



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

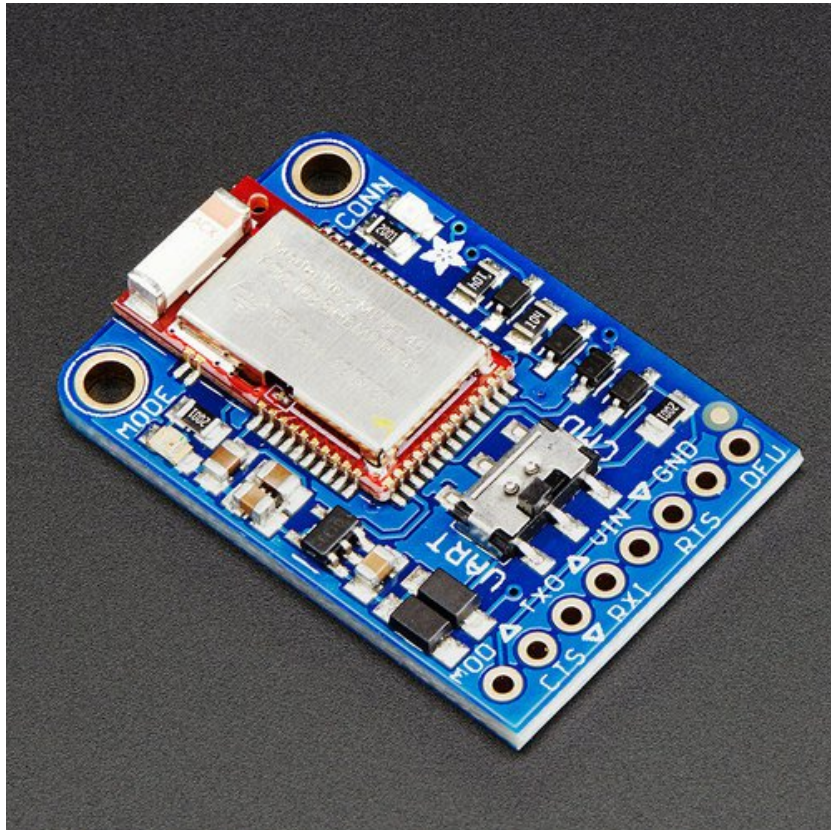
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



□

Introducing the Adafruit Bluefruit LE UART Friend

Created by Kevin Townsend



Last updated on 2017-04-24 08:30:30 PM UTC

Guide Contents

Guide Contents	2
Introduction	9
So it's a Fancy Pants Wireless UART Adapter?	9
Download our free Android/iOS app and you're ready to rock!	10
You can do a lot more too!	10
Why Use Adafruit's Module?	10
Technical Specifications	10
Pinouts	12
Power Pins	12
UART Pins	12
Other Pins	12
Reverse Side Breakouts	13
Assembly	14
Prepare the header strip:	14
Add the breakout board:	14
And Solder!	15
Wiring	16
Wiring for Arduino Uno	16
Wiring for an Arduino Micro or Leonardo	16
Wiring for an FTDI Cable	16
Factory Reset	18
Factory Reset via DFU Pin	18
FactoryReset Sample Sketch	18
AT+FACTORYRESET	19
Factory Reset via FCTR Test Pad	19
DFU Updates	21
Adafruit Bluefruit LE Connect	21
Current Measurements	22
Test Conditions	22
Fast Advertising Mode	22
Slow Advertising Mode	22
Connected Mode (UART)	23

Software	24
Configuration!	25
Which board do you have?	25
Bluefruit Micro or Feather 32u4 Bluefruit	25
Feather M0 Bluefruit LE	25
Bluefruit LE SPI Friend	26
Bluefruit LE UART Friend or Flora BLE	26
Configure the Pins Used	26
Common settings:	27
Software UART	27
Hardware UART	27
Mode Pin	27
SPI Pins	27
Software SPI Pins	27
Select the Serial Bus	27
UART Based Boards (Bluefruit LE UART Friend & Flora BLE)	27
SPI Based Boards (Bluefruit LE SPI Friend)	28
ATCommand	29
Opening the Sketch	29
Configuration	29
Running the Sketch	30
BLEUart	32
Opening the Sketch	32
Configuration	32
Running the Sketch	33
HIDKeyboard	37
Opening the Sketch	37
Configuration	37
Running the Sketch	38
Bonding the HID Keyboard	38
Android	39
iOS	40
OS X	41
Controller	43

Opening the Sketch	43
Configuration	43
Running the Sketch	44
Using Bluefruit LE Connect in Controller Mode	44
Streaming Sensor Data	45
Control Pad Module	46
Color Picker Module	47
HeartRateMonitor	49
Opening the Sketch	49
Configuration	49
If Using Hardware or Software UART	50
Running the Sketch	50
nRF Toolbox HRM Example	51
CoreBluetooth HRM Example	52
UriBeacon	54
Opening the Sketch	54
Configuration	54
Running the Sketch	55
HALP!	56
Data Mode	57
Switching Command/Data Mode via +++	58
Command Mode	60
Hayes/AT Commands	60
Test Command Mode '=?'	60
Write Command Mode '=xxx'	60
Execute Mode	61
Read Command Mode '?'	61
Dynamically Switching Modes via +++	61
Standard AT	62
AT	62
ATI	62
ATZ	62

