



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

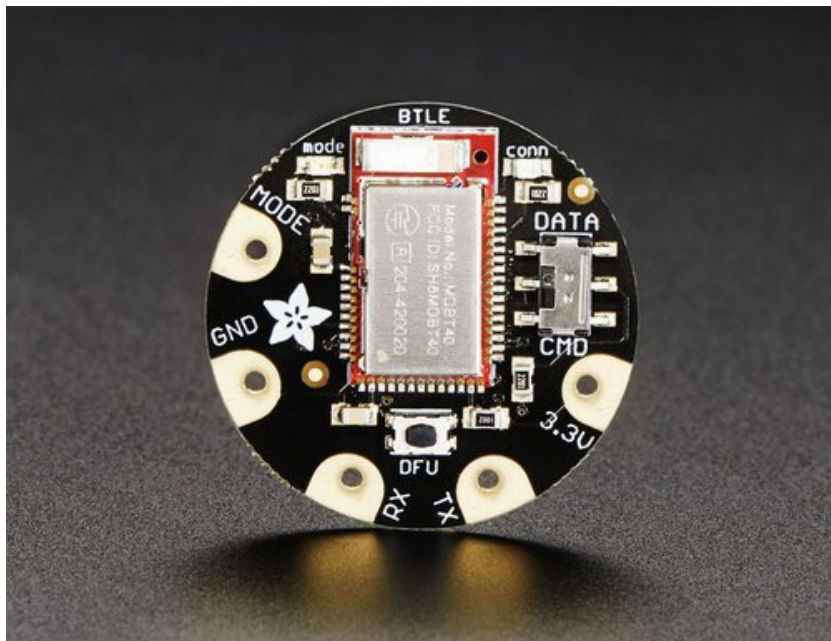
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



□

## Adafruit Flora Bluefruit LE

Created by lady ada



Last updated on 2016-09-28 08:14:24 PM UTC

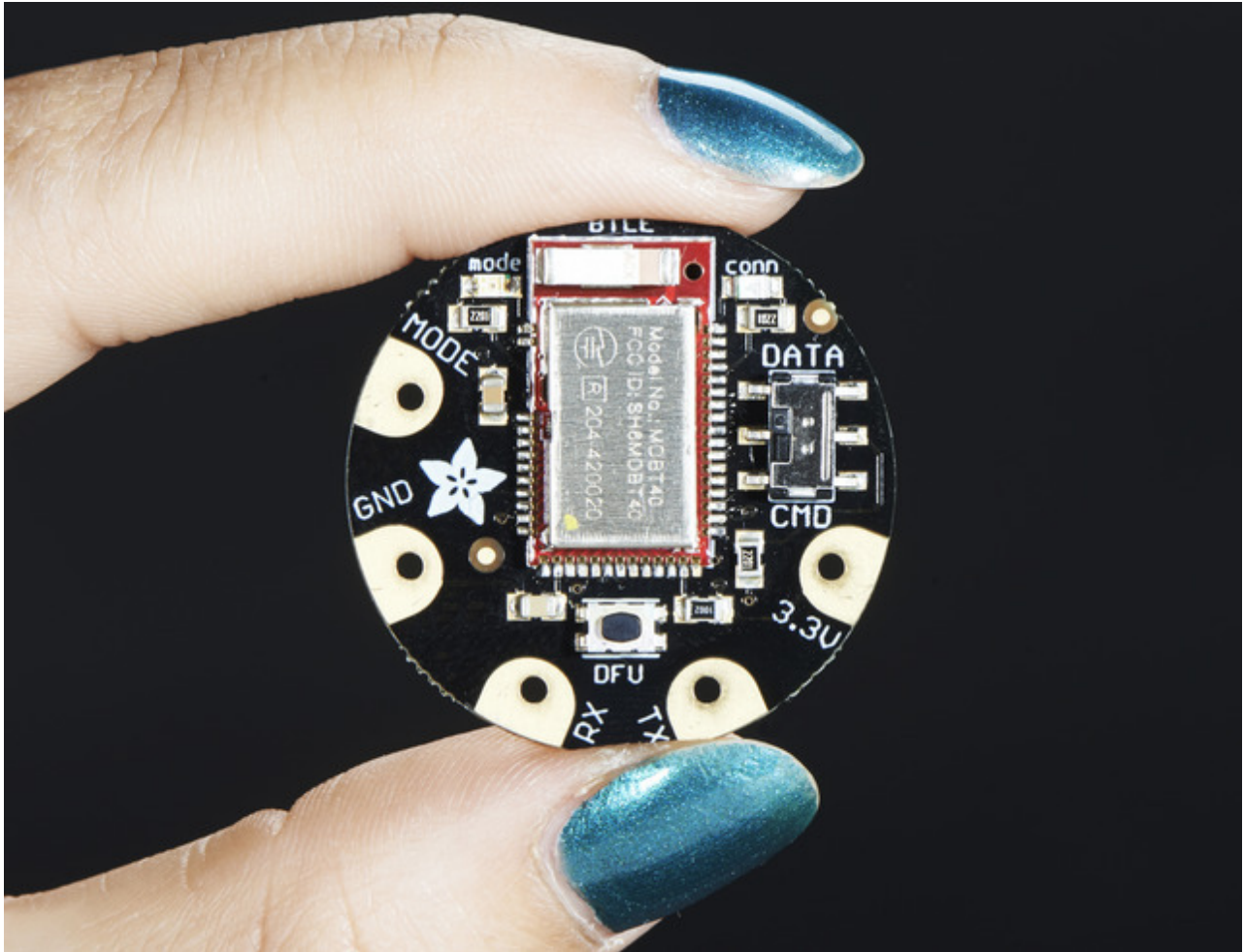
# Guide Contents

Guide Contents	2
Overview	5
Get started fast with the Bluefruit App	6
You can do a lot more too!	7
Pinouts	9
Power Pads	9
Data Pads	9
Other Pads	10
Reverse Side	11
Sewing	12
Factory Reset	13
Factory Reset via DFU Button	13
FactoryReset Sample Sketch	13
AT+FACTORYRESET	14
Factory Reset via F.RST Test Pad	15
Firmware Updates	16
Adafruit Bluefruit LE Connect	16
Installing Software	17
Example Code	18
Configuring for Flora	18
ATCommand	19
Opening the Sketch	19
Configuration	20
Running the Sketch	21
BLEUart	24
Opening the Sketch	24
Configuration	25
Running the Sketch	26
Controller	30
Opening the Sketch	30
Configuration	31

Running the Sketch	32
Using Bluefruit LE Connect in Controller Mode	32
Streaming Sensor Data	34
Control Pad Module	35
Color Picker Module	36
HIDKeyboard	39
Opening the Sketch	39
Configuration	40
Running the Sketch	41
Bonding the HID Keyboard	42
Android	42
iOS	44
OS X	46
UriBeacon	49
Opening the Sketch	49
Configuration	50
Running the Sketch	51
HeartRateMonitor	53
Opening the Sketch	53
Configuration	54
If Using Hardware or Software UART	55
Running the Sketch	55
nRF Toolbox HRM Example	57
CoreBluetooth HRM Example	59
HALP!	60
Data Mode	62
Switching Command/Data Mode via +++	63
Command Mode	65
Hayes/AT Commands	65
UART Service	66
Downloads	67
Schematic	67

