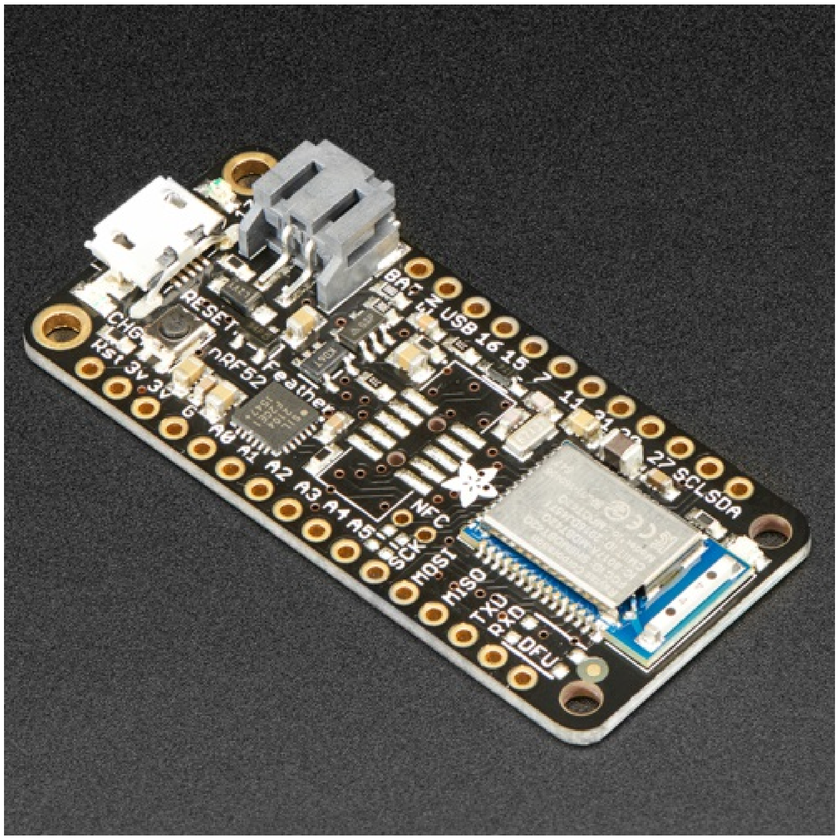
Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

Contact us

Bluefruit nRF52 Feather Learning Guide

Created by Kevin Townsend



Last updated on 2017-04-27 07:29:12 PM UTC

Guide Contents

Introduction

The **Adafruit Feather nRF52 Bluefruit** is our latest easy-to-use all-in-one Bluetooth Low Energy board, with a native-bluetooth chip, the nRF52832! It's our take on an 'all-in-one' Arduino-compatible + Bluetooth Low Energy with built in USB and battery charging.

This chip has twice the flash, SRAM and performance of the earlier nRF51-based Bluefruit modules. Best of all it has Arduino IDE support so there is no 'helper' chip like the ATmega32u4 or ATSAMD21. Instead, this chip is programmed directly! It's got tons of awesome peripherals: plenty of GPIO, analog inputs, PWM, timers, etc. Leaving out the extra microcontroller means the price, complexity and power-usage are all lower/better. It allows you to run code directly on the nRF52832, straight from the Arduino IDE as you would with any other MCU or Arduino compatible device. A single MCU means better performance, lower overall power consumption, and lower production costs if you ever want to design your own hardware based on your Bluefruit nRF52 Feather project!

The chips are pre-programed with an auto-resetting bootloader so you can upload quickly in the Arduino IDE with no button-pressing. Want to program the chip directly? You can use our command line tools with your favorite editor and toolchain.

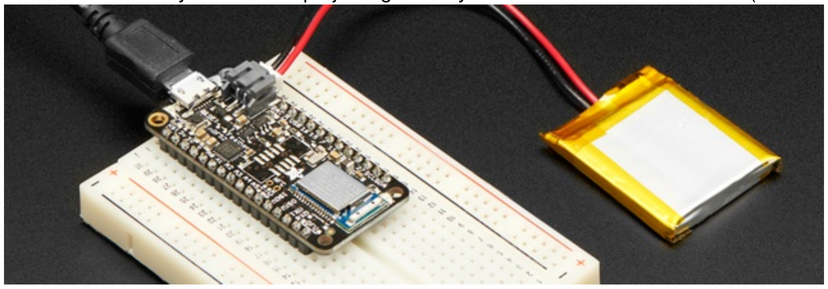
And to get you up and running quickly, we've done all the heavy lifting of getting the low level BLE stack into shape so that you can focus on your project from day one!

nRF52832 Technical Details

- 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 (http://adafru.it/vaJ) 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
- Optional SWD connector for debugging
- Works out of the box with just about all of our Adafruit FeatherWings! (http://adafru.it/vby) (Wings that require the UART like the GPS FeatherWing won't work)

Further technical details available in the nRF52832 (http://adafru.it/vaJ) product specification.

Like all of our Feather boards, the Bluefruit nRF52 Feather includes on board USB-based LIPO charging, and has a standard LIPO battery connector to make your wireless projects genuinely 'wireless' at no additional cost (aside from the LIPO cell itself).



nRF51 or nRF52 Bluefruit Devices?

The Bluefruit nRF52 Feather (based on the nRF52832 (http://adafru.it/vaJ) SoC) is quite different from the earlier nRF51822 based Bluefruit products (Bluefruit M0 Feather (http://adafru.it/t6a), etc.), both of which will continue to exist.