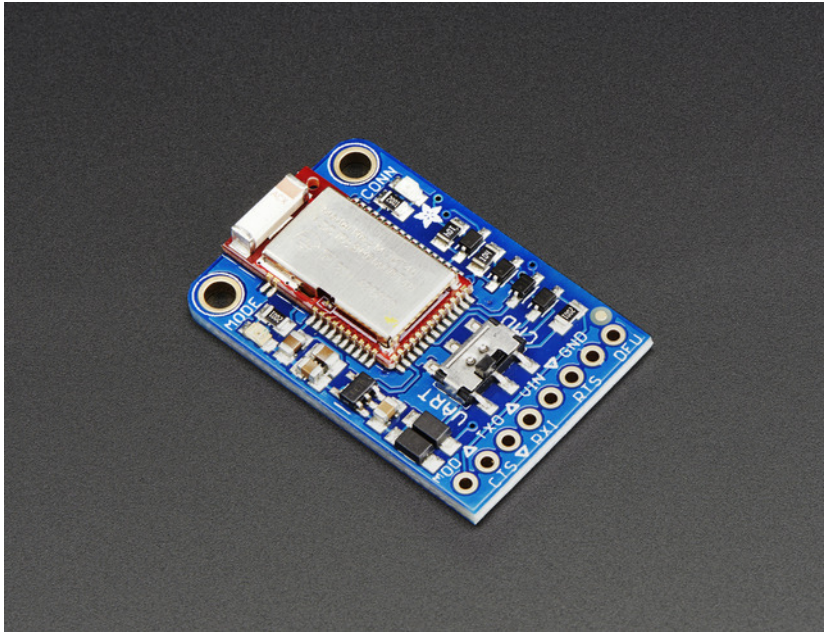
ATE	62
+++	63
General Purpose	64
AT+FACTORYRESET	64
AT+DFU	64
AT+HELP	64
AT+NVMWRITE	64
AT+NVMREAD	65
AT+MODESWITCHEN	65
Hardware	66
AT+BAUDRATE	66
AT+HWADC	66
AT+HWGETDIETEMP	66
AT+HWGPIO	67
AT+HWGPIOMODE	67
AT+HWI2CSCAN	68
AT+HWVBAT	68
AT+HWRANDOM	68
AT+HWMODELED	68
AT+UARTFLOW	69
Beacon	70
AT+BLEBEACON	70
AT+BLEURIBEACON	71
Deprecated: AT+EDDYSTONEENABLE	72
AT+EDDYSTONEURL	72
AT+EDDYSTONECONFIGEN	72
AT+EDDYSTONESEVICEEN	73
AT+EDDYSTONEBROADCAST	73
BLE Generic	74
AT+BLEPOWERLEVEL	74
AT+BLEGETADDRTYPE	74
AT+BLEGETADDR	74

AT+BLEGETPEERADDR	75
AT+BLEGETRSSI	75
BLE Services	76
AT+BLEUARTTX	76
TX FIFO Buffer Handling	76
AT+BLEUARTTXF	77
AT+BLEUARTRX	77
AT+BLEUARTFIFO	78
AT+BLEKEYBOARDEN	78
AT+BLEKEYBOARD	78
AT+BLEKEYBOARDCODE	79
Modifier Values	79
AT+BLEHIDEN	79
AT+BLEHIDMOUSEMOVE	80
AT+BLEHIDMOUSEBUTTON	80
AT+BLEHIDCONTROLKEY	81
AT+BLEHIDGAMEPADEN	81
AT+BLEHIDGAMEPAD	82
AT+BLEMIDIEN	82
AT+BLEMIDIRX	82
AT+BLEMIDITX	83
AT+BLEBATTEN	83
AT+BLEBATTVAL	83
BLE GAP	84
AT+GAPCONNECTABLE	84
AT+GAPGETCONN	84
AT+GAPDISCONNECT	84
AT+GAPDEVNAME	84
AT+GAPDELBONDS	85
AT+GAPINTERVALS	85
AT+GAPSTARTADV	86
AT+GAPSTOPADV	86
AT+GAPSETADVDATA	86

BLE GATT	88
GATT Limitations	88
AT+GATTCLEAR	88
AT+GATTADDSERVICE	88
AT+GATTADDCHAR	89
AT+GATTCHAR	90
AT+GATTLIST	91
AT+GATTCHARRAW	92
Debug	93
AT+DBGMEMRD	93
AT+DBGNVMRD	93
AT+DBGSTACKSIZE	93
AT+DBGSTACKDUMP	93
History	96
Version 0.7.7	96
Version 0.7.0	96
Version 0.6.7	97
Version 0.6.6	97
Version 0.6.5	98
Version 0.6.2	98
Version 0.5.0	98
Version 0.4.7	99
Version 0.3.0	99
GATT Service Details	100
UART Service	100
UART Service	101
Characteristics	101
TX (0x0002)	101
RX (0x0003)	101
Software Resources	102
Bluefruit LE Client Apps and Libraries	102
Bluefruit LE Connect (http://adafru.it/f4G) (Android/Java)	102
Bluefruit LE Connect (http://adafru.it/f4H) (iOS/Swift)	102

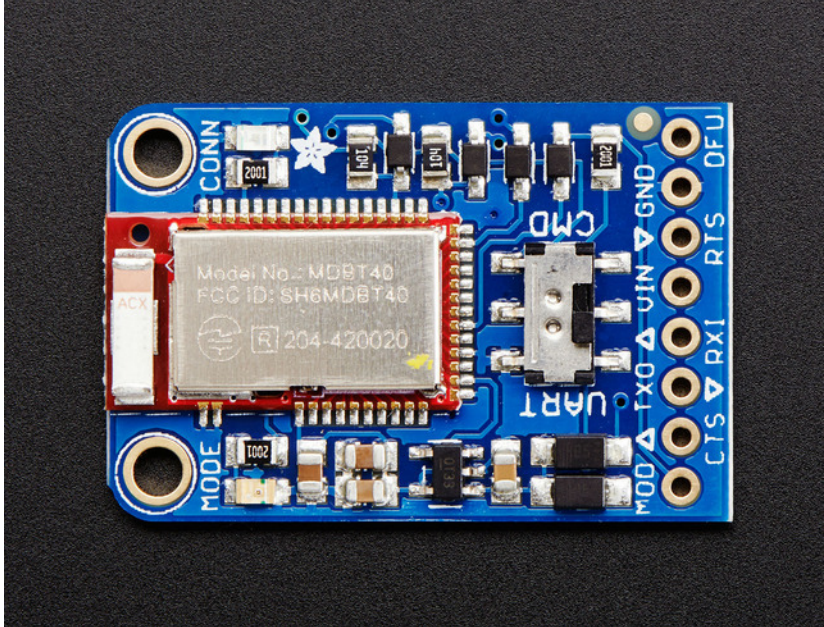
Bluefruit LE Connect for OS X (http://adafru.it/o9F) (Swift)	102
Bluefruit LE Command Line Updater for OS X (http://adafru.it/pLF) (Swift)	103
Deprecated: Bluefruit Buddy (http://adafru.it/mCn) (OS X)	103
ABLE (http://adafru.it/ijB) (Cross Platform/Node+Electron)	104
Bluefruit LE Python Wrapper (http://adafru.it/fQF)	104
Debug Tools	105
AdaLink (http://adafru.it/fPq) (Python)	105
Adafruit nRF51822 Flasher (http://adafru.it/fVL) (Python)	105
BLE FAQ	106
Bluefruit LE Connect (Android)	107
Nordic nRF Toolbox	107
Adafruit_nRF51822_Flasher	107
Device Recovery	112
How to Recover a Bluefruit Board	112
1. Force DFU Mode at Startup	112
2. Update the Bluefruit Firmware	112
BLEFRIEND32 Firmware (UART, 32KB SRAM)	112
BLESPIFRIEND Firmware (SPI)	112
3. Flash a Test Sketch	112
4. Perform a Factory Reset	112
Still Having Problems?	113
Downloads	114
Files	114
Schematic	114
Board Layout	114

Introduction



Would you like to add powerful and easy-to-use Bluetooth Low Energy to your robot, art or other electronics project? Heck yeah! With BLE now included in modern smart phones and tablets, its fun to add wireless connectivity. So what you really need is the new Adafruit Bluefruit LE UART Friend!

The Bluefruit LE UART Friend makes it easy to add Bluetooth Low Energy connectivity to anything with a hardware or software serial port. We even have nice hardware flow control so you won't have to think about losing data. Connect to your Arduino or other microcontroller or even just a standard FTDI cable for debugging and testing.



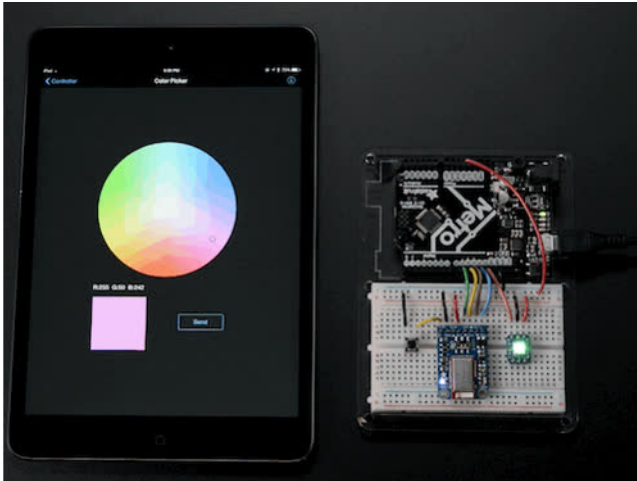
This multi-function module can do quite a lot! For most people, they'll be very happy to use the standard Nordic UART RX/TX connection profile. In this profile, the Bluefruit acts as a data pipe, that can 'transparently' transmit back and forth from your iOS or Android device. You can use our [iOS App](http://adafru.it/iCi)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

So it's a Fancy Pants Wireless UART Adapter?

The board is capable of much more than just sending strings over the air! Thanks to an easy to learn [AT command set](http://adafruit.it/iCG) (<http://adafruit.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://adafruit.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!

Download our free Android/iOS app and you're ready to rock!

Using our Bluefruit [iOS App](http://adafruit.it/iCi) (<http://adafruit.it/iCi>) or [Android App](http://adafruit.it/f4G) (<http://adafruit.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://adafruit.it/iCI) (<http://adafruit.it/iCI>), [quaternion/accelerometer/gyro/magnetometer or location \(GPS\)](#) (<http://adafruit.it/iCI>), and an 8-button [control game pad](http://adafruit.it/iCI) (<http://adafruit.it/iCI>).



You can do a lot more too!

- [The Bluefruit can also act like an HID Keyboard](http://adafruit.it/iCJ) (<http://adafruit.it/iCJ>) (for devices that support BLE HID)
- [Can become a BLE Heart Rate Monitor](http://adafruit.it/iCK) (<http://adafruit.it/iCK>) (a standard profile for BLE) - you just need to add the pulse-detection circuitry
- [Turn it into a UriBeacon](http://adafruit.it/iCL) (<http://adafruit.it/iCL>), 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://adafruit.it/iCM) (<http://adafruit.it/iCM>) Use any Android or iOS device to get updates and install them!

Why Use Adafruit's Module?

There are plenty of BLE modules out there, with varying quality on the HW design as well as the firmware.

One of the biggest advantages of the Adafruit Bluefruit LE family is that **we wrote all of the firmware running on the devices ourselves from scratch.**

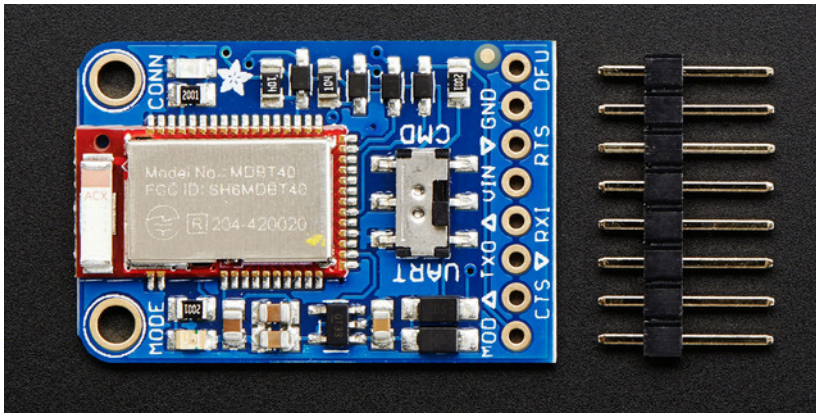
We control every line of code that runs on our modules ... and so we aren't at the mercy of any third party vendors who may or may not be interested in keeping their code up to date or catering to our customer's needs.

Because we control everything about the product, we add features that are important to our customers, can solve any issues that do come up without begging any 3rd parties, and we can even change Bluetooth SoCs entirely if the need ever arises!