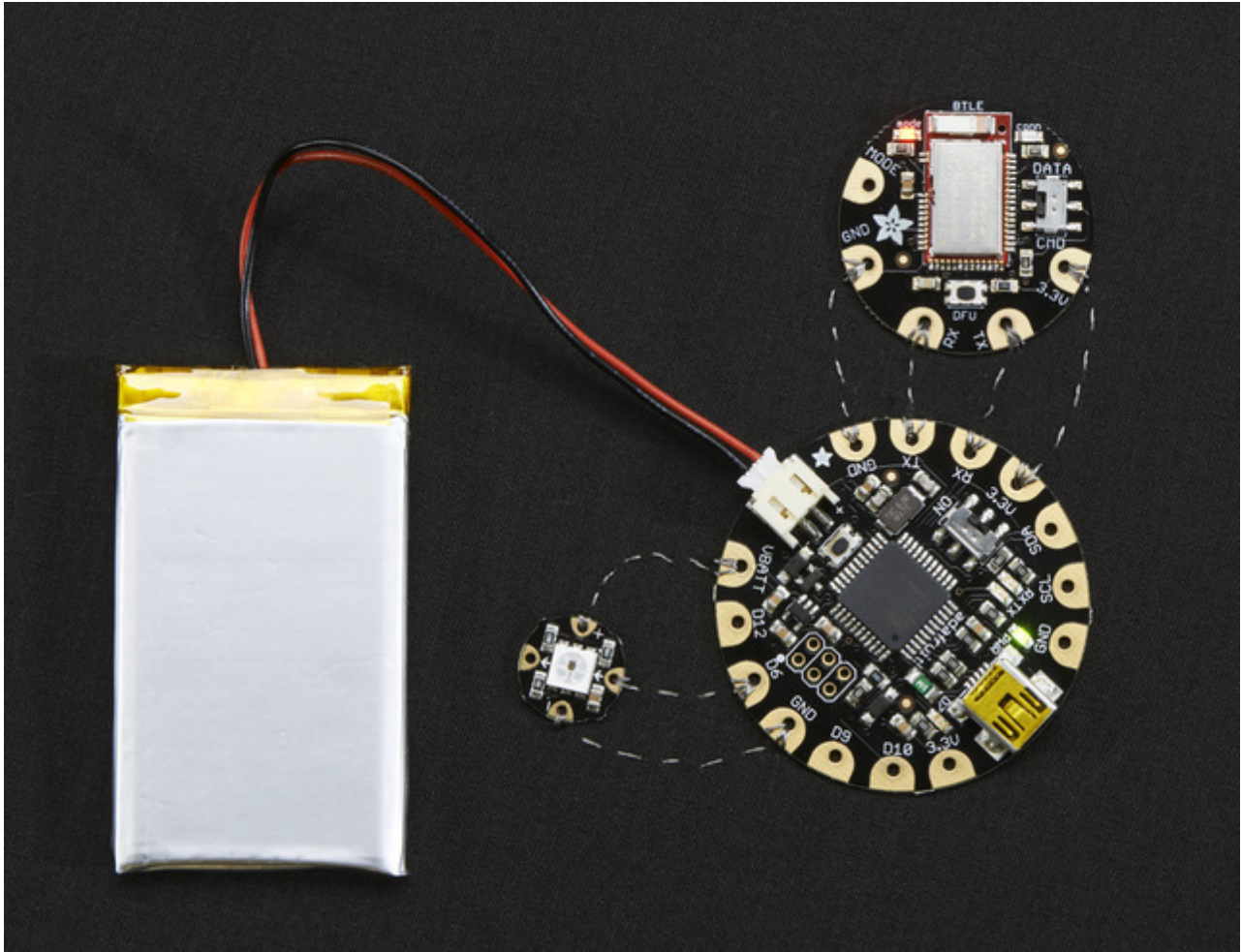


# Overview



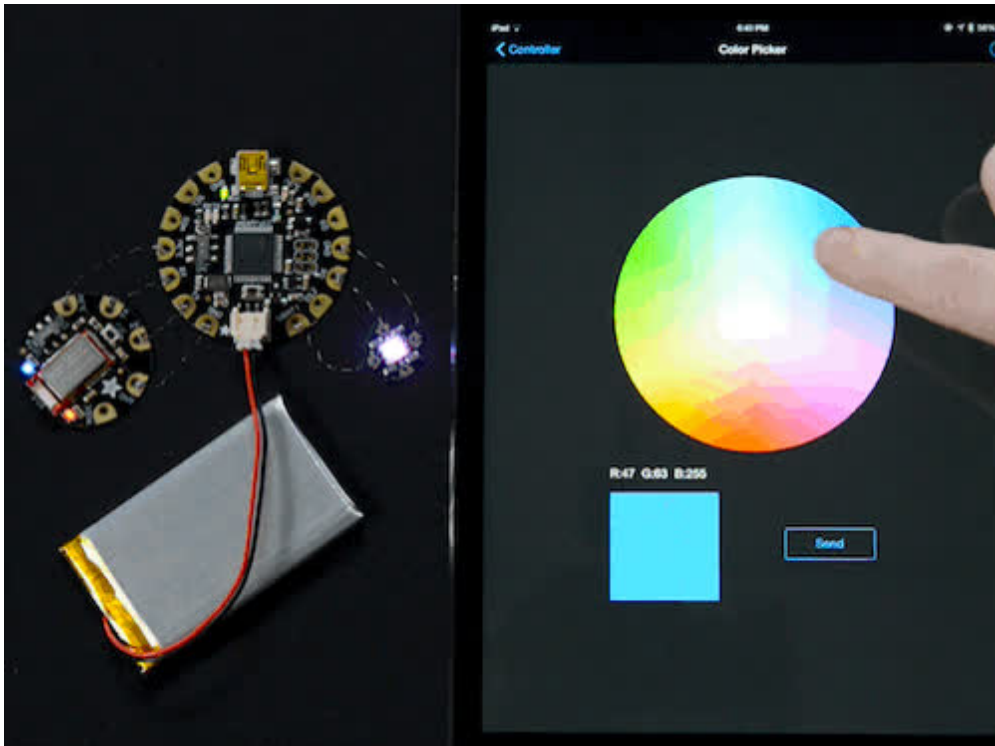
Would you like to add powerful and easy-to-use Bluetooth Low Energy to your wearable FLORA project? Heck yeah! With BLE now included in modern smart phones and tablets, its fun to add wireless connectivity. So what you really need is the new Adafruit Flora Bluefruit LE!

The Flora Bluefruit LE makes it easy to add Bluetooth Low Energy connectivity to your Flora. Sew 4 traces (or solder 4 wires) and BooM! Bluetooth Low Energy!



## Get started fast with the Bluefruit App

Using our Bluefruit [iOS App](http://adafru.it/iCi) or [Android App](http://adafru.it/f4G), you can quickly get your interactive 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). After you connect to the Bluefruit, you can send commands wirelessly in under 10 minutes

