

在 Arduino 平台使用八卦单机版 (Sino:bit)

作者 Dave Astels 译者 Yang Yuan (twitter: @Hello3Planet)



原文最后更新时间 2017-12-06 10:15:35 PM UTC

译文最后更新时间 2017-12-19

目录

目录	2
概览	3
装上板子亮起来	4
安装 Windows 7 驱动	4
Download Arduino IDE	4
Install SoftDevice onto SinoBit	4
Add NRF5x Board Support	4
Install the Sino:bit library.	5
Select Board and Upload	5
Buttons	9
NOTE:	10
Other GPIO	11
Alligator Clip Pads	11
UART	12
I2C	12
Expansion Connector	12
Accelerometer and Magnetometer	14
Magnetometer	14
Accelerometer	16
Adafruit Libraries	19
Download BLE Peripheral library	19
Download Adafruit GFX library	19
Download Adafruit_Microbit library	20
LED Matrix	21
Analog Input	23
Bluetooth	26
Bluetooth UART	27
Install Library & Example Code	27

Bluetooth Connection	28
Bluetooth Plotter	32
Install Library & Example Code	32
Bluetooth Controller	35
Install Library & Example Code	35
Logging Temperature to Adafruit IO	38
Create a Microbit Temperature Feed	38
Temperature Logger Sketch	38
Test UART Mode	39
HALP!!!	45

概览

引用自 <https://github.com/sinobitorg/hardware>

八卦单机板是为中国打造的一款微型可编程计算机。是基于德国的 Calliope mini 获得官方准许后衍生出来的项目。在构思中国版本的时候有几个想法，其中一点就是把原有的 5X5LED 矩阵改为 12X12。这样可支持多国语言，不仅只是英语，也可以显示中文，日本，印度语，阿拉伯语和其他不以拉丁语为基础的语言。如果保留原有的只是 5X5 矩阵，其他国家的孩子就无法体验到第一次显示屏弹出他们自己语言的“Hello, World”的那种激动兴奋的心情。

八卦单机板是由 Naomi Wu 所构思，一位开源硬件的传道者和 DIY 爱好者。由深圳本地为创客和硬件爱好者以合同形式提供代工业务的电子公司 Elecrow Technology 所设计制作。

你知道吗？酷酷的八卦单机板又有了新玩法！现在可以用 Arduino IDE 给八卦单机板写程序了哦。就让我们跟随这篇教程学习如何在八卦单机板上点亮 LED、读取内置温度传感器、通过蓝牙收发数据，还可以上传数据记录到 Adafruit.IO！

八卦单机板衍生自衍生于 BBC Micro:bit 的 [Calliope Mini](#)，采用同样的 Nordic nRF51822 MCU、同样的加速度传感器和磁力传感器，也同样拥有两个用户可读的按钮输入和可自定义内容的 LED 矩阵。八卦单机板板载了 MicroUSB 下载接口，同时提供 JST 3V 电池接口以便脱离电脑使用。在需要与其他硬件进行更为稳固的连接时，也可以使用板上预留的 2.54mm 间距 2x13 排针接口。

八卦单机板和 Calliope Mini 和 BBC Microbit 相比，有多处不同之处，有一些甚至是大刀阔斧的改进。

八卦单机板也预留了适配鳄鱼夹的孔位，但足足有 8 个：除去电源和地，还有足足 6 个输入输出接口。而且 8 个接口排布于八边形的八个角，彼此间距较大，在使用时也更容易意外短路。

以及，八卦单机板的超大改变：LED 矩阵从 5x5 扩大到了 12x12。这样做的出发点是显示中文等非罗马字符，也有诸多额外好处（后文详述）。

要干脆、别犹豫！拿起你的八卦单机板跟着教程嗨起！

PS: 八卦单机板还有个特殊之处——它是开源硬件联盟（OSHWA）在中国认证的第一款硬件产品！

装上板子亮起来！ 安装 Windows 7 驱动程序

如果你在用 Windows 7，则需要安装这个驱动。

如果你使用的是 Mac，Windows 8, 8.1, 10 或者 Linux，则可直接跳过本步骤。

下载 mBed 串口驱动程序（英文页面）

以上页面的 Adafruit 短连接 <https://adafru.it/pNa>

下载 Arduino IDE

需要下载桌面版的 Arduino IDE. 请确保下载最新版本

下载 Arduino IDE

以上页面的 Adafruit 短连接 <https://adafru.it/fvm>

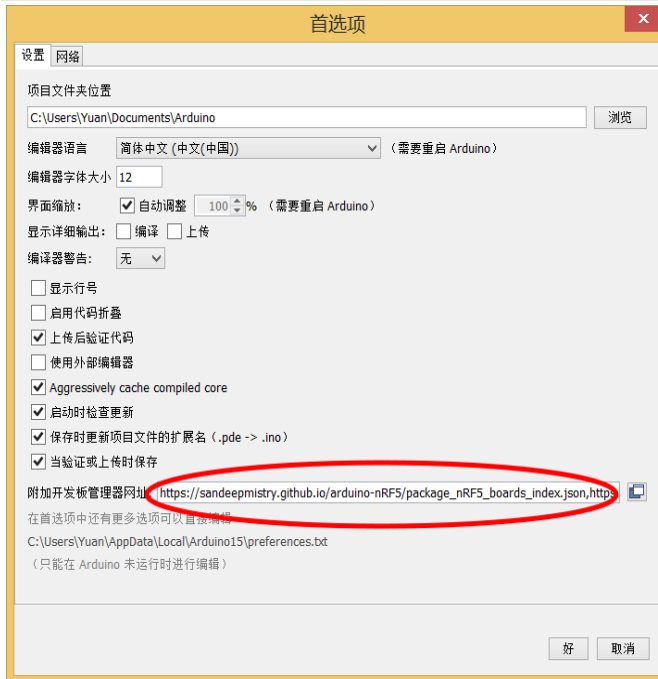
向八卦单片机烧录 SoftDevice 固件

需要安装 SoftDevice 固件才能使用 Arduino 软件与八卦单片机通讯。如果你之前在八卦单片机上使用了 MicroPython，则需要重新烧录 SoftDevice 固件。

向八卦单片机烧录 SoftDevice 固件并不困难，可以参考 [Sandeep Mistry 的说明（英文）](#)。我在选择编程器时选择了 CMSIS-DAP，在 SoftDevice 固件烧入以后，我编程器选项选回了 USBTinyISP。

添加 NRF5x 开发板支持

Micro:bit 使用的 nRF51 架构不被 Arduino 原生支持，但是我们可以可以在 Arduino IDE 下简单地添加支持文件。



打开 **Arduino** 的 **文件 > 首选项** 菜单，在附加开发板管理器网址中添加这个网址：

https://sandeepmistry.github.io/arduino-nRF5/package_nRF5_boards_index.json

如果附加开发板管理器网址列表中已有其他网址，可以在添加时用逗号进行分隔。



打开 **工具 > 开发板 > 开发板管理器** 菜单，搜索 **Nordic**，安装 **Nordic Semiconductor nRF5 Boards by Sandeep Mistry** 开发板配置文件。

安装八卦单片机板库文件

八卦单片机板目前还需要独立的库文件进行支持，但应该很快被纳入 nRF5 主板支持列表。目前的使用过程中，我们可以把八卦单片机板看做带有附加功能的 micro:bit，八卦单片机板库文件提供的就是对这部分附加功能的支持。

测试版的八卦单片机支持库文件在 <https://github.com/dastels/Sinobit>，需要下载并手动安装。点击以下链接可以直接下载对应的 zip 文件。

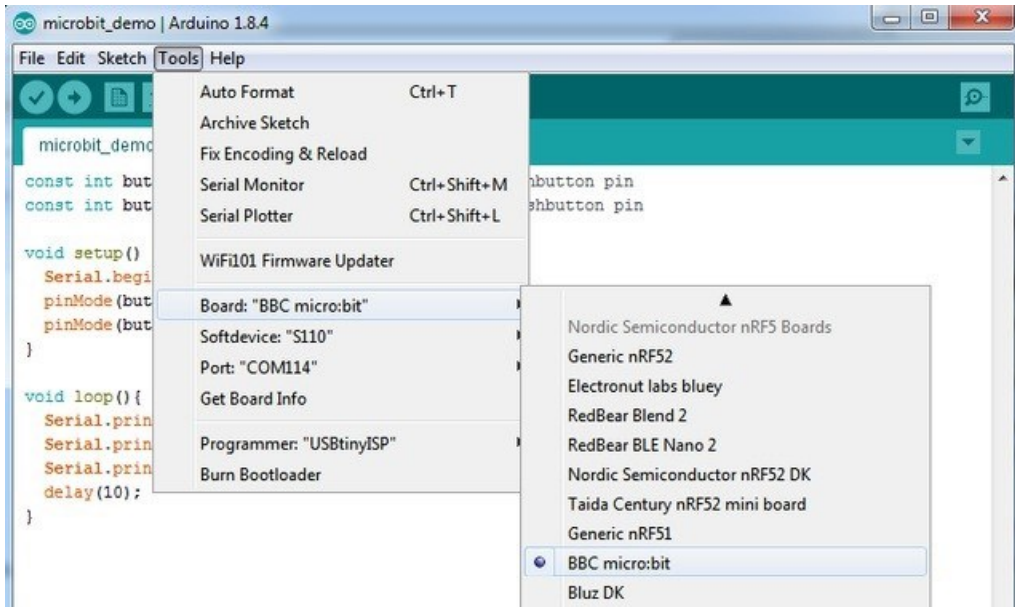
下载八卦单片机板库文件

以上页面的 Adafruit 短连接 <https://adafru.it/Afh>

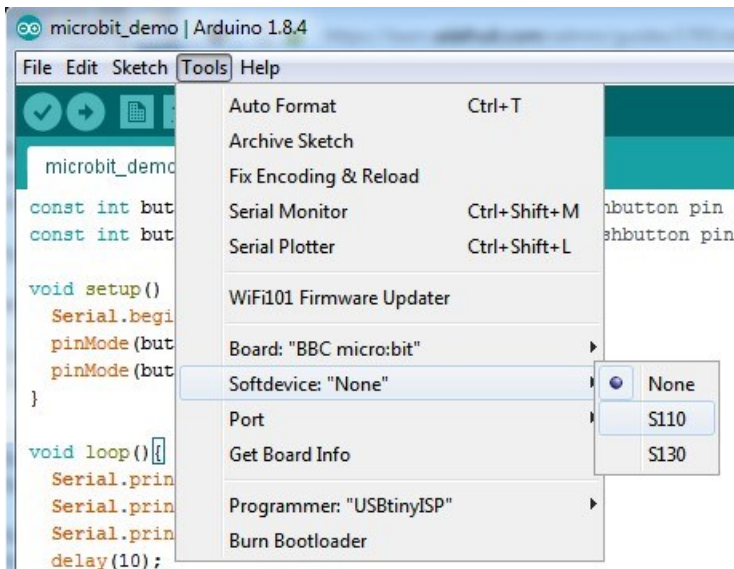
将下载到的 zip 文件解压，并将解压出的文件夹重命名为 Sinobit，确认其中包含 src 和 example 文件夹，并把这个 Sinobit 文件夹复制到你的 Arduino 默认项目文件夹下的 libraries 子文件夹中。如果之前从未安装过附加库文件，可能需要手动创建 libraries 文件夹。文件拷贝完成后，重启 Arduino IDE 即可生效。

选取开发板并上传

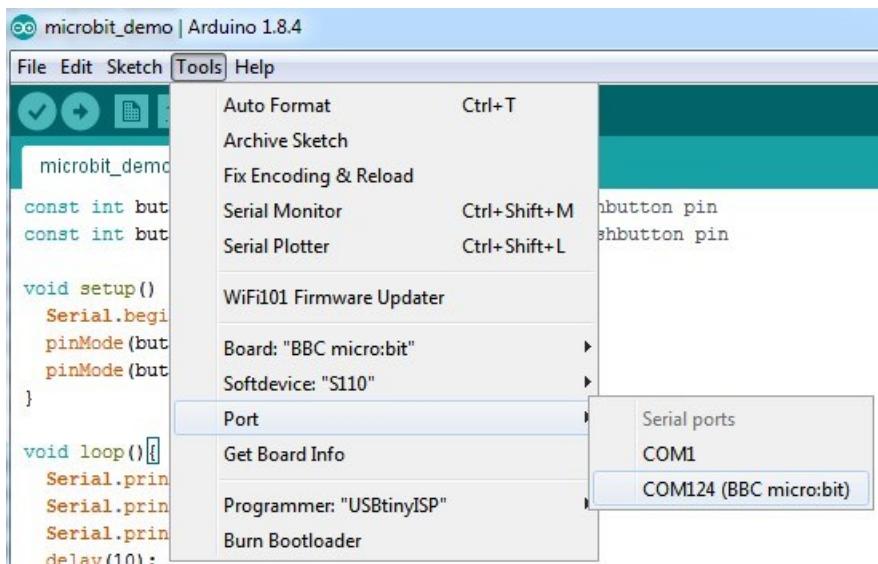
在开发板菜单选择 BBC micro:bit，



并在开发板下方的 Softdevice 选择 S110



选择 micro:bit 对应的 COM 接口。



并用以下代码创建 blink 示例项目，并上传。

```
#include <sinobit.h>

Sinobit matrix = Sinobit();

void setup() {
  Serial.begin(9600);
  Serial.println("Sinobit is ready!");

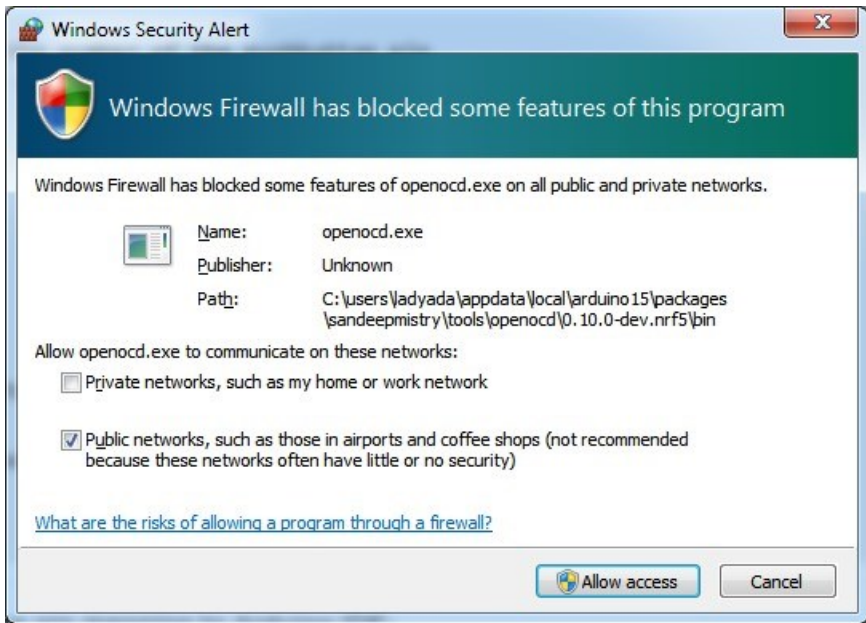
  matrix.begin();
  delay(100);
  matrix.clearScreen();
}

void loop() {
  Serial.println("blink!");

  // set a pixel
  matrix.drawPixel(0, 0, 1);
  matrix.writeScreen();
  delay(500);

  // clear a pixel!
  matrix.drawPixel(0, 0, 0);
  matrix.writeScreen();
  delay(500);
}
```