From a hardware perspective, the nRF52 Feather is based on a much more powerful ARM Cortex M4F processor, with 512KB flash, 64KB SRAM and hardware floating point acceleration ... whereas the earlier nRF51822 is based on the smaller ARM Cortex M0 core (fewer internal instructions), with 256KB flash and either 16KB or 32KB SRAM.

More importantly, the design approach that we took with the nRF52 is completely different:

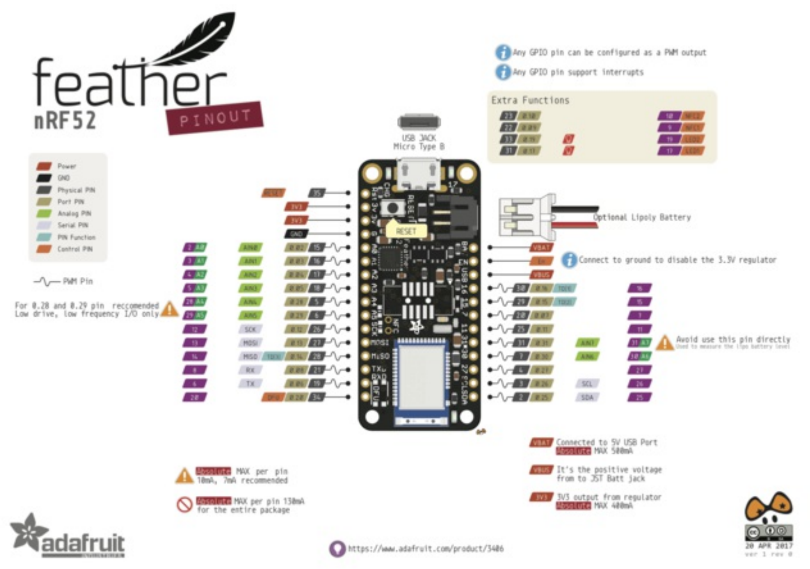
- nRF51 based Bluefruit boards run as modules that you connect to via an external MCU (typically an Atmel 32u4 or a SAMD21), sending AT style commands over SPI or UART.
- **With the nRF52, you run all of your code directly on the nRF52832 and no external MCU is used or required!**

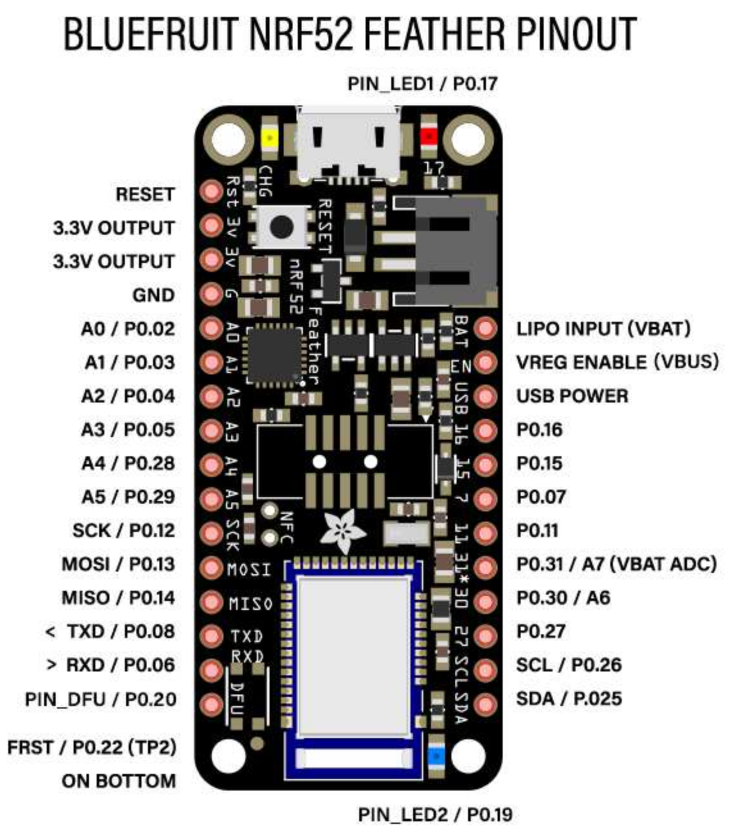
This change of design helps keep the overall costs lower, allows for far better performance since you aren't limited by the SPI or UART transport channel, and can help improve overall power consumption.

As a tradeoff, it also means a completely different API and development process, though!

nRF51 Bluefruit sketches won't run on nRF52 Bluefruit hardware without modification! The two device families have different APIs and programming models, and aim to solve your wireless problems in two different ways.

Device Pinout


feather
nRF52
PINOUT
Any GPIO pin can be configured as a PWM output
Any GPIO pin support interrupts
Extra Functions
USB JACK Micro Type B
RESET
Optional Lipoly Battery
Connect to ground to disable the 3.3V regulator
Avoid use this pin directly
PWM Pin
For 0.28 and 0.29 pin reccomended Low drive, low frequency I/O only
Connected to 5V USB Port
It's the positive voltage from to JST Batt Jack
3V3 output from regulator
https://www.adafruit.com/product/3406
adafruit
20 APR 2017



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 nR52 BSP, and further improvements are planned here.

