a string ignoring the case	80
Grep recursively with specific file types	80
Field Debugging Tips	82
Debugging Crash Dumps	82
Option 1: Debugging Crash Dumps with GDB	82
Option 2: Debugging Crash Dumps with objdump	83
Debugging Repeatable Crashes	83
Adafruit_Mynewt	85
Installing Adafruit_Mynewt	85
Provided Apps	85
Helper Modules	85
External Content	86
Mynewt Bootloader	86
Build and Package Management	86
Downloads	87
Module Details	87
Schematic	87

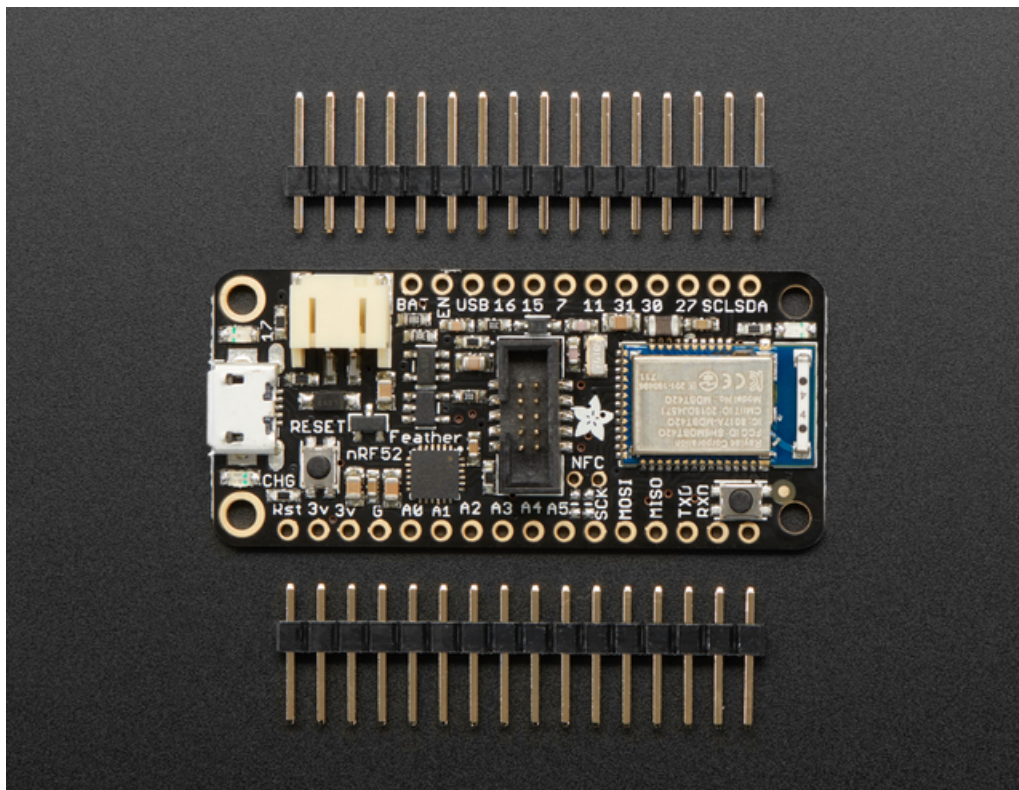
Overview



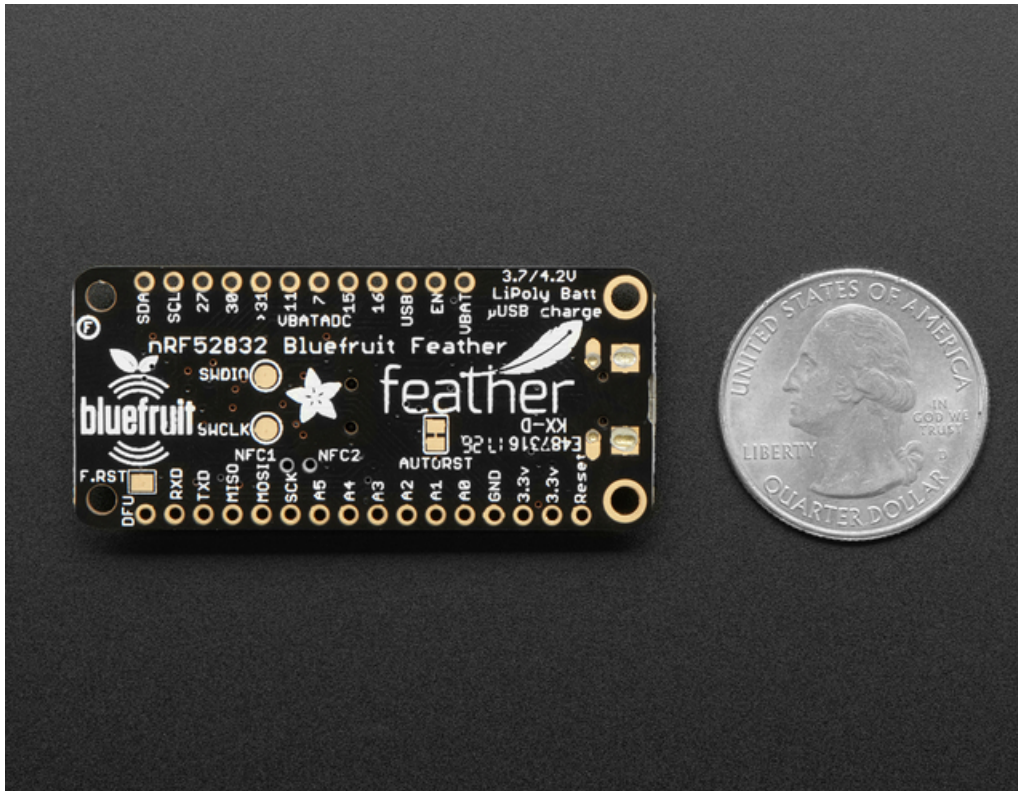
The **Adafruit Feather nRF52 Pro** is our latest Bluetooth Low Energy board for advanced projects and users who want to use a **fully open source Bluetooth Low Energy 5.0 stack**. It has a native-bluetooth chip, the nRF52832, as well as built in USB Serial and battery charging! [We have other boards in the Feather family, check'em out here.](#)