如果你的 Windows 防火墙弹出关于 **openocd.exe** 的安全警告，请选择“允许访问”以便代码能够正确上传。上传成功后，八卦单片机左上角的 LED 就会闪动哦。



按钮用起来

用以下代码创建一个新的项目，便可以在串口监视器看到按钮按下的状态。

打开串口监视器，设置波特率为 9600bps，按下 MicroUSB 插座边的 RST 按钮来重启板载的程序，在欢迎信息后按下板载的按钮，就可以看到按钮的状态。

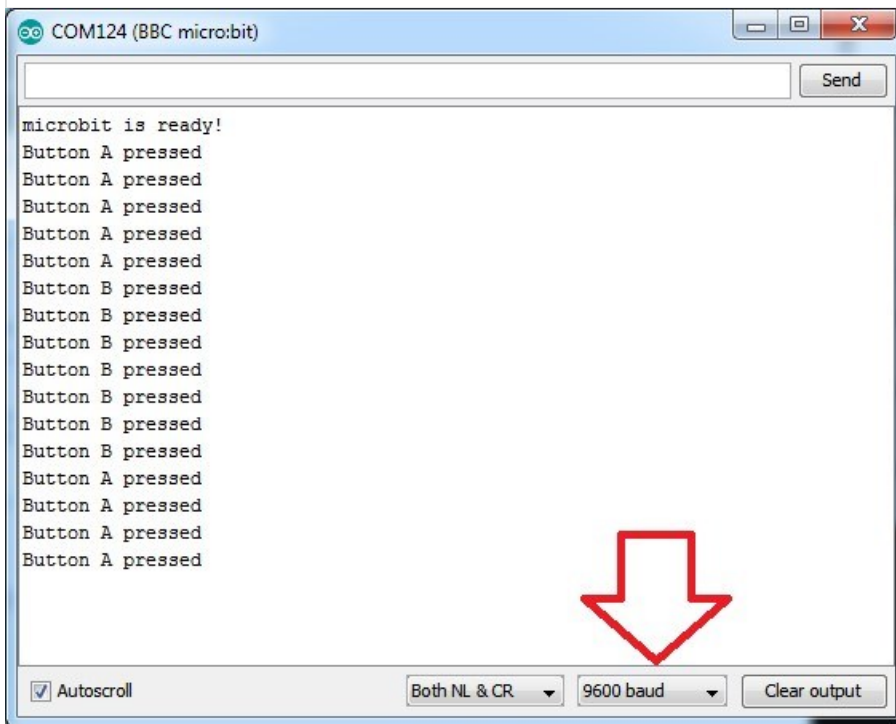
注意：在八卦单片机板上，按钮 A 是 LED 矩阵右侧的按钮，而按钮 B 是 LED 矩阵左侧的按钮。

```
const int buttonA = 5; // the number of the pushbutton pin
const int buttonB = 11; // the number of the pushbutton pin

void setup() {
  Serial.begin(9600);
  Serial.println("microbit is ready!");

  pinMode(buttonA, INPUT);
  pinMode(buttonB, INPUT);
}

void loop(){
  if (!digitalRead(buttonA)) {
    Serial.println("Button A pressed");
  }
  if (!digitalRead(buttonB)) {
    Serial.println("Button B pressed");
  }
  delay(10);
}
```



如果你选择在项目中包含 `sinobit.h`，也可以直接用该文件中为按钮定义的常量（`SINOBIT_BUTTON_A`、`SINOBIT_BUTTON_B`）。

```
#include <sinobit.h>

void setup() {
  Serial.begin(9600);
  Serial.println("microbit is ready!");

  pinMode(SINOBIT_BUTTON_A, INPUT);
  pinMode(SINOBIT_BUTTON_B, INPUT);
}

void loop(){
  if (!digitalRead(SINOBIT_BUTTON_A)) {
    Serial.println("Button A pressed");
  }
  if (!digitalRead(SINOBIT_BUTTON_B)) {
    Serial.println("Button B pressed");
  }
  delay(10);
}
```

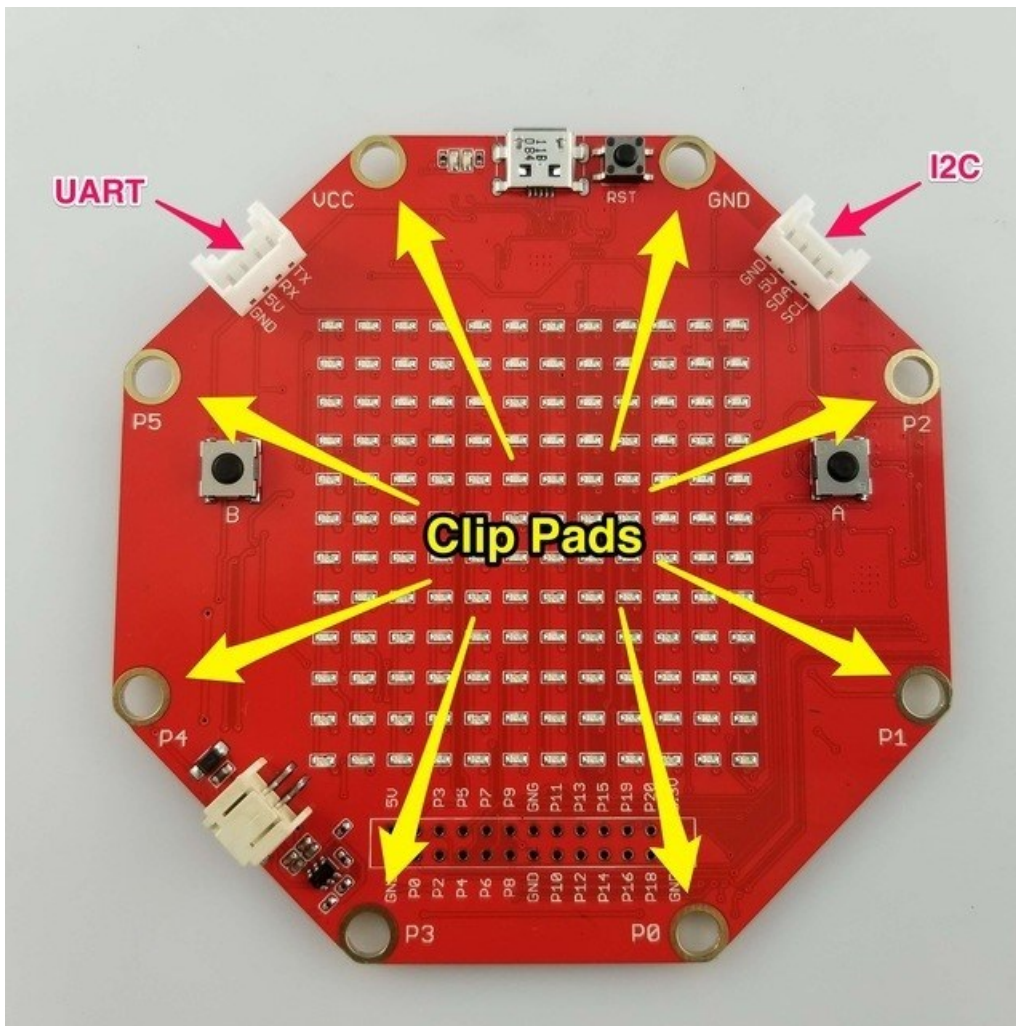
注意：

八卦单片机板上的按钮输出的是“触发低电平”信号，也就是说按钮按下输出低（逻辑假），松开时输出高（逻辑真）。

其他 GPIO

八卦单机板上有一些不同的输入输出方式，接下来我们来一起看一看。

(GPIO:General Purpose Input and Output)



鳄鱼夹适配焊盘

在八卦单机板的边缘有 8 个适配鳄鱼夹的焊盘，3V (VCC) 和地线 (GND)，以及标记 P0-P5 的六个输入输出接口，足有 micro:bit 的两倍之多。像 Calliope 一样，这 8 个焊盘设计得彼此分开，以避免意外短接。库文件里也提供了引用这些焊盘的常量。

下面的示例会依次在 P0-P5 端口发送脉冲，永不停息。

```
#include <sinobit.h>

void setup() {
  pinMode(SINOBIT_PAD_P0, OUTPUT);
  pinMode(SINOBIT_PAD_P1, OUTPUT);
  pinMode(SINOBIT_PAD_P2, OUTPUT);
  pinMode(SINOBIT_PAD_P3, OUTPUT);
  pinMode(SINOBIT_PAD_P4, OUTPUT);
  pinMode(SINOBIT_PAD_P5, OUTPUT);
}

void pulse(int io_pin)
{
  digitalWrite(io_pin, HIGH);
  delay(100);
  digitalWrite(io_pin, LOW);
  delay(100);
}

void loop() {
  pulse(SINOBIT_PAD_P0);
  pulse(SINOBIT_PAD_P1);
  pulse(SINOBIT_PAD_P2);
  pulse(SINOBIT_PAD_P3);
  pulse(SINOBIT_PAD_P4);
  pulse(SINOBIT_PAD_P5);
}
```

UART 接口

八卦单机版左上的 4pin 接口是 UART 接口，管脚从左至右分别是 GND，5v，Rx，和 Tx。

RX 同时与板边的 P1 焊盘和 I/O 扩展端口（下文详述）上的 P1 管脚连接。类似的，TX 与板边的 P2 焊盘和 I/O 扩展端口上的 P2 管脚连接。这样的连接方式下，你可以用鳄鱼夹、4pin 线或是扩展板来连接 UART 接口，更加灵活。

I2C 接口

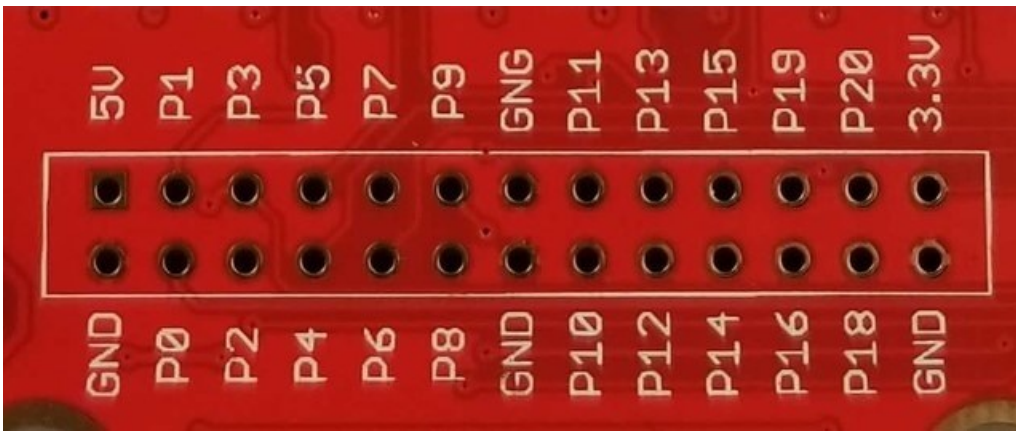
与 UART 接口类似，I2C 信号可由八卦单机板右上的 4pin 接口引出，SDA、SCL 分别连接到 I/O 扩展端口的 P19、P20 管脚。但与 UART 接口不同的是，I2C 接口并没有连接到鳄鱼夹用的焊盘上。

I/O 扩展端口

I/O 扩展端口使用 2x13 2.54mm 连接器，你可以在板子正面（有 LED 矩阵的一面）焊上排针以使用跳线连接到面包板制作原型，也可以在板子背面焊上排母来让出 LED 矩阵、挂载扩展板。在板子背面用排针也是可以的，但是我个人更愿意用排母，以免某个管脚与桌上或工作台上的金属物件接触时，让你的板子遇到化作一缕青烟怪 Sparky（英文）。

所有可用的 I/O 输入信号都连接到这个扩展端口：数字与模拟连接、I2C 接口、UART 接口、SPI 接口以及地线、3.3V 和 5V。

和鳄鱼夹焊盘一样，八卦单机板的库文件定义了这些信号的引用名称。



板上引出这些连接和 micro:bit 的用法一样，但显著的不同（后文详述）在于在八卦单机板上 LED 矩阵的使用并不影响 I/O 输入输出管脚的使用，而且，兼容鳄鱼夹的焊盘数量也更多。

- Pin #0 - 兼容鳄鱼夹- 模拟输入，
- Pin #1 - 兼容鳄鱼夹- 模拟输入， 也可用作 Rx
- Pin #2 - 兼容鳄鱼夹- 模拟输入， 也可用作 Tx
- Pin #3 - 兼容鳄鱼夹， 模拟输入，
- Pin #4 - 兼容鳄鱼夹， 模拟输入，
- Pin #5 - 也可用作按钮 A
- Pin #6
- Pin #7
- Pin #8
- Pin #9
- Pin #10 - 兼容鳄鱼夹， 模拟输入，
- Pin #11 - 也可用作按钮 B
- Pin #12
- Pin #13 - 也可用作 SPI 的 CLOCK 管脚
- Pin #14 - 也可用作 SPI 的 MISO 管脚
- Pin #15 - 也可用作 SPI 的 MOSI 管脚
- Pin #16
- Pin #19 - 也可用作 I2C 的 SCL (CLOCK)管脚
- Pin #20 - 也可用作 I2C 的 SDA (DATA)管脚

这个扩展端口可能看起来还有些不够清楚，但鉴于目前硬件板子只到 v1 版本，后续版本里会把这些细节理清。

加速度计与磁力计

磁力计



我们从磁力计芯片 MAG3110 开始

MAG3110 资料手册白皮书（英文）

以上页面的 Adafruit 短连接 <https://adafru.it/z4B>

我们可以用对应的 Arduino 库来与之通讯。

通过以下链接下载 Sparkfun 的 MAG3110 Arduino 库文件吧！

下载 Sparkfun MAG3110 展开板 Arduino 库(ZIP 文件)