Technical Specifications

- ARM Cortex M0 core running at 16MHz
- 256KB flash memory
- 32KB SRAM
- Peak current draw
- Transport: UART @ 9600 baud with HW flow control **(CTS+RTS required!)**
- 5V-safe inputs (Arduino Uno friendly, etc.)
- On-board 3.3V voltage regulation
- Bootloader with support for safe OTA firmware updates
- Easy AT command set to get up and running quickly

Pinouts



HW flow control (CTS+RTS) should always be used with the nRF51822. The UART peripheral block is designed in a way that more or less requires HW flow control for reliable UART communication, with a very small internal buffer and tight timing limitations due to the SoftDevice architecture (Nordic's proprietary Bluetooth Low Energy stack).

Power Pins

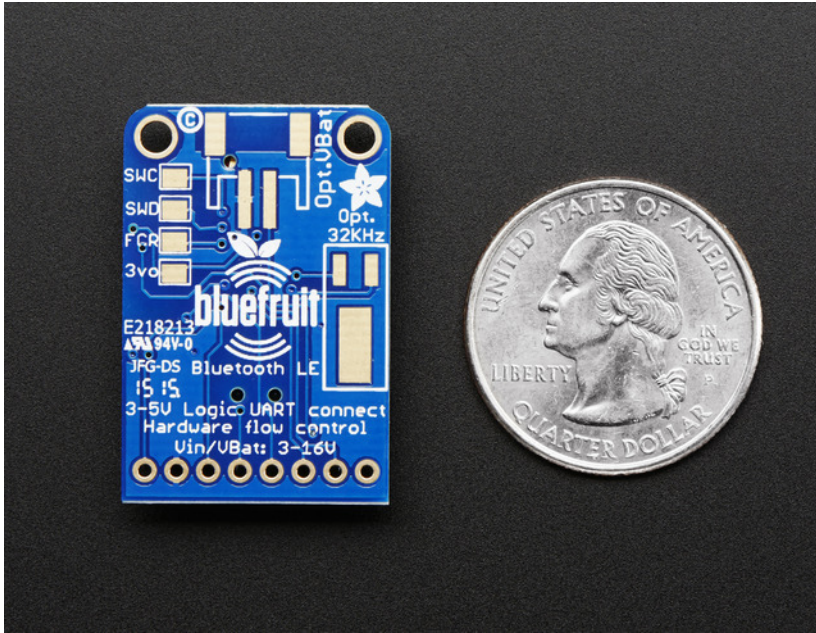
- **VIN**: This is the power supply for the module, supply with 3.3-16V power supply input. This will be regulated down to 3.3V to run the chip
- **GND**: The common/GND pin for power and logic

UART Pins

- **TXO** - This is the UART Transmit pin *out* of the breakout (Bluefruit LE --> MCU), it's at 3.3V logic level.
- **RXI** - This is the UART Receive pin *into* the breakout (MCU --> Bluefruit LE). This has a logic level shifter on it, you can use 3-5V logic.
- **CTS** - Clear to Send hardware flow control pin *into* the breakout (MCU --> Bluefruit LE). Use this pin to tell the Bluefruit that it can send data back to the microcontroller over the TXO pin. **This pin is pulled high by default and must be set to ground in order to enable data transfer out! If you do not need hardware flow control, tie this pin to ground** it is a level shifted pin, you can use 3-5V logic
- **RTS** - Read to Send flow control pin *out* of the module (Bluefruit LE --> MCU). This pin will be low when its fine to send data to the Bluefruit. In general, at 9600 baud we haven't seen a need for this pin, but you can watch it for full flow control! This pin is 3.3V out

Other Pins

- **MOD**: Mode Selection. The Bluefruit has two modes, Command and Data. You can keep this pin disconnected, and use the slide switch to select the mode. Or, you can control the mode by setting this pin voltage, it will override the switch setting! High = Command Mode, Low = UART/DATA mode. This pin is level shifted, you can use 3-5V logic
- **DFU**: Setting this pin low when you power the device up will force the Bluefruit LE module to enter a special firmware update mode to update the firmware over the air. Once the device is powered up, this pin can *also* be used to perform a factory reset. Wire the pin to GND for >5s until the two LEDs start to blink, then release the pin (set it to 5V or logic high) and a factory reset will be performed.



Reverse Side Breakouts

On the back we also have a few breakouts!

Opt VBat: If you fancy, [you can solder on a JST 2-PH connector \(http://adafru.it/1769\)](http://adafru.it/1769), this will let you easily plug in a Lithium Ion or other battery pack. This connector pad is diode protected so you can use both Vin and VBat and the regulator will automatically switch to the higher voltage

Opt. 32 KHz: If you're doing some funky low power work, [we wanted to give you the option of solderin in a 32khz oscillator. \(http://adafru.it/f4U\)](http://adafru.it/f4U) Our firmware doesn't support it yet but its there!

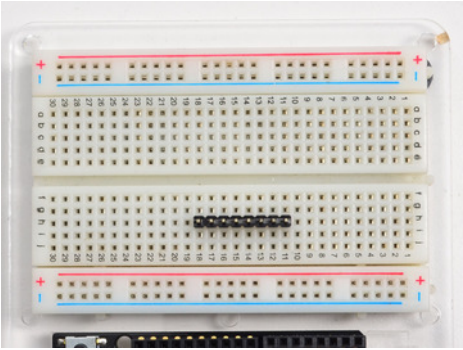
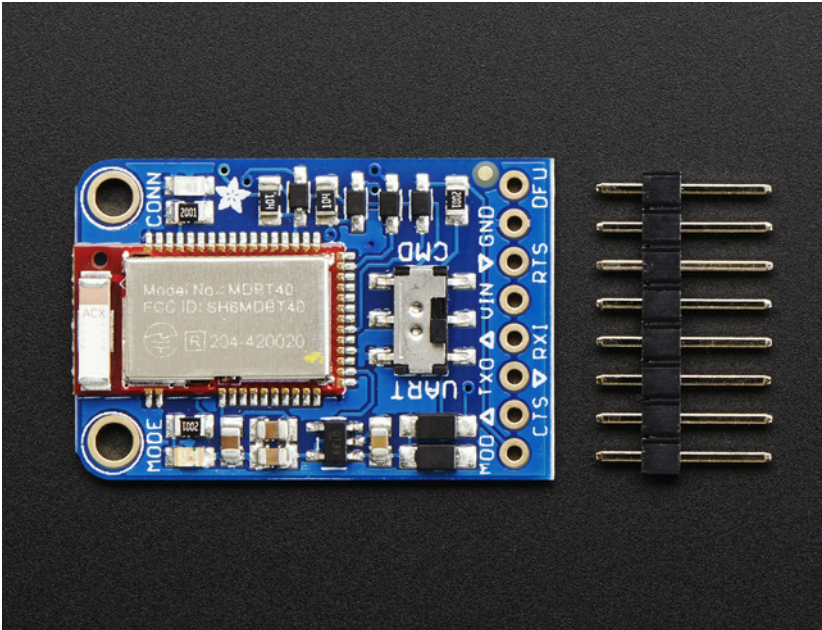
SWC: This is the SWD clock pin, 3v logic - for advanced hackers!