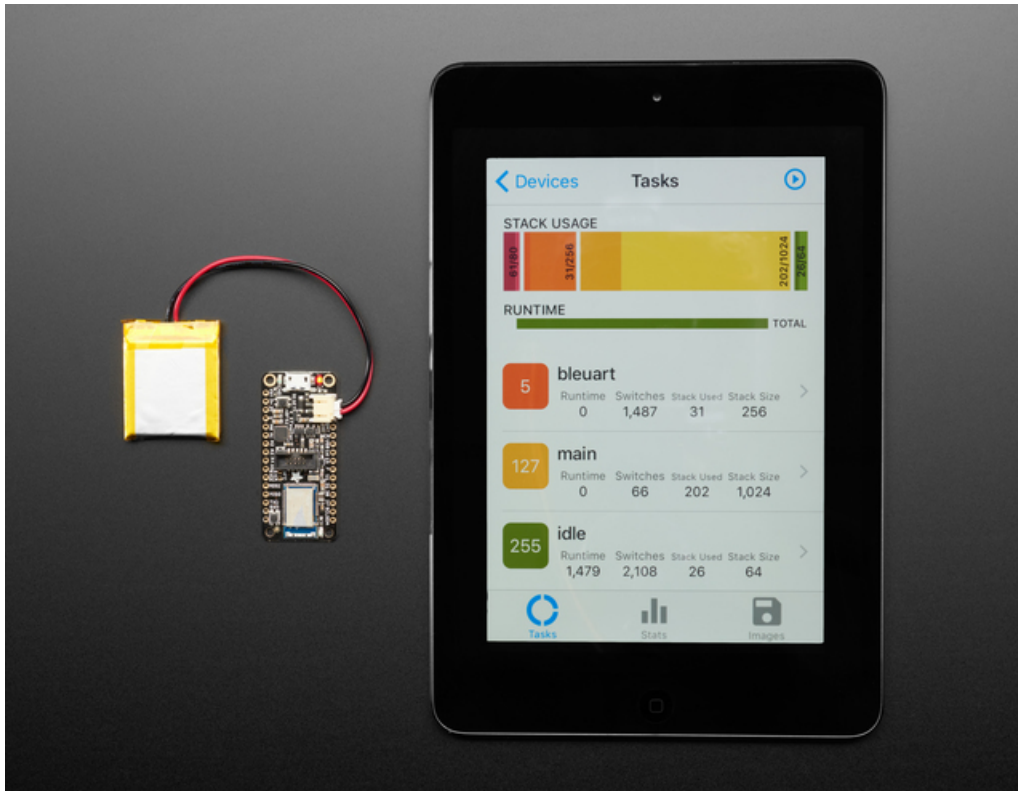
This is a special Feather - unlike the rest of the Feather family, this board is not for use with Arduino IDE. Instead, it is for use with Mynewt only! We have programmed it with the Mynewt bootloader and updated the hardware to add an SWD connector and an additional DFU button. [If you want to use the nRF52 with Arduino IDE please check out the Bluefruit nRF52 Feather which works out-of-the-box with our Arduino board support package.](#) This Feather is for advanced users only, you will be interacting with the Mynewt RTOS rather than Arduino and you cannot easily go back-and-forth without an SWD programmer.



This chip has twice the flash, SRAM and performance of the earlier nRF51-based modules, and can do a lot of heavy lifting. That extra processing power and memory means you can now run an RTOS for fast development of complex projects. This board was designed to work with the [Apache Mynewt](#) operating system, which is released by the Apache Foundation under the permissive and commercial-friendly Apache license. Apache Mynewt includes a number of professionally written networking stacks (BLE 5.0, [OIC/OCF](#), etc.), development tools, and professional project management features like a secure bootloader that allows you to cryptographically sign firmware images and verify them during device updates.



The Adafruit Feather nRF52 Pro ships pre-programmed with the Mynewt serial bootloader that allows you to flash firmware to the device directly from the command-line using the on-board USB Serial converter and the 'newtmgr' tool, so you can get started right away with no additional hardware required. For more advanced debugging, however, you can use a Segger J-Link and the on-board SWD debug connector, which gives you access to a more powerful set of development tools and options.



We also have a custom [Mynewt Manager](#) application that we've written for iOS that allows you to perform over the air firmware updates, and get live updates of the tasks running in the task manager, or to see any statistics generated by the internal reporting systems that are part of the OS.

Features:

- ARM Cortex M4F (with HW floating point acceleration) running at 64MHz
- 512KB flash and 64KB SRAM
- **Built in USB Serial converter for fast and efficient programming and debugging**
- Bluetooth Low Energy compatible 2.4GHz radio (Details available in the [nRF52832](#) product specification)
- **FCC / IC / TELEC certified module**
- Up to +4dBm output power
- 1.7v to 3.3v operation with internal linear and DC/DC voltage regulators
- 19 GPIO, 8 x 12-bit ADC pins, up to 12 PWM outputs (3 PWM modules with 4 outputs each)
- Pin #17 red LED for general purpose blinking
- Power/enable pin
- 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
- 4 mounting holes
- Reset button
- SWD connector for debugging
- 100% open source firmware when used with [Apache Mynewt](#)

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+.