以上页面的 Adafruit 短连接 <https://adafru.it/z4D>

附上安装库文件的教程（英文）

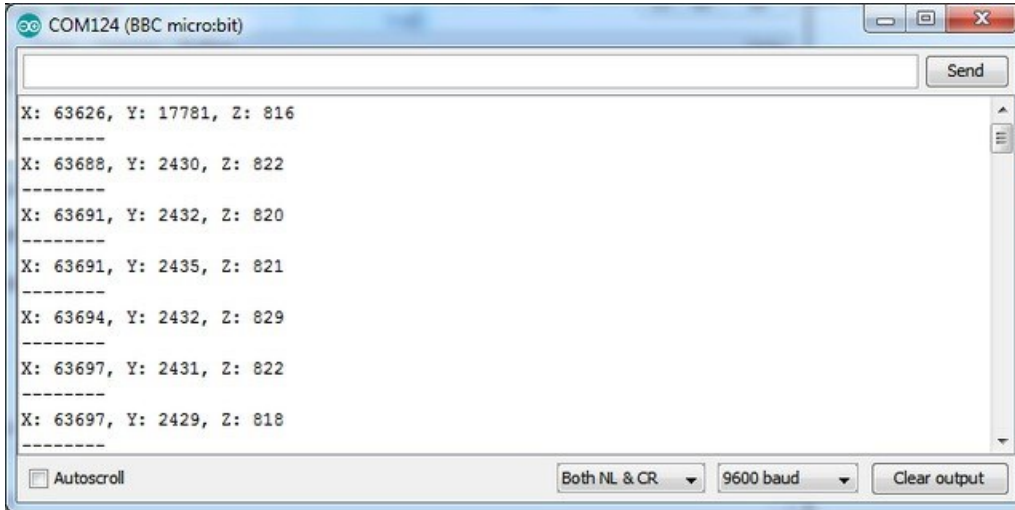
重启 Arduino IDE，上传库中自带的 [MAG3110 示例文件](#)，便可开始使用驱动磁力计。建议从下面列出的[基础范例](#)开始。

```
/* *****  
* SparkFun_MAG3110_Basic  
* Triple Axis Magnetometer Breakout - MAG3110  
* Hook Up Guide Example  
*  
* Utilizing Sparkfun's MAG3110 Library  
* A basic sketch that reads x y and z readings  
* from the MAG3110 sensor  
*  
* George B. on behalf of SparkFun Electronics  
* Created: Sep 22, 2016  
* Updated: n/a  
*  
* Development Environment Specifics:  
* Arduino 1.6.7  
*  
* Hardware Specifications:  
* SparkFun MAG3110  
* Bi-directional Logic Level Converter  
* Arduino Micro  
*  
* This code is beerware; if you see me (or any other SparkFun employee) at the  
* local, and you've found our code helpful, please buy us a round!  
* Distributed as-is; no warranty is given.  
* ***** /  
#include <SparkFun_MAG3110.h>  
MAG3110 mag = MAG3110(); //Instantiate MAG3110  
void setup() {  
  Serial.begin(9600);  
  
  mag.initialize(); //Initializes the mag sensor  
  mag.start(); //Puts the sensor in active mode  
}  
  
void loop() {  
  int x, y, z;  
  //Only read data when it's ready  
  if(mag.dataReady()) {  
    //Read the data  
    mag.readMag(&x, &y, &z);  
  
    Serial.print("X: ");  
    Serial.print(x);  
    Serial.print(", Y: ");
```

```
Serial.print(y);
Serial.print(", Z: ");
Serial.println(z);

Serial.println("-----");
}
}
```

把这段代码上传到八卦单机版，便可以在串口监视器看到类似下图的原始数据输出。



需要注意的是，在这个示例中，内置的磁力计并未经过校准，所以 X, Y, Z 的值会变来变去，但当你扭动、翻转八卦单机板的时候，读数值会随之上下波动。（所以磁力计在使用时都必须校准）

加速度计

除了磁力计，八卦单机板也板载了一块三轴加速度计。

可以用 MMA8653 库文件来与之通讯

Adafruit MMA8653 Arduino 库文件 (ZIP 文件)

以上页面的 Adafruit 短连接 <https://adafru.it/z5a>

普通地下载并安装库文件即可。

之后，让我们（重启 IDE）示例文件走起！


```

/*
 * MMA845XQ test code
 * (C) 2012 Akafugu Corporation
 *
 * This program is free software; you can redistribute it and/or modify it under the
 * terms of the GNU General Public License as published by the Free Software
 * Foundation; either version 2 of the License, or (at your option) any later
 * version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT ANY
 * WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
 * PARTICULAR PURPOSE. See the GNU General Public License for more details.
 *
 */

#include "Wire.h"
#include "MMA8653.h"

MMA8653 accel;

void setup() {
  Serial.begin(9600);

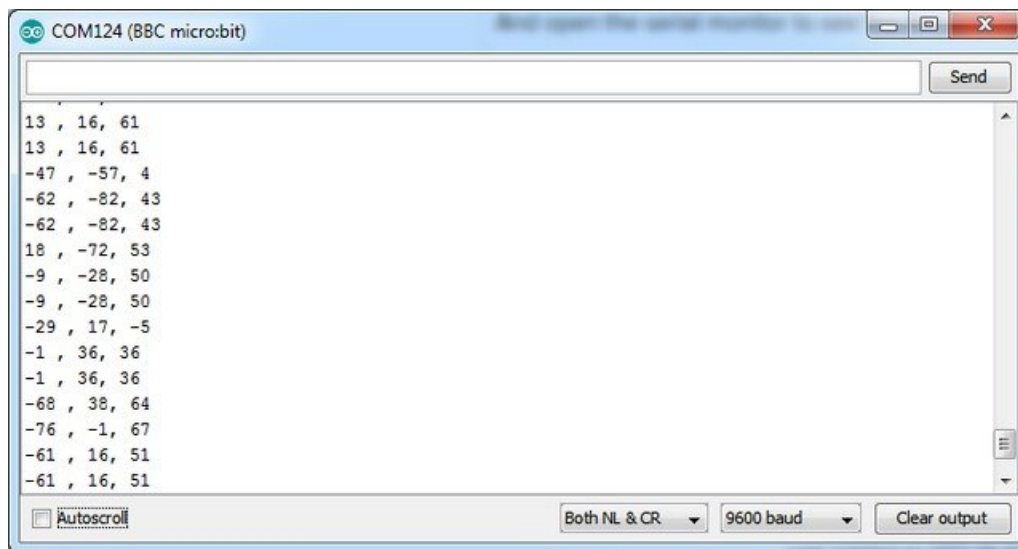
  Serial.println("microbit accel test");

  accel.begin(false, 2); // 8-bit mode, 2g range
}

void loop() { accel.update();
  Serial.print(accel.getX()); Serial.print(", ");
  Serial.print(accel.getY()); Serial.print(", ");
  Serial.println(accel.getZ());
  delay(100);
}

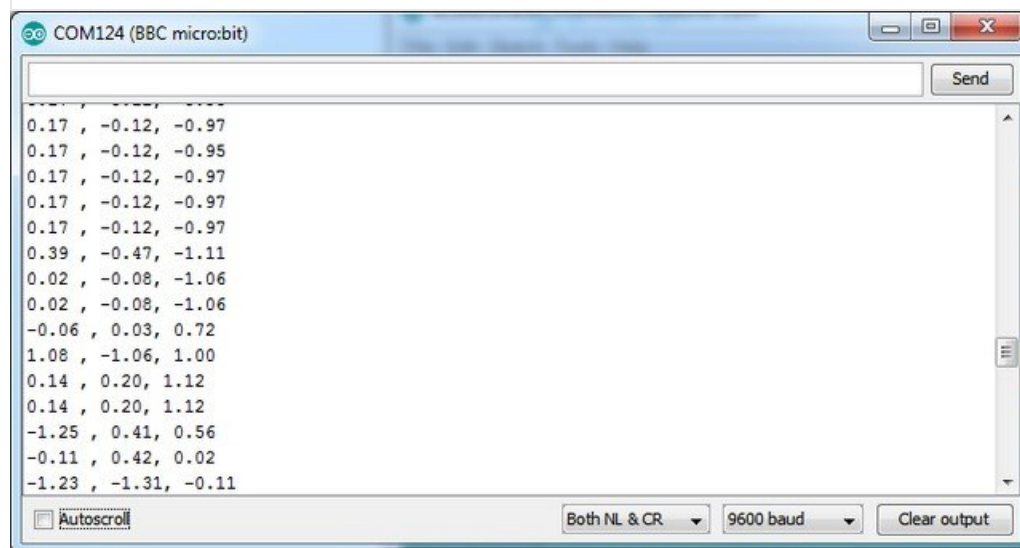
```

打开串口监视器，就能看到 X、Y、Z 轴加速度原始数据。



这个库有些老旧，也不太完整，所以当下暂时只能使用 8 位模式输出数据。如果你需要把输出的原始数据转换成加速度值，可以用这段代码：

```
void loop() {  
  accel.update();  
  Serial.print((float)accel.getX() * 0.0156); Serial.print(", ");  
  Serial.print((float)accel.getY() * 0.0156); Serial.print(", ");  
  Serial.println((float)accel.getZ() * 0.0156);  
  
  delay(100);  
}
```



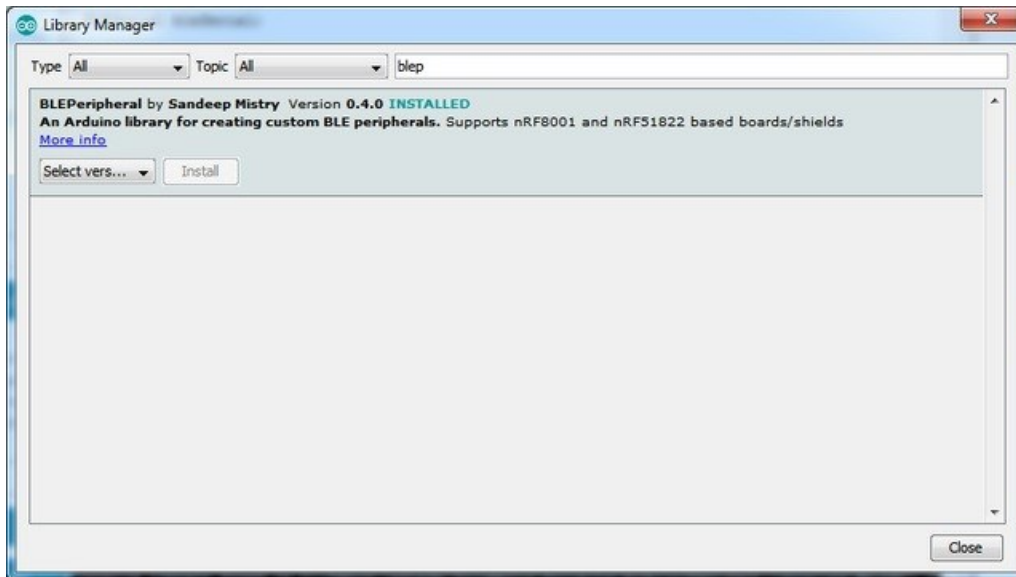
Adafruit 库文件

准备好玩更复杂的项目了吗？或许你需要个一揽子包圆的助手库文件帮你处理内部温度传感器、LED 矩阵和蓝牙连接。

所以我们写了这个封装库把这一切管理起来。除此之外，还需要安装一些额外的支持库，比如 Adafruit GFX。

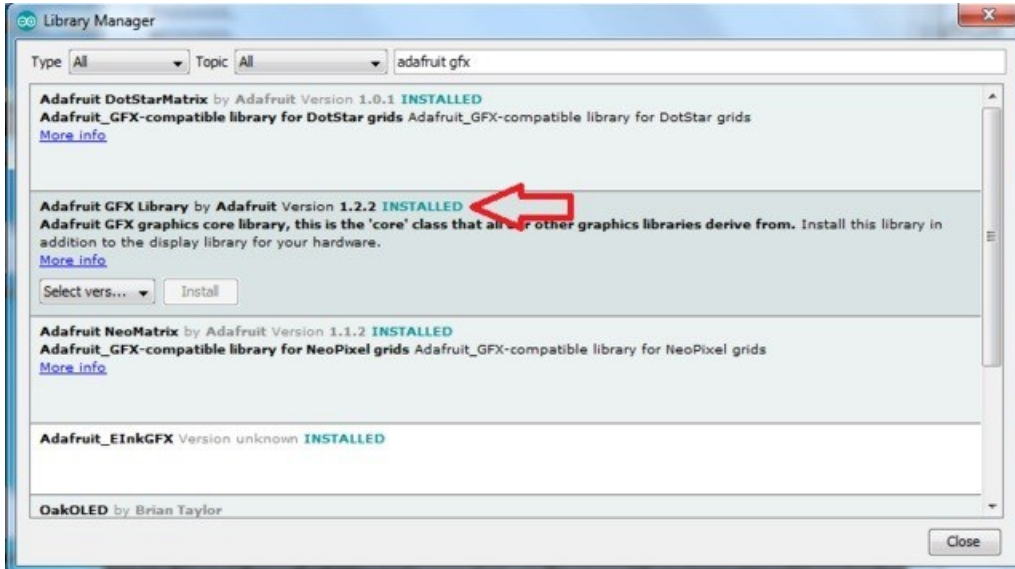
下载 BLE 蓝牙外设库(BLE Peripheral)

BLE 蓝牙外设库(BLE Peripheral)可以在 项目 > 加载库 > 管理库 的库管理器里搜索并安装，关键字 BLEP。



下载 Adafruit 图像特效库(Adafruit GFX)

依照同样方式，从 Arduino 库管理器里，下载安装 Adafruit 图像特效库(Adafruit GFX)，关键字 Adafruit GFX。



下载 Adafruit_Microbit 库

要使用 LED 矩阵或者蓝牙连接功能的话，可以从我们的 GitHub 下载 Adafruit_Microbit 库。既可以[直接访问我们的 GitHub 页面来下载](#)，也可以直接点击下面的链接下载 ZIP 文件。

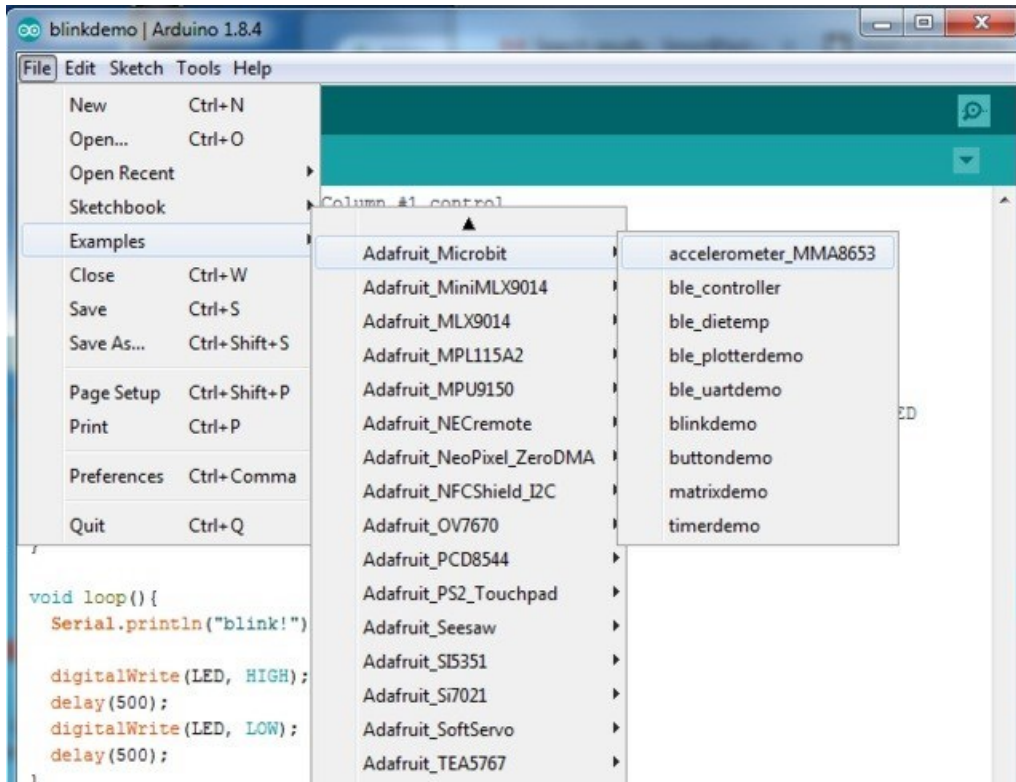
下载 Adafruit Microbit 库

以上页面的 [Adafruit 短连接 https://adafru.it/zqD](https://adafru.it/zqD)