I2C Pins

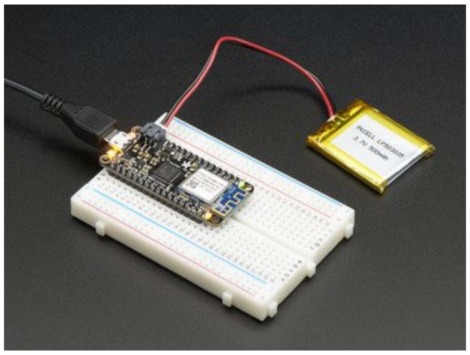
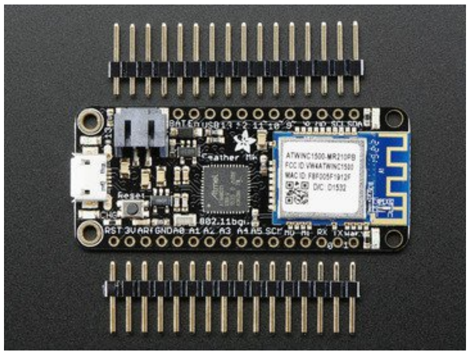
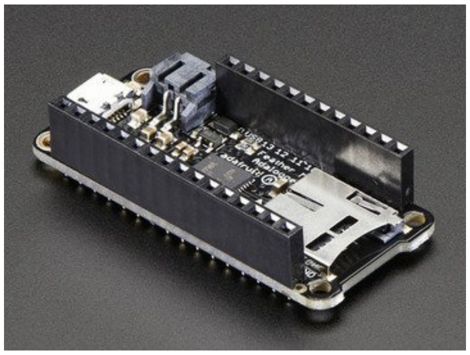
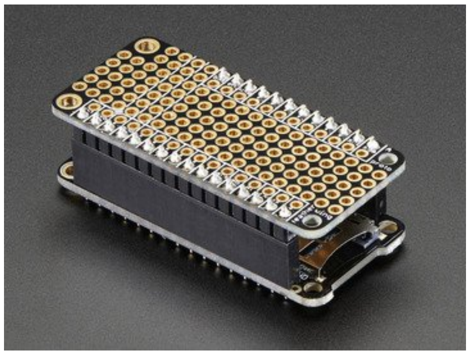
I2C pins on the nRF52832 require external pullup resistors to function, which are not present on the Adafruit nRF52 Feather by default. You will need to supply external pullups to use these. All Adafruit I2C breakouts have appropriate pullups on them already, so this normally won't be an issue for you.

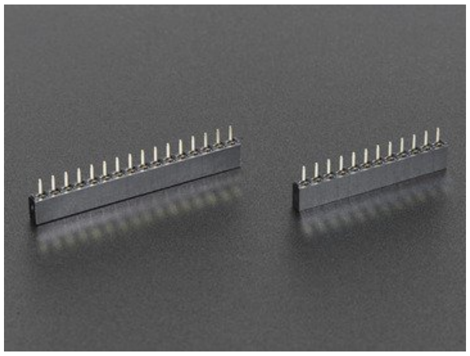
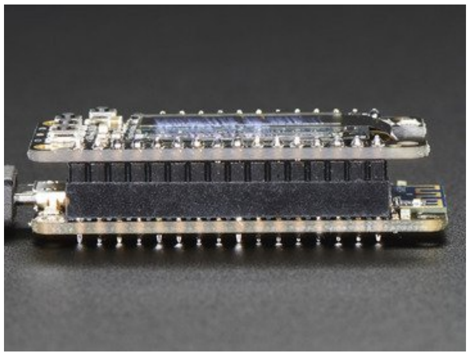
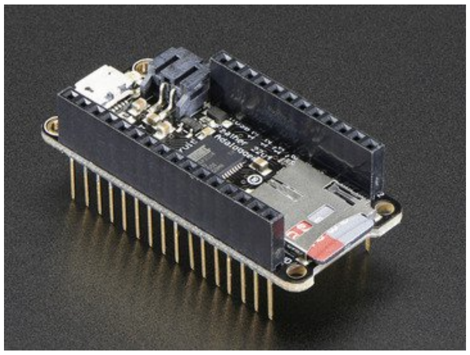
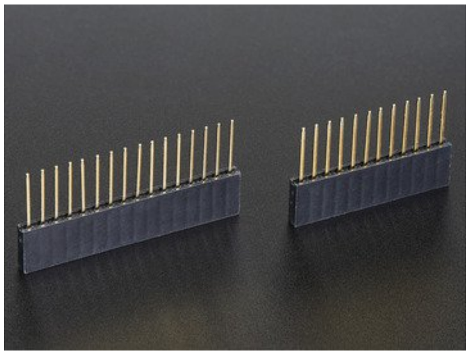
Assembly

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

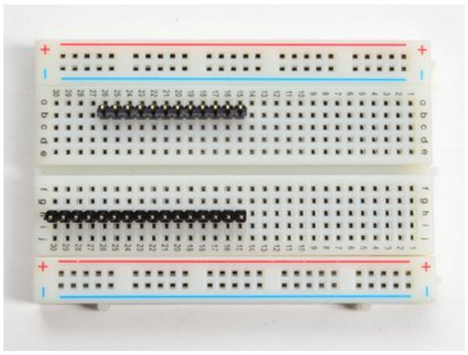
Header Options!