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 it's 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.

The Power of Bluefruit LE

The nRF52 Pro uses an nRF52832 chipset from Nordic, which can be used as both a main microcontroller and a bluetooth low energy interface. For most people, they'll be very happy to use the standard Nordic UART RX/TX connection profile - example code is provided! 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](#) or [Android App](#), or [write your own to communicate with the UART service](#).

The board is capable of much more than just sending strings over the air! Thanks to the Apache Mynewt operating system, you have full control over how the device behaves, including the ability to define and manipulate your own [GATT Services and Characteristics](#), or change the way that the device advertises itself for other Bluetooth Low Energy devices to see.

Use the Bluefruit App to get your project started

Using our Bluefruit [iOS App](#) or [Android App](#), you can quickly get your project prototyped by using your iOS or Android phone/tablet as a controller. This data can be read over BLE and processed directly by the nRF52 microcontroller

Comes fully assembled and tested, with a USB Serial bootloader that lets you quickly use it from the command line. We also toss in some header so you can solder it in and plug into a solderless breadboard. [Lipoly battery and MicroUSB cable not included](#) (but we do have lots of options in the shop if you'd like!)

Bluetooth Certification Warning

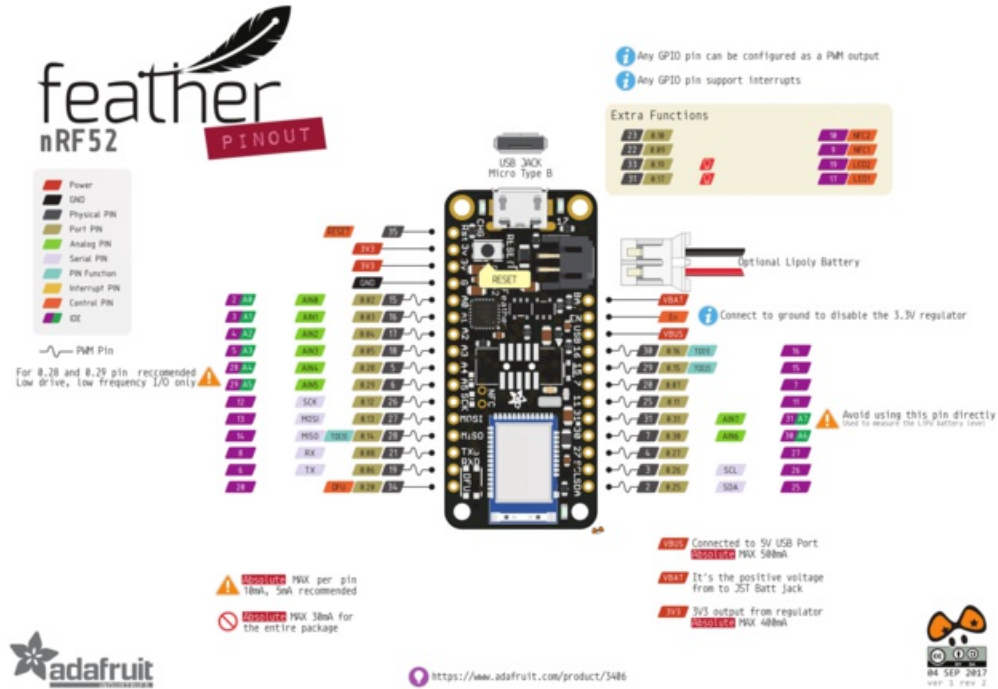
Because of complex and expensive Bluetooth SIG certification requirements, the nRF52 Pro doesn't ship with any default Bluetooth functionality on the post-production firmware, since at present the Mynewt BLE stack has not gone through the certification process with the Bluetooth SIG. We aren't able to ship BLE functionality out of the box without this certification. In order to use Mynewt in a production product, you will need to go through the certification process for your product, although you are free to develop locally with Mynewt in a test environment.

The only Bluetooth Low Energy stack certified to run on the nRF52 modules with no additional work on your part is the Nordic SoftDevice, which is used in the Arduino variant of these boards and can be flashed onto the nRF52 Pro using your choice of ARM SWD debugger, such as the [Segger J-Link](#) (Adafruit bootloader and Nordic SD hex files are [available here](#)).