SWD: This is the SWD data pin, 3v logic - for advanced hackers!

3Vo: This is the output from the 3V regulator, for testing and also if you really need regulated 3V, up to 250mA available

FCR: This is the factory reset pin. When all else fails and you did something to really weird out your module, tie this pad to ground while powering up the module and it will factory reset. You should try the DFU reset method first tho (see that tutorial page)

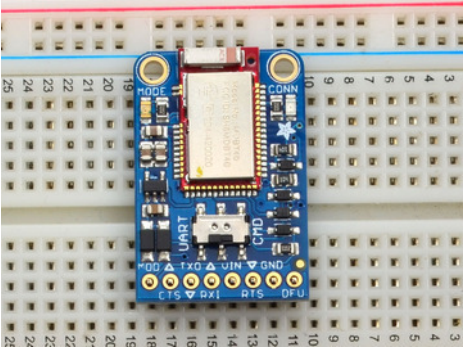
Assembly



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**

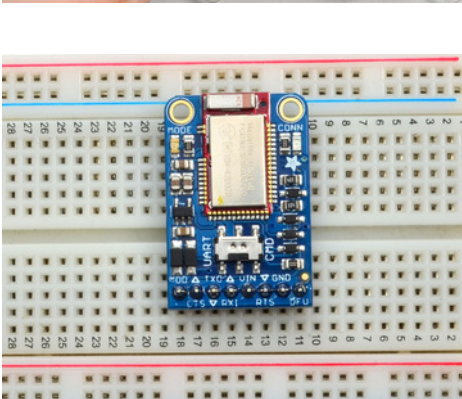
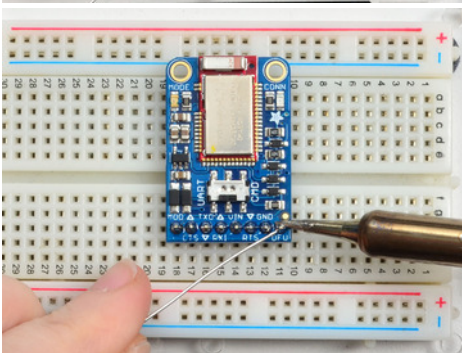
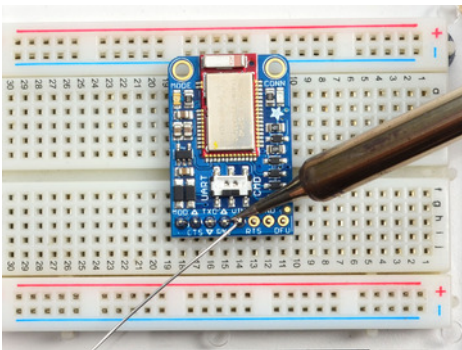
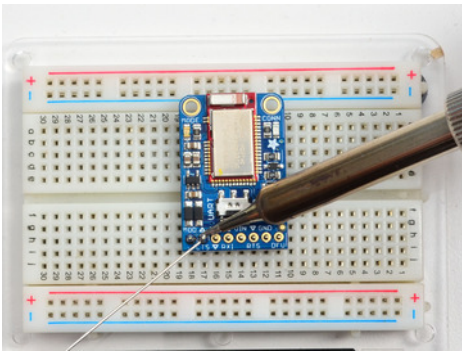
-



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.

Solder the longer power/data strip first

(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

Wiring

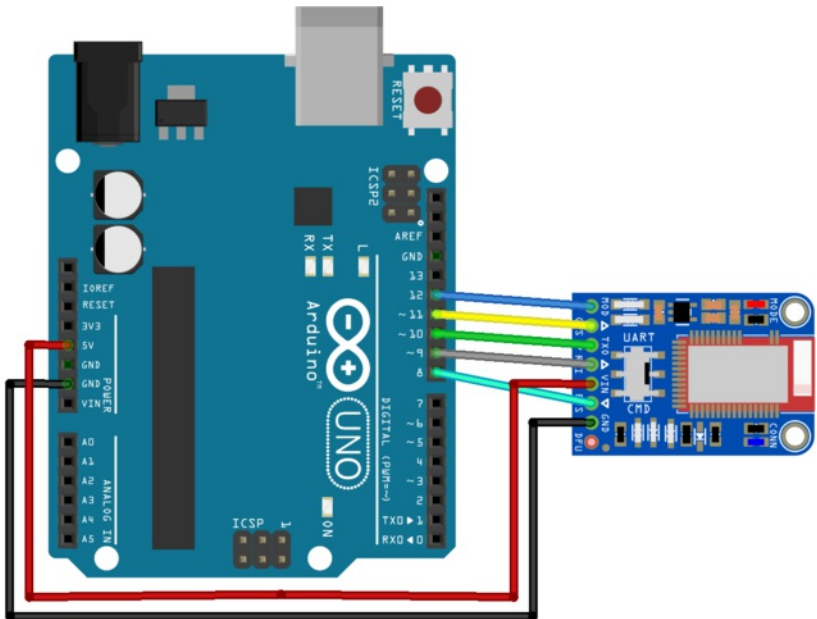
You can use the Bluefruit LE UART friend with any microcontroller with 3 or 5V logic, but we'll be demoing it with an Arduino UNO. Depending on whether your microcontroller has a hardware or software UART, adjust pins as necessary!

Wiring for Arduino Uno

To connect the Bluefruit LE UART Friend to your Arduino Uno using the default pinout in our sample sketches, connect the pins up as follows:

- **MOD** to **Pin 12**
- **CTS** to **Pin 11**
- **TXO** to **Pin 10**
- **RXI** to **Pin 9**
- **VIN** to **5V**
- **RTS** to **Pin 8**
- **GND** to **GND**

The wiring diagram below shows how this might look on your system:



All of these pins are 'flexible' and you can change them around as necessary after you get your setup running nicely, but we recommend starting out with our default wiring.

Wiring for an Arduino Micro or Leonardo

To connect the Bluefruit LE UART Friend to an Arduino Micro or Leonardo, use the following pinout and make sure you've selected HW UART as the constructor in your sample sketches (they default to SW Serial):

- **1/TX** to **RXI**
- **0/RX** to **TXO**
- **+5V** on the Micro/Leonardo to **VIN**
- **GND** to **GND**
- Connect **CTS** on the Bluefruit LE Module to **GND**

The HW Serial constructor can be seen below, and should be uncommented in the sample sketches:

```
/* ...or hardware serial, which does not need the RTS/CTS pins. Uncomment this line */
Adafruit_BluefruitLE_UART ble(BLUEFRUIT_HWSERIAL_NAME, BLUEFRUIT_UART_MODE_PIN);
```

Wiring for an FTDI Cable

Since the UART Friend is, well, a serial port, you can use an FTDI Friend or cable to quickly connect using any serial console. You won't get MODE or DFU connections, so don't forget to flick the mode switch as necessary if you want to be in a particular mode.

Simply insert an FTDI cable directly into the Bluefruit LE UART Friend header by using the six pins in the middle, being careful to align the power pins up correctly (the red wire to VIN and the black wire to GND):



Factory Reset

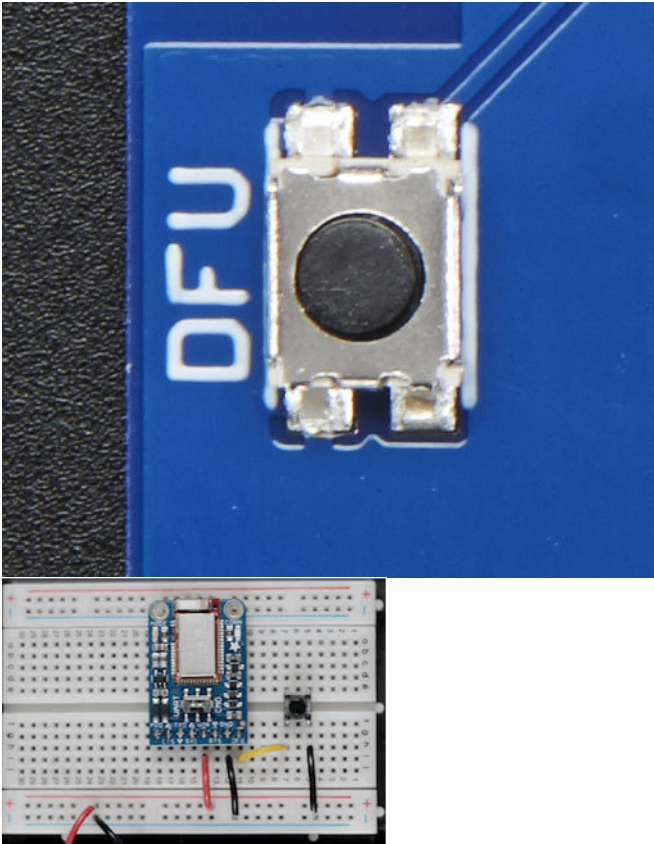
There are several methods that you can use to perform a factory reset on your Bluefruit LE module if something gets misconfigured, or to delete persistent changes like UriBeacon or advertising payload changes, etc.

These methods are the same for both UART and SPI versions of Bluefruit LE

Factory Reset via DFU Pin

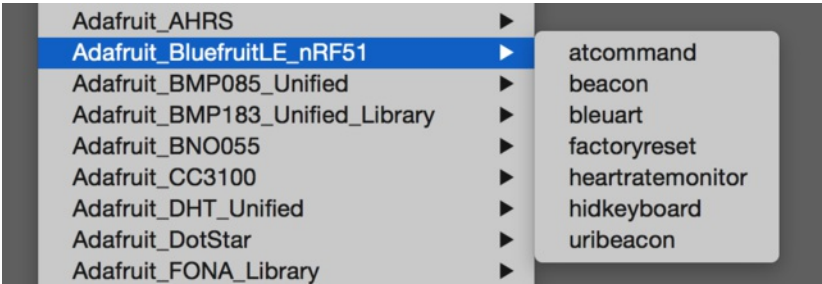
If you hold the DFU pin low (set the pin to GND) for **>5 seconds**, the red and blue LEDs next to the module will start blinking and the device will perform a factory reset as soon as you release the DFU pin (disconnecting it from GND).

If you have a DFU button instead of a pin, just hold the button down.

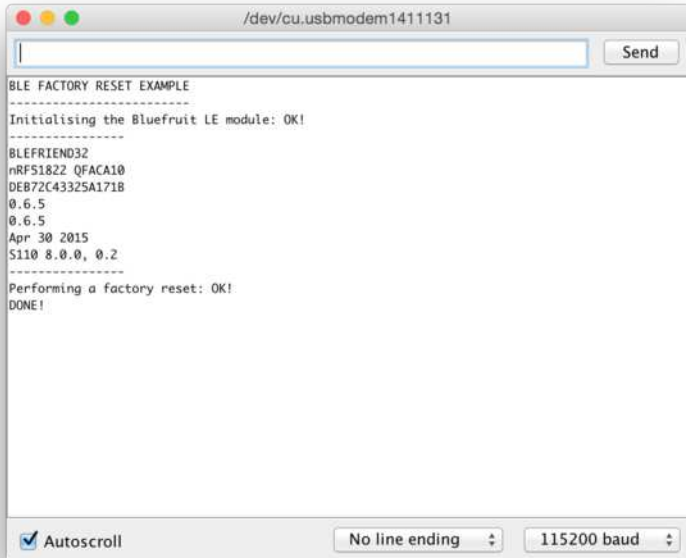


FactoryReset Sample Sketch

There is a FactoryReset sample sketch in the Adafruit Bluefruit LE library, which can be access in the **File > Examples > Adafruit_BluefruitLE_nRF51** folder ([See the Software section of this tutorial \(http://adafru.it/iCj\)](http://adafru.it/iCj) for instructions on installing the library):



Upload this sketch and open the Serial Monitor and it should perform a factory reset for you:



AT+FACTORYRESET

You can also perform a factory reset by sending the **AT+FACTORYRESET** command to your Bluefruit LE module in your favorite terminal emulator or using the [ATCommand](http://adafru.it/iCk) (<http://adafru.it/iCk>) example sketch.

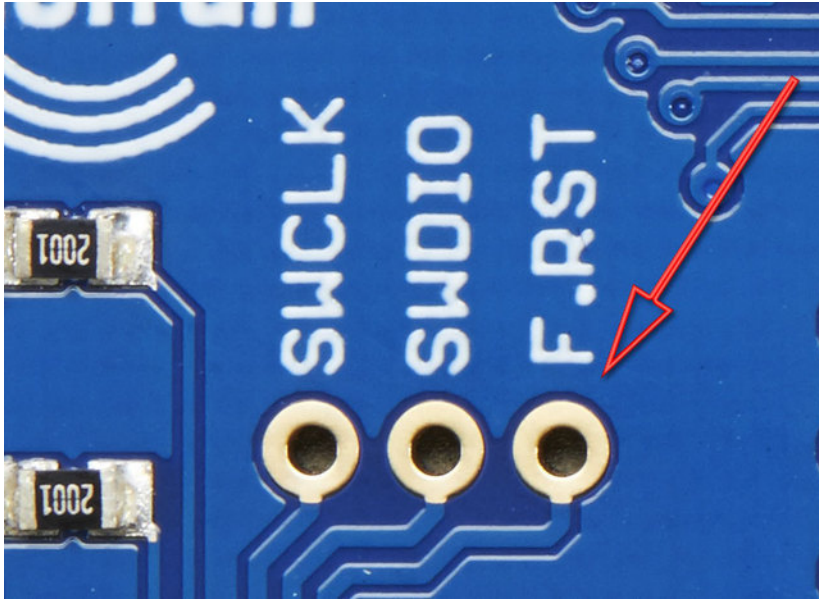
```
AT+FACTORYRESET
OK
```

This command will also cause the device to reset.

Factory Reset via FCTR Test Pad

On the bottom of the Bluefruit LE Friend board or shields there is a test pad or pin that exposes the Factory Reset pin on the modules (marked **FCR** or **F.RST**). Setting this pad low when the device is powered up will cause a factory reset at startup.





DFU Updates

We're constantly working on the Bluefruit LE firmware to add new features, and keep up to date with what customers need and want.

To make sure you stay up to date with those changes, we've included an easy to use over the air updater on all of our nRF51 based Bluefruit LE modules.

Adafruit Bluefruit LE Connect

Updating your Bluefruit LE device to the latest firmware is as easy as installing [Adafruit's Bluefruit LE Connect application](http://adafru.it/f4G) (<http://adafru.it/f4G>) (Android) from the Google Play Store or [Bluefruit LE Connect for iOS](http://adafru.it/f4H) (<http://adafru.it/f4H>) from the Apple App Store.

Any time a firmware update is available, the application will propose to download the latest binaries and take care of all the details of transferring them to your Bluefruit device, as shown in the video below:

Current Measurements

The following tables give you an idea of the average current draw in three different operating modes with the Bluefruit LE UART Friend:

- Fast Advertising Mode (the first 30 seconds after power up)
- Slow Advertising Mode (>30s since power up)
- Connected Mode (using the UART profile in this case)

Test Conditions

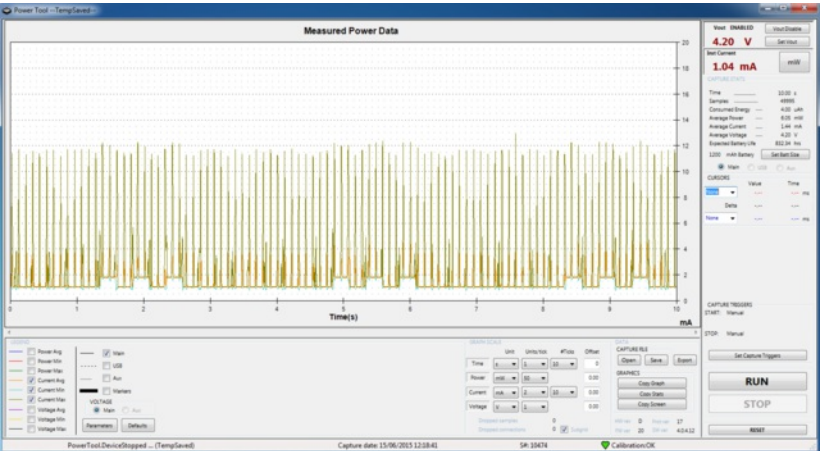
The board was powered from a fully charged **1200mAh 4.2V LIPO** (<http://adafru.it/dyW>) cell running at 4.2V. Power efficiency will generally increase as the LIPO battery voltage drops, so 4.2V should be considered a worst case scenario.

- **Power Supply:** 1200mAh 4.2V LIPO Cell (at 4.2V)
- **Firmware:** 0.6.2

Fast Advertising Mode

The first 30 seconds that the Bluefruit LE UART Friend is powered up it will enter 'Fast Advertising Mode', which sends an advertising packet once every 100ms.

- **Average Current:** 1.44mA
- **Peak Current:** 13.5mA
- **Expected Battery Life:** 832 hours (~34.6 days)

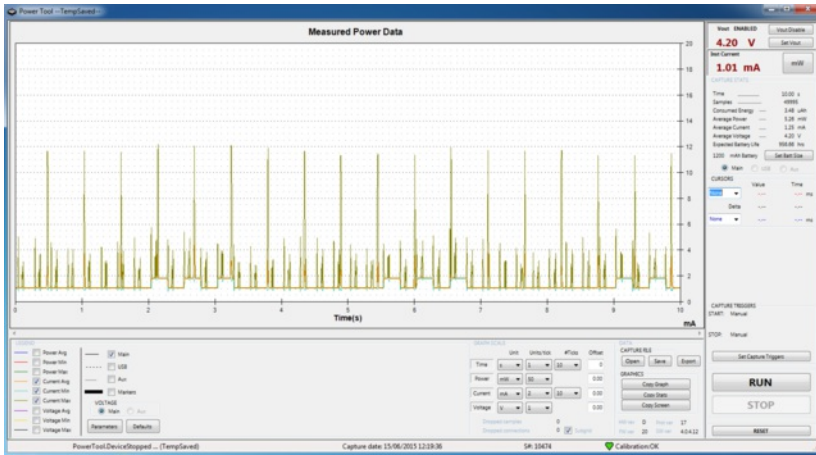


The MODE LEDs can be seen blinking in regular intervals as the three rectangular bumps at the bottom of the chart.

Slow Advertising Mode

After 30 seconds the Bluefruit LE UART Friend will enter 'Slow Advertising Mode', which sends an advertising packet once every 546.25ms.

- **Average Current:** 1.25mA
- **Peak Current:** 13.5mA
- **Expected Battery Life:** 956 hours (~40 days)



Connected Mode (UART)

The following measurements illustrate the average current draw in connected mode (with the connected LED enabled). UART mode was used as a test case.

- **Average Current:** 1.86mA
- **Peak Current:** 15.2mA
- **Expected Battery Life:** 645 hours (~26.8 days)



Software

In order to try out our demos, you'll need to download the Adafruit BLE library for the nRF51 based modules such as this one (a.k.a. Adafruit_BluefruitLE_nRF51)

You can check out the code here at github (<http://adafru.it/f4V>) but its likely easier to just download by clicking:

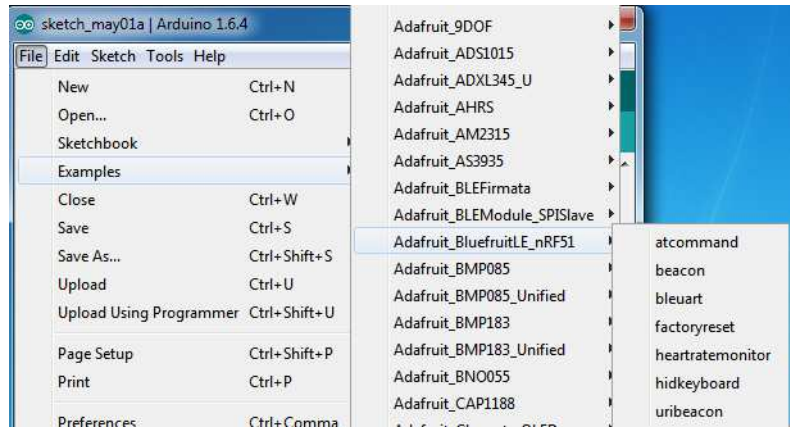
[Download Adafruit_BluefruitLE_nRF51](http://adafru.it/f4W)
<http://adafru.it/f4W>

Rename the uncompressed folder **Adafruit_BluefruitLE_nRF51** and check that the **Adafruit_BluefruitLE_nRF51** folder contains **Adafruit_BLE.cpp** and **Adafruit_BLE.h** (as well as a bunch of other files)

Place the **Adafruit_BluefruitLE_nRF51** library folder your **arduinofolder/libraries/** folder. You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

We also have a great tutorial on Arduino library installation at: <http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use> (<http://adafru.it/aYM>)

After restarting, check that you see the library folder with examples:

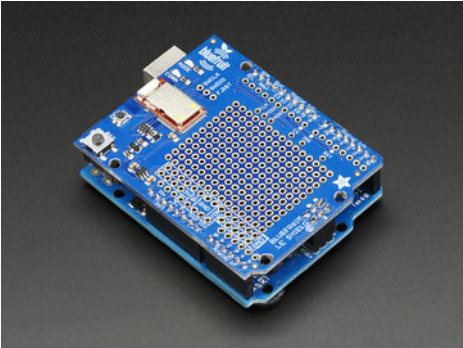


Configuration!

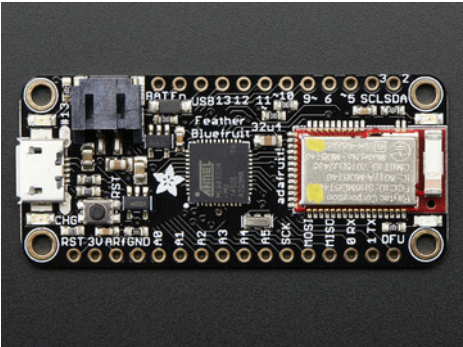
Before you start uploading any of the example sketches, you'll need to CONFIGURE the Bluefruit interface - there's a lot of options so pay close attention!

Which board do you have?

There's a few products under the Bluefruit name:

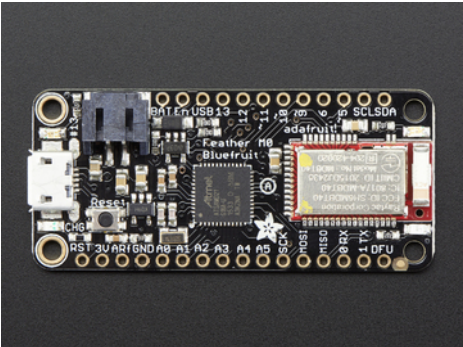


If you are using the Bluefruit LE Shield then you have an **SPI-connected NRF51822** module. You can use this with **Atmega328** (Arduino UNO or compatible), **ATmega32u4** (Arduino Leonardo, compatible) or **ATSAMD21** (Arduino Zero, compatible) and possibly others.
Your pinouts are **Hardware SPI, CS = 8, IRQ = 7, RST = 4**



Bluefruit Micro or Feather 32u4 Bluefruit

If you have a Bluefruit Micro or Feather 32u4 Bluefruit LE then you have an **ATmega32u4** chip with **Hardware SPI, CS = 8, IRQ = 7, RST = 4**



Feather M0 Bluefruit LE

If you have a Feather M0 Bluefruit LE then you have an **ATSAMD21** chip with **Hardware SPI, CS = 8, IRQ = 7, RST = 4**