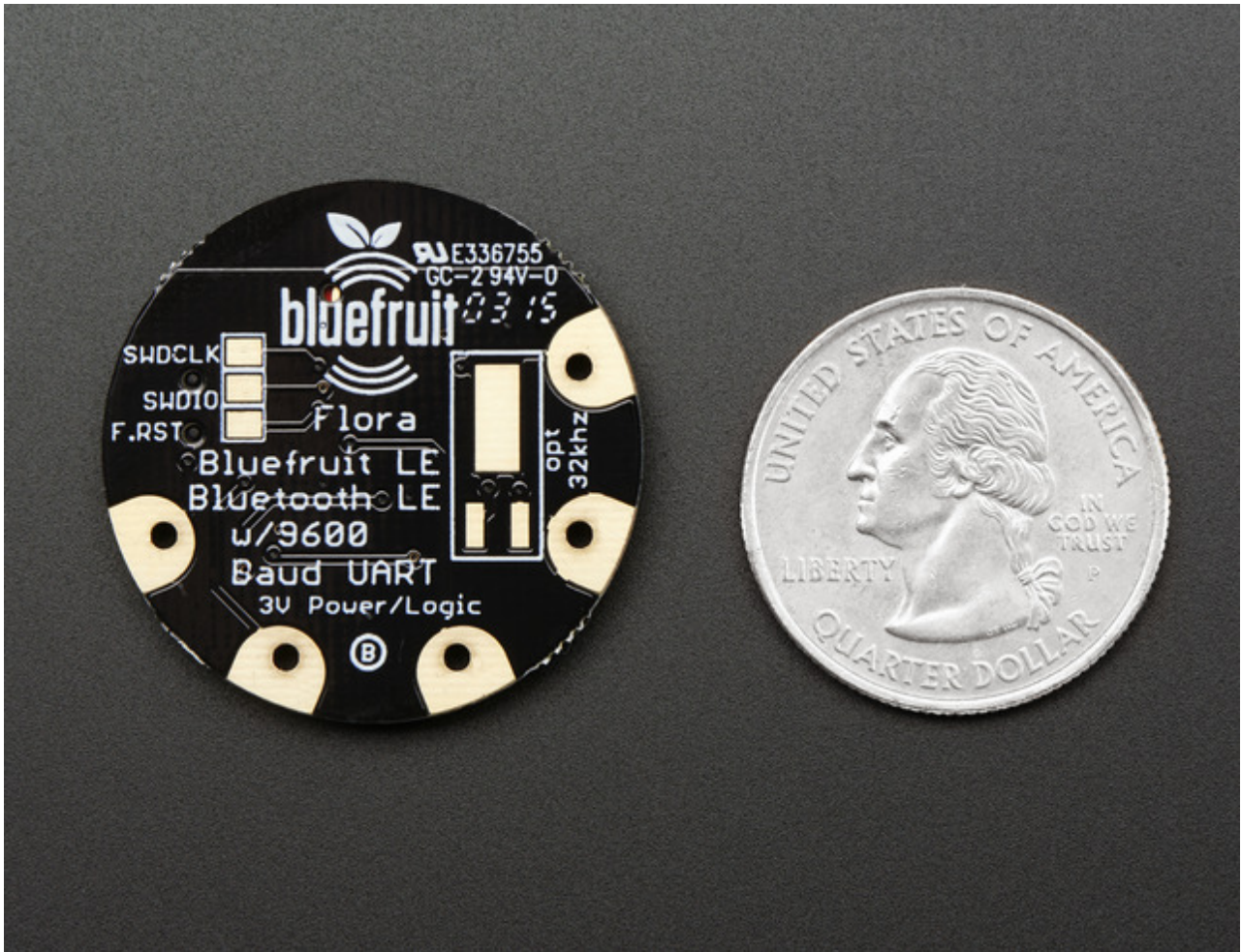


For advanced hackers, they'll be very happy to use the standard Nordic UART RX/TX connection profile. In this profile, the Bluefruit acts as a data pipe, that can 'transparently' transmit back and forth from your iOS or Android device. You can use our [iOS App](http://adafru.it/iCi) (<http://adafru.it/iCi>) 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>).

## You can do a lot more too!

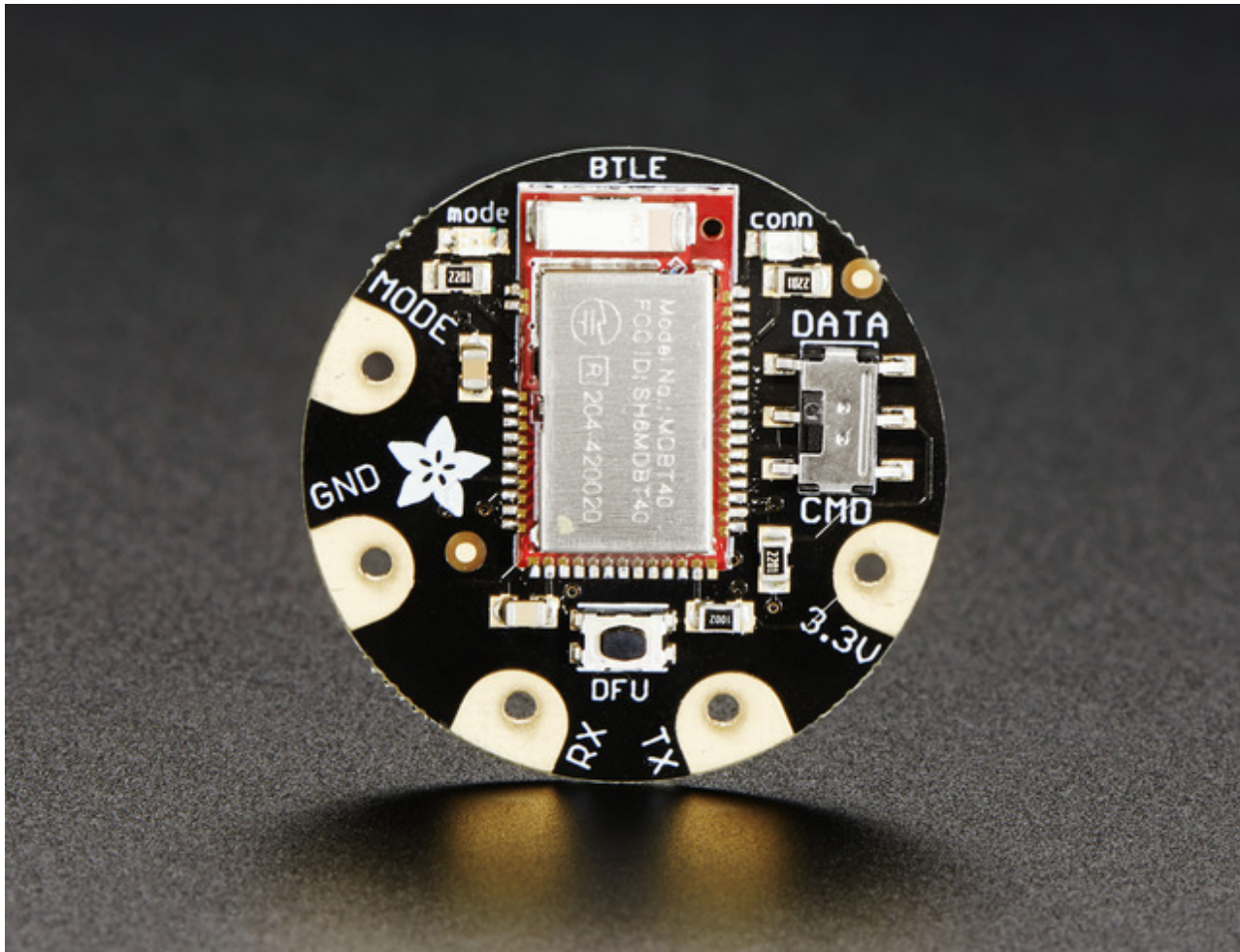
- [The Bluefruit can also act like an HID Keyboard](http://adafru.it/iCJ) (<http://adafru.it/iCJ>) (for devices that support BLE HID)
- [Can become a BLE Heart Rate Monitor](http://adafru.it/iCK) (<http://adafru.it/iCK>) (a standard profile for BLE) - you just need to add the pulse-detection circuitry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





[This is the same module and firmware as our BLE UART Friend](http://adafru.it/2479)(<http://adafru.it/2479>) but in a nice rounded shape, so you can switch between the two and have the same working code. This Bluefruit LE does not have the hardware flow control pins so it is best used with a microcontroller with hardware serial support (like, y'know, the Flora!)

# Pinouts



## Power Pads

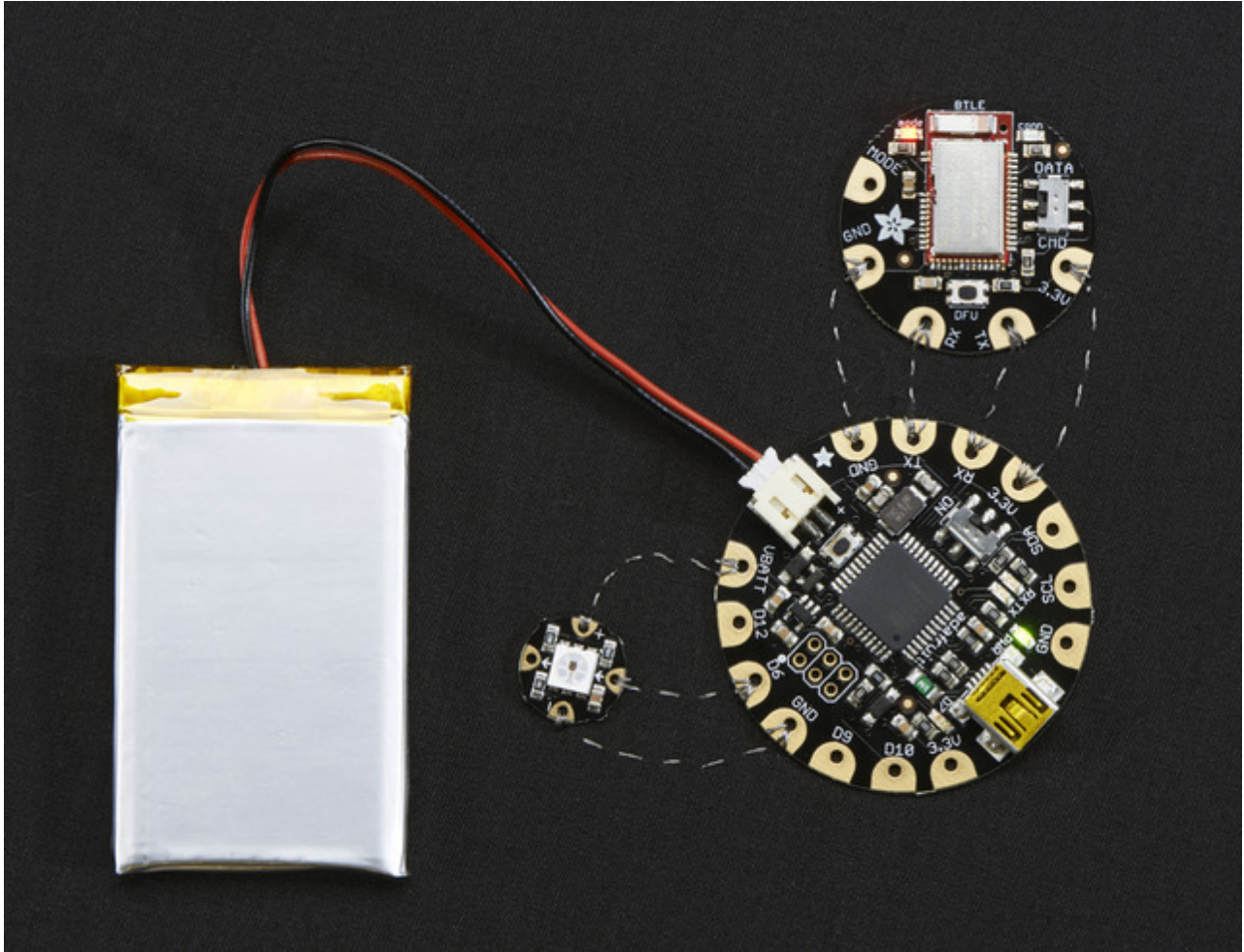
- **3.3V**: This is the power supply for the module, supply with 3.3V power supply input. You can dip down to maybe 2.7V and up to 3.6V or so, but regulated 3.3V is ideal
- **GND**: The common/GND pad for power and logic