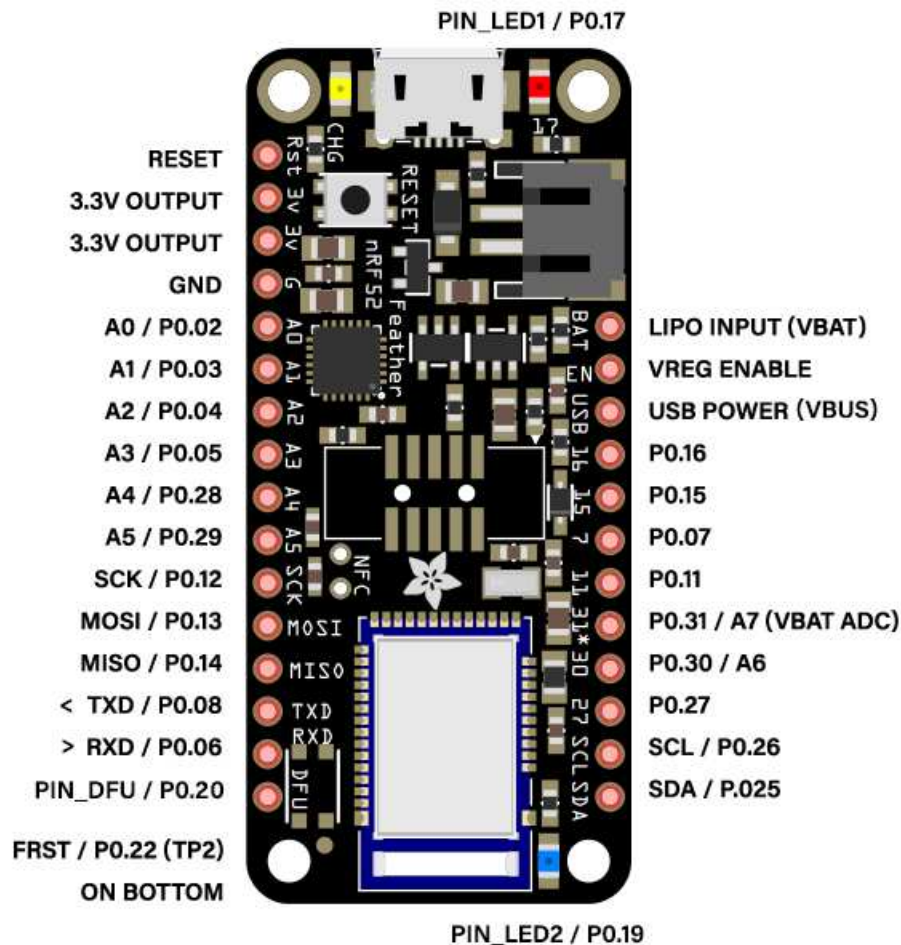
When the Mynewt stack is certified by the Bluetooth SIG (no ETA on if or when that will happen since it's a long and expensive process), Bluetooth functionality will be enabled on shipping devices, but at present only USB Serial is used in the default firmware on production devices, and the Nordic SD should be used in commercial products if you are not able to go through the certification process yourself. There are no HW restrictions on the nRF52 Pro boards, we simply have our hands tied on what we can ship and claim coming off the production line.

UPDATE (Sept 17 2017): Runtime, one of the main contributors to the Mynewt core codebase, have contracted an external company to certify nimble (the Mynewt BLE stack). Once the certification process is completed, the results and certificate details will be posted here.

Device Pinout



BLUEFRUIT NRF52 FEATHER PINOUT



Special Notes

The following pins have some restrictions that need to be taken into account when using them:

- **PIN_DFU / P0.20:** If this pin is detected to be at GND level at startup, the board will enter a special serial bootloader mode and will not execute any user code, going straight into bootloader mode. If you wish to use this pin as a standard GPIO, make sure that it is pulled high with a pullup resistor so that your code will execute normally when the MCU starts up.
- **P0.31 / A7:** This pin is hard wired to a voltage-divider on the LIPO battery input, allow you to safely measure the LIPO battery level on your device. If possible, **you should avoid using this pin directly.**
- **FRST/P0.22:** Setting this pin to GND at startup will cause the device to perform a factory reset at startup, erasing and config data as well as the user sketch. At the next reset, you should enter serial bootloader mode by default, since no user sketch will be present. You can use this to recover 'bricked' boards, but if you don't wish to do this be careful not to have FRST low at startup. By default, a weak internal pull-up resistor is enabled on this pin during the bootloader phase.

Power Pins

- **3.3V Output:** This two pins are connected to the output of the on board 3.3V regulator. They can be used to

supply 3.3V power to external sensors, breakouts or Feather Wings.

- **LIPO Input (VBAT):** This is the voltage supply off the optional LIPO cell that can be connected via the JST PH connector. It is nominally ~3.5-4.2V.
- **VREG Enable:** This pin can be set to GND to disable the 3.3V output from the on board voltage regulator. By default it is set high via a pullup resistor.
- **USB Power (VBUS):** This is the voltage supply off USB connector, nominally 4.5-5.2V.

Analog Inputs

The 8 available analog inputs can be configured to generate 8, 10 or 12-bit data (or 14-bits with over-sampling), at speeds up to 200kHz (depending on the bit-width of the values generated), based on either an internal 0.6V reference or the external supply.

The following default values are used:

- **Default voltage range:** 0-3.6V (uses the internal 0.6V reference with 1/6 gain)
- **Default resolution:** 10-bit (0..4095)

Unlike digital functions, which can be remapped to any GPIO/digital pin, the ADC functionality is tied to specified pins, labelled as A* in the image above (A0, A1, etc.).

PWM Outputs

Any GPIO pin can be configured as a PWM output, using the dedicated PWM block.

Three PWM modules can provide up to 12 PWM channels with individual frequency control in groups of up to four channels.

Please note that DMA based PWM output is still a work in progress in the initial release of the nRF52 BSP, and further improvements are planned here.

I2C Pins

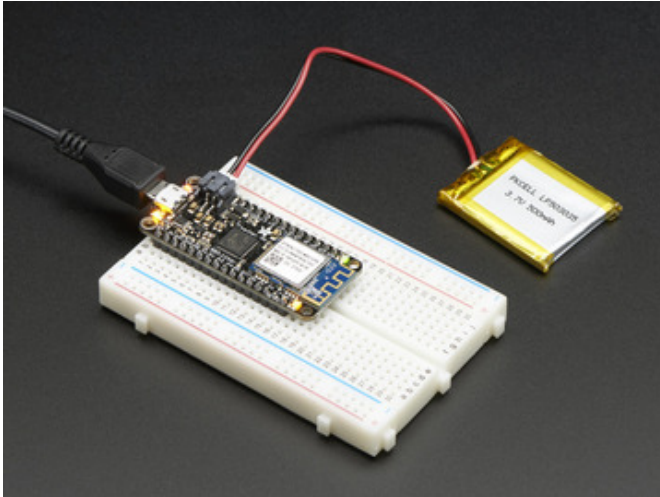
I2C pins on the nRF52832 require external pullup resistors to function, which are not present on the Adafruit nRF52 Feather by default. You will need to supply external pullups to use these. All Adafruit I2C breakouts have appropriate pullups on them already, so this normally won't be an issue for you.