把解压出的文件夹重命名为 Adafruit_Microbit，确认其中直接包含 Adafruit_Microbit.cpp 和 Adafruit_Microbit.h，并把 Adafruit_Microbit 文件夹复制到你的 Arduino 默认项目文件夹下的 libraries 子文件夹中。如果之前从未安装过附加库文件，可能需要手动创建 libraries 文件夹。文件拷贝完成后，重启 Arduino IDE 即可生效。

如果你在安装 Arduino 库时遇到困难，也可以参考我们的教程（英文）：
<http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use>

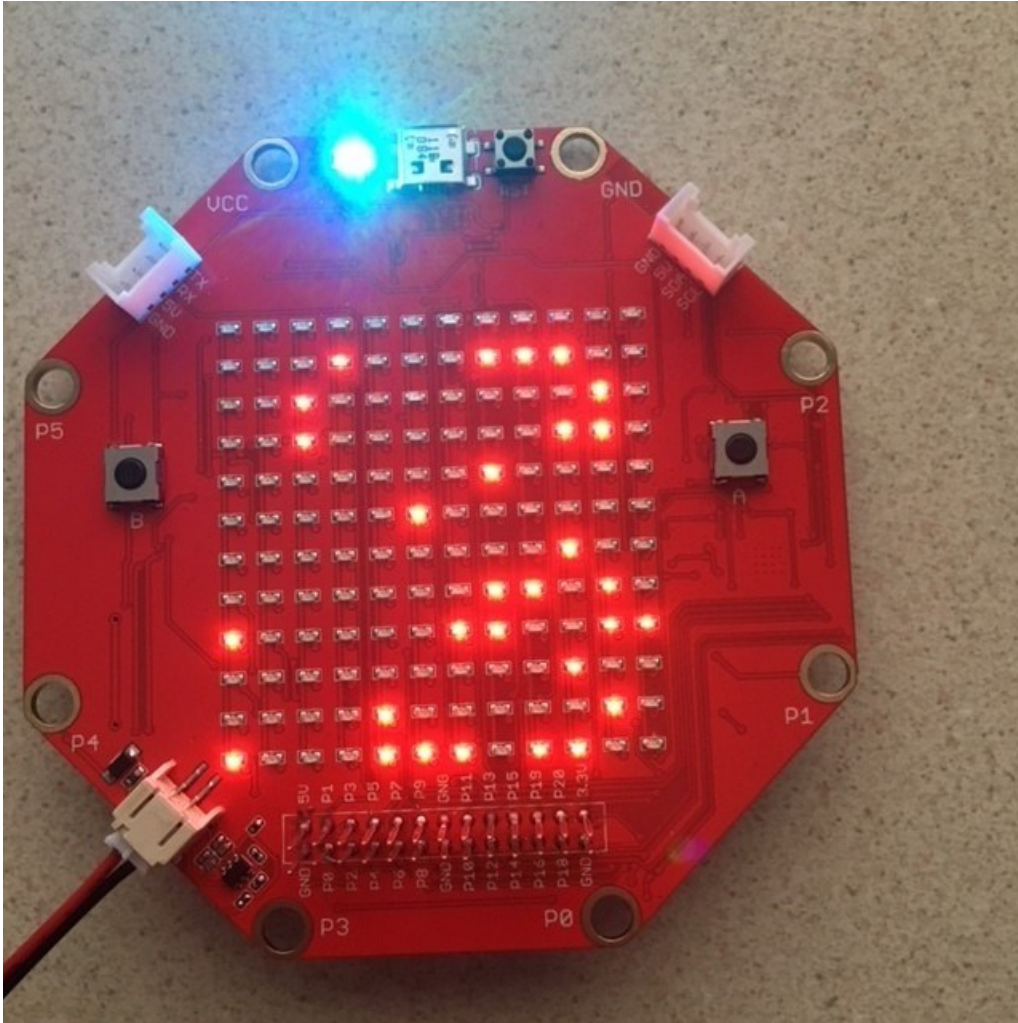
重启 Arduino IDE 之后，便可在 文件 > 示例 > Adafruit_Microbit 目录下找到库文件的自带示例。



LED 矩阵

八卦单机板与 micro:bit 的 LED 矩阵有一些比较重要的不同之处。

最明显的区别是八卦单机板的 LED 矩阵是 $12 \times 12 = 144$ 个而非 $5 \times 5 = 25$ 个。对于非罗马字符集来说， 5×5 点阵太小了，所以我们引入更大的 LED 矩阵来支持更多字符集。即便你不需要更多字符集，更大的显示空间总是意外地好用。八卦单机板的库与 GFX 库搭配相得益彰，能最大程度地利用起大矩阵在 GFX 图元显示上的优势。为了展示这一优势，我们在库文件中集成了 Conway 的生命游戏，该游戏的原代码在（英文页面）：<https://learn.adafruit.com/minicommodore-pet-with-charlieplexed-led-matrix/software>。



Conway 生命游戏展示

在八卦单片机板上，为了节约 MCU 的资源，LED 矩阵并不需要 micro:bit 一样用 MCU 来管理和控制。八卦单片机板使用 Holtek HT1632 芯片来管理 LED 矩阵，MCU 只需告诉 HT1632 每一个 LED 需要呈现的状态即可，能够省出更多计算资源来执行更加关键的代码。

八卦单片机板的库文件已经把一切幕后工作集成到位，可以直接调用 GFX 图元。以下示例便是一个与 micro:bit 稍有不同的 LED 矩阵指示按钮状态的示例。

```

#include <sinobit.h>

Sinobit matrix = Sinobit();

void setup() {
  matrix.begin();
  delay(100);
  matrix.clearScreen();
  matrix.setTextWrap(false);

  pinMode(SINOBIT_BUTTON_A, INPUT);
  pinMode(SINOBIT_BUTTON_B, INPUT);
}

void loop() {
  if (!digitalRead(SINOBIT_BUTTON_A)) {
    matrix.drawLine(7, 1, 11, 5, 1);
    matrix.drawLine(7, 10, 11, 6, 1);
  } else {
    matrix.drawLine(7, 1, 11, 5, 0);
    matrix.drawLine(7, 10, 11, 6, 0);
  }

  if (!digitalRead(SINOBIT_BUTTON_B)) {
    matrix.drawLine(4, 1, 0, 5, 1);
    matrix.drawLine(4, 10, 0, 6, 1);
  } else {
    matrix.drawLine(4, 1, 0, 5, 0);
    matrix.drawLine(4, 10, 0, 6, 0);
  }

  matrix.writeScreen();
}

```

模拟输入

八卦单片机板和 micro:bit 使用同样的 MCU，所以它们的模拟输入功能是相同的。（详见文中 GPIO 部分）

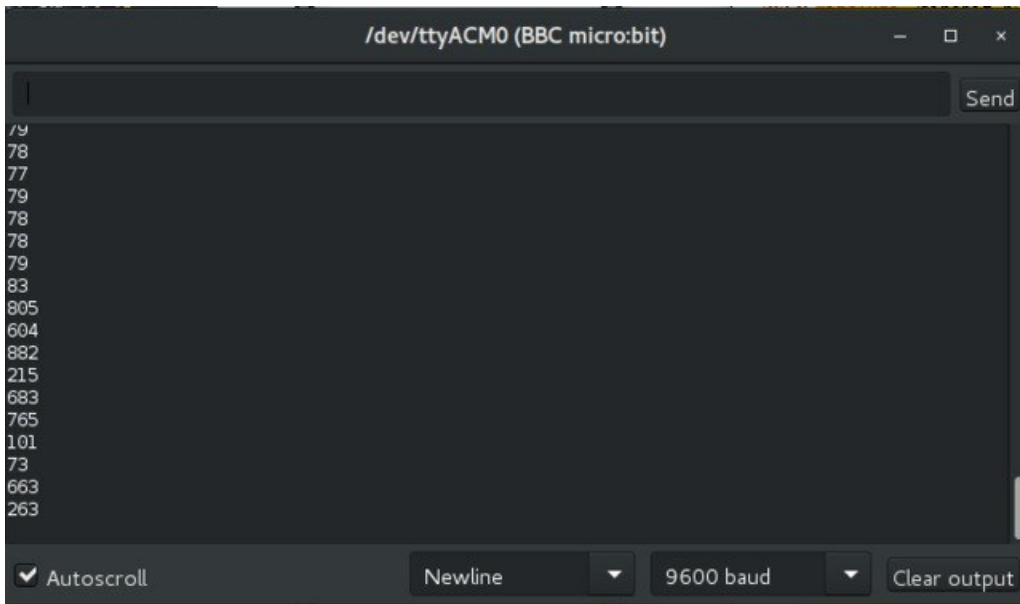
```

int analogPin = 0;
int val = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  delay(1000);
  val = analogRead(analogPin);
  Serial.println(val);
}

```



大尺寸的 LED 矩阵赋予了我们更多现实图像的潜能，比如模拟输入的直方图。


```

#include <sinobit.h>

Sinobit matrix = Sinobit();

int analogPin = SINOBIT_PAD_P0;
int readings[12];

void setup()
{
  matrix.begin();
  delay(100);
  matrix.clearScreen();
  for (int i = 0; i < 12; i++) {
    readings[i] = 0;
  }
}

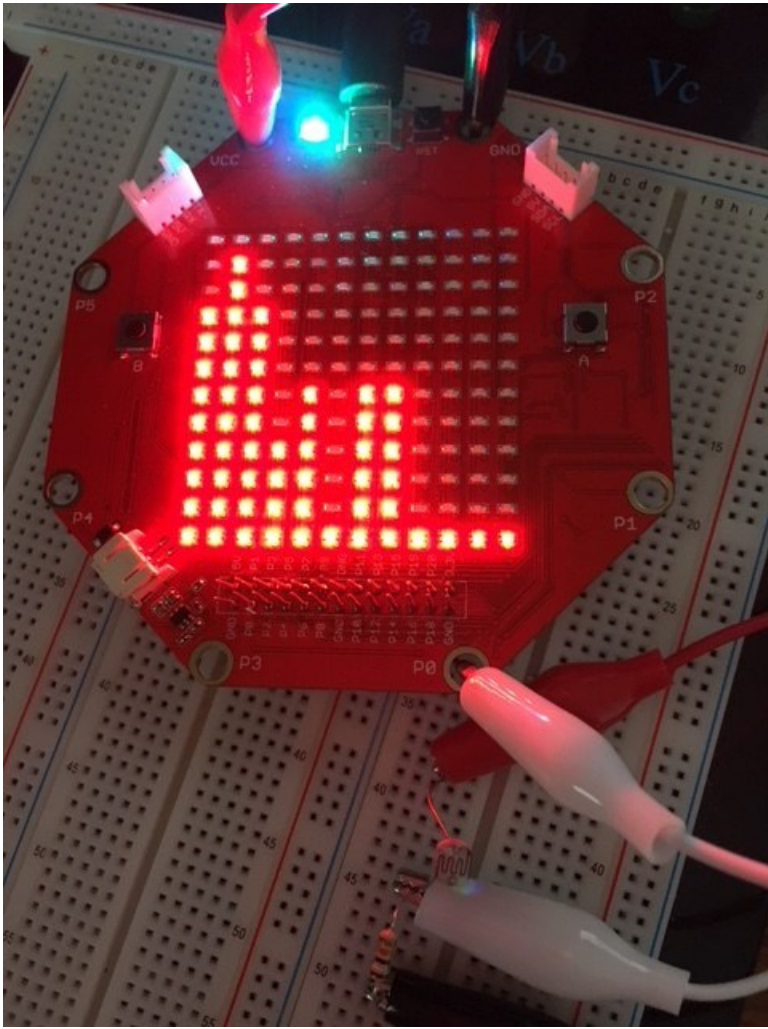
void loop()
{
  delay(1000);

  for (int i = 0; i < 11; i++) {
    readings[i] = readings[i + 1];
  }
  // experience showed that my photocell never got to a max reading
  // which is why I use 12 as the upper limit instead of 11 as expected
  readings[11] = map(analogRead(analogPin), 0, 1023, 0, 12);

  for (int i = 0; i < 12; i++) {
    if (readings[i] == 0) {
      matrix.drawLine(i, 11, i, 0, 0);
    } else if (readings[i] == 11) {
      matrix.drawLine(i, 11, i, 0, 1);
    } else {
      matrix.drawLine(i, 11, i, 12 - readings[i], 1);
      matrix.drawLine(i, 11 - readings[i], i, 0, 0);
    }
  }

  matrix.writeScreen();
}

```



蓝牙

八卦单片机板和 micro:bit 使用同样的 MCU，所以它们的蓝牙通讯能力也是相同的。你可以用八卦单片机板和其他八卦单片机板、micro:bit、Calliope 或是你的智能手机、平板通讯。以下示例在手机一侧使用 Adafruit Bluefruit APP。



蓝牙 UART 串口

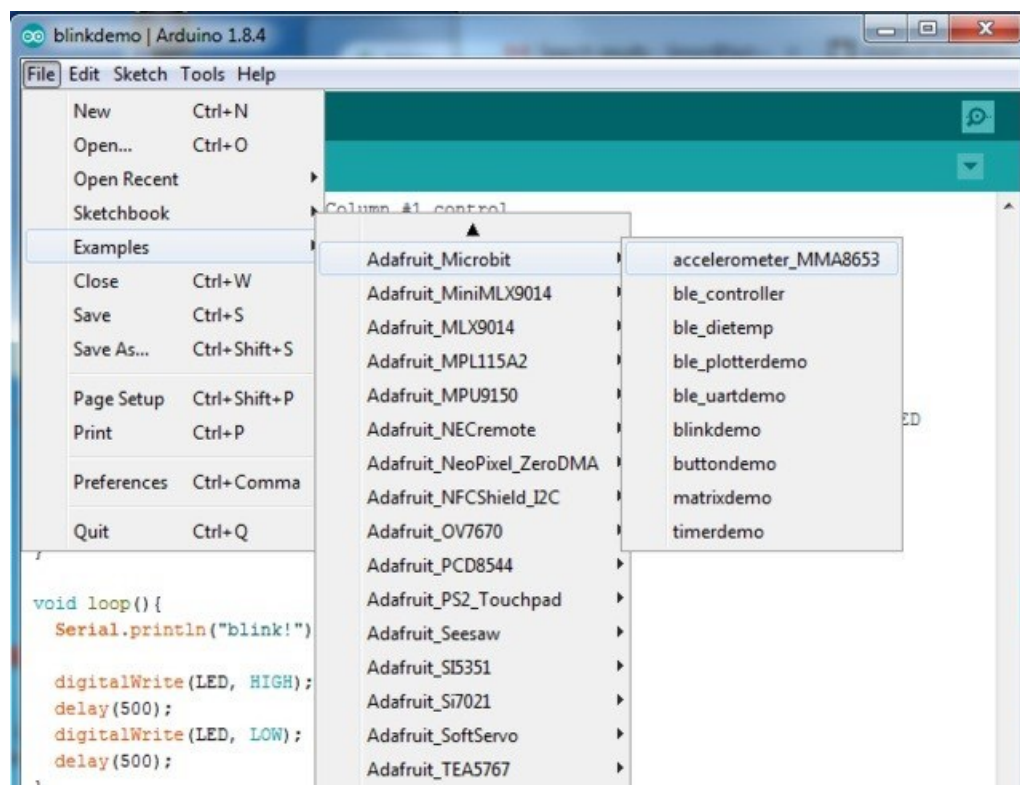
八卦单片机的主芯片集成了 BLE 低功耗蓝牙，助你实现各种炫酷的无线项目。

Adafruit BlueFruit Connect APP 简单易用，可在 [iOS App store](#) 或 [Android 应用市场](#) 下载到。

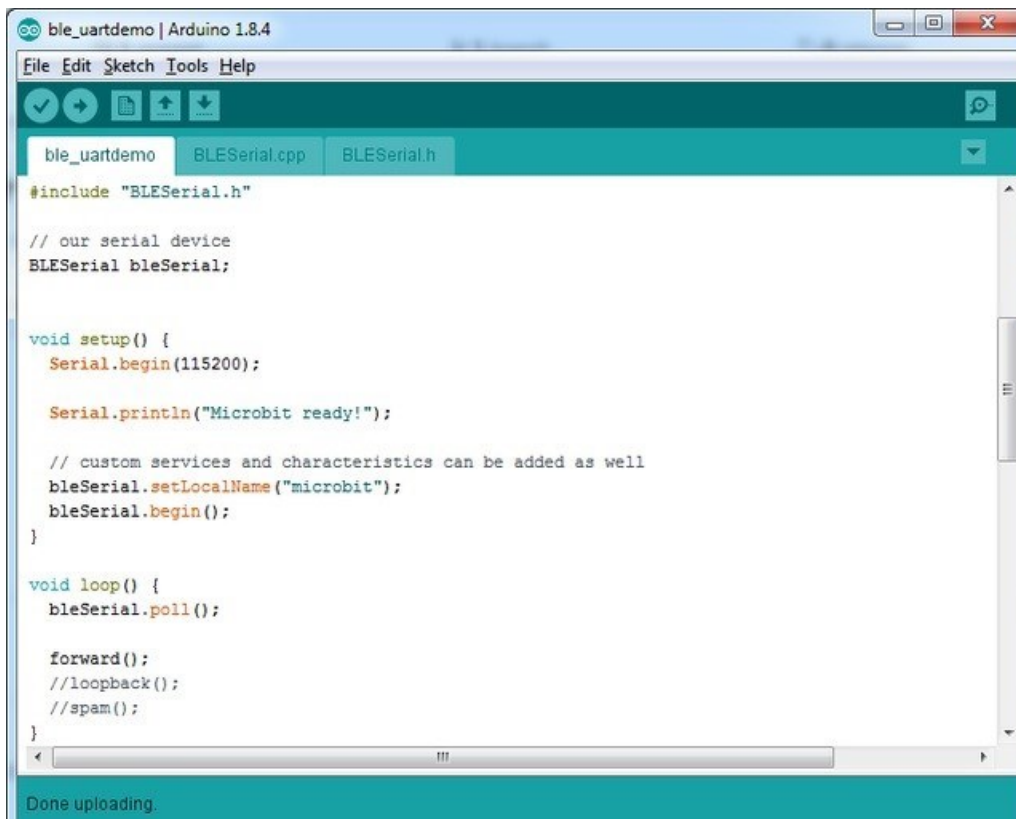
如果在连接中遇到困难，请通读 [APP 连接与使用教程（英文）](#) 以便更好理解其工作原理。

安装库文件与示例代码

首先，需要安装 Adafruit 辅助库文件，随后在示例文件找到我们的 BLE 示例文件。



加载蓝牙 UART 示例文件 (ble_uartdemo)



找到以下 3 行:

```
forward();
//loopback();
//spam();
```

修改为:

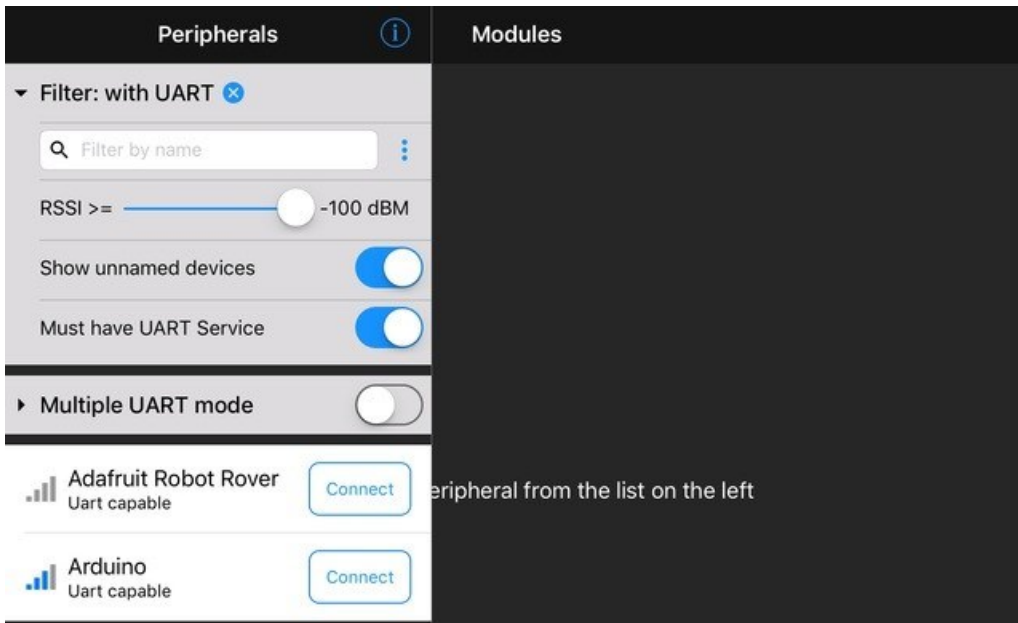
```
//forward();
loopback();
spam();
```

如此这般，你的八卦单片机板便会每秒外发一次数据，便于测试。把代码上传到板上，就可以开始了。

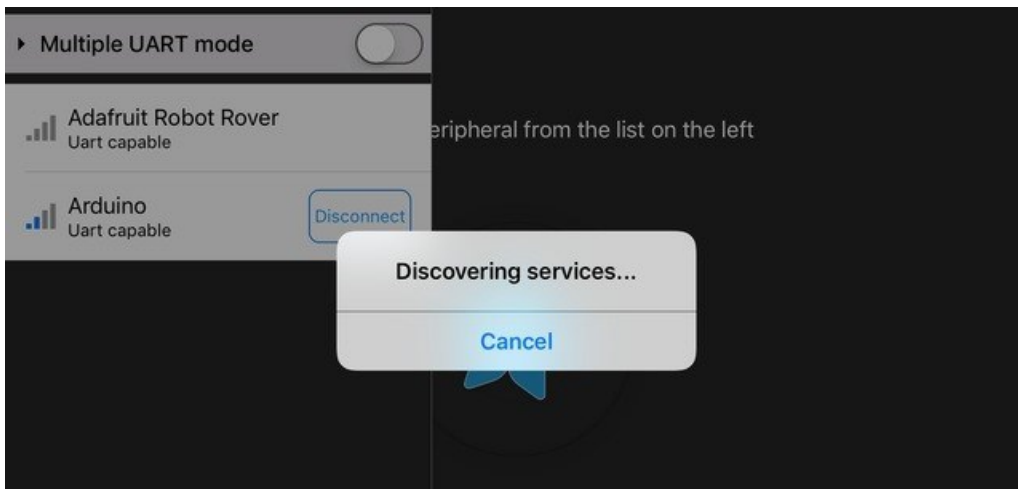
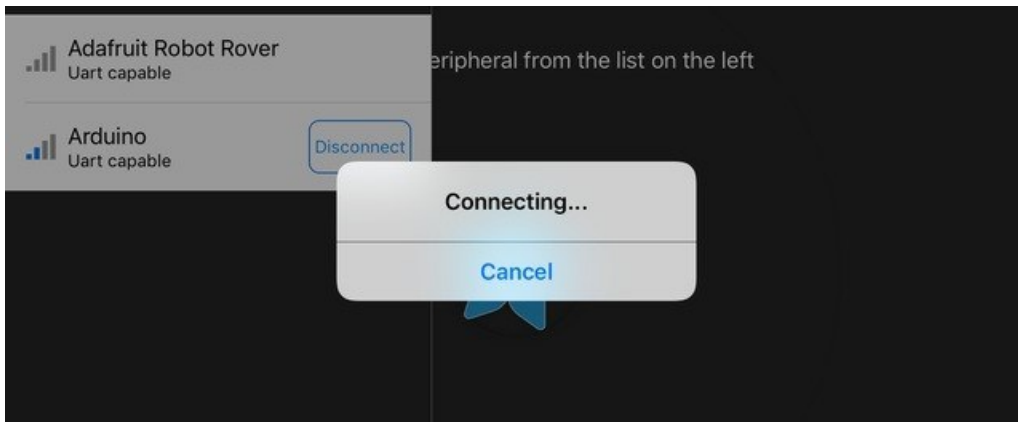
蓝牙连接

代码上传到八卦单片机板后，请打开 **Adafruit Bluefruit Connect APP**。

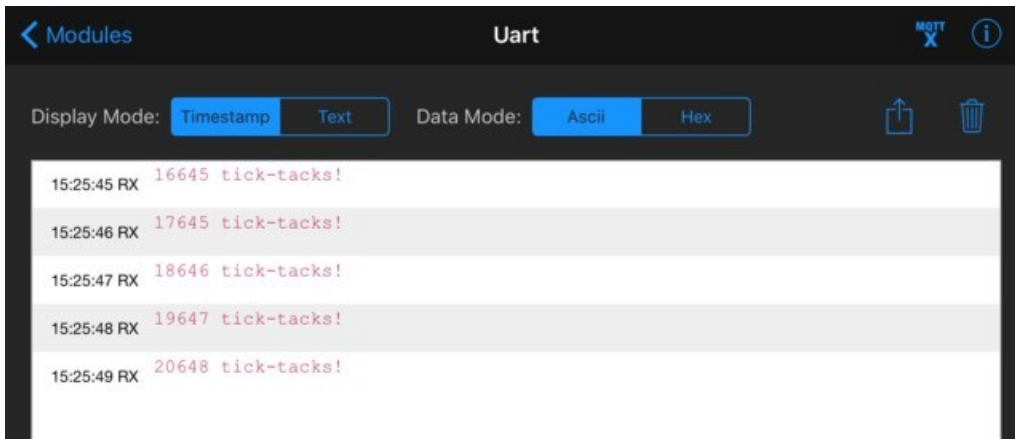
从左侧下方的菜单选择 **micro:bit**，它有时候也会显示为 **UART** 或者 **Arduino**。



点击 Connect 按钮来连接。



之后从模块列表中选择 UART，进入 Timestamp 时间戳模式，如果一切正常的话，你会每秒收到一个消息。



测试正常后，回到项目代码，并把之前的三行修改回：

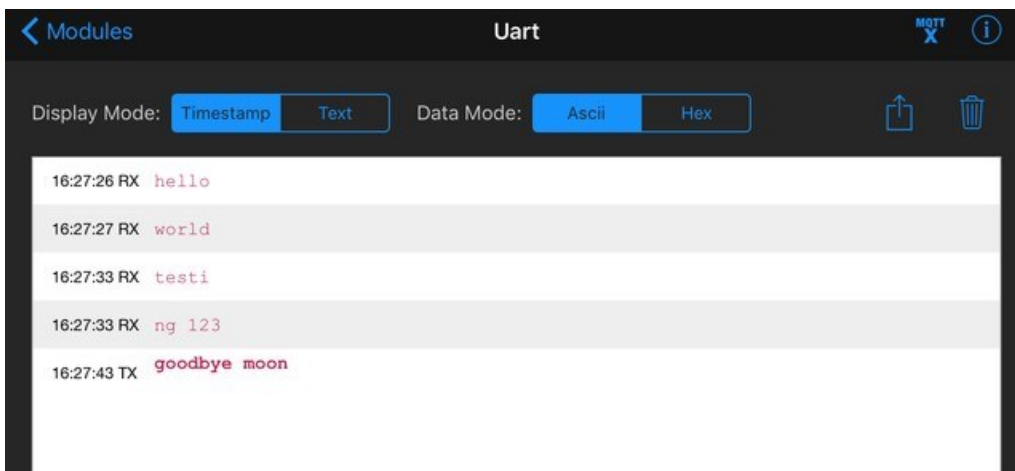
```
forward();  
//loopback();  
//spam();
```

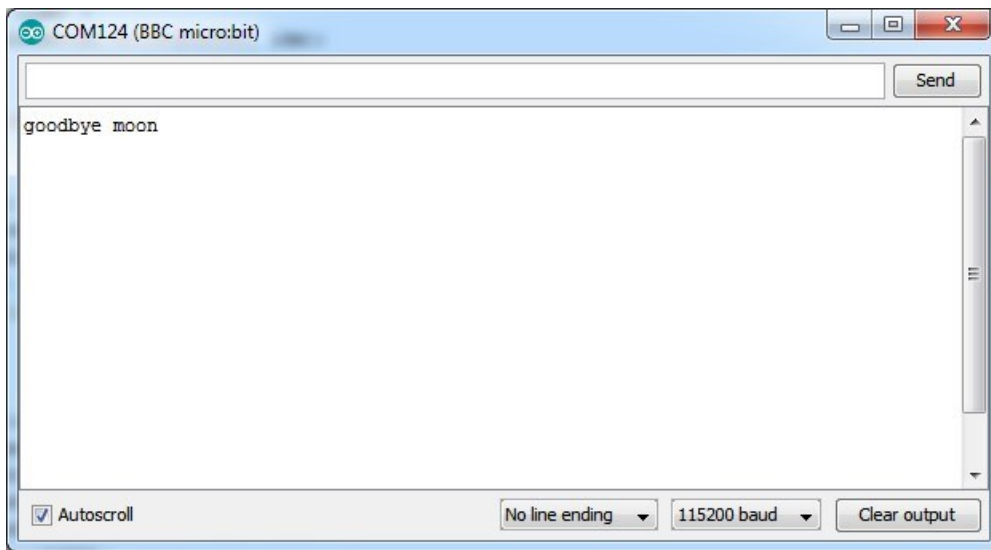
重新上传程序，手机端 APP 会提示你已断开连接，只需回到设备列表点击从回到设备扫描界面即可。

在 APP 主界面打开串口控制台（serial console），波特率设置为 115200。

之后再（连接八卦单机板）进入 UART 模式，便可以从 APP 一侧向 micro:bit 所连接的电脑上的串口监视器发送、接收信息。请注意 micro:bit 的 UART 接口比较古怪：如果通过串口监视器‘一次性’发送超过 10 个字符，可能会导致 micro:bit 卡死。

（译者注：本段内容针对 micro:bit，尚未在八卦单机板 Sino:bit 测试）





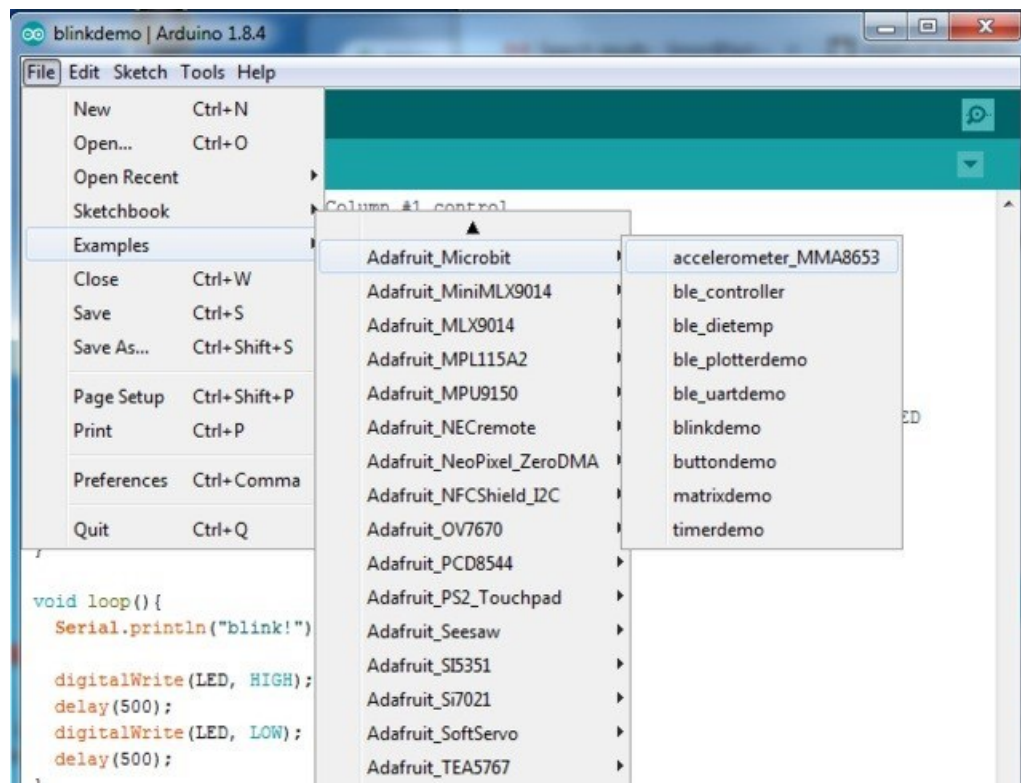
这个功能跑通之后，不妨试一试我们的控制器（ble_controller）示例哦。

蓝牙绘图仪

Bluefruit APP 内置了绘图仪功能，在前文所述的 UART 示例调试好之后，便可以用这个功能来把八卦单片机板收集到的数据转化为可视化的图形！

安装库与示例代码

如果安装库遇到问题，请回到文档前部，查阅“下载 Adafruit_Microbit 库”一节。



加载蓝牙绘图仪示例文件 (ble_plotterdemo)


```
ble_plotterdemo | Arduino 1.8.4
File Edit Sketch Tools Help
ble_plotterdemo BLESerial.cpp BLESerial.h

void setup() {
  Serial.begin(115200);

  Serial.println("Plotter demo ready!");

  accel.begin(false, 2); // 8-bit mode, 2g range

  // custom services and characteristics can be added as well
  bleSerial.setLocalName("microbit UART");
  bleSerial.begin();
}

void loop() {
  bleSerial.poll();

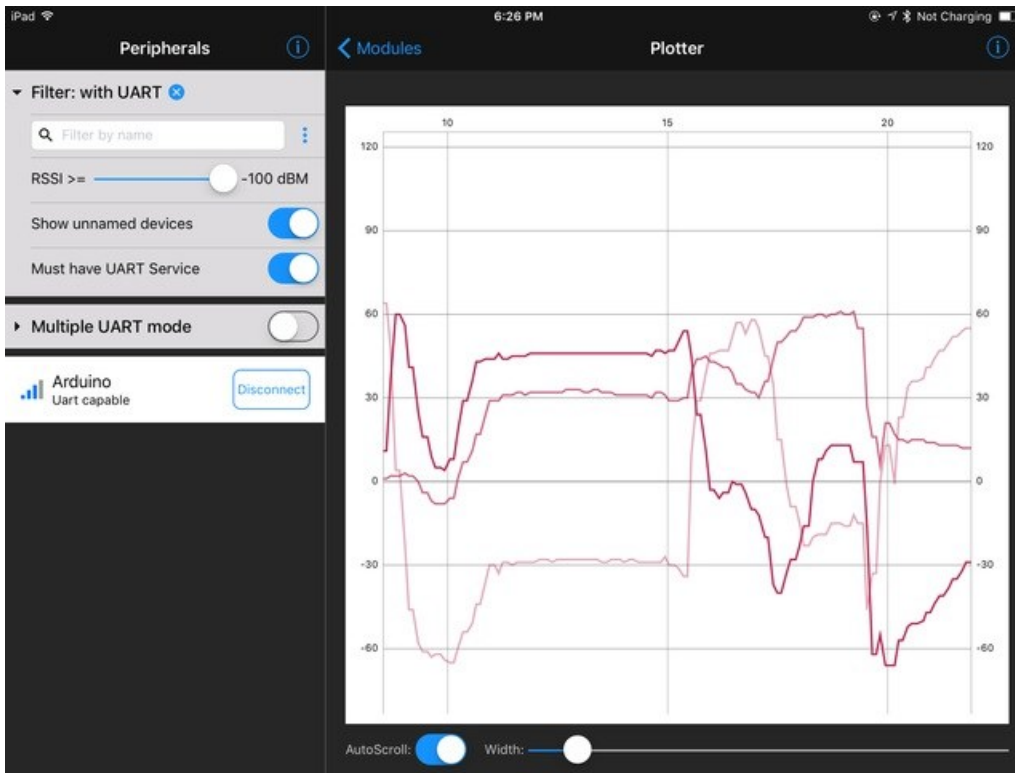
  accel.update();

  // print the data on serial port
  Serial.print(accel.getX()); Serial.print(", ");
  Serial.print(accel.getY()); Serial.print(", ");
  Serial.println(accel.getZ());

  // send it over bluetooth
  bleSerial.print(accel.getX()); bleSerial.print(",");
}
```

之后在手机端 APP 一侧选择绘图仪模块（Plotter），之后便可以看到 X,Y 和 Z 的值出现在屏幕，并开始滚动。

你可以用任意需要的数据进行绘图，只需要调用 `bleSerial.print()` 函数，以逗号分隔的形式发送数据，并在每一个数据集尾部调用 `bleSerial.println()` 函数，便可以每一组数据集绘制出图像。



所以如果你需要绘制 X,Y,Z 轴加速度的合并向量 $\sqrt{x^2 + y^2 + z^2}$ ，可以用下面这段代码：

```

void loop() {
  bleSerial.poll();

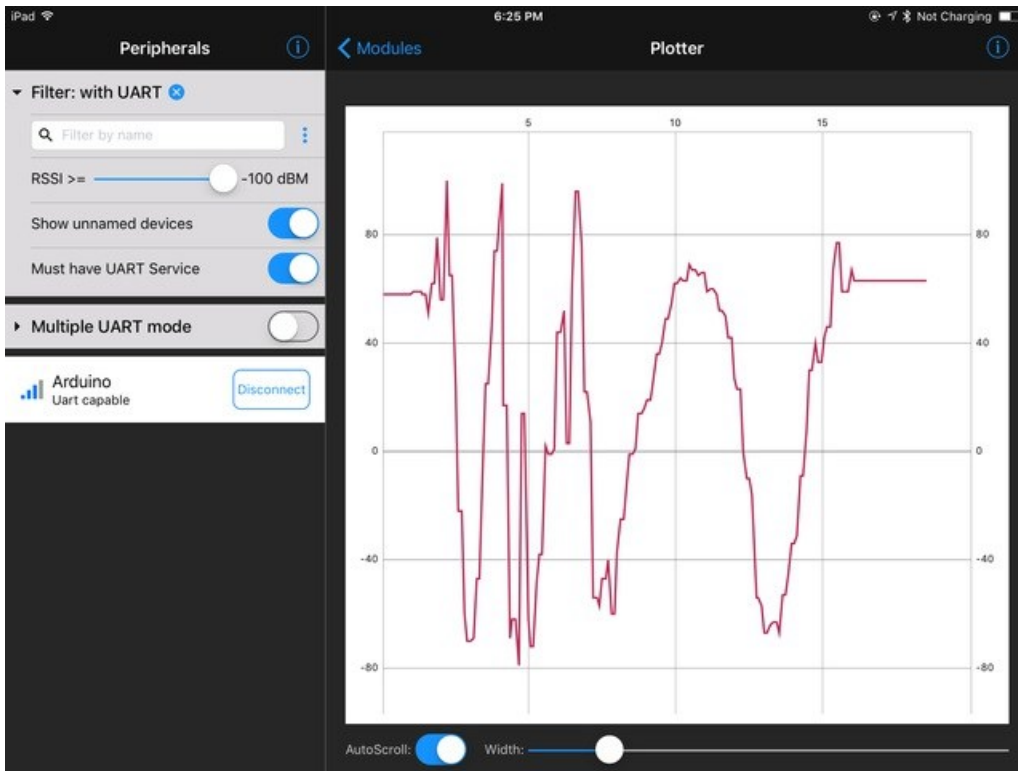
  accel.update();
  // print the data on serial port
  Serial.print(accel.getX()); Serial.print(", ");
  Serial.print(accel.getY()); Serial.print(", ");
  Serial.println(accel.getZ());

  float vector = (accel.getX() * accel.getX()) + (accel.getY() * accel.getY()) + (accel.getZ() * accel.getZ());
  vector = sqrt(vector);

  // send it over bluetooth
  bleSerial.println(vector);

  delay(100);
}

```



蓝牙控制器

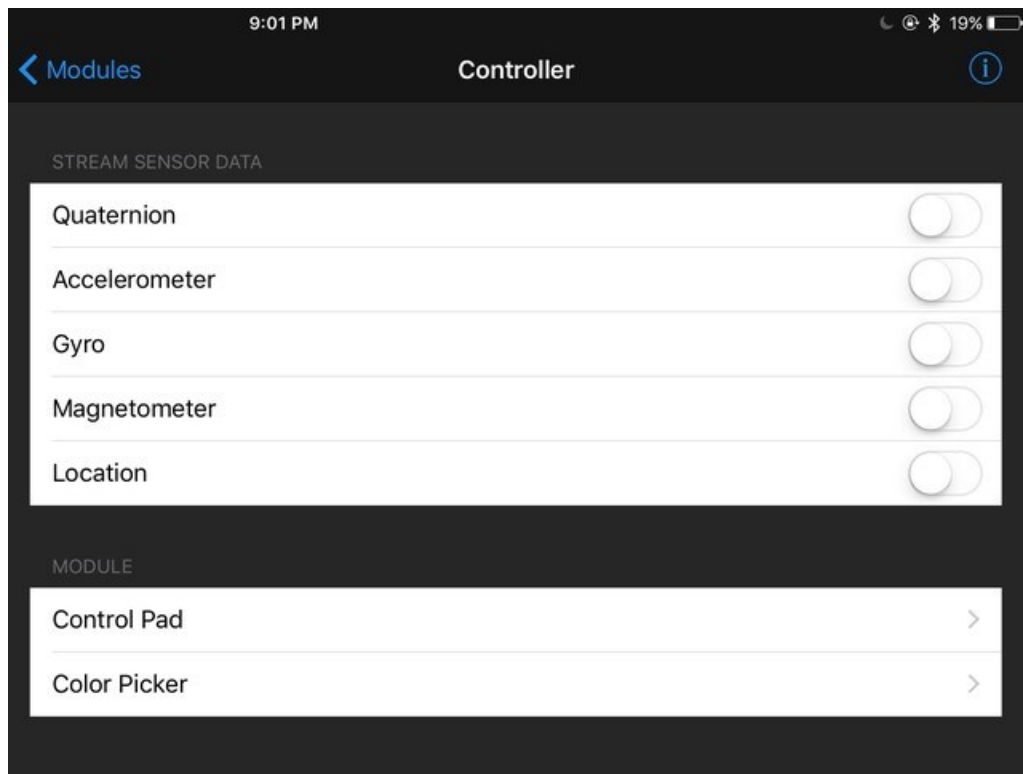


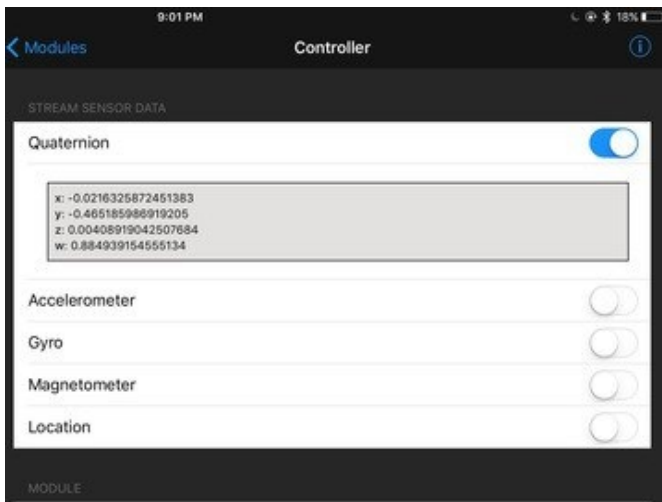
对于控制类项目，可以使用 APP 端的控制器功能，它集成了大量实用的交互功能，能让你下一个 LED 或机器人项目易如反掌。

安装库与示例代码

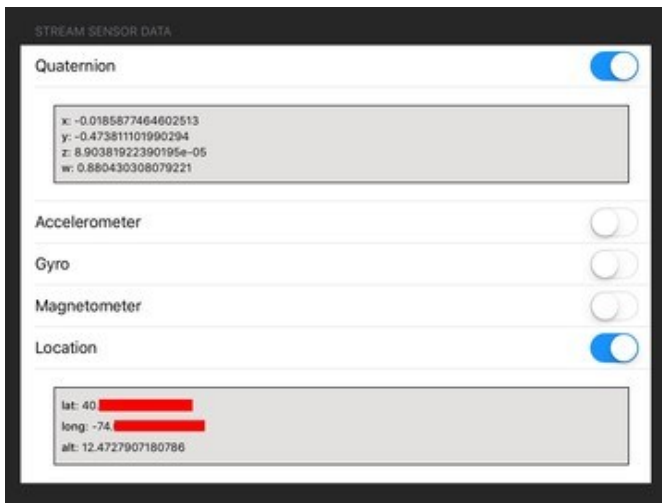
如果安装库遇到问题，请回到文档前部，查阅“下载 Adafruit_Microbit 库”一节。

打开蓝牙控制器示例文件（ble_controllerdemo），加载到八卦单片机板，APP 连接蓝牙。

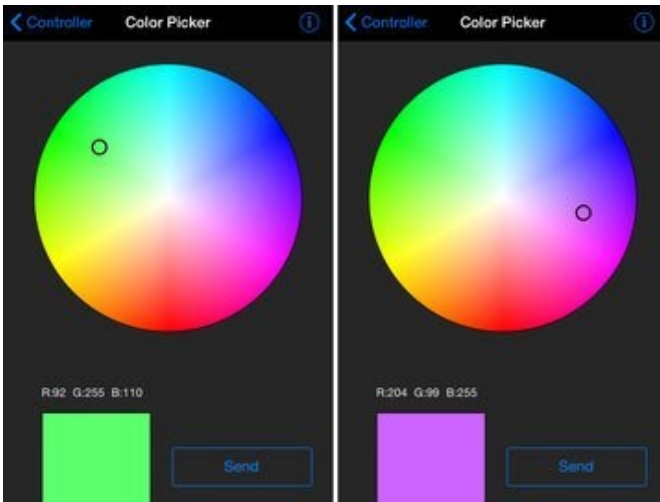




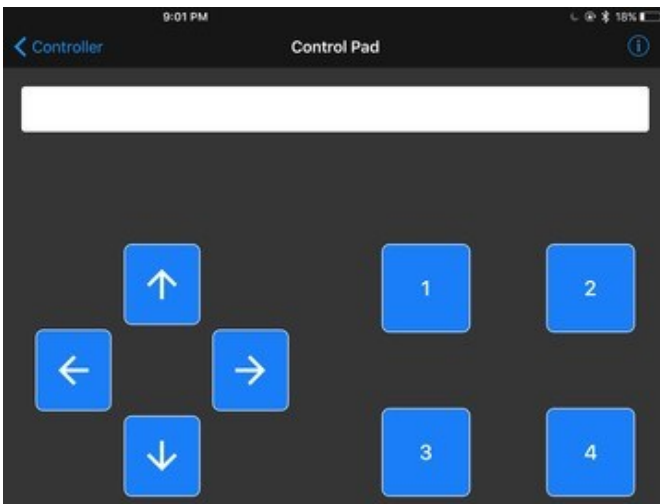
前五个选项允许你把平板或手机的数据串流到八卦单片机板，比如方位角，GPS 定位等等。既可以单个使用，也可以五个数据全开。



底部的 2 个模块（取色器和控制手柄）也可以随时调用，点击打开对应的界面：



取色器模块可以用来在色轮上任意取色，并把对应的 24 位颜色值通过 BLE 发送到八卦单片机板。



控制手柄界面有 8 个虚拟按键，每次按下和释放按键都会向八卦单片机板发送信号。

如果有数据从八卦单片机板一侧**发回**，则会显示在手柄界面上方的文本框。

你可以在连接了八卦单片机板的电脑上开启串口监视器来查看收到的数据。

```
COM10 (BBC micro:bit)
Controller demo ready!
Quat 0.35 0.01 0.00 0.94
Quat 0.35 0.01 -0.00 0.94
Quat 0.35 0.01 -0.00 0.94
Quat 0.36 0.01 -0.00 0.93
Quat 0.35 0.01 -0.00 0.93
Quat 0.17 -0.12 -0.04 0.98
Quat -0.04 0.03 -0.12 0.99
Quat 0.41 0.50 -0.03 0.76
Quat 0.57 0.17 0.17 0.78
Quat 0.19 -0.29 -0.04 0.94
Quat 0.38 0.10 -0.02 0.92
Quat 0.42 0.11 -0.01 0.90
Button 5 pressed
Button 5 released
Button 3 pressed
Button 3 released
Button 4 pressed
Button 4 released
Button 7 pressed
Button 7 released
RGB #9647FF
RGB #54FF45
RGB #FF3B46
Autoscroll
No line ending
115200 baud
Clear output
```



记录温度并发送到 Adafruit IO

通过蓝牙发送数据到手机并且绘图，并用手机对八卦单片机板进行控制十分方便，然而如果需要长期数据存储，或是远程控制，要怎么办呢？

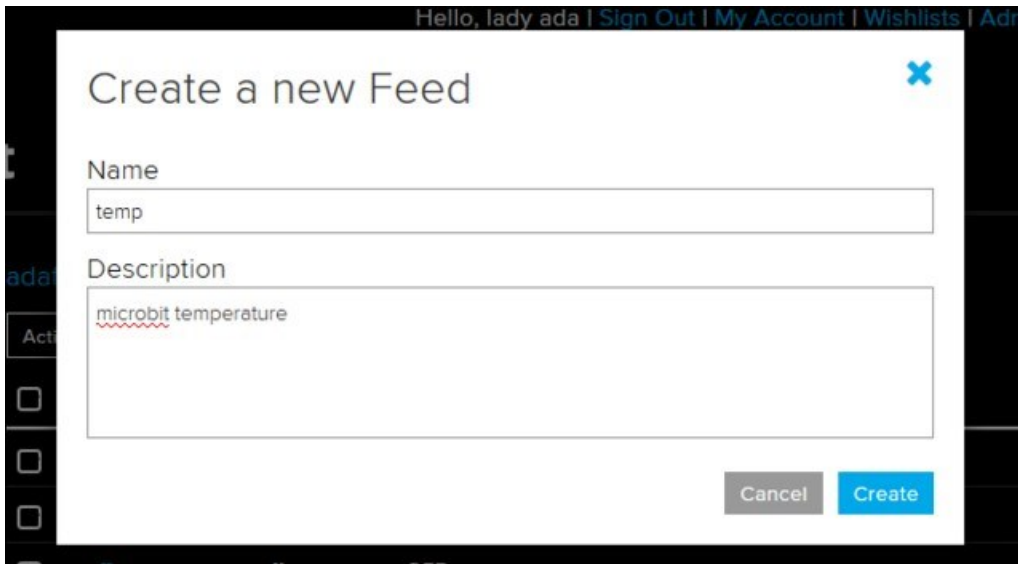
简单！用 Adafruit IO 来创建图标和控制台，和 BlueFruit APP 一样免费哦！

在继续之前，请[注册 Adafruit IO 账户](#)

关于 Adafruit IO 的细节不在此详述，请查阅相关教程。

创建八卦单片机板温度数据源

我们需要“空间”来存放温度，在这里我们叫它 temp。



温度记录项目文件

打开 BLE 核心温度示例（ble_dietemp）

这段代码会读取芯片内部的温度计读数。它并不完全精准，但会在温度上升、下降时对应体现到数值的上升和下降。这个示例不需要外部硬件，所以用起来比较友好。

```
#define OVERSAMPLE 50

void loop() {
  bleSerial.poll();

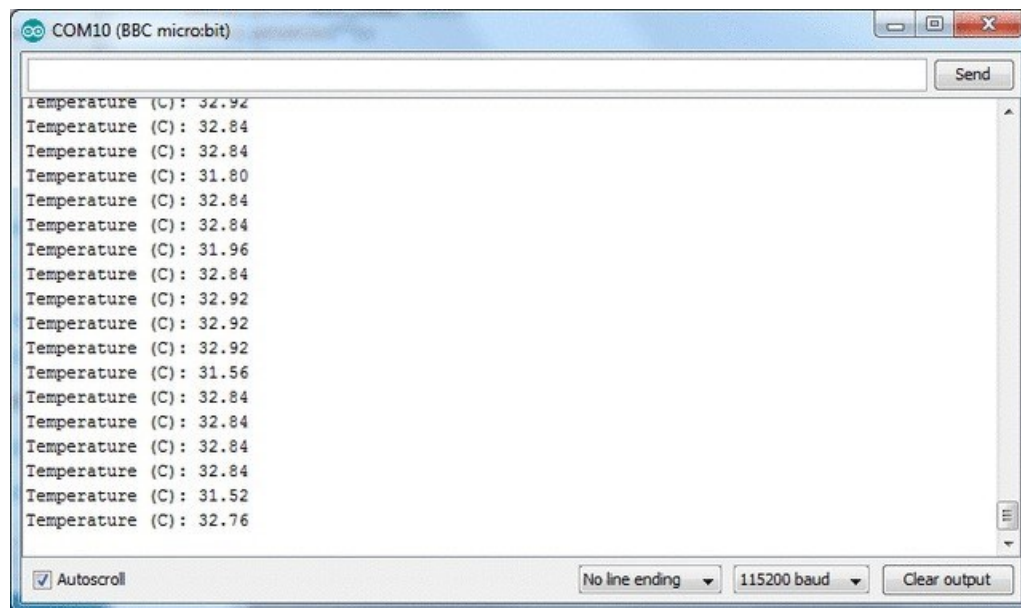
  // Take 'OVERSAMPLES' measurements and average them!
  float avgtemp = 0;
  for (int i = 0; i < OVERSAMPLE; i++) {
    int32_t temp;
    do {
      temp = temperature_data_get();
    } while (temp == 0); // re run until we get valid data
    avgtemp += temp;
    delay(1);
  }
  avgtemp /= OVERSAMPLE;

  Serial.print("Temperature (C): ");
  Serial.println(avgtemp/100.0F); // Float value since temp is in 0.25°C steps

  // Send just the raw reading over bluetooth
  bleSerial.println(avgtemp/100.0F);
}
```

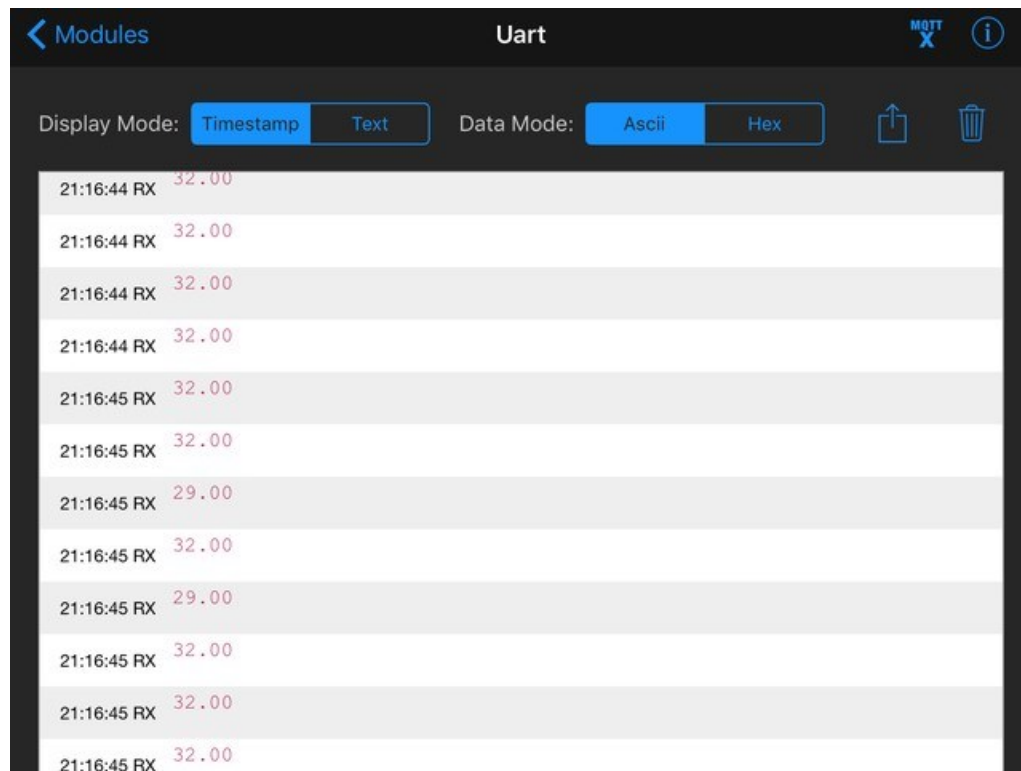
请注意在本示例中，先连续读取 50 个温度值并取平均值，然后在每两次数据上报之间等待 5000 毫秒（5 秒）。Adafruit IO 会限定可以上传和存储的数据量，所以需要把程序执行的节奏放慢。

上传示例文件，并打开串口监视器，便可以在此复查读取到的温度。

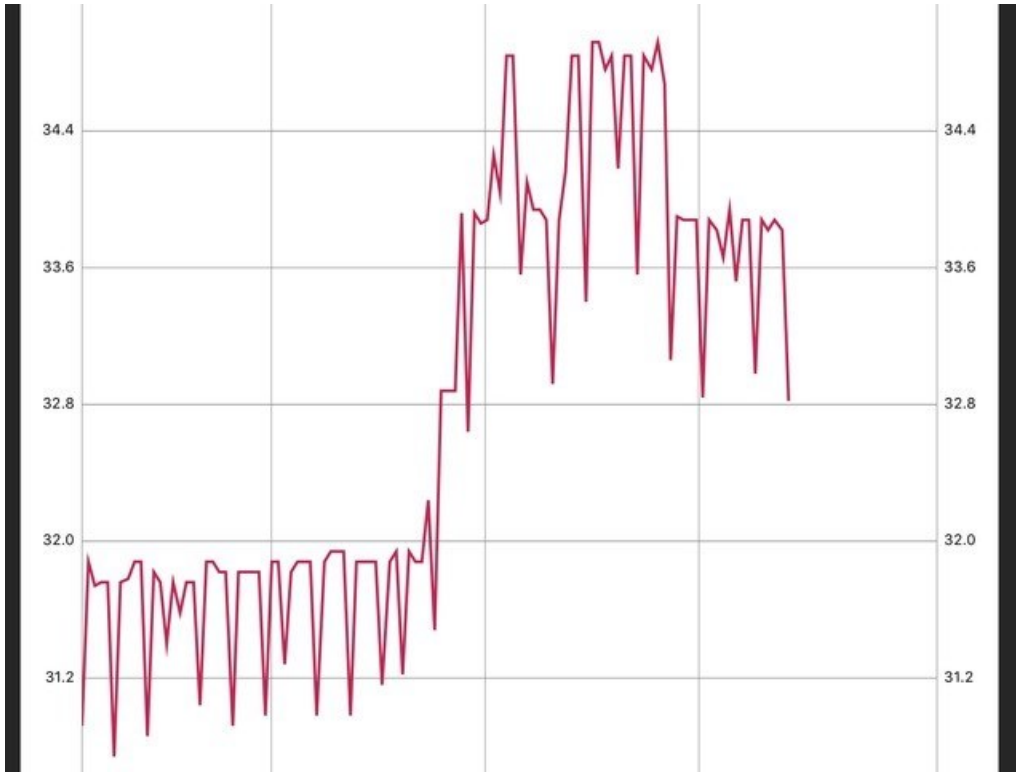


测试 UART 模式

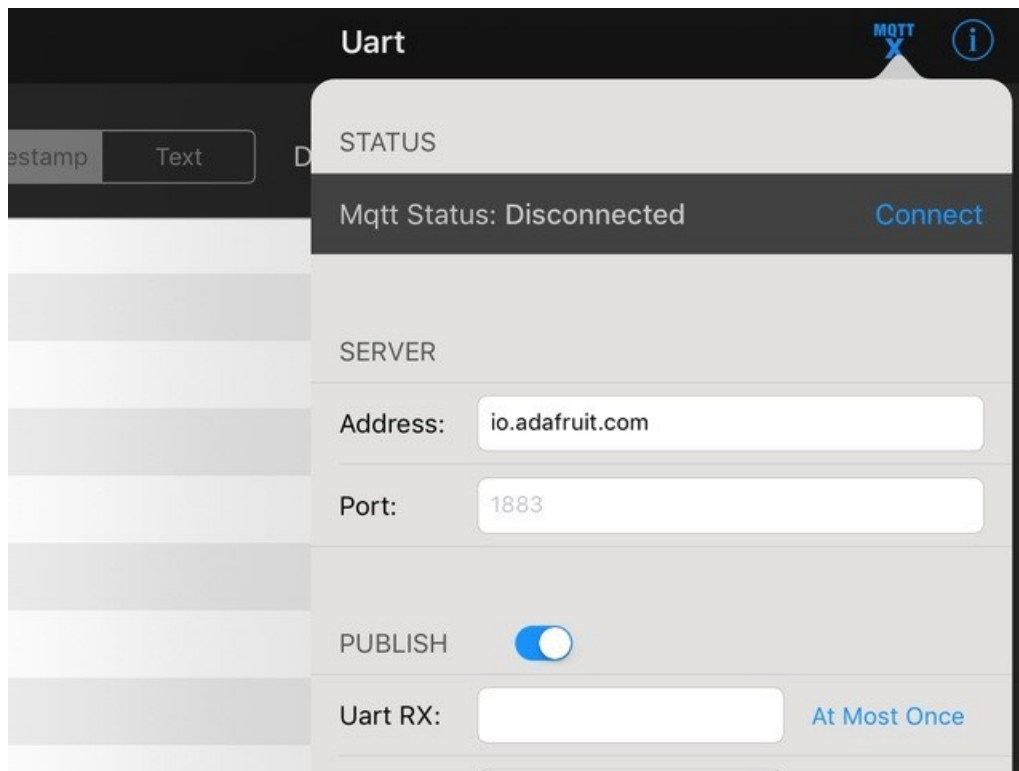
用 Adafruit Bluefruit Connect APP 连接到你的八卦单片机板，选择 UART 模式，应当会看到数据持续慢慢传来。



你可以利用这些数据进行绘图，但注意度数并不精准。如果你用台灯烘烤 nRF51 主芯片，温度读数会随之缓缓上升。

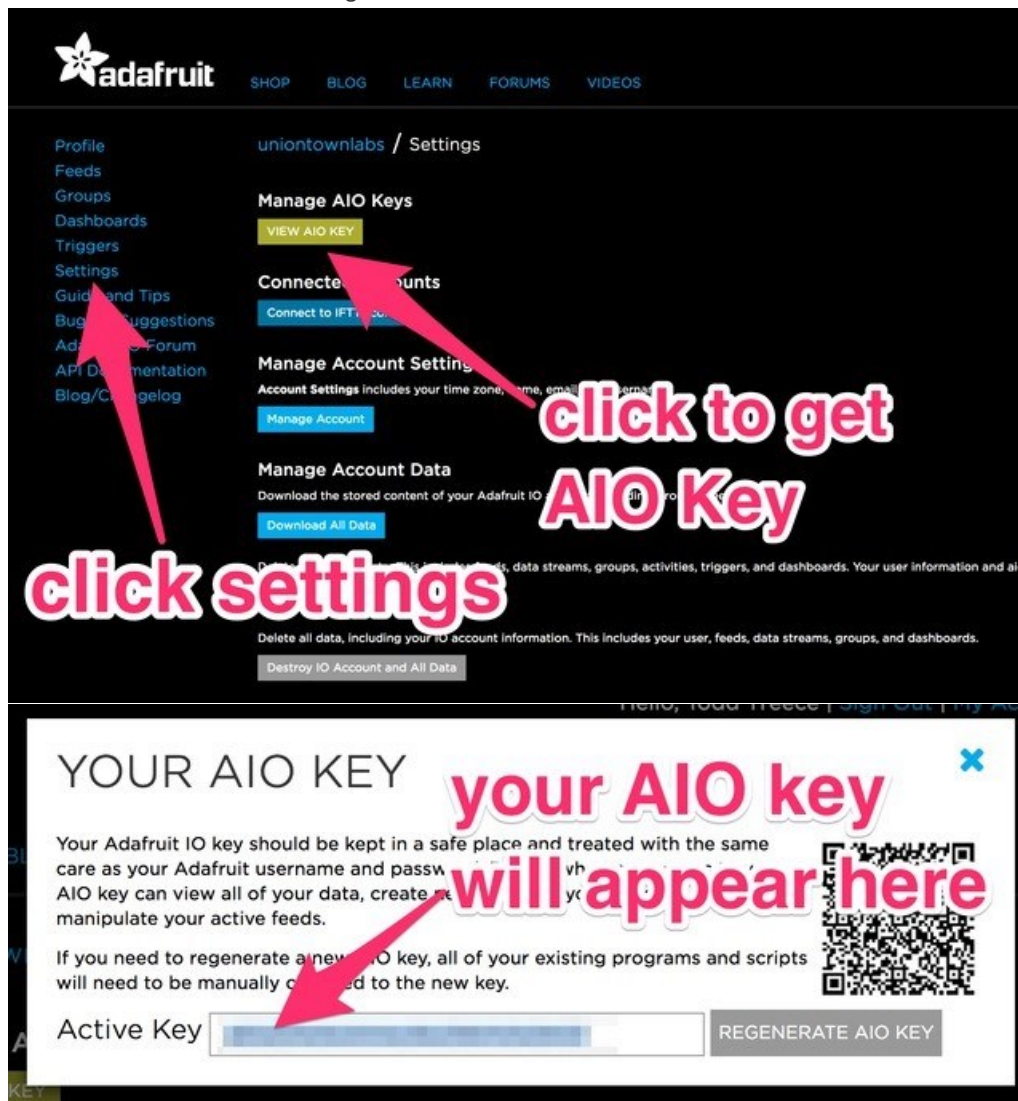


之后回到手机 APP，进入 UART 模式并点击右上角的 MQTT 按钮，

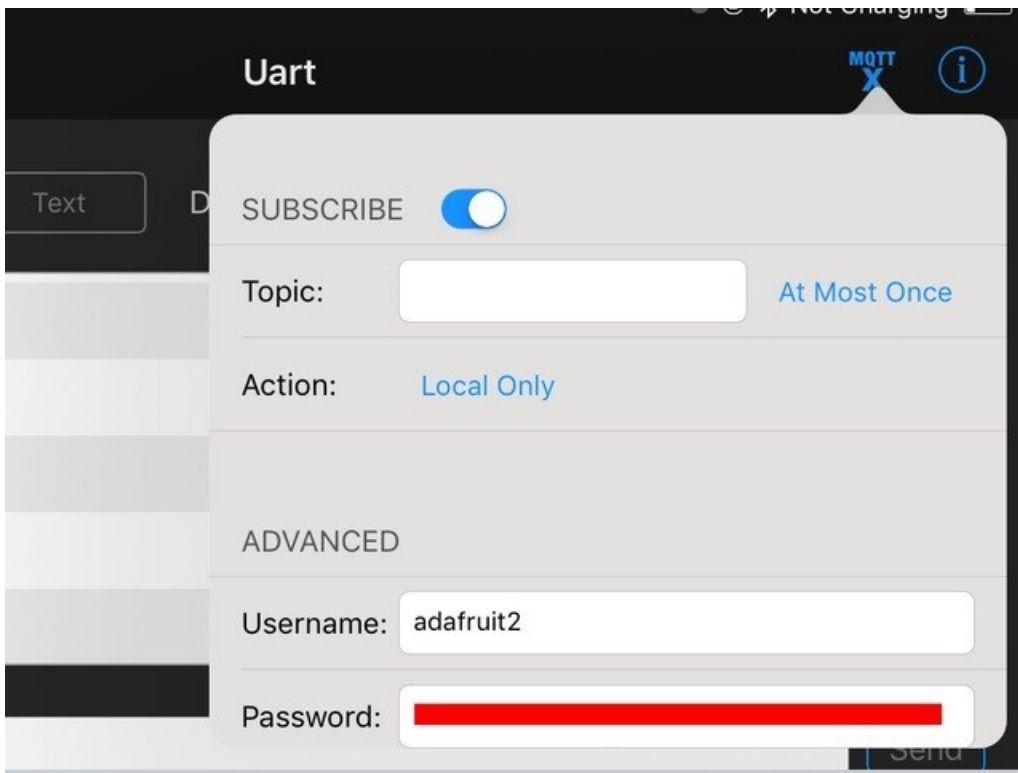


MQTT 服务器和端口都会被预设到 Adafruit IO。

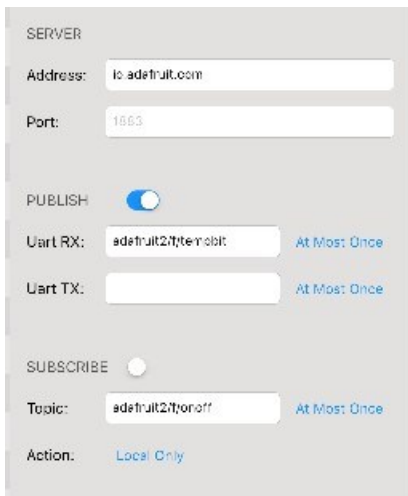
在你的 Adafruit IO 设置 (Settings) 页面找到你的 AIO 密钥。



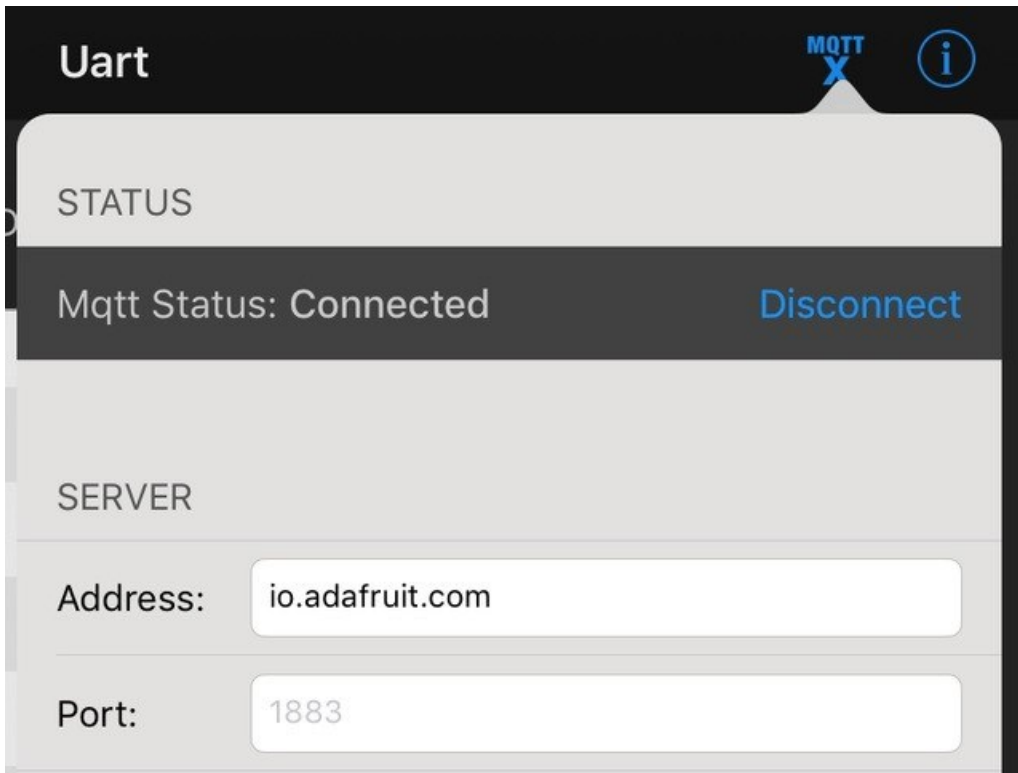
并把对应的密钥填写到 APP 的这个页面 (即便这里显示的是密码 Password, 也要填写 AIO 密钥)



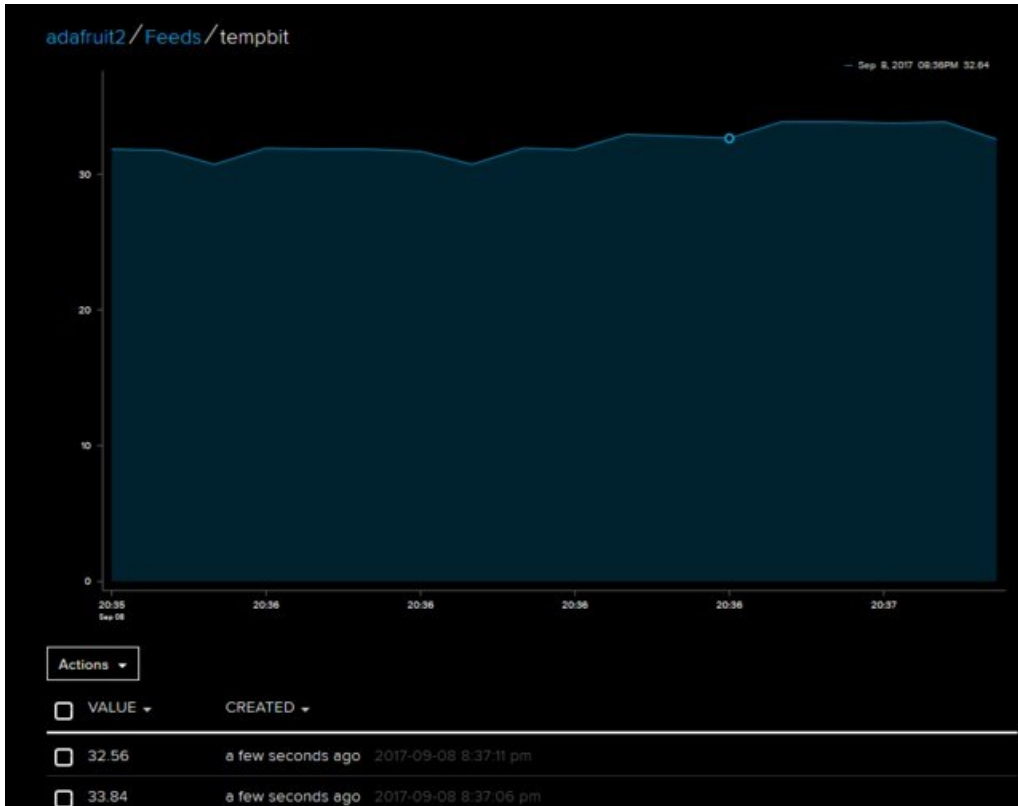
之后输入你的数据源名称 用户名/f/temp，用户名就是你在 Adafruit IO 的用户名，/f/后面的名称是刚才在页面添加项目时的名称，将这个字段填写到发布（Publish）下的 UART RX 标签即可。（订阅（Subscribe）数据目前有一些小 bug，所以我们把同样的数据源先放在这里。）



最后点击顶部的连接（Connect）按钮，



几分钟后刷新你的 Adafruit IO 数据源页面，就能看到你的数据源源不断地串流到网站啦。



好耶！接下来你可以创建公共的数据仪表盘并分享（英文）。

HALP!!!

If you're having issues, you may want to check Sandeep's installation guide for the nRF5x package which may have more details (in case there are updates)

Some people reported that their microbit did not have a softdevice on it already (which seems odd but is possible!) You can try installing this hex file which will use MakeCode to install a softdevice. Just drag it onto the MICROBIT disk drive

microbit-adv.hex

<https://adafru.it/zwf>

There's also instructions here on how to manually install a softdevice or in the off chance you want softdevice 130 instead of the 'standard' s110 (see [Flashing SoftDevice](#))

Last Updated: 2017-12-06 10:15:34 PM UTC