## Data Pads

- **TX** - This is the UART Transmit pin *out* of the breakout (Bluefruit LE --> Flora RX), it's at 3.3V logic level.
- **RX** - This is the UART Receive pin *into* the breakout (Flora TX --> Bluefruit LE) it

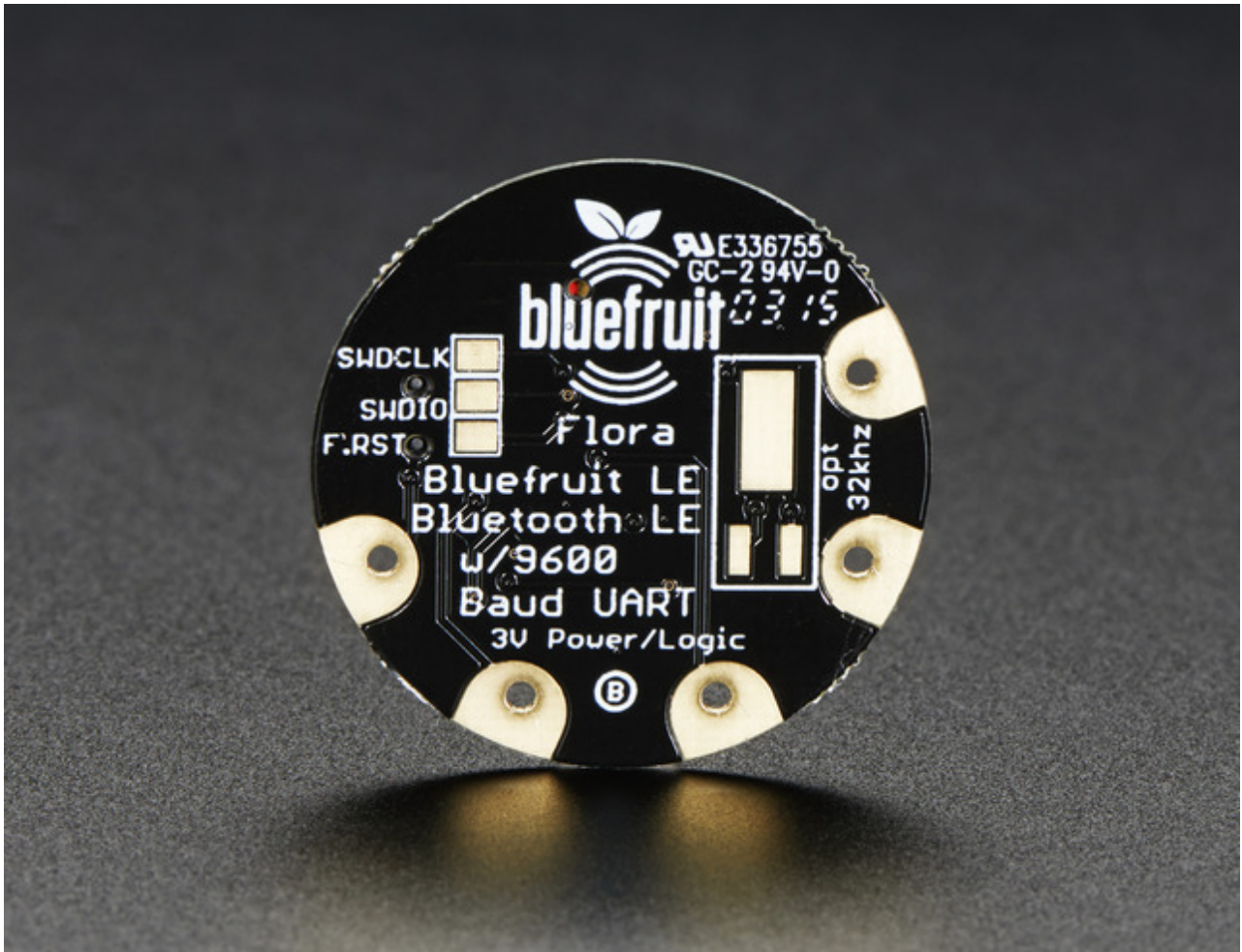
requires 3.3V logic level.

These 4 pads are in perfect order to connect directly to your Flora!



## Other Pads

- **MODE:** 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 requires 3.3V logic



## Reverse Side

On the back we also have a few breakouts!

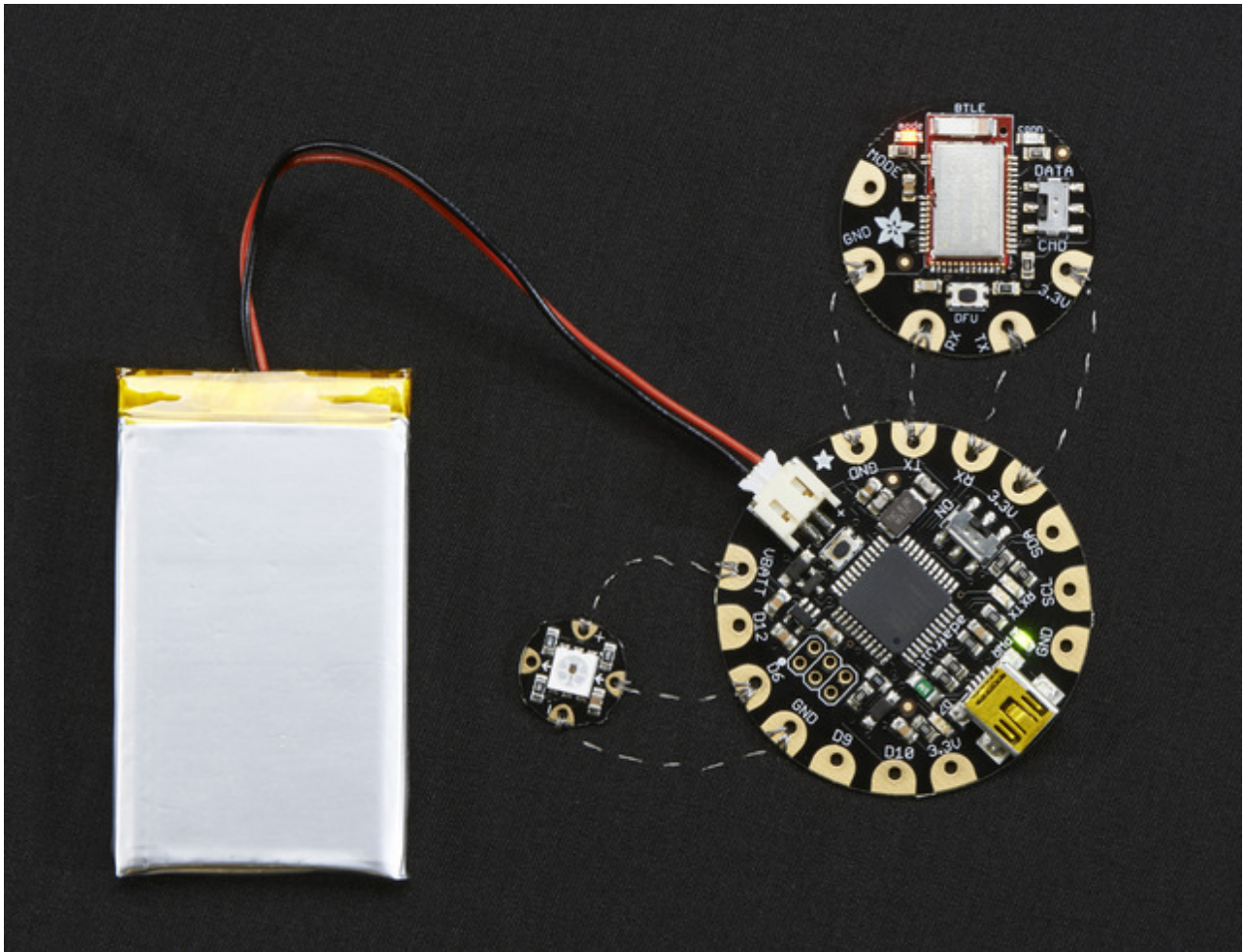
**Opt. 32 KHz:** If you're doing some funky low power work, [we wanted to give you the option of solderin in a 32khz oscillator.](http://adafru.it/f4U) (<http://adafru.it/f4U>) Our firmware doesn't support it yet but its there!