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

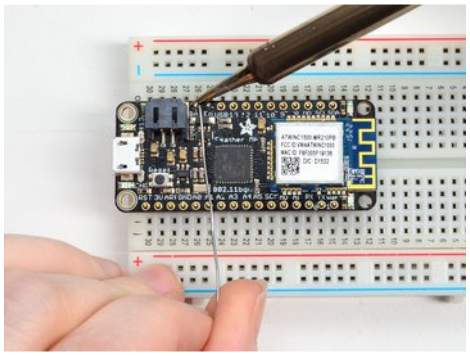
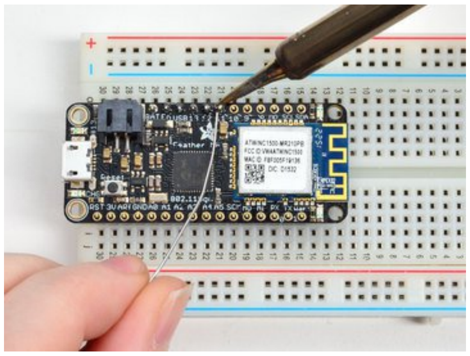
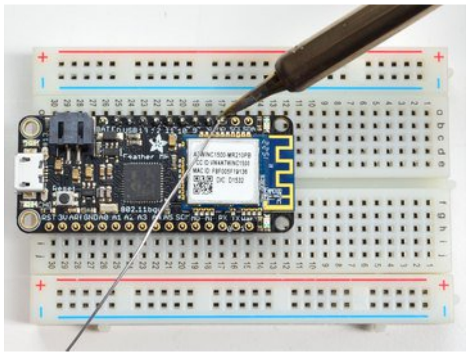
- 

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

- 
- 
- 

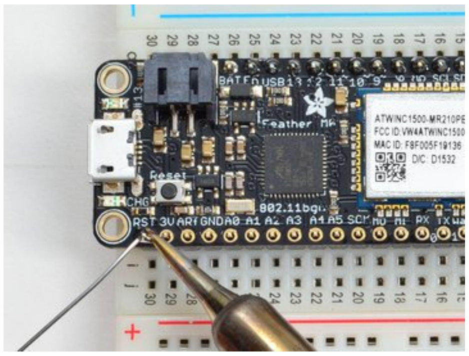
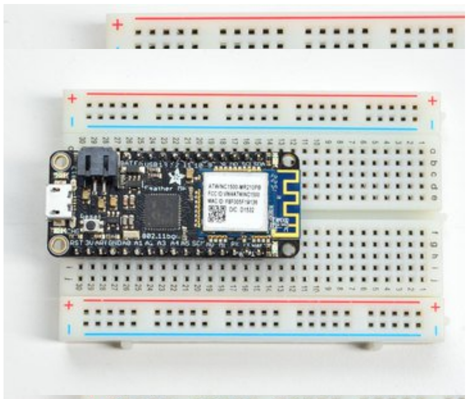
pads

And Solder!

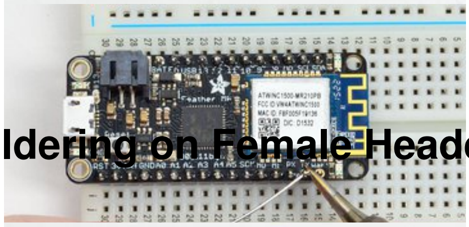
Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our[*Guide to Excellent Soldering*](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

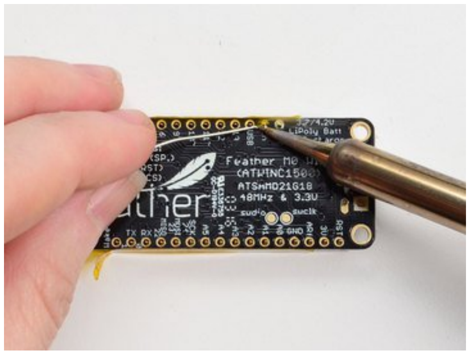
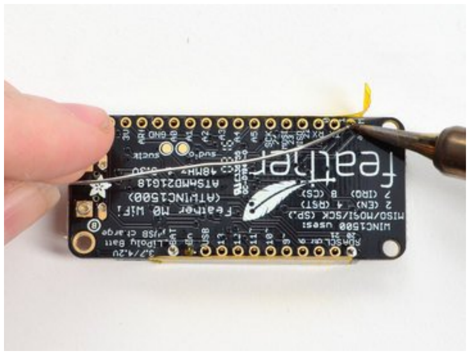
- 

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

- 
-
- 

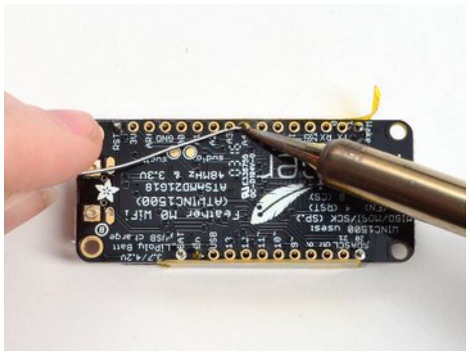
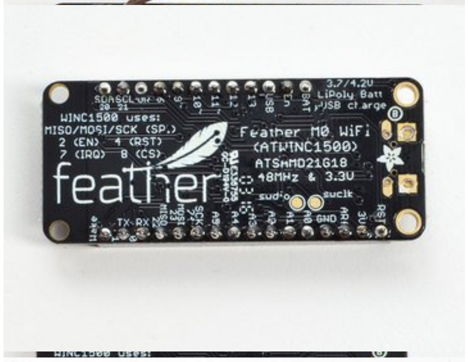
Flip & Tack Solder

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
-
-

Arduino BSP Setup

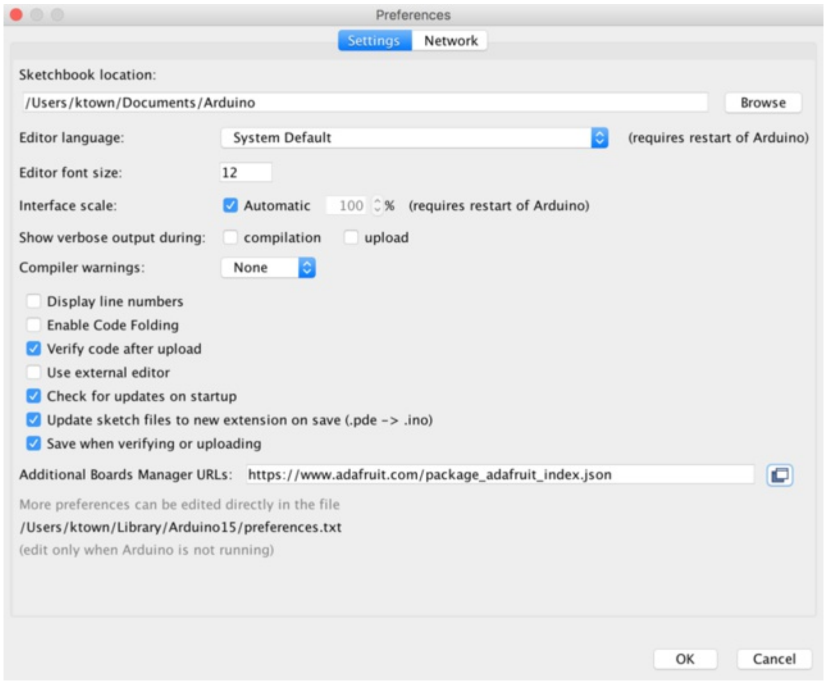
You can install the Adafruit Bluefruit nRF52 BSP in two steps:

nRF52 support requires at least Arduino IDE version 1.6.12! Please make sure you have an up to date version before proceeding with this guide! Please consult the FAQ section at the bottom of this page if you run into any problems installing or using this BSP!

1. BSP Installation

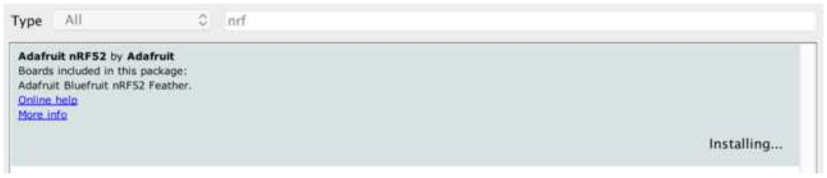
Recommended: Installing the BSP via the Board Manager

- Download and install the Arduino IDE (http://adafru.it/fvm) (At least **v1.6.12**)
- Start the Arduino IDE
- Go into Preferences
- Add https://www.adafruit.com/package_adafruit_index.json as an '**Additional Board Manager URL**' (see image below)



Sketchbook location: /Users/ktown/Documents/Arduino
Editor language: System Default (requires restart of Arduino)
Editor font size: 12
Interface scale: ✓ Automatic 100 % (requires restart of Arduino)
Show verbose output during: compilation upload
Compiler warnings: None
Display line numbers
Enable Code Folding
✓ Verify code after upload
Use external editor
✓ Check for updates on startup
✓ Update sketch files to new extension on save (.pde -> .ino)
✓ Save when verifying or uploading
Additional Boards Manager URLs: https://www.adafruit.com/package_adafruit_index.json
More preferences can be edited directly in the file
/Users/ktown/Library/Arduino15/preferences.txt
(edit only when Arduino is not running)

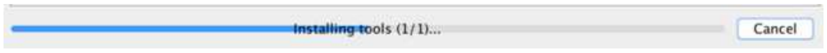
- Restart the Arduino IDE
- Open the **Boards Manager** option from the **Tools -> Board** menu and install '**Adafruit nRF52 by Adafruit**' (see image below)



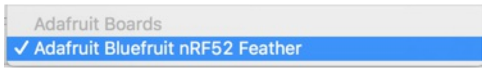
Adafruit nRF52 by **Adafruit**
Boards included in this package:
Adafruit Bluefruit nRF52 Feather.

It will take up to a few minutes to finish installing the cross-compiling toolchain and tools associated with this BSP.

The delay during the installation stage shown in the image below is normal, please be patient and let the installation terminate normally:



- Once the BSP is installed, select '**Adafruit Bluefruit nRF52 Feather**' from the **Tools -> Board** menu, which will update your system config to use the right compiler and settings for the nRF52:



2. Third Party Tool Installation

The following third party tools must also be installed on your system to allow you to work with the Bluefruit nRF52 Feather:

This step is only required on OS X and Linux. If you are using Windows, a pre-built 32-bit binary of nrfutil is already included in the BSP that should work out of the box for most setups.
You will need to have both Python and pip available on your system to use the tools below!

nrfutil (OS X and Linux Only)

This is a python wrapper (http://adafru.it/vaG) for Nordic's **nrfutil**, which is used to flash boards using the built in serial bootloader.

To install this tool, open a terminal or command prompty window and go into the folder where the BSP was installed in step one above.

Depending on your operating system. The BSP should be located in one of the following paths:

- **Windows**: %APPDATA%\Local\Arduino15\packages\adafruit\hardware\nrf52
- **OS X**: ~/Library/Arduino15/packages/adafruit/hardware/nrf52
- **Linux**: ~/.arduino15/packages/adafruit/hardware/nrf52

The path above will also include a sub-folder with the version number (ex. `0.5.0`). Be sure to enter that sub-folder as well, for example:

```
$ cd ~/Library/Arduino15/packages/adafruit/hardware/nrf52
$ ls
0.5.0/
$ cd 0.5.0
```

Next go into the tools/nrfutil-0.5.2 folder in the path above, and run the following commands to make nrfutil available to the Arduino IDE:

```
$ cd tools/nrfutil-0.5.2
$ sudo pip install -r requirements.txt
$ sudo python setup.py install
```

Don't install nrfutil from the pip package (ex. `sudo pip install nrfutil`). The latest nrfutil does not support DFU via Serial, and you should install the local copy of 0.5.2 included with the BSP via the `python setup.py install` command above.

3. Advanced Option: Manually Install the BSP via 'git'

If you wish to do any development against the core codebase (generate pull requests, etc.), you can also optionally install the Adafruit nRF52 BSP manually using 'git', as decribed below:

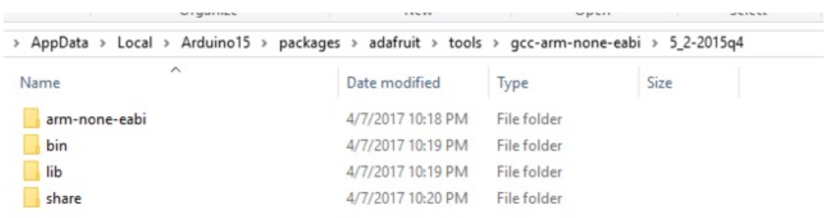
Adafruit nRF52 BSP via git (for core development and PRs only)

1. Go to the sketchbook folder on your command line, which should be one of the following:
   OS X: ~/Documents/Arduino
   Linux: ~/Arduino
   Windows: ~/Documents/Arduino
2. Create a folder named hardware/Adafruit, if it does not exist, and change directories into it.
3. Clone the Adafruit_nRF52_Arduino (http://adafru.it/vaF) repo in the folder described in step 2:
   ```
   git clone git@github.com:adafruit/Adafruit_nRF52_Arduino.git
   ```
4. This should result in a final folder name like '~/Documents/Arduino/hardware/Adafruit/Adafruit_nRF52_Arduino' (OS X).
5. Restart the Arduino IDE

You may also need to manually install the GCC ARM toolchain if it isn't already present:

1. Download the cross compiler: https://github.com/adafruit/Adafruit_nRF52_Arduino/releases/tag/gcc-5_2-2015q4 (http://adafru.it/vuB)
2. Extract the file to %APPDATA%\Local\Arduino15\packages\adafruit\tools\gcc-arm-none-eabi
3. Rename the extracted folder from gcc-arm-none-eabi-5_2-2015q4 to 5_2-2015q4

You should end up with the following folder structure for the cross-compiler:



AppData > Local > Arduino15 > packages > adafruit > tools > gcc-arm-none-eabi > 5_2-2015q4

Name	Date modified	Type	Size
arm-none-eabi	4/7/2017 10:18 PM	File folder	
bin	4/7/2017 10:19 PM	File folder	
lib	4/7/2017 10:19 PM	File folder	
share	4/7/2017 10:20 PM	File folder	

BSP FAQs

The following FAQs may be useful if you run into any problems:

Windows Related

If you are using BSP 0.6.0 or greater, there are no known issues with the BSP installation process on Windows. Please update to the latest version if you currently have an earlier release.

OS X Related

I can compile and link sketches on OS X, but nrfutil gives me the following error: 'AttributeError: 'int' object has no attribute 'value''?

Depending on your system setup and Python version, you may need to make a manual adjustment to a file in nrfutil, which is used when compiling and flashing files from the Arduino IDE.

Open the following file (please note that the BSP version number in the path may be different!): ~/Library/Arduino15/packages/adafruit/hardware/nrf52/0.5.1/tools/nrfutil-0.5.2/nordicsemi/dfu/init_packet.py ... and make the following changes:

```
@@ -79,7 +79,7 @@
-       for key in sorted(self.init_packet_fields.keys(), key=lambda x: x.value):
+        for key in sorted(self.init_packet_fields.keys(), key=lambda x: x):

@@ -94,7 +94,8 @@
-       for key in sorted(self.init_packet_fields.keys(), key=lambda x: x.value):
+        for key in sorted(self.init_packet_fields.keys(), key=lambda x: x):
```

Linux Related

On Linux I'm getting 'arm-none-eabi-g++: no such file or directory', even though 'arm-none-eabi-g++' exists in the path specified. What should I do?

This is probably caused by a conflict between 32-bit and 64-bit versions of the compiler, libc and the IDE. The compiler uses 32-bit binaries, so you also need to have a 32-bit version of libc installed on your system (details (http://adafru.it/vnE)). Try running the following commands from the command line to resolve this:

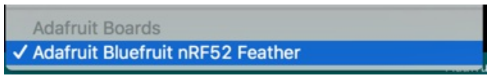
```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6:i386
```

Arduino Board Setup

Once you have the Bluefruit nRF52 BSP setup on your system, you need to select the appropriate board, which will determine the compiler and expose some new menus options:

1. Select the Board Target

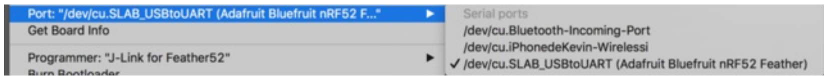
- Go to the **Tools** menu
- Select **Tools > Board > Adafruit Bluefruit nRF52 Feather**



2. Select the USB CDC Serial Port

Finally, you need to set the serial port used by Serial Monitor and the serial bootloader:

- Go to **Tools** > **Port** and select the appropriate SiLabs device


Port: "/dev/cu.SLAB_USBtoUART (Adafruit Bluefruit nRF52 F..."
Get Board Info
Programmer: "J-Link for Feather52"
Burn Bootloader
Serial ports
/dev/cu.Bluetooth-Incoming-Port
/dev/cu.iPhonedeKevin-Wirelessi
✓ /dev/cu.SLAB_USBtoUART (Adafruit Bluefruit nRF52 Feather)

If you don't see the SiLabs device listed, you may need to install theSiLabs CP2104 driver (http://adafru.it/vaH) on your system.

3. Run a Test Sketch

At this point, you should be able to run a test sketch from the**Examples** folder, or just flash the following blinky code from the Arduino IDE:

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

This will blink the pin #17 red LED on the Feather

Using the Bootloader

This page is for information purposes only. Normally the bootloader will work transparently and automatically from the Arduino IDE, requiring no manual intervention on your behalf.

The Bluefruit nRF52 Feather includes a customized version of the Nordic bootloader that enables serial support, over the air (OTA) DFU support, and various fail safe features like factory reset when the FRST pin is grounded at startup.

The bootloader that all Bluefruit nRF52 Feathers ships with allows you to flash user sketches to the nRF52832 using only the CP2104 USB to serial adapter populated on the board.

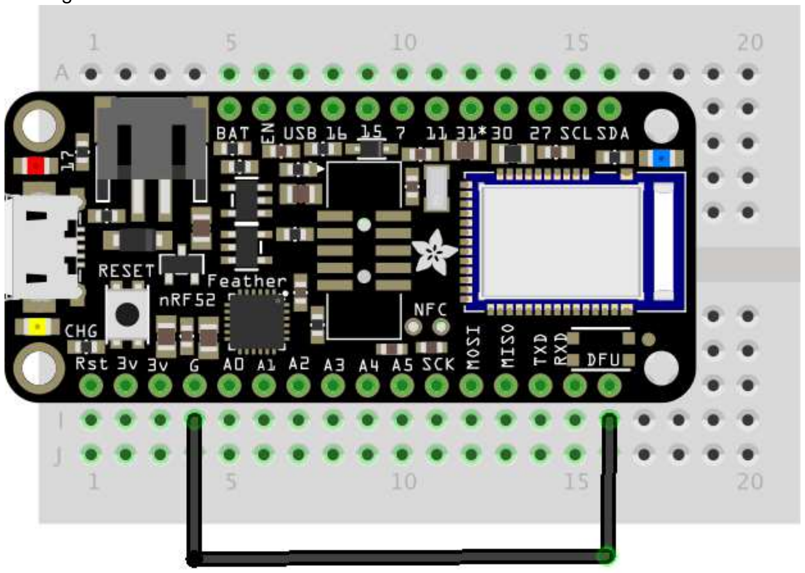
Forcing Serial Boot Mode

The Bluefruit nRF52 Feather is designed to briefly enter serial bootloader mode for a short delay every time the device comes out of reset, and the DTR line on the CP2104 USB to Serial adapter will trigger a reset every time the Serial Monitor is opened. This means that you can normally flash a user sketch to the nRF52 with no manual intervention on your part at a HW level.

If you need to force the serial bootloader mode, however, you can connect the**DFU** pin to **GND** at startup, which will force you to enter serial bootloader mode and stay in that mode until you reset or power cycle the board.

This can be used to recover bricked boards where a bad user sketch has been uploaded, since you will enter serial bootloader mode without executing the user sketch, and you can flash a new sketch directly from the Arduino IDE.

Forcing the serial bootloader can often be used to recover bricked devices.



Factory Reset

The Bluefruit nRF52 Feather has an optional FRST pad on the bottom of the PCB.

If you brick your device, you can solder a wire to the**FRST** pad, connecting it to **GND**. When a GND state is detected at power up the following actions will be performed:

- The user application flash section will be erased
- The user 'App Data' section that stores non volatile config data will be erased

This will cause the device to enter serial bootloader mode at startup, and the user sketch or config data that caused the device to stop responding should be removed.

Be sure to disconnect the pin from GND after a successful factory reset!

Advanced: OTA DFU Bootloader

While this is only recommended for advanced users, you can also force OTA (Over The Air) DFU bootloader mode to enable OTA updates using BLE and Nordic's proprietary update protocol (which is support by both Nordic mobile apps, and out own Bluefruit LE Connect).

To force OTA DFU mode, set both FRST and DFU to GND at startup Power cycling the board will cause the device to boot up into OTA DFU mode.

This option is not actively support nor recommended by Adafruit, and we are still working on making this as safe as possible for users via our Bluefruit LE Connect application. Use OTA DFU at your own risk knowing you can brick your device and may need a Segger J-Link or similar device to regain control of it!

Flashing the Bootloader

All Adafruit nRF52 boards chip with the bootloader pre-flashed. This page is provided for information purposes only!

All Bluefruit nRF52 Feather boards and Bluefruit nRF52 modules ship with the serial bootloader pre-flashed, so this page is normally not required when setting your device and system up.

The information provided here is only intended for for rare cases where you may want or need to reflash the bootloader yourself, and have access to the HW required to do so.

You will need a Segger J-Link to flash the bootloader to the nRF52832 SoC!

Third Party Tool Requirements

To burn the bootloader from within the Arduino IDE, you will need the following tools installed on your system and available in the system path:

JLink Drivers and Tools

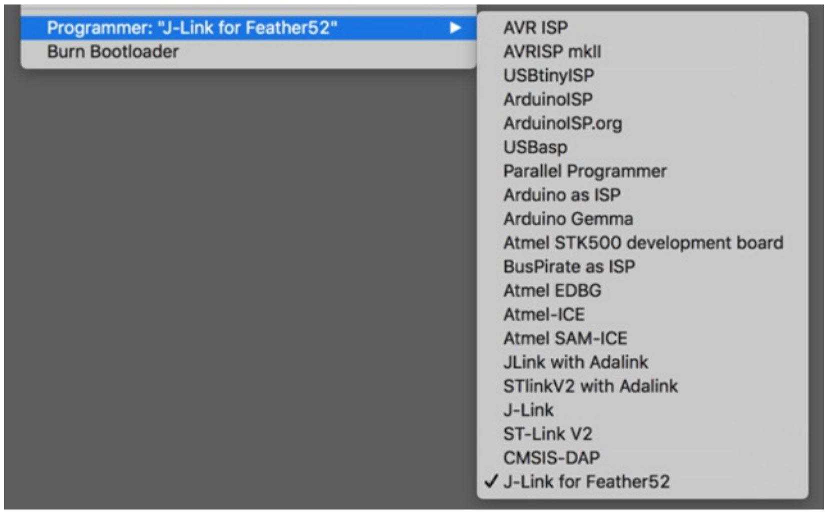
Download and install the JLink Software and Documentation Pack (http://adafru.it/val) from Segger, which will also install a set of command line tools.

Burning the Bootloader from the Arduino IDE

Once the tools above have been installed and added to your system path, from the Arduino IDE:

- Select `**Tools > Board > Adafruit Bluefruit Feather52**`
- Select `**Tools > Programmer > J-Link for Feather52**`
- Select `**Tools > Burn Bootloader**` with the board and J-Link connected

The appropriate **Programmer** target and **Burn Bootloader** button can be seen below:



Manually Burning the Bootloader via nrfjprog

You can also manually burn the bootloader from the command line, using `nrfjprog` from Nordic.

You can either download nRF5x-Command-Line-Tools (http://adafru.it/vaJ) for OSX/Linux/Win32, or use the version that ships with the BSP in the **tools/nrf5x-command-line-tools** folder.

Run the folllwing commands, updating the path to the .hex file as appropriate:

```
$ nrfjprog -e -f nrf52
$ nrfjprog --program bootloader_with_s132.hex -f nrf52
$ nrfjprog --reset -f nrf52
```

You should see something similar to the following output, followed by a fast blinky on the status LED to indicate that you are in DFU/bootloader mode since no user sketch was found after the device reset:

All commands below were run from 'tools/nrf5x-command-line-tools/osx/nrfjprog'

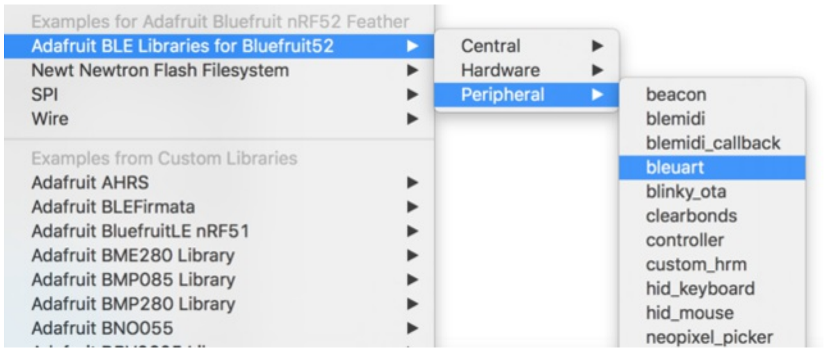
```
$ ./nrfjprog -e -f nrf52
Erasing code and UICR flash areas.
Applying system reset.

$ ./nrfjprog --program  ../../../../bin/bootloader/bootloader_v050_s132_v201.hex -f nrf52
Parsing hex file.
Reading flash area to program to guarantee it is erased.
Checking that the area to write is not protected.
Programing device.

$ ./nrfjprog --reset -f nrf52
Applying system reset.
Run.
```

OS X Note: You may need to create a symlink in `/usr/local/bin` to the
`nrfjprog` tool wherever you have added it. You can run the following command, for example:

```
$ ln -s $HOME/prog/nordic/nrfjprog/nrfjprog /usr/local/bin/nrfjprog
```

Examples

There are numerous examples available for the Bluefruit nRF52 Feather in the**Examples** menu of the nRF52 BSP, and these are always up to date. You're first stop looking for example code should be there:



Example Source Code

The latest example source code is always available and visible on Github, and the public git repository should be considered the definitive source of example code for this board.

Click here to browse the example source code on Github
http://adafru.it/vaK

Documented Examples

To help explain some common use cases for the nRF52 BLE API, feel free to consult the example documentation in this section of the learning guide:

- **Advertising: Beacon** - Shows how to use the BLEBeacon helper class to configure your Bleufruit nRF52 Feather as a beacon
- **BLE UART: Controller** - Shows how to use the**Controller** utility in our Bluefruit LE Connect apps to send basic data between your peripheral and your phone or tablet.
- **Custom: HRM** - Shows how to defined and work with a custom GATT Service and Characteristic, using the officially adopted Heart Rate Monitor (HRM) service as an example.