Assembly

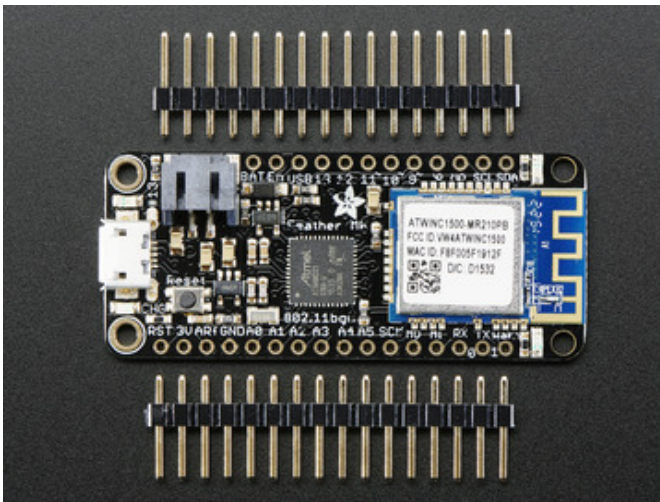
We ship Feathers fully tested but without headers attached - this gives you the most flexibility on choosing how to use and configure your Feather

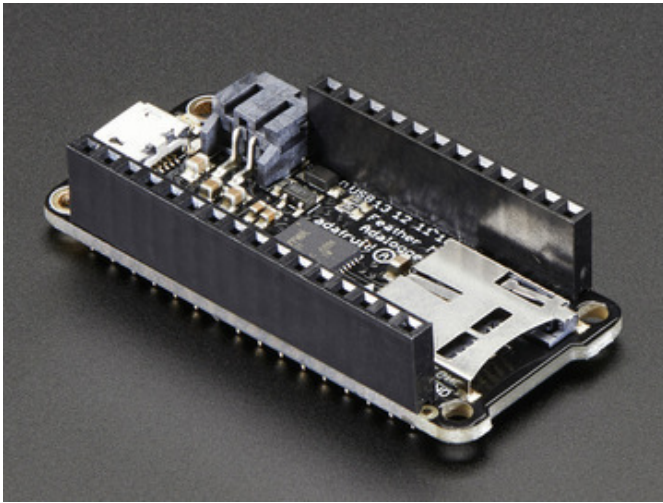
Header Options!

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

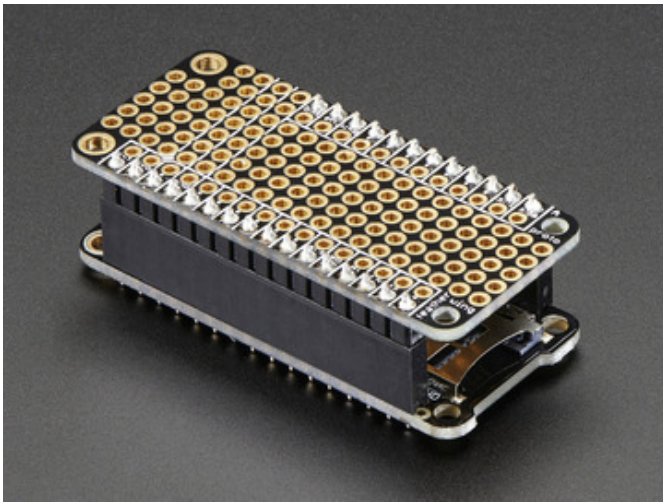


The first option is soldering in plain male headers, this lets you plug in the Feather into a solderless breadboard

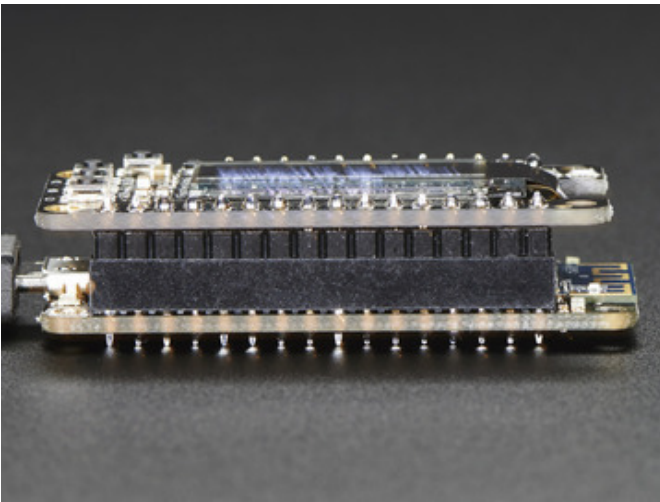
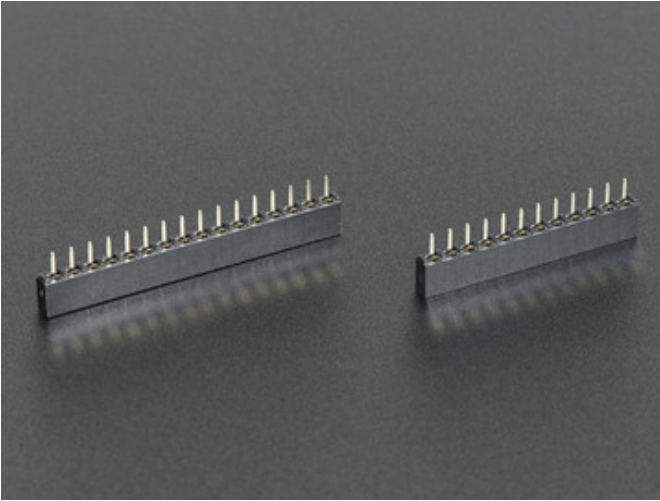