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

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

**F.RST:** 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)

# Sewing



Sewing the Flora BLE to a Flora is very straight-forward: the four required pads line up so just sew those in short straight lines from one module to another. the **MODE** pad is not required for most uses so you can leave that unconnected.

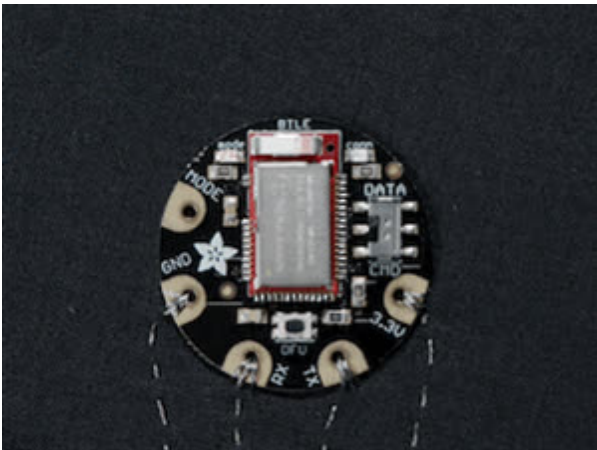
When you power it on, you should see the red LED on the BLE blink a few times and then pause & repeat, so you know that it has power!

# 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.

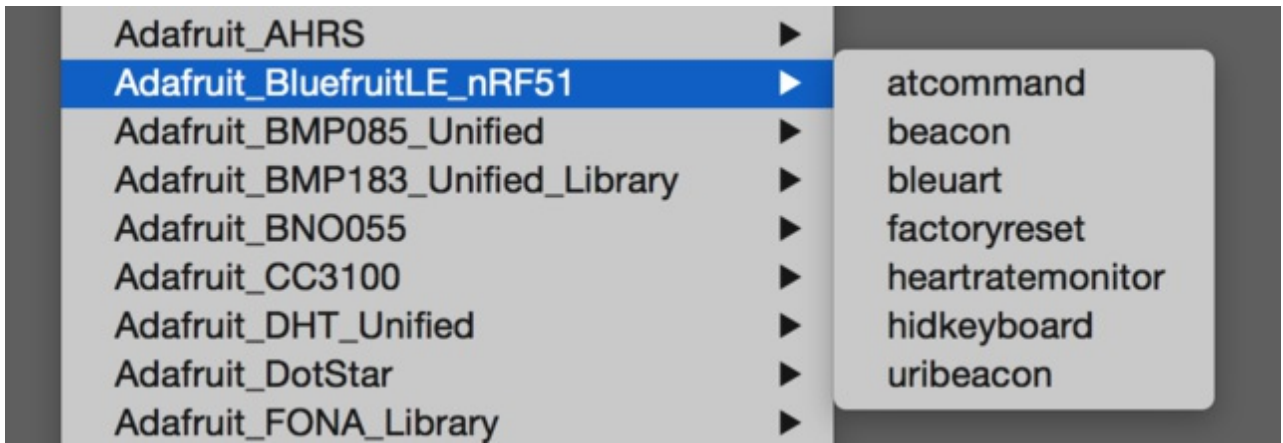
## Factory Reset via DFU Button

Once the module is powered up, if you hold the DFU button down for **5 seconds**, the blue LED on the module will start blinking and the device will perform a factory reset as soon as you release the DFU button.

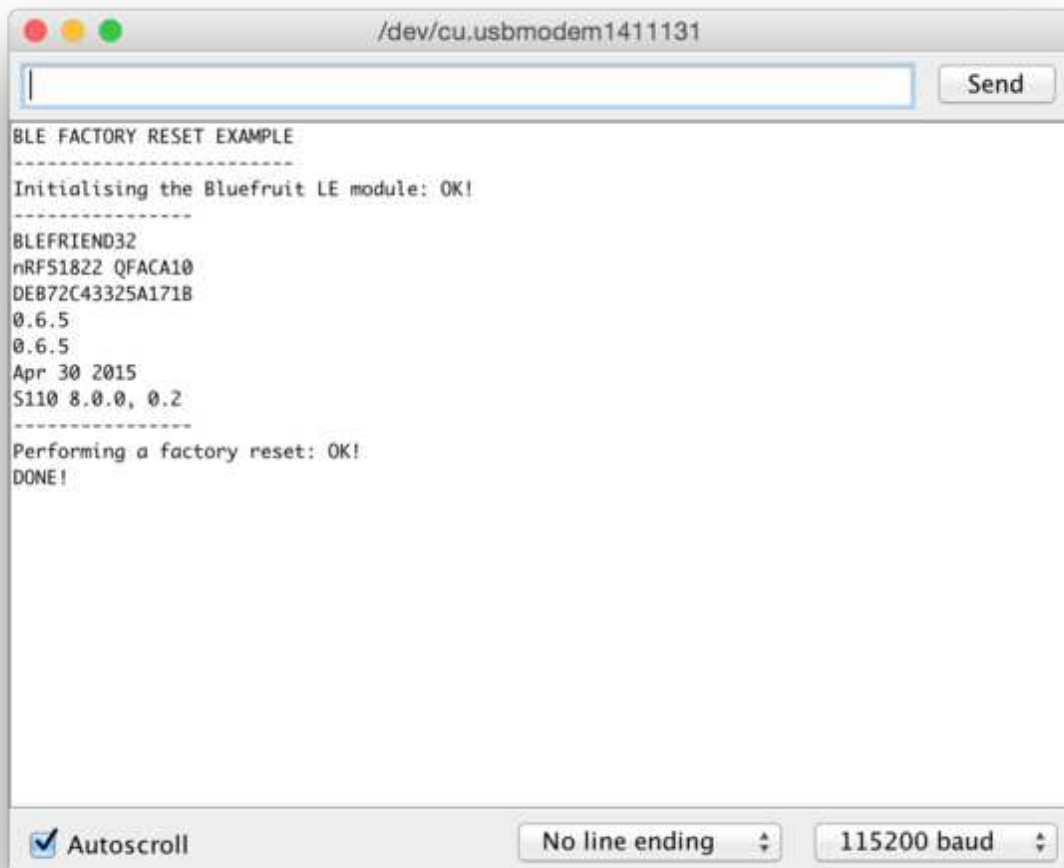


## FactoryReset Sample Sketch

There is a FactoryReset sample sketch in the Adafruit Bluefruit LE library, which can be accessed in the **File > Examples > Adafruit\_BluefruitLE\_nRF51** folder:



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



## AT+FACTORYRESET

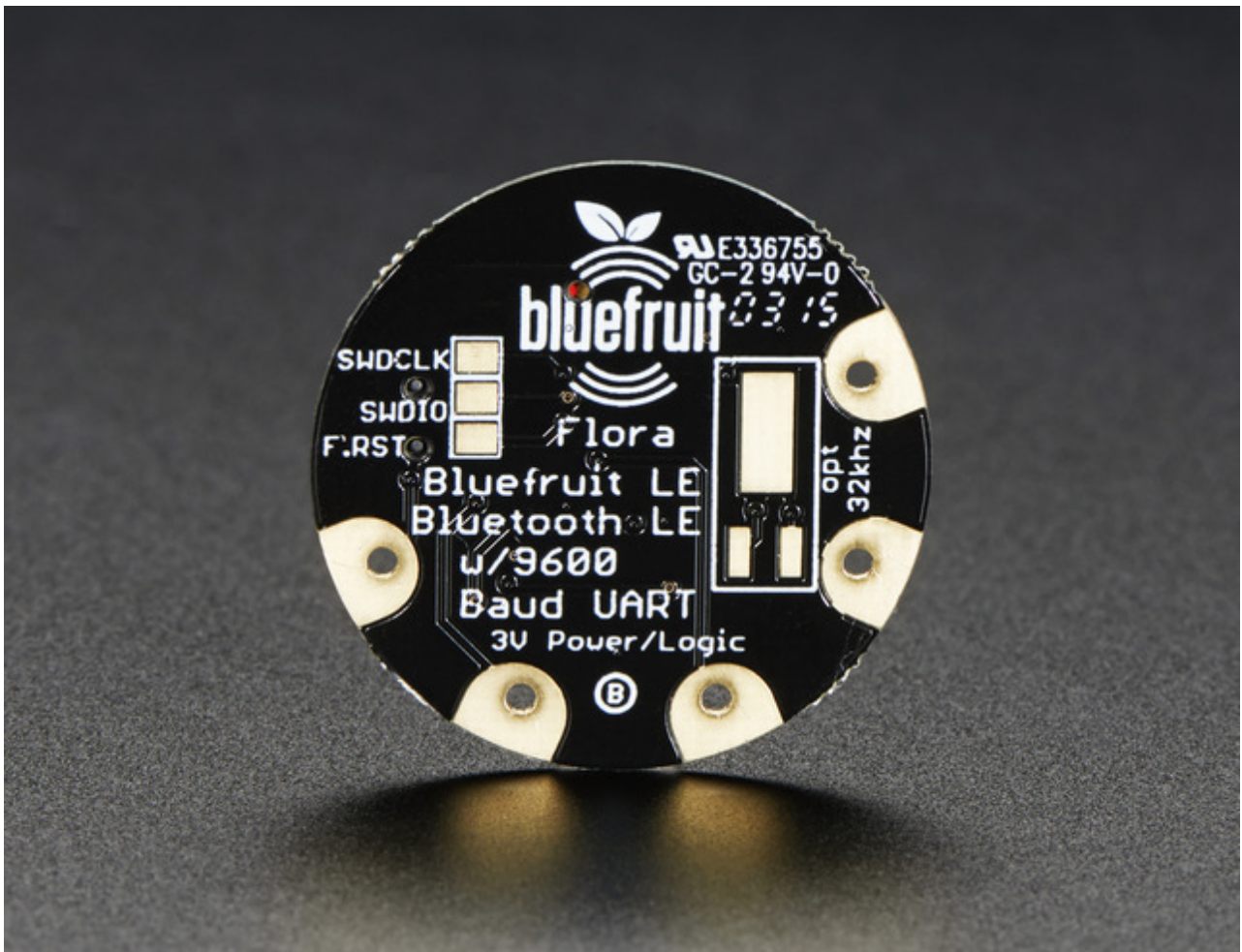
You can also perform a factory reset by sending the **AT+FACTORYRESET** command to your Bluefruit LE module in your favorite terminal emulator or using the [ATCommand](http://adafru.it/iCk) (<http://adafru.it/iCk>) example sketch. Make sure its in command mode first, then type in:

```
AT+FACTORYRESET
OK
```

This command will also cause the device to reset.

## Factory Reset via F.RST Test Pad

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







# Firmware 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>) from the Google Play Store. (An updated iOS version with DFU support will be available shortly!)

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:

# Installing 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), (<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)

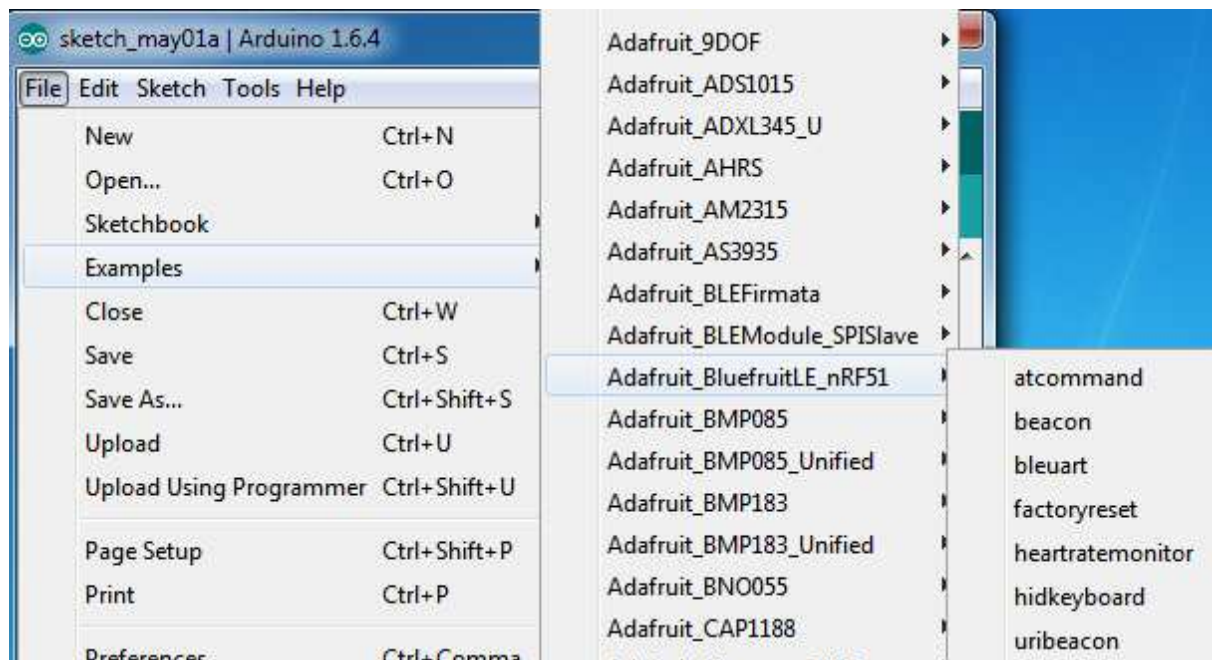
Place the **Adafruit\_BluefruitLE\_nRF51** library folder your *arduinorsketchfolder/libraries/* folder.

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:





# Example Code

We have tons of examples that will get you going with the Bluefruit, and its really easy! Since we have one library for both wearable and non-wearable Bluefruit's, the example code is the same but may require some minor tweaks to adjust for Flora.

Most importantly, **the flora bluefruit is for use with Hardware Serial only** and does not have flow control pins. Also, **we expect you will not use the mode pad** so you can keep the switch set to CMD and then change into Data mode by sending **+++** .