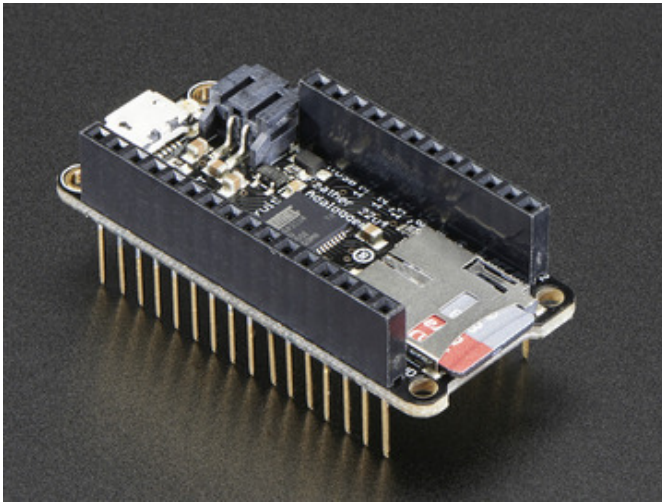


Another option is to go with socket female headers. This won't let you plug the Feather into a breadboard but it will let you attach featherwings very easily

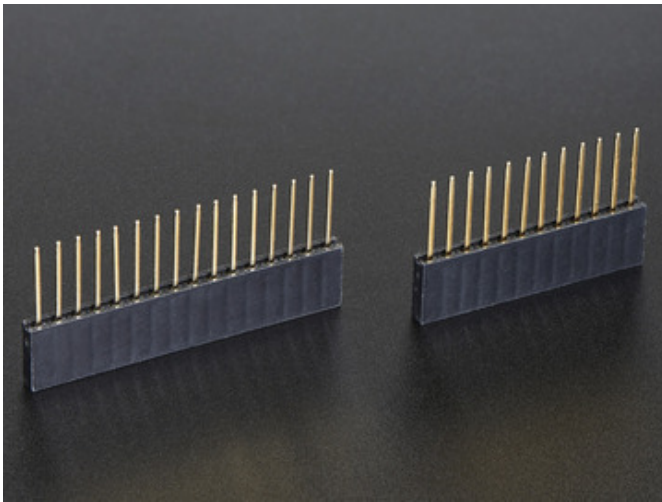


We also have 'slim' versions of the female headers, that are a little shorter and give a more compact shape

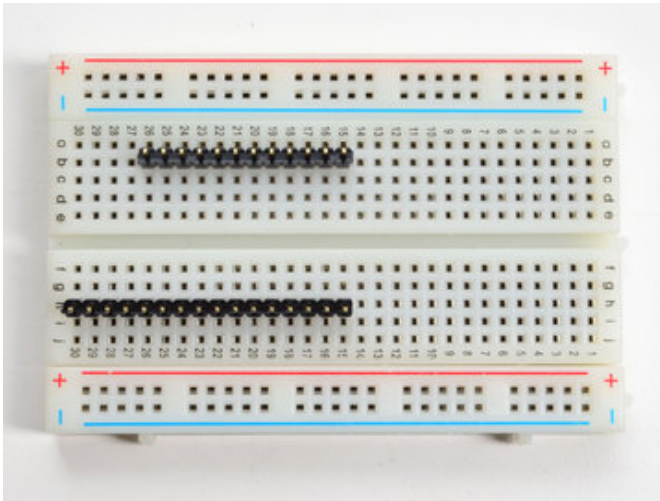




Finally, there's the "Stacking Header" option. This one is sort of the best-of-both-worlds. You get the ability to plug into a solderless breadboard *and* plug a featherwing on top. But its a little bulky

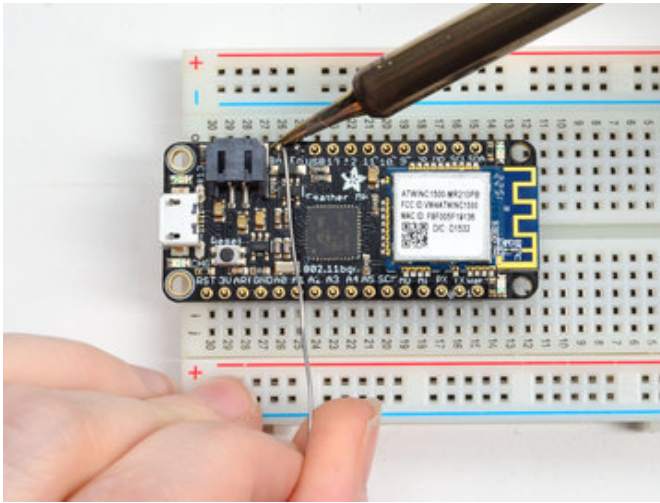


Soldering in Plain Headers



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**



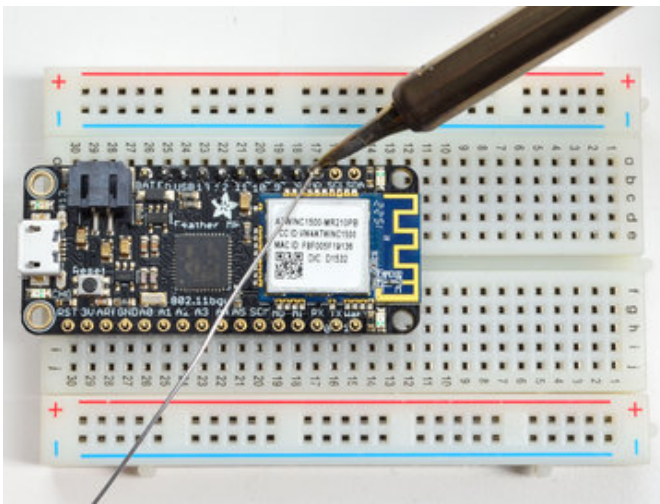
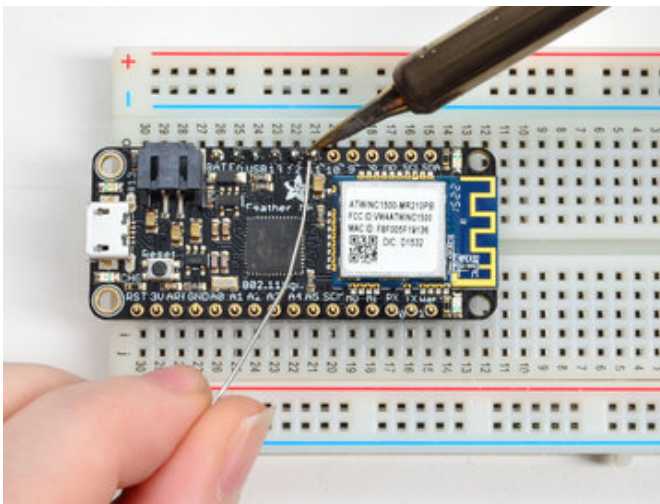
Add the breakout board:

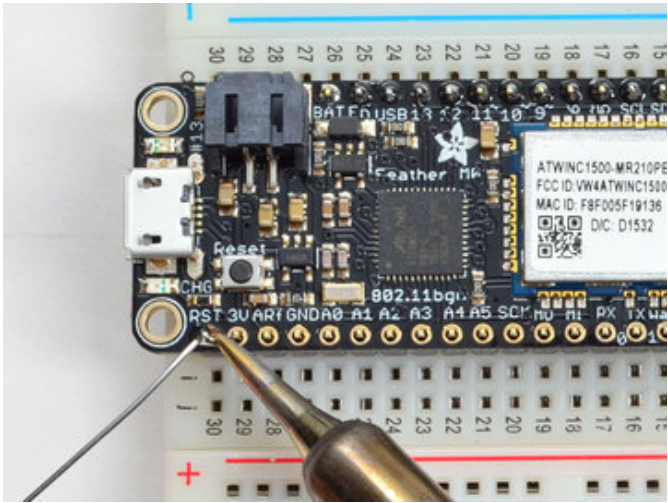
Place the breakout board over the pins so that the short pins poke through the breakout pads

And Solder!

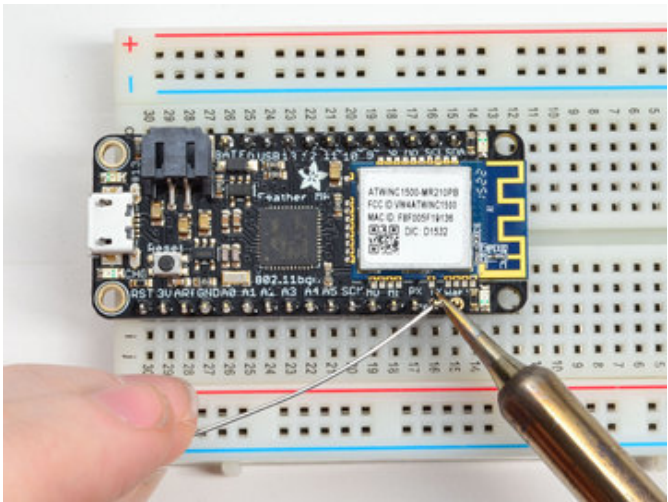
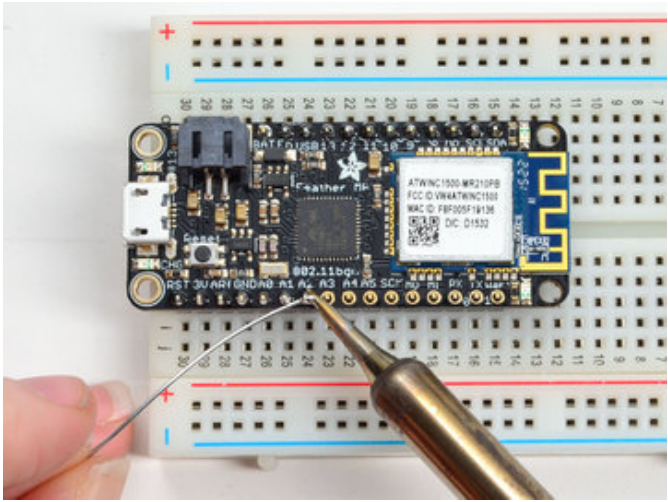
Be sure to solder all pins for reliable electrical contact.

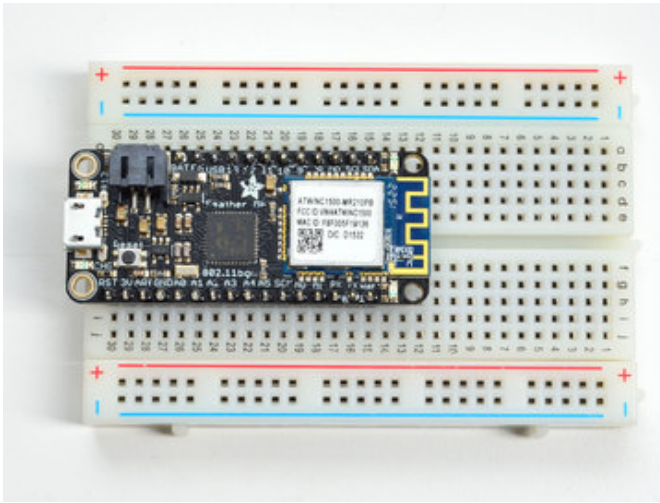
(For tips on soldering, be sure to check out our [Guide to Excellent Soldering](https://adafruit.it/aTk) (<https://adafruit.it/aTk>)).





Solder the other strip as well.





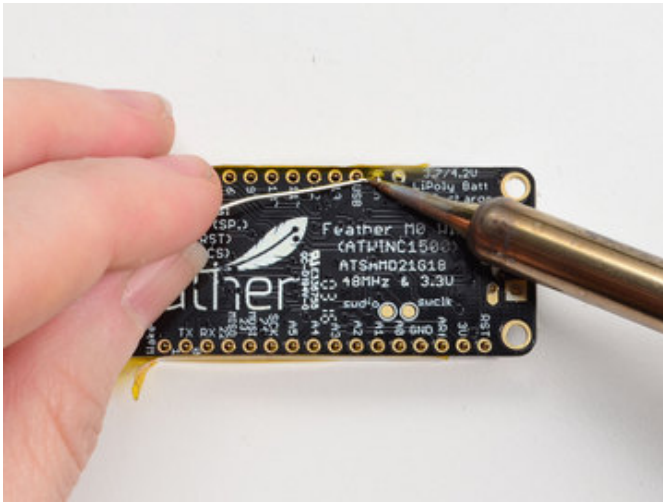
You're done! Check your solder joints visually and continue onto the next steps

Soldering on Female Header



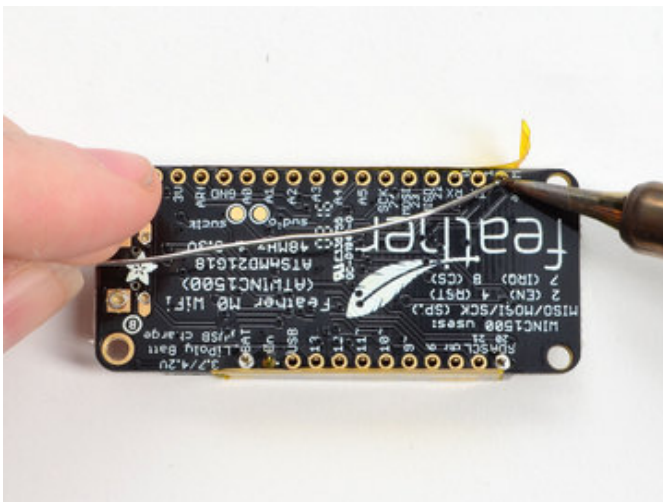
Tape In Place

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

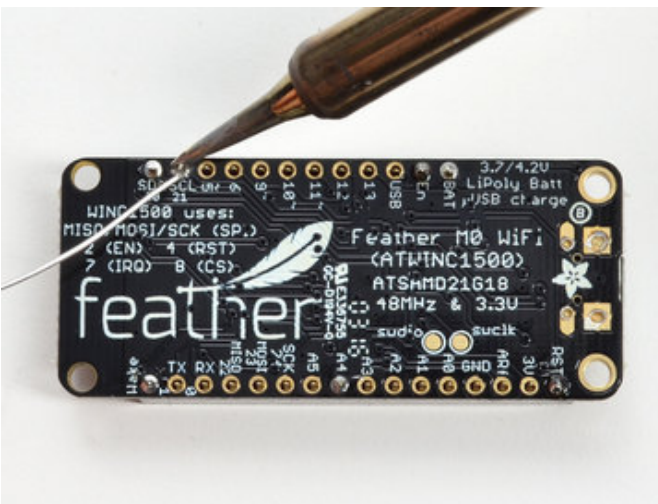
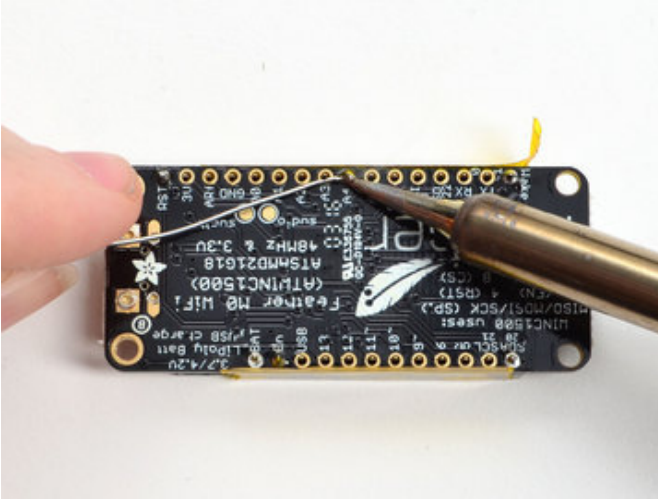
After flipping over, solder one or two points on each strip, to 'tack' the header in place



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](https://adafruit.it/aTk) (<https://adafruit.it/aTk>)).



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