## Configuring for Flora

For all of the examples, look near the top for a line like this:

```
//#define BLUEFRUIT_HWSERIAL_NAME      Serial1
```

and remove the // comment marks. Do the same for:

```
//Adafruit_BluefruitLE_UART ble(BLUEFRUIT_HWSERIAL_NAME, BLUEFRUIT_UART_MODE_PIN);
```

Next, remove the code for the SoftwareSerial type, find these lines and delete them:

```
/* Create the bluefruit object, either software serial... */
```

```
SoftwareSerial bluefruitSS = SoftwareSerial(BLUEFRUIT_SWUART_TXD_PIN, BLUEFRUIT_SWUART_RXD_PIN);
```

```
Adafruit_BluefruitLE_UART ble(bluefruitSS, BLUEFRUIT_UART_MODE_PIN,  
    BLUEFRUIT_UART_CTS_PIN, BLUEFRUIT_UART_RTS_PIN);
```

Finally, unless you happen to be using the MODE pad, make sure **BLUEFRUIT\_UART\_MODE\_PIN** is set to -1

```
#define BLUEFRUIT_UART_MODE_PIN      -1 // Not used with FLORA
```

Same with RTS and CTS, find these lines and set both to -1

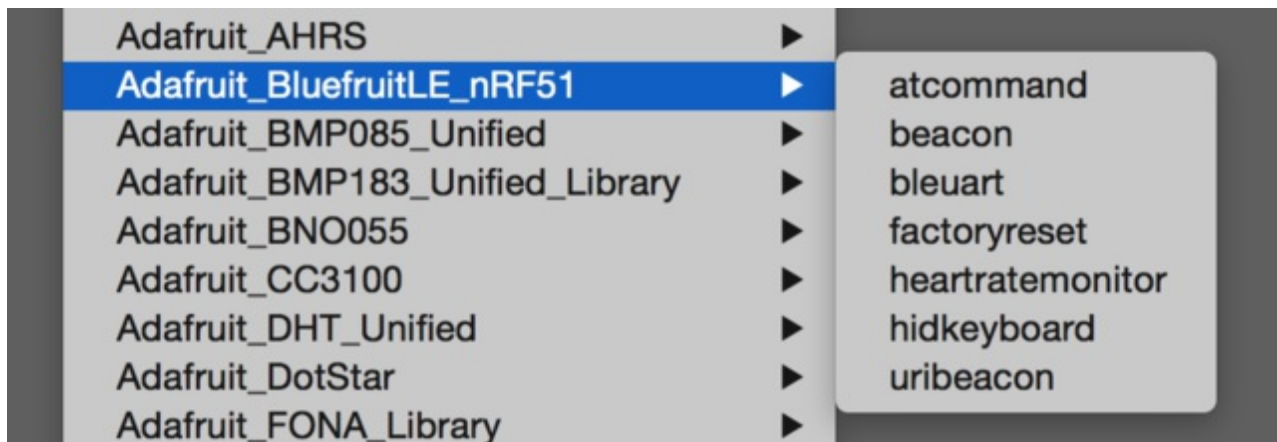
```
#define BLUEFRUIT_UART_CTS_PIN      -1 // Not used with FLORA  
#define BLUEFRUIT_UART_RTS_PIN      -1 // Not used with FLORA
```

# ATCommand

The **ATCommand** example allows you to execute AT commands from your sketch, and see the results in the Serial Monitor. This can be useful for debugging, or just testing different commands out to see how they work in the real world. It's a good one to start with!

## Opening the Sketch

To open the ATCommand sketch, click on the **File > Examples > Adafruit\_BluefruitLE\_nRF51** folder in the Arduino IDE and select **atcommand**:



This will open up a new instance of the example in the IDE, as shown below:



## Configuration

Check the **Configuration!** page earlier to set up the sketch for Software/Hardware UART or Software/Hardware SPI. The default is hardware SPI

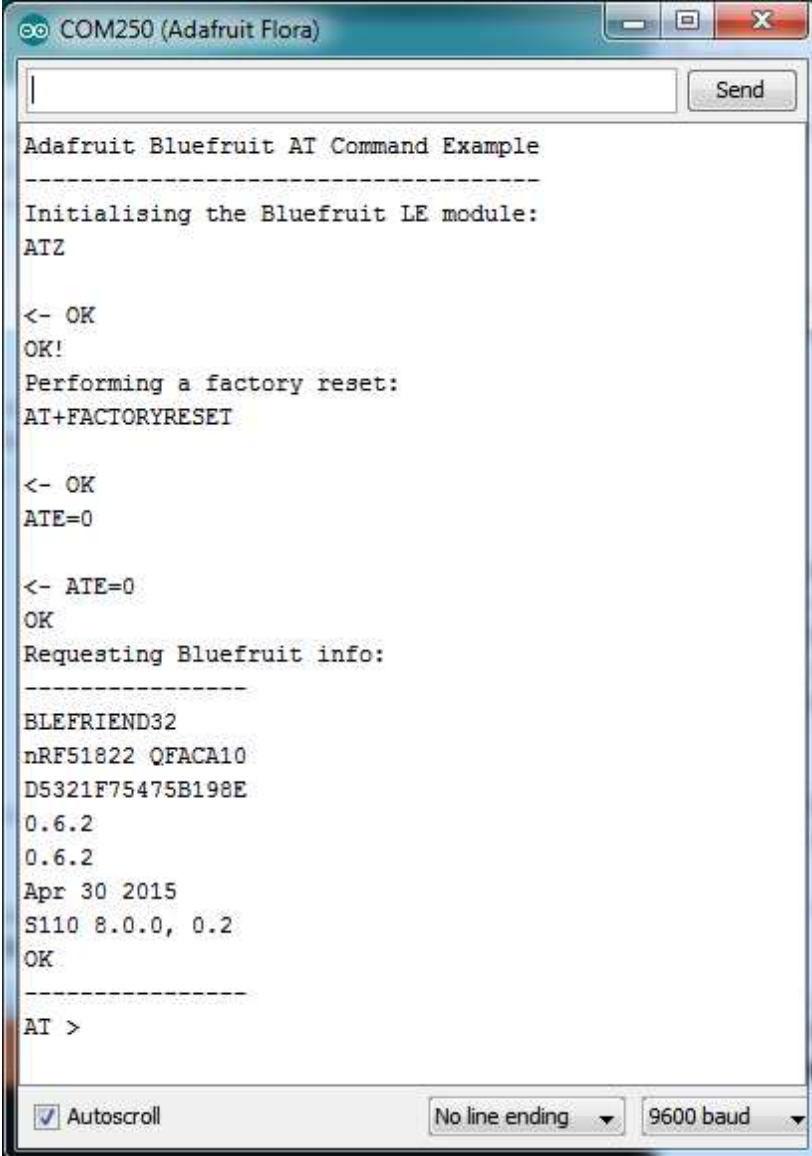
If using software or hardware Serial UART:

- This tutorial does not need to use the MODE pin, **make sure you have the mode**

- **switch in CMD mode** if you do not configure & connect a MODE pin
- Don't forget to also **connect the CTS pin on the Bluefruit to ground if you are not using it!** (The Flora has this already done)

## Running the Sketch

Once you upload the sketch to your board (via the arrow-shaped upload icon), and the upload process has finished, open up the Serial Monitor via **Tools > Serial Monitor**, and make sure that the baud rate in the lower right-hand corner is set to **115200**:



```
COM250 (Adafruit Flora)
|
| Send
Adafruit Bluefruit AT Command Example
-----
Initialising the Bluefruit LE module:
ATZ

<- OK
OK!
Performing a factory reset:
AT+FACTORYRESET

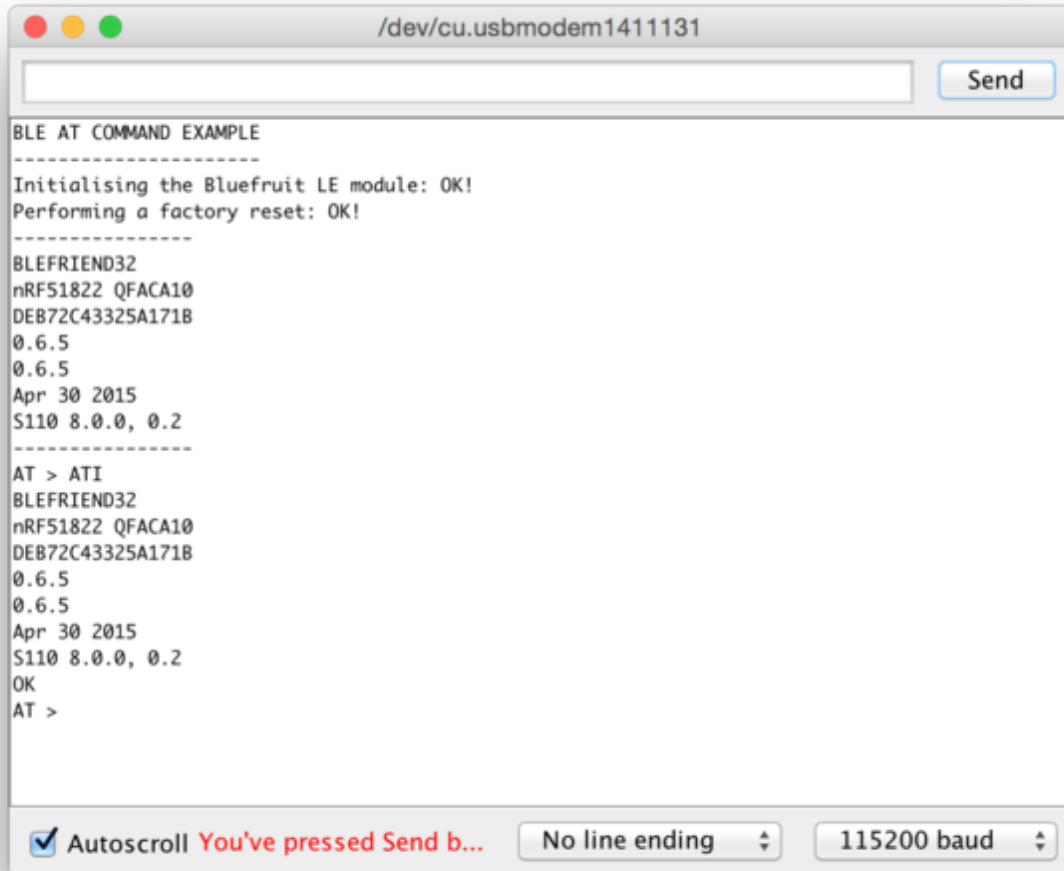
<- OK
ATE=0

<- ATE=0
OK
Requesting Bluefruit info:
-----
BLEFRIEND32
nRF51822 QFACA10
D5321F75475B198E
0.6.2
0.6.2
Apr 30 2015
S110 8.0.0, 0.2
OK
-----
AT >
```

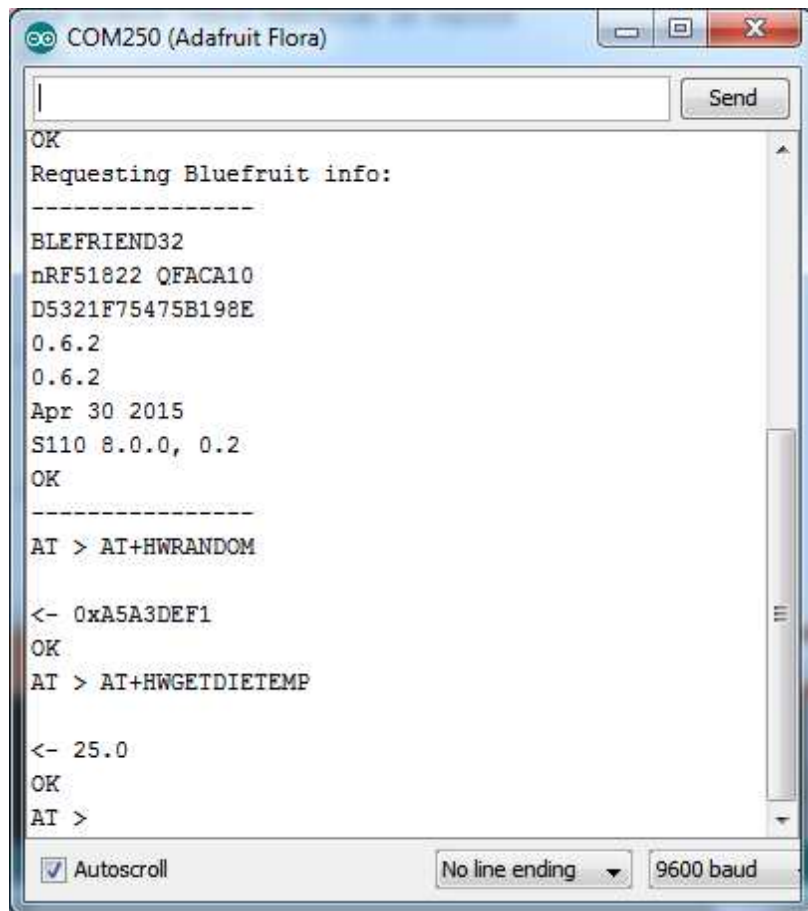
Autoscroll  No line ending 9600 baud

To send an AT command to the Bluefruit LE module, enter the command in the textbox at the top of the Serial Monitor and click the **Send** button:

The response to the AT command will be displayed in the main part of the Serial Monitor. The response from 'ATI' is shown below:



You can do pretty much anything at this prompt, with the AT command set. Try **AT+HELP** to get a list of all commands, and try out ones like **AT+HWGETDIETEMP** (get temperature at the nRF51822 die) and **AT+HWRANDOM** (generate a random number)



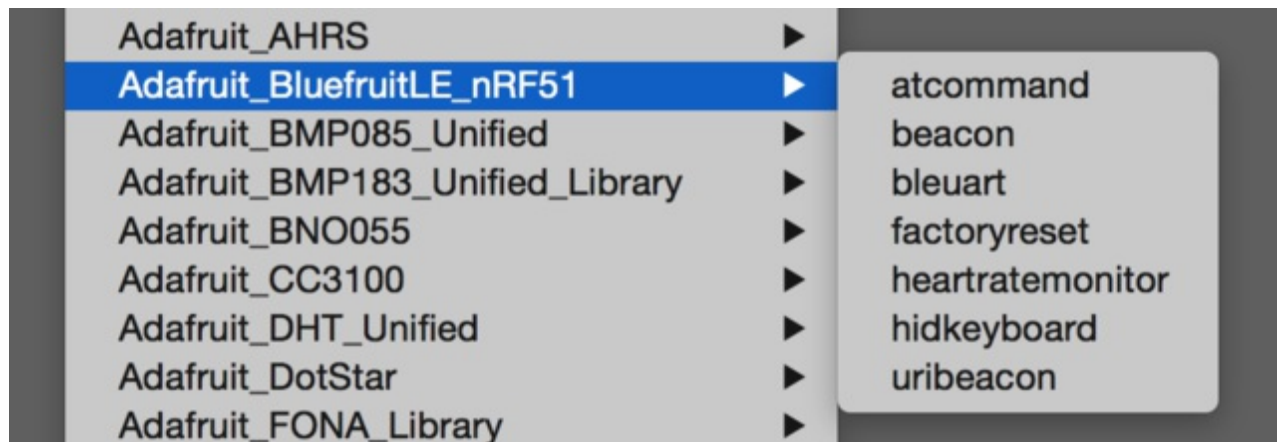


# BLEUart

The **BLEUart** example sketch allows you to send and receive text data between the Arduino and a connected Bluetooth Low Energy Central device on the other end (such as your mobile phone using the **Adafruit Bluefruit LE Connect** application for [Android](http://adafru.it/f4G) or [iOS](http://adafru.it/f4H) in UART mode).

## Opening the Sketch

To open the ATCommand sketch, click on the **File > Examples > Adafruit\_BluefruitLE\_nRF51** folder in the Arduino IDE and select **bleuart\_cmdmode**:



This will open up a new instance of the example in the IDE, as shown below:

```
bleuart_cmdmode | Arduino 1.6.4
File Edit Sketch Tools Help
bleuart_cmdmode $
/*!
  @file    bleuart_cmdmode.ino
  @author  hathach, ktown (Adafruit Industries)

  This demo will show you how to send and receive data in COMMAND mode
  (without needing to put the module into DATA mode or using the MODE pin)
*/
#include <string.h>
#include <Arduino.h>
#include <SPI.h>
#include <SoftwareSerial.h>

#include "Adafruit_BLE.h"
#include "Adafruit_BLE_HWSPI.h"
#include "Adafruit_BluefruitLE_UART.h"

// If you are using Software Serial....
// The following macros declare the pins used for SW serial, you should
// use these pins if you are connecting the UART Friend to an UNO
#define BLUEFRUIT_SWUART_RXD_PIN    9    // Required for software serial!
#define BLUEFRUIT_SWUART_TXD_PIN    10   // Required for software serial!
#define BLUEFRUIT_UART_CTS_PIN      11   // Required for software serial!
#define BLUEFRUIT_UART_RTS_PIN      -1   // Optional, set to -1 if unused

// If you are using Hardware Serial
// The following macros declare the Serial port you are using. Uncomment this
// line if you are connecting the BLE to Leonardo/Micro or Flora
```

Done compiling.

Global variables use 498 bytes of dynamic memory.

19 Adafruit Flora on COM250

## Configuration

Check the **Configuration!** page earlier to set up the sketch for Software/Hardware UART or Software/Hardware SPI. The default is hardware SPI

If using software or hardware Serial UART: