

2016



TI SimpleLink SensorTag 开发套件教程

手把手教你玩转 TI SimpleLink SensorTag 开发

摘要

新 SensorTag 现在包括 10 个低功耗 MEMS 传感器，采用小尺寸红色封装。它可以使用 DevPack 实现扩展，易于添加您自己的传感器或传动器。通过蓝牙智能连接到云，在 3 分钟内在线获得传感器数据。SensorTag 开箱即可使用，带有 iOS 和 Android 应用，无需编程经验即可开始操作。



电子发烧友开发板试用平台
最专业的电子论坛
2016/7/4





目录

| | |
|---|-----|
| 产品介绍 | 3 |
| 内容简介 | 7 |
| 开箱介绍 | 8 |
| TI SimpleLink SensorTag 套件初亮相 | 8 |
| 上电试机篇 | 18 |
| TI SimpleLink SensorTag 套件的全部功能简介 | 18 |
| TI SimpleLink SensorTag 上的磁传感器/蜂鸣器 | 29 |
| TI SimpleLink SensorTag 上的压力/红外/数字温度传感器 | 37 |
| TI SimpleLink SensorTag 上的 9 轴运动跟踪器件 | 48 |
| 云端体验篇 | 54 |
| 固件升级 OTA | 59 |
| 开发环境搭建 | 68 |
| BLE 协议栈的安装 | 72 |
| 服务浏览 (1) | 77 |
| 服务浏览 (2) | 87 |
| 运动生理监测：羽毛球 | 97 |
| 看源码 | 100 |
| 安卓 app 代码一观 | 113 |
| 总结 | 117 |

产品介绍



新 SensorTag IoT 套件邀请您将自己的云连接产品创意付诸实践。新 SensorTag 现在包括 10 个低功耗 MEMS 传感器，采用小尺寸红色封装。它可以通过 DevPack 实现扩展，易于添加您自己的传感器或传动器。

通过蓝牙智能连接到云，在 3 分钟内在线获得传感器数据。SensorTag 开箱即可使用，带有 iOS 和 Android 应用，无需编程经验即可开始操作。

新 SensorTag 以 CC2650 无线 MCU 为基础，功耗比此前的蓝牙智能产品低 75%。因此，SensorTag 可以使用电池供电，一个纽扣电池就可以使用几年。

蓝牙智能 SensorTag 包含 iBeacon 技术。该技术使您的电话可以根据 SensorTag 数据和物理位置启动应用程序并自定义内容。访问 www.TI.com/sensortag 获得更多信息并下载软件和设计文件。



开发板试用平台 <http://bbs.elecfans.com/try.html>

SensorTag 基于 SimpleLink 超低功耗 CC2650 无线 MCU，可支持蓝牙智能、6LoWPAN 和 ZigBee 的开发。此 SensorTag 的一种特性是允许开发人员在不同的 2.4 GHz 技术之间切换，只需以无线方式直接从 SensorTag 应用程序中加载新的软件映像即可达此目的。

一，SensorTag 特点：

1，一个助力物联网的开发平台

将传感器数据与云连接相整合的下一代物联网开发套件

2，凭借安全的云连接，轻松实现物联网

使用 SensorTag 应用程序（包括 iOS 和 Android 版本）让数据在三分钟内连接到云

3，一个套件，多种无线连接技术

4，SensorTag 提供两种变体，每种变体具有不同的特性：

- 支持蓝牙智能、6LoWPAN 和 ZigBee 的多标准（目前已可用）
- Wi-Fi®（即将推出）

5，低功耗或无电池运行

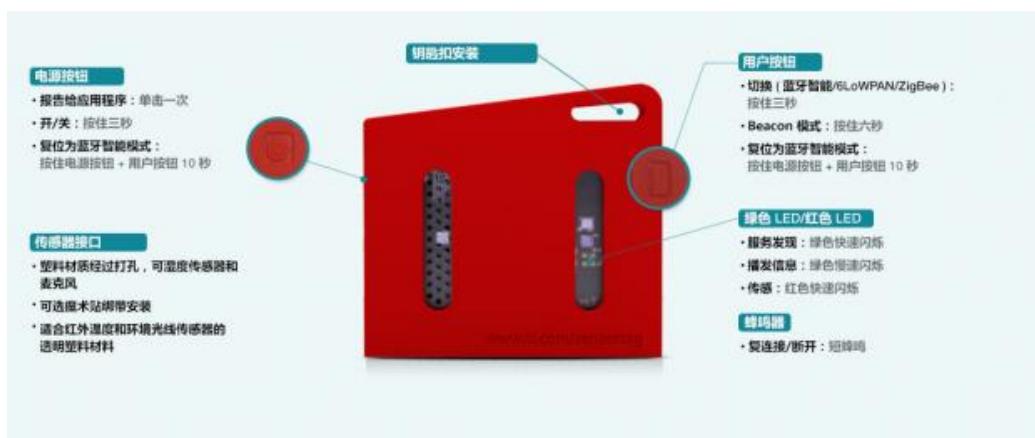
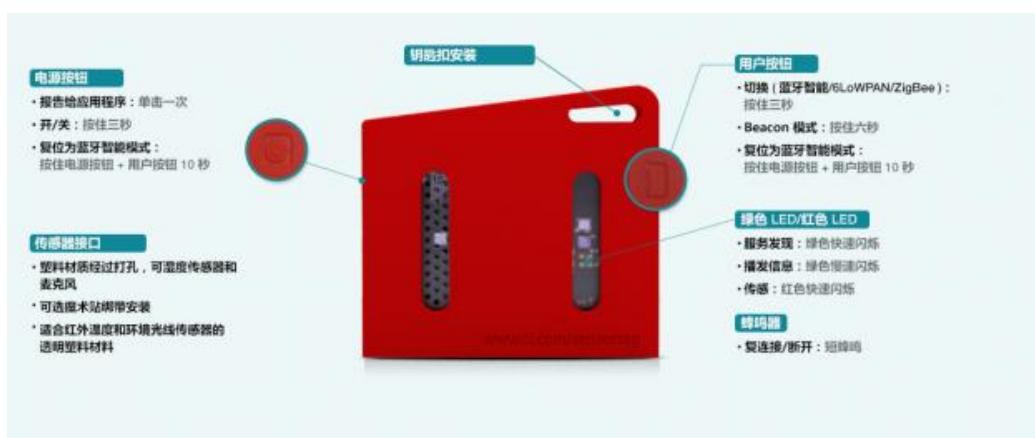
一年电池寿命和无电池运行（适用于蓝牙智能、6LoWPAN 和 ZigBee）

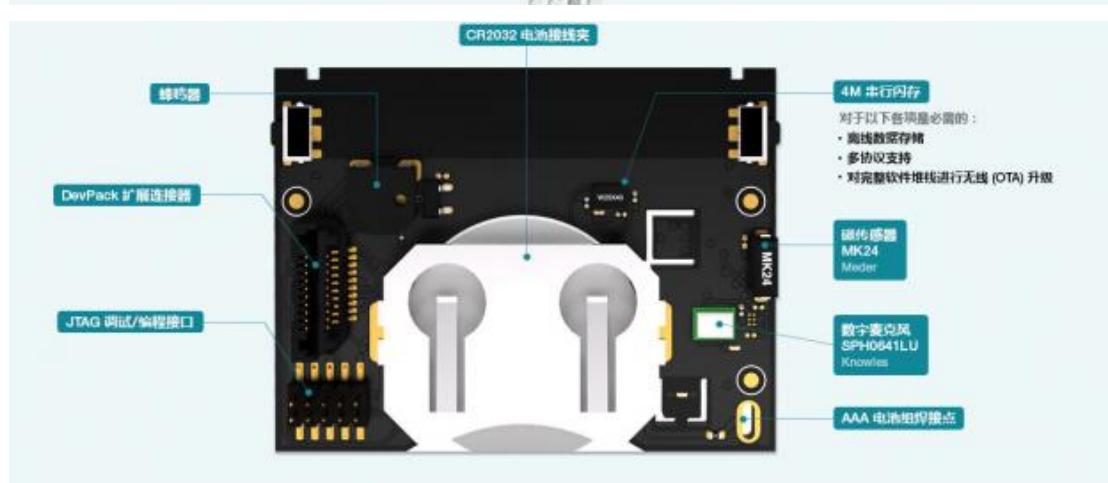
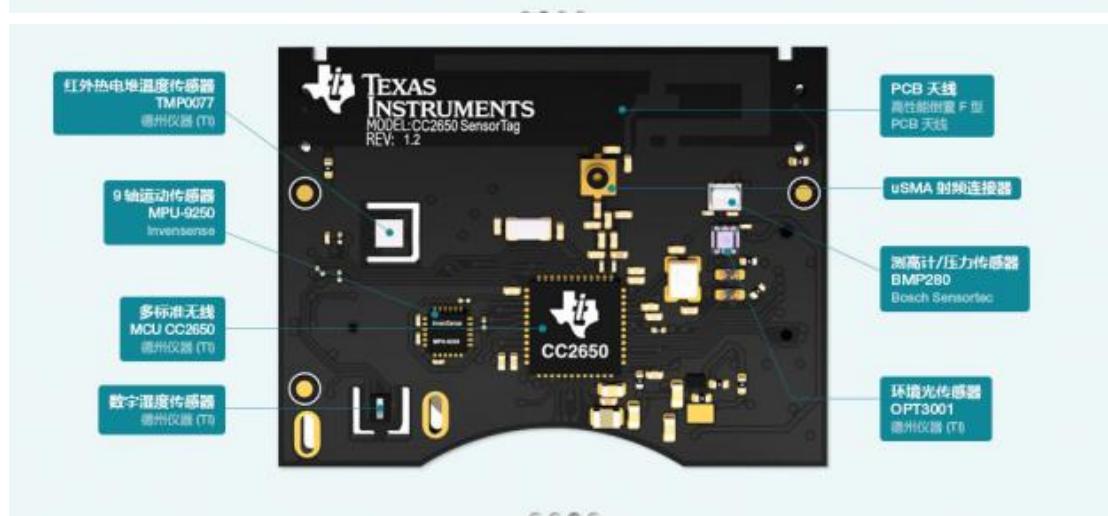
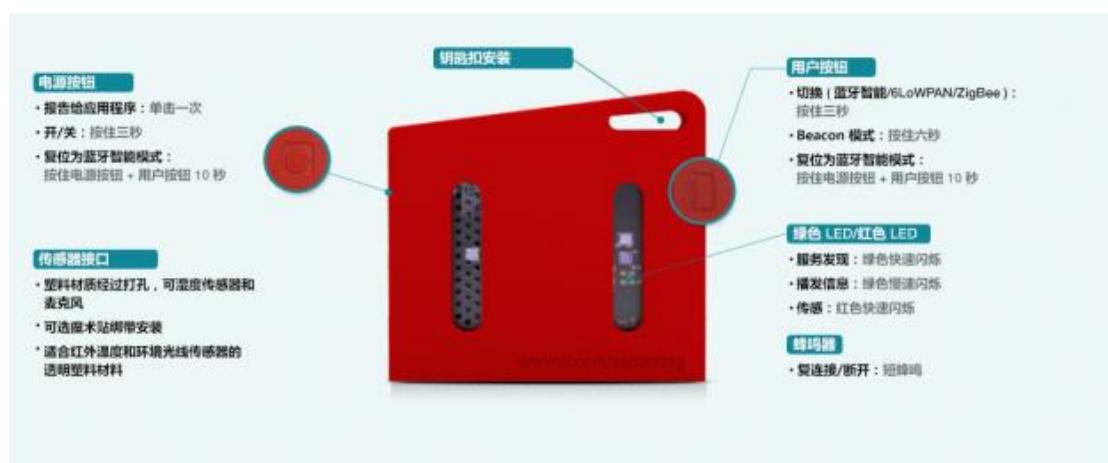
6，在 SensorTag 应用程序中或在线查看传感器数据
消除软件与硬件之间的障碍

7，访问开源设计文件和软件

共享您的设计或浏览其他人共享的其他产品理念

二，SensorTag 结构说明







开发板试用平台 <http://bbs.elecfans.com/try.html>

内容简介

本资料收录于电子发烧友论坛开发板试用平台精华篇编制而成，专门为创客、电子爱好者而备。文中讲解如何建立开发环境，如何编译系统，如何开发项目等教程。文字特意介绍了很多极具挑战性和实用性的案例，欢迎大家共同学习。

本文内容主要均来自电子发烧友论坛，并且每篇文章都注明了作者网名和原文链接。我们欢迎各位玩家复制传播本手册，但不得擅自摘抄部分或全部内容用作商业用途，违者必究，电子发烧友保留本手册的解释和修改权。



开发板试用平台 <http://bbs.elecfans.com/try.html>

开箱介绍

TI SimpleLink SensorTag 套件初亮相

作者 : @ [birdinskydzfsy](#)

地址 : http://bbs.elecfans.com/jishu_575376_1_1.html

按照惯例，开箱篇还是上图片为主，

这个是外包装盒，掂起来很轻，很普通的纸盒子，不过这个看起来不起眼儿的盒子里可是暗藏玄机，一会儿你就知道了



盒子背面，主要告诉我们有个锂电池



咦？打开盒子是个圆窟窿，好吧，这个很常见，SensorTag 已经小荷



开发板试用平台 <http://bbs.elecfans.com/try.html>

才露尖尖角了，



咦？拿下圆窟窿里边是个方窟窿，Oh,my god,这不是暗示天圆地方吧！！

这么大的咒语在，TI 还能不高大上啊 !!!



SensorTag 完全是被纸盒悬在包装的几何中心，我认为不要说这个东

从车上掉下来，就是从 100 米高空扔下来，里边的 SensorTag 也伤不了分毫。有两个说明书，全是英文，还有寥寥几行日文。。。有英文就好办



这是 SensorTag 快速启动说明，说实话我现在有点蒙圈了，这怎么玩要啊？看说明吧，估计 TI 早就想到我要蒙，给我了张快速玩耍的说明图文并茂，看看吧，第一步，下载 SensorTag 的软件到 iPhone 上，这个我早就装了，第二步，拉下那个塑料膜，这才发现这个膜，电池已经装上了，为了防止电能无 谓的浪费，加了膜隔离电极，看看人家，这是绿色人文精神啊，第三步，打开 app，快速体验！



很好很好，已经不蒙圈了，体验之前再仔细欣赏下这件巧夺天工的作品。这是那张膜。



正面观，就像个文件夹，右下方印有 TI 的 logo，橡胶材质外壳，摸

起来很细腻，有弹性，有厚度，有手感，有内涵，嗯，四有新人 !!!



背面观，外壳开了两道透气孔，这里应该是传感器集中的地方，左边的应该是红外温度和湿度的，需要和空气接触，开了许多洞洞，右边的应该是测高、气压和感光的，



左上 45 度观，左边有个按钮，应该是[电源](#)按钮，很明显有个国际通

用的图形。



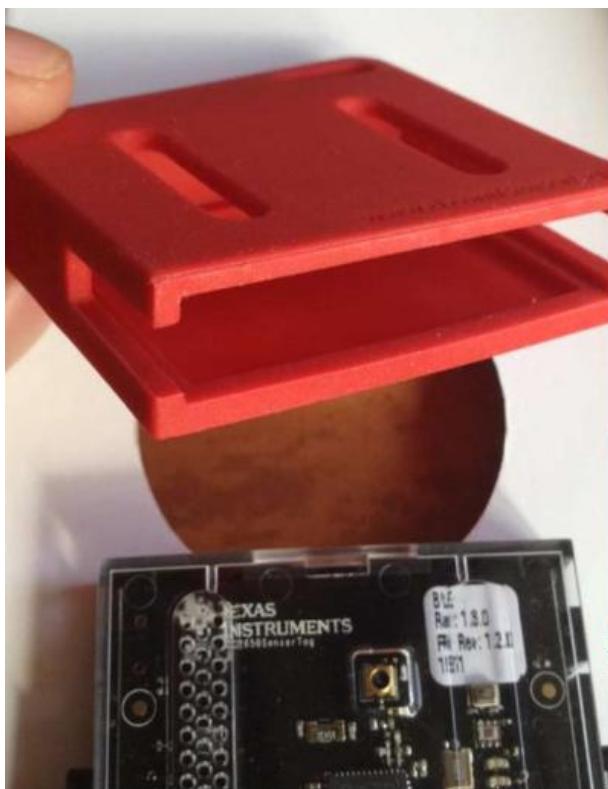
右上 45 度观，这里也有个按钮，据说是用户按钮



前下 45 度观，SensorTag 机芯可以从底部抽出来



外壳与机芯，抽出机芯，有很大摩擦力，要掐住机芯硬壳两边，轻捏外壳使之变形离开机芯慢慢抽出，所以在地球上机芯滑出来的可能性基本为 0。



机芯被安置在很精美的 PVC 壳子里，背面黑色不透明，正面无色透明，

上下有卡扣，打开需轻掰。



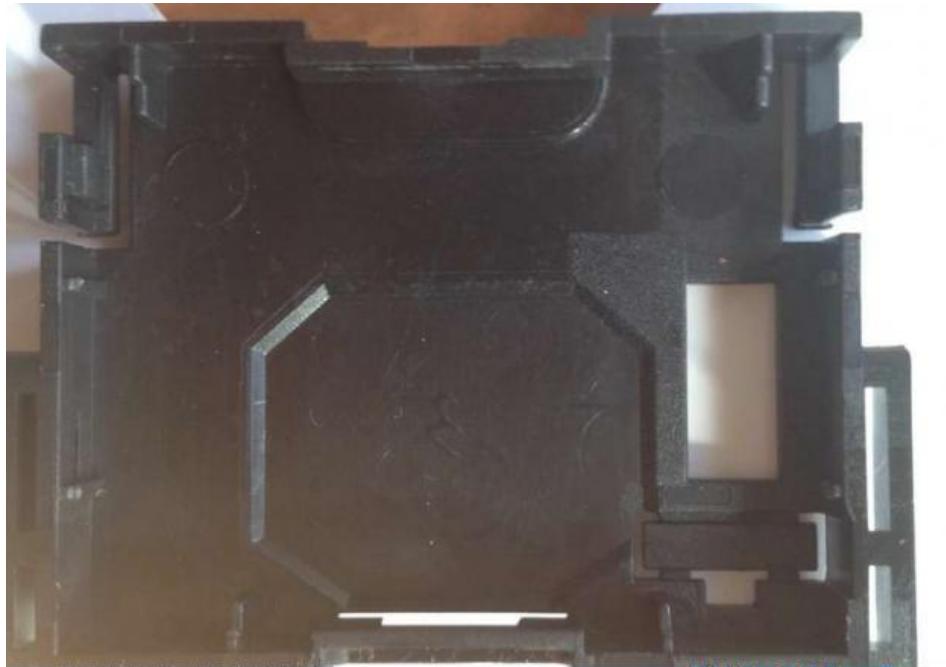
TI SimpleLink SensorTag 核心板素颜照，印刷清晰，做工精良，边交无毛刺。

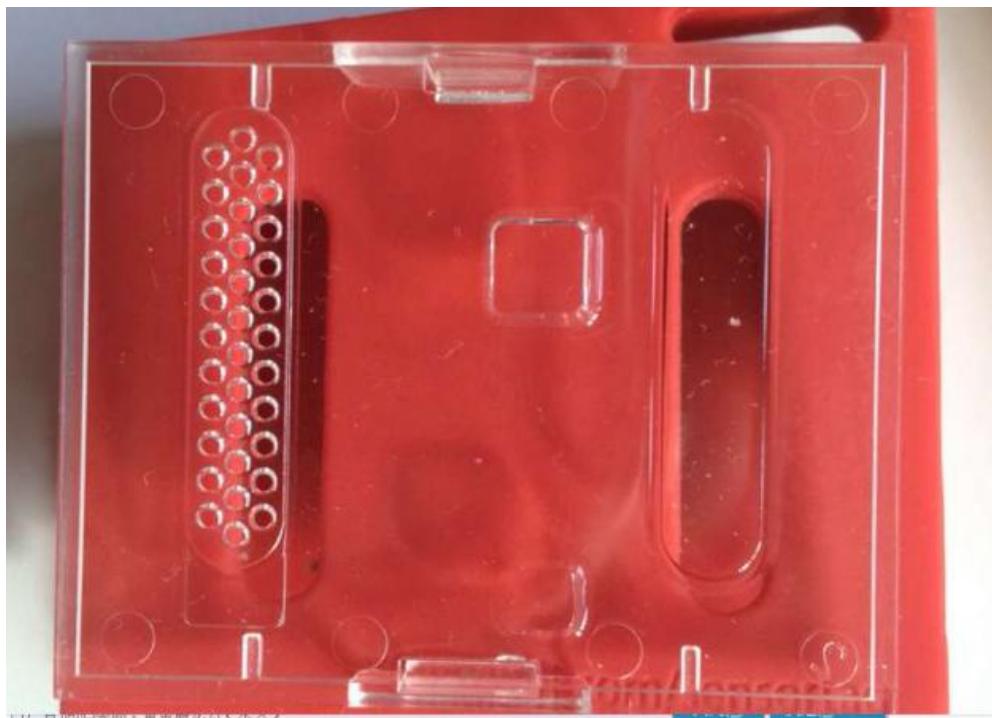


TI SimpleLink SensorTag 核心板背面大头照，干净，印刷的白色字迹清晰工整无涂抹痕迹。



不得不看的 PVC 外壳，如果说苹果的产品设计是圆滑无棱角，那这款 TI 产品的特点就是棱角分明，虽然有棱角，但是都经过仔细抛光，没有毛刺，用料讲究，黑也要黑得均匀。





整体回顾下，核心被保护在硬质的 PVC 壳子中，外面又包有柔软的橡胶质外壳，如果挂在钥匙或书包上，最大限度地缓冲了外力的冲击，想想我们自己的身体结构，也是内脏被保护在骨骼围城的空腔中，外面是柔软的脂肪皮肤。TI 不是在效法自然吧？

没有过多的色彩，只有黑、白、红，三种色调，黑色神秘而深沉，白色纯洁而神圣，红色活泼而热情；黑与白，既是阴与阳，这是个广泛存在的哲学，计算机的世界就是个阴与阳组成的二进制的世界，在这个世界里有 TI 人火一样的热情，火一样的不断向上的工作热忱。从精致的作品中当然能够看出作者精致的内心，向 TI 人致敬。

资料收集

所有资料均可以在 ti 官网上下载。地址是：

http://www.ti.com.cn/ww/wireless_connectivity/sensortag2015/index.html

上电试机篇

TI SimpleLink SensorTag 套件的全部功能简介

作者 : @ [birdinskydzfsy](#)

地址 : http://bbs.elecfans.com/jishu_575382_1_1.html

上一篇瞻仰了 SensorTag 的风采，这一篇开始来看看内涵。我的手机是 4s 的，下的 app 是 ios 的。



把 SensorTag 请出来，按一下边上的[电源](#)按钮，右边传感器区域绿色

的 led 会闪烁，表示待连接状态，如果长时间没有连接，就不闪了，好像是自动关机了。



接下来打开 app，首先得先打开蓝牙，否则会有提示你打开，

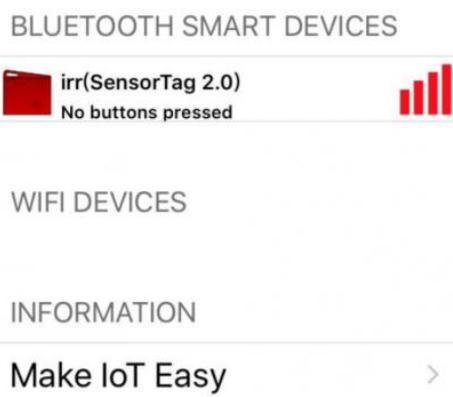


连接后，是这个样子，我们一点点看，最上边的就是已连接的设备，

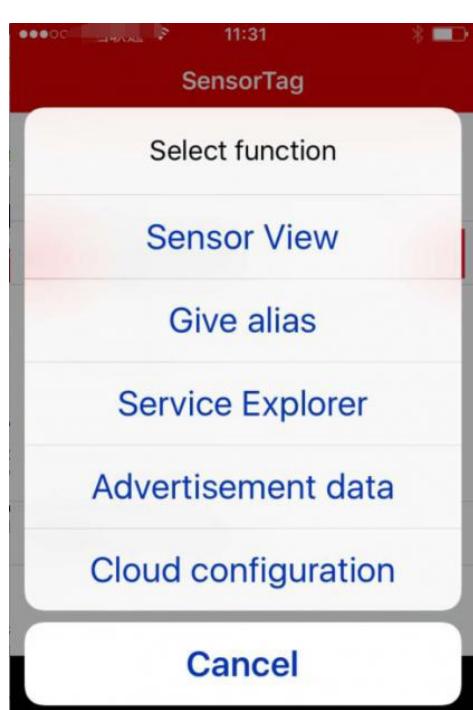


开发板试用平台 <http://bbs.elecfans.com/try.html>

右边是信号强度，左边是设备名和小图标“irr(SensorTag2.0)”，设备的名字是可以自己起的。

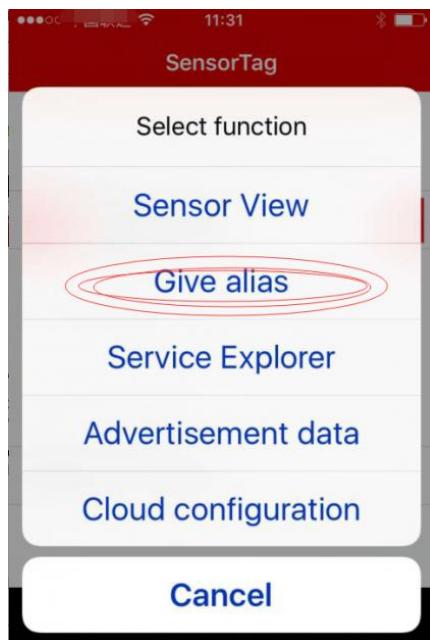


戳一下“SensorTag2.0”会弹出菜单，略懂英文的童鞋都能看懂，



先看下第三行，其实就是第二个选项，因为第一行是“选择功能”，显

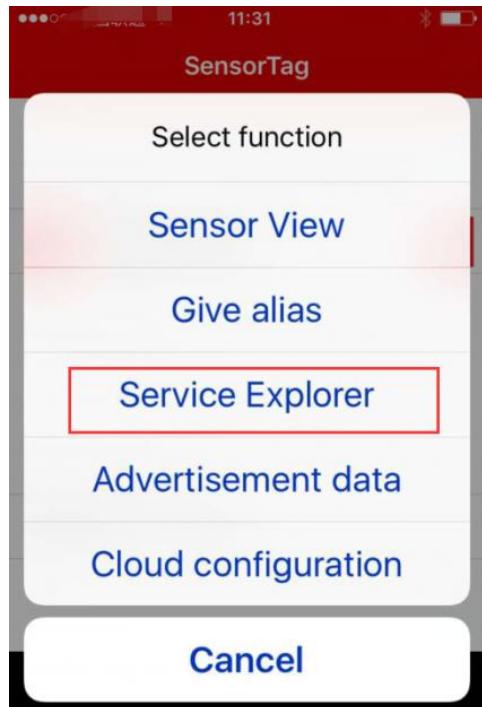
然是个菜单标题，要是我就把这个字号再整大点，也许这是 ti 的 ui 风格，不过对于一个测试功能的 app 来说，这无关紧要，也许人家就是凸显功能项，悟空，为师是不是又聒噪了？



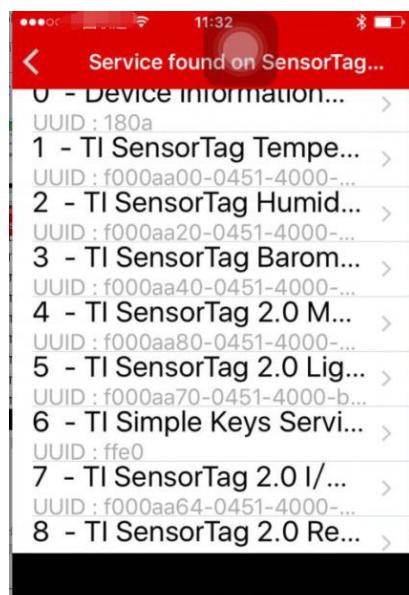
戳完后，尽情地命名吧，为了迎合女友的心意，我用了 irr，安定团结是发展的前提！！！



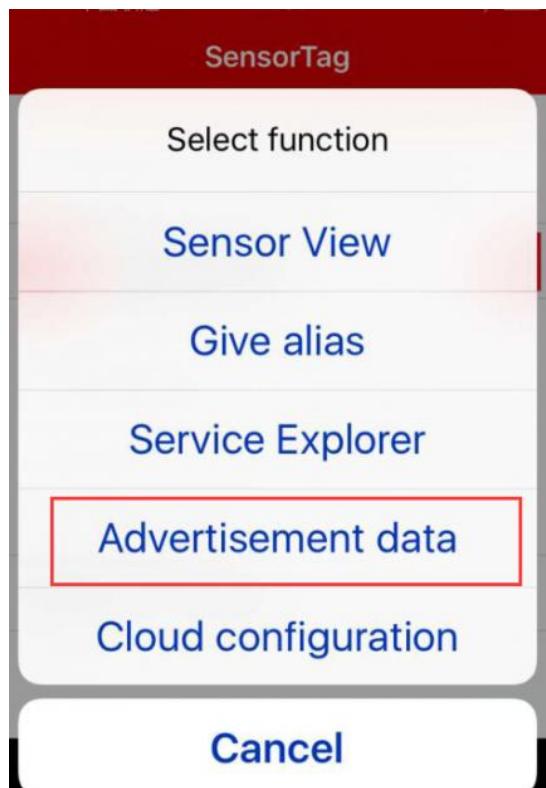
第三项是“服务浏览”



戳进去，是十一个服务，我们浏览下，看看那未完全显示的英文们也能猜个八九不离十，驱动信息，温度，湿度，气压，运动(4 -TI SensorTag 2.0 Movement....这就露了一个 M)，光照，按键，IO，Register 应该是寄存器，还有两个没在图上的是连接控制和 OAD，这个东就浏览到这，后文再详细看看。



第四项，公示信息，



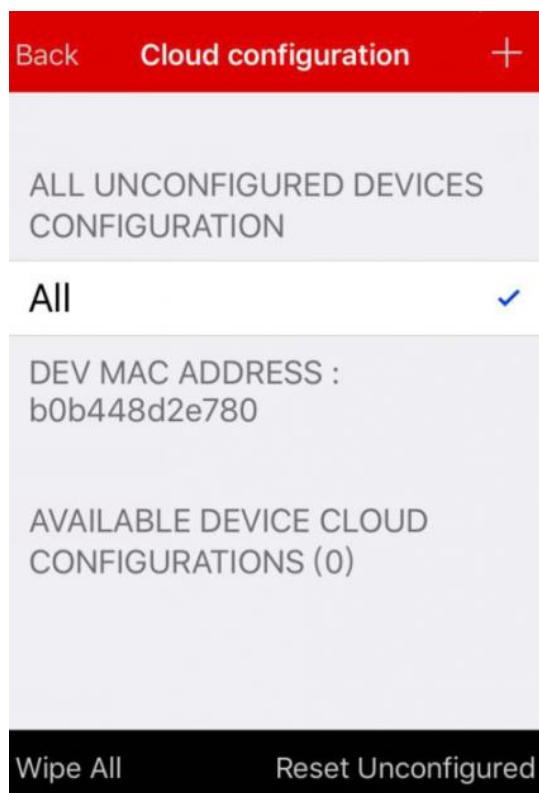
戳进去是一些似懂非懂的值





开发板试用平台 <http://bbs.elecfans.com/try.html>

第五项是云配置信息，戳进去是这个样子的，里边好几级菜单，



最后看看第一项，“查看传感器数据”，就这么翻吧





开发板试用平台 <http://bbs.elecfans.com/try.html>

这是云和按键服务，云的还没试过，按左右两边按键的时候界面会有提示

The screenshot shows the SensorTag Cloud View interface. At the top, there's a red header bar with a back arrow, the text "SensorTag", and an "Edit" button. Below the header, the title "Cloud View" is centered. On the left, there are two icons: a red folder with a small red square icon and a white cloud with a black "X". To the right of these icons is a button labeled "Push to cloud" with a toggle switch. Below the icons, the text "Interval 1.0s" is displayed. In the center, there's a section titled "TI Simple Keys Service" with four icons: a red folder with a red arrow pointing down, three red squares, and a small device icon. Below this section is a "Notify" button. Further down, there's a section titled "Ambient Temperature" with a Texas Instruments logo. The main content area displays the text "24.0°C" next to a thermometer icon.

这是温度传感器和红外温度传感器，红外可以扫描几十厘米范围内的温度，比如一杯热饮，

The screenshot shows the SensorTag Cloud View interface. At the top, there's a red header bar with a back arrow, the text "SensorTag", and an "Edit" button. Below the header, the title "Cloud View" is centered. On the left, there are two icons: a red folder with a small red square icon and a thermometer icon. To the right of these icons is the temperature value "24.0°C" with a right-pointing arrow. Below the icons, the text "Interval : 1.0s" is displayed. In the center, there's a section titled "IR Temperature" with a thermometer icon. The main content area displays the text "18.6°C" next to a thermometer icon. Below the temperature values, there are two "Interval : 1.0s" labels.



开发板试用平台 <http://bbs.elecfans.com/try.html>

这是湿度和气压、高度，对着 SensorTag 哈一口气，湿度就往上涨



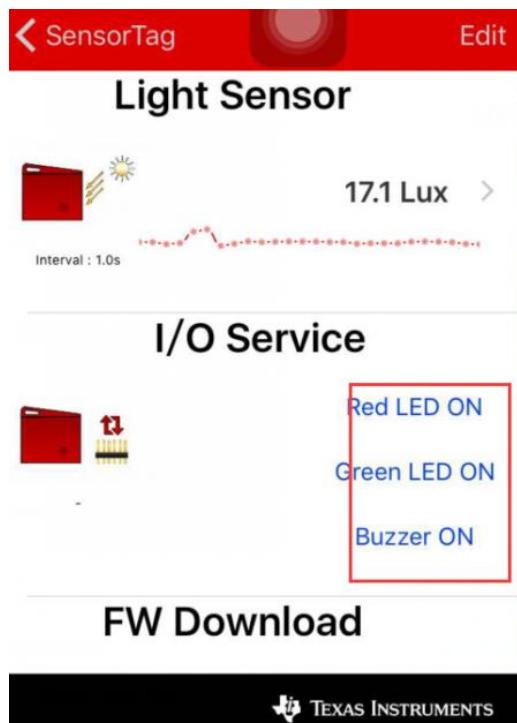
这是 9 轴传感器的运动参数，有个开关打开后，晃动下传感器会开始传递参数，静置一会，就不传值了





开发板试用平台 <http://bbs.elecfans.com/try.html>

这是光照和 IO 口，用手捂住 SensorTag 光照强度就变 0 了，下边有三个 IO 开关，两个 led 和一个蜂鸣器，



这俩没看懂，上边的好像是下载固件的吧，下边的是控制传输速度的？





开发板试用平台 <http://bbs.elecfans.com/try.html>

这是设备信息，还有信号强度的吧

< SensorTag Edit
/20.00ms

Device Information

System ID : ff:b0:b4:48:00:00:d2:e7:80
Model Number : CC2650 SensorTag
Serial Number : N.A.
FW rev. : 1.20 (Jul 20 2015)
HW rev. : PCB 1.2/1.3
SW rev. : N.A.
Manufacturer Name : Texas Instruments
IEEE 11073-20601 reg. : þ
PnP ID :

RSSI

Interval 1.0s

-53dBm

这次概要地浏览了一下，东西太丰富了，接下来慢慢仔细研读。



开发板试用平台 <http://bbs.elecfans.com/try.html>

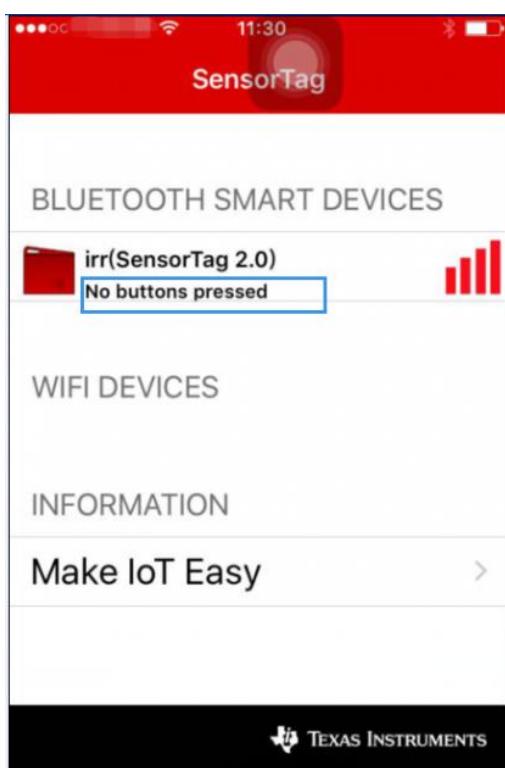
TI SimpleLink SensorTag 上的磁传感器/蜂鸣器

作者 : @ [birdinskydzfsy](#)

地址 : http://bbs.elecfans.com/jishu_575765_1_1.html

上一篇中蜻蜓点水般地看了 SensorTag 的全部功能，所有功能一带而过，这回我们仔细研究下各个功能。

人脑是天生为处理图像而生的，还是上图，我们从 SensorTag 上的传感器开始看，打开 app,连接，这是主界面，我估计要是再来一个 SensorTag，应该是两项，N 个就是 N 项，推测而已，有机会童鞋们试试，告诉我是不是这样。童鞋们注意，那个红色文件夹右边下半部有一行英 文，“No buttons pressed”，这不是乱写的，现在确实“没有按钮按下”，什么意思呢？我们轻按 SensorTag 右边的用户按钮，开箱篇提到了，有[电源](#)按钮和用户按钮，我们轻按用户按钮，

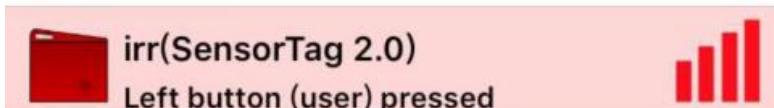




看到变化没？？？变成“Left button(user)pressed”了，显然，人家界面上每行字都是有深度的，不是乱写的，虽然是个传感器测试 app 的 demo, 做得也很用心。

SensorTag

BLUETOOTH SMART DEVICES



WIFI DEVICES



开发板试用平台 <http://bbs.elecfans.com/try.html>

这是按下电源按钮的效果，为了截屏，我多按了一会儿说明书上讲的明白，三秒开关机，童鞋们注意到，除了显示“Right button(power)pressed”外，项目变灰了，因为断电了，按了电源键关机了，于是 app 上灰掉了，提示你要想再用打开电源，什么是细节决定成败？这就是。

SensorTag

BLUETOOTH SMART DEVICES



irr(SensorTag 2.0)

Right button (power) pressed



WIFI DEVICES

干脆我们把按键的全部都看完，还记得这个吧，是不是有童鞋有想法了，

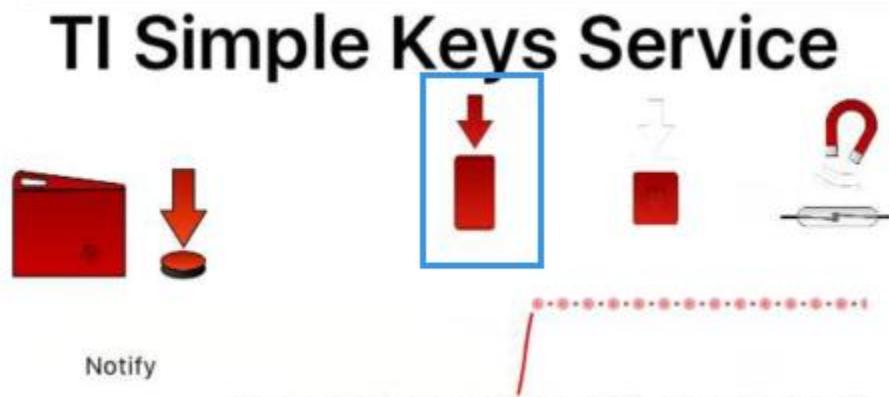
TI Simple Keys Service



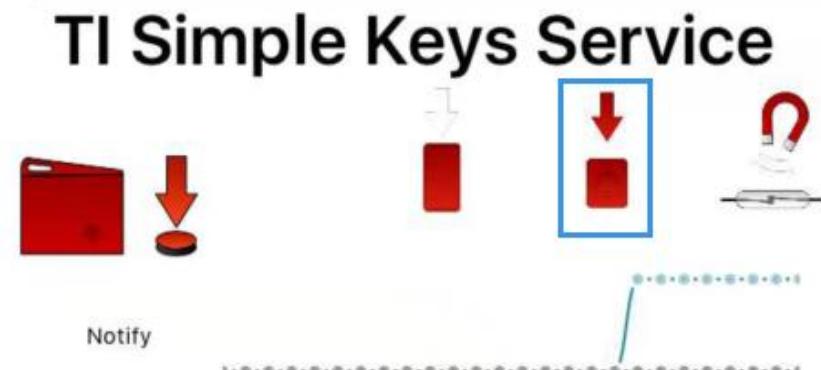
Notify



先上图，边看边说，这是按下用户键的效果图，



这是按下电源键的效果图，这两个键按下时，对应的图标上方，会有红箭头闪一下，同时下方会划出高电位波形图，童鞋们注意那个电源键的 icon，里边隐约有一个小圈圈是开关标志，因为我处理了图像可能不清楚了，所以，再次说明，TI 做的东西很细致，连一个图标都不放过，电子妹 妹们你们会怎么样选男盆友呢？“三看”，看头看脚看指甲，如果头发整齐，鞋子干净，指甲缝儿里没有泥儿，那么这个哥哥可以考虑，用哲学的观点，内外是统一 的，有什么样的外在就有神马样的内在，TI 做的东西精致不是偶然的。看完此贴的男童鞋们泡妹前请把头发鞋子指甲缝打扫干净噢 !!



第三个图标，童鞋们也认得，是个干簧管，是滴，是个磁力计，MK24系列干簧传感器，一个感应磁力的开关，开始认为对磁场强度有点要求，这种白板上用的磁力贴好像不行，放上去没反应，

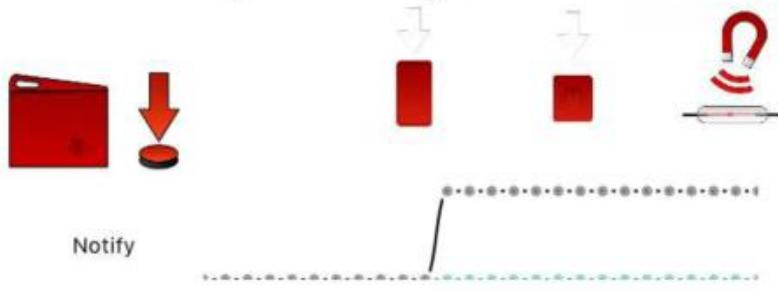


我用了这种擦玻璃的东 ,里边有两块大磁铁 ,玻璃外一块 ,里边一块 ,
用于高层建筑擦玻璃 ,尽管我一回也没有用过 ,靠近后 ,



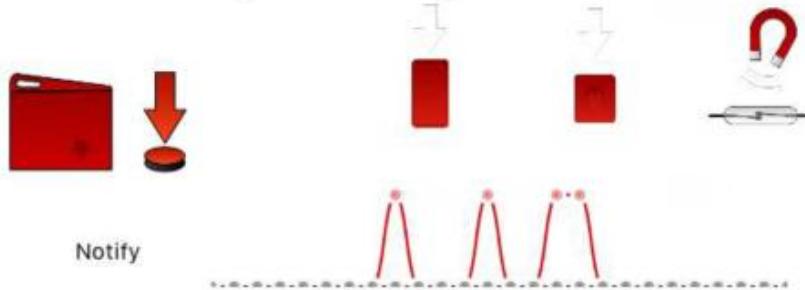
发现没，起作用了，黑色的波形线，后来看了板子结构，干簧管在侧边，靠近电源键，于是，磁力贴靠近侧边试试也有反应了，虽然一个干簧管只有通断两种状态，没法测量磁场强度，但是感觉利用这东可以做成反坦克地雷，人站上去没事，坦克这种家伙那个履带的磁性足矣引爆了，不知道有没有这种雷，记得《地雷战》里 演过鬼子用探雷器（其实就是磁性探雷器，干簧管遇到铁壳雷吸合发现目标）把铁雷都找出来了，石头雷测不出来，不过最终还是难逃人民的智慧，用头发丝雷破解了探雷器，这探雷器和鬼子也一起飞上天了。后来的电子探雷器就更高级了，利用各种传感器，检测雷区的理化性质变化发现目标。又扯远了。

TI Simple Keys Service



再来几个，这是用户键按下几次，

TI Simple Keys Service



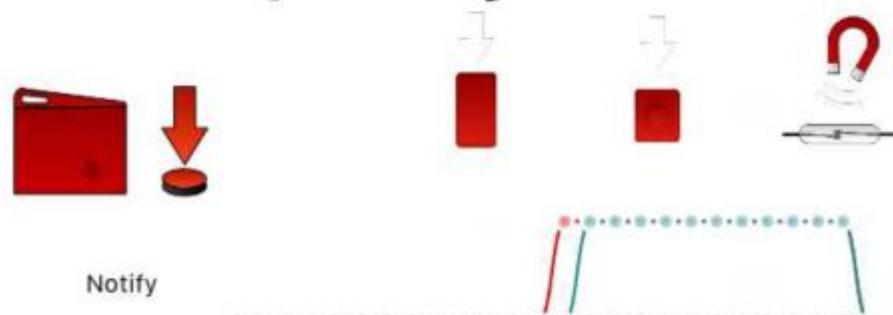


开发板试用平台 <http://bbs.elecfans.com/try.html>

这是电源键和用户键同时按下，红色的是电源，蓝色的是用户，TI

早就想到你会同时按，所以用不同颜色的波形区分

TI Simple Keys Service



既然说了按键，那就再接着说按键，刚才说的是硬件按键，这回说软按键，如图，

I/O Service



拉灯是我们永恒的话题，TI 也没有放过拉灯，测试 IO，仍然用的点灯和蜂鸣器，我戳，我戳，开了两盏灯

I/O Service

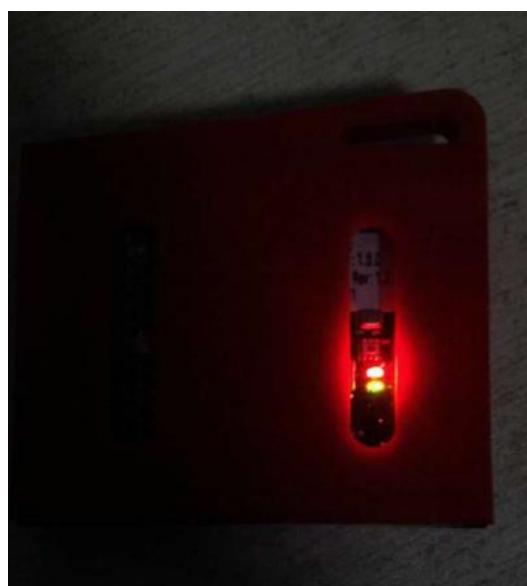


Red LED OFF
Green LED OFF

Buzzer ON

红灯绿灯都亮了，红的确实很红，绿的确实很绿，截完图好像失真了。

蜂鸣器打开后是比较低沉的声音，能听到，且不刺耳，我不得不再次说 TI 的好话，童鞋们可能玩过好多[开发板](#)，我也有好些块，我玩耍的时候找一块胶带，先把蜂鸣器粘上几层，因为示例程序一开，那就是刺耳的尖叫声，唉，保护耳朵，热爱生命。TI 就想得周全，知道你要 测蜂鸣器，给你个音量合适，频率合适的声音，作为测试者，听到声音即可，我知道有那个功能，没有必要扯着嗓子吼，最终让它怎么叫看我的心情了，元芳你怎么 看？



TI SimpleLink SensorTag 上的压力/红外/数字温度传感器

作者 : @ [birdinskydzfsy](#)

地址 : http://bbs.elecfans.com/jishu_576126_1_1.html

上回说了 SensorTag 上的开关和 IO Server 的 led 和蜂鸣器的软开关 , 这回我们说点啥 ? 等待篇里我们还提到了一个 app, 就是这个 ,



这个好像是给这个东东用的 ,



我们的连不上。好在有能连上的版本，而且功能貌似更为强大。

接着说，我们看看比较熟悉的几个物理值，湿度、温度、气压、高度，这个是相对湿度，这个就不必特别说明了，大家都懂的，前文也说了，对着 SensorTag，以后我们叫她小 T 吧，我少打几个字，对着小 T 哈气，于是湿度会增加，这里咱们玩个特别的，对着小 T 猛哈一阵气。

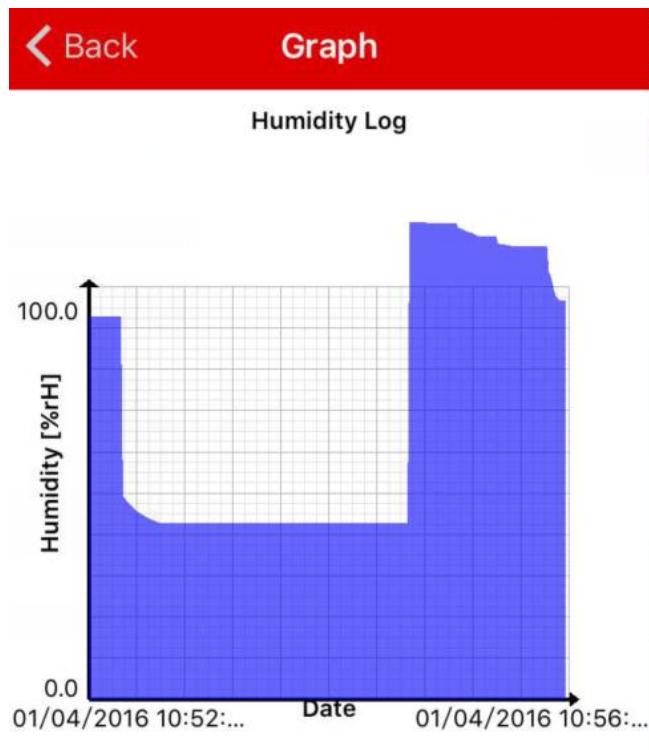


如下图，看着是不是有点鬼异啊？湿度值超了 100%了。。。。。

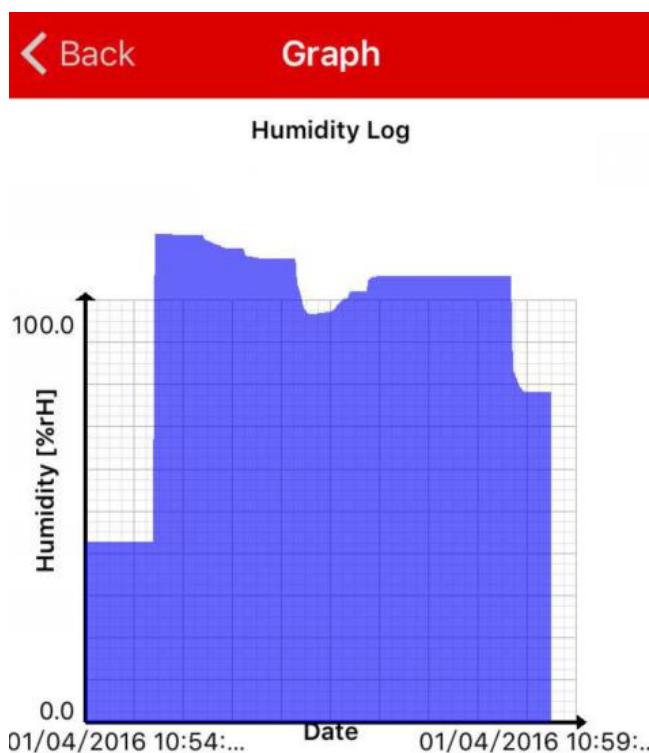


这是湿度值数据的图表，显示了一段时间内湿度的变化，戳一下湿度那项就可以进去看，上图最右边有一个“>”，凡是有这个符号的说明

还有内容可以看。



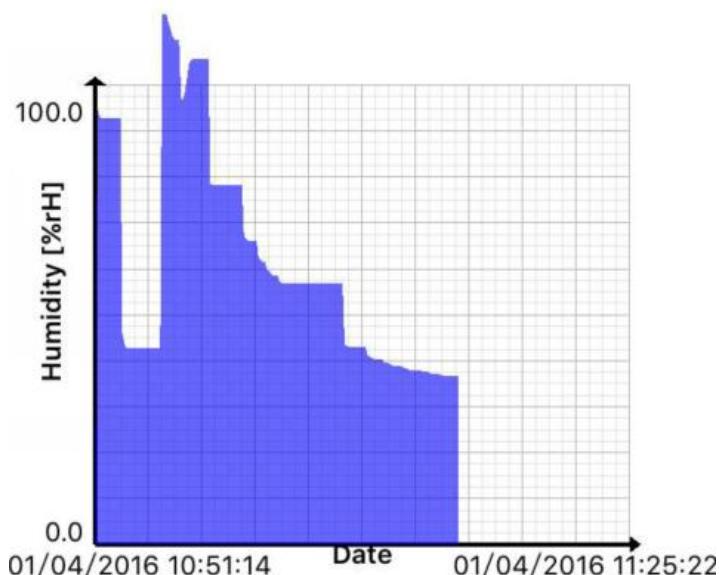
过了一阵，湿度开始下降，如图，如果你觉得图表比较长，一屏显示不下，可以用两根手指缩小



就是这样紫，在这里我们产生了一个疑问，相对湿度咋超 100% 啦？？



Humidity Log

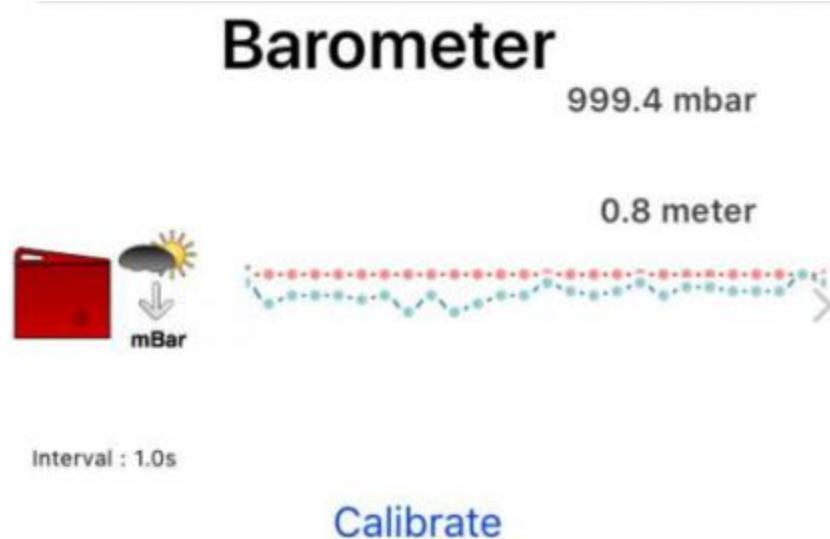


按理说这是不可能的，就像物质的纯度，中学化学童鞋们都学过，纯净物是理论上存在而实际是不存在的，有网上说什么 UFO 用的纯度百分之好几百的铁呀镁呀的，好吧，我相信你。在地球上，用来自做芯片的单晶硅，是很纯的了，一百万硅原子里可能掺杂了一个别的原子。我们的小 T 用的是 HDC1000YPA 数字温湿度传感器，读取温湿度值的时候按规定的时序操作时钟线和数据线读取。湿度传感器的原理，是利用物质的电阻-湿度特性制作的，常见的比如氯化锂电湿敏元件，质量比较好的元件湿度量程能达到 2%RH~100%RH，其他的材料还有碳湿敏元件、氧化铝、陶瓷等等。不管撒元件，应该都不能被水淹了吧，测量湿度值被 水淹了也没有实际意义，所以，湿度值超过 100%

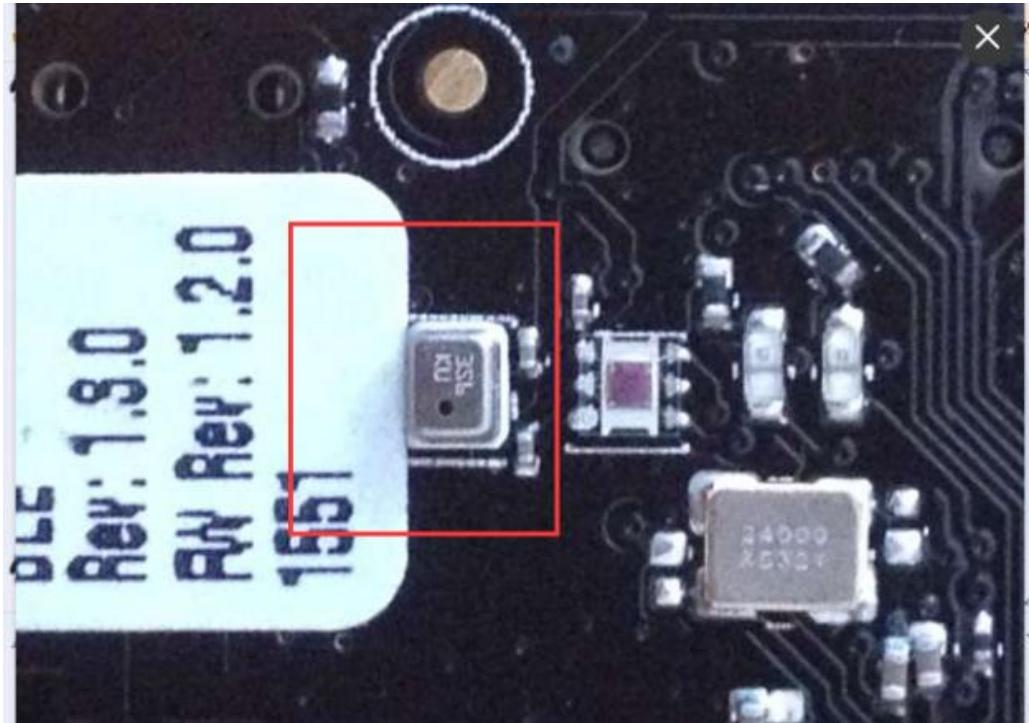
应该说明了个问题，哈气太猛了，传感器上凝结了水滴，造成元件在非正常环境下工作，电-湿度特性混乱，数值也就不准了，等慢慢水滴蒸发，湿度值就会恢复正常。

这是气压和高度。

这里有两个值，上边的是气压值，mbar 是毫巴，压强单位，1000 毫巴 (mbar)=100000 帕斯卡(Pa) ,首都地处华北平原，气压在 1000 毫巴也就是 1000 百帕左右比较正常，阴天下雨可能会低一些；下边的值是高 度，高度我一开始不太清楚，这个值单位是米，但是 0 米是什么地方呢？



为了搞清楚这个问题，先从传感器入手，这是小 T 上的 BMP280 气压传感器。



气压传感器啥原理呢？空气压缩机的气压传感器主要的传感元件是一个对气压传感器内的强弱敏感的薄膜和一个顶针开控制，[电路](#)方面它连接了一个柔性电阻器。当被测气体的压力降低或升高时，这个薄膜变形带动顶针，同时该电阻器的阻值将会改变。很多空气的气压传感器的主要部件为变容式硅膜盒。当该变容硅膜盒外界大气压力发生变化时顶针动作，单晶硅膜盒随着发生弹性变形，从而引起硅膜盒平行板电容器电容量的变化。开来这东也是有电阻式的电容式和有电容式的，简单理解就是气压变化引起元件形变，引发电容或电阻变化。

测气压是没啥问题了，那这东怎么测出高度来呢？还得感谢物理老师，童鞋们回忆下，初中物理学过是否记得，海平面处，指甲盖大的地方，如果有一个空气柱的话，这空气柱质量有 1KG 重，站得

越高气压越低，越低气压越高，青藏高原上几十度水就开了，气压低造成了沸点升高..... 太乱了，整理下，我们要的是高 度和大气压有一定关系，一般地，随着高度增加，气压下降，高度降低，气压上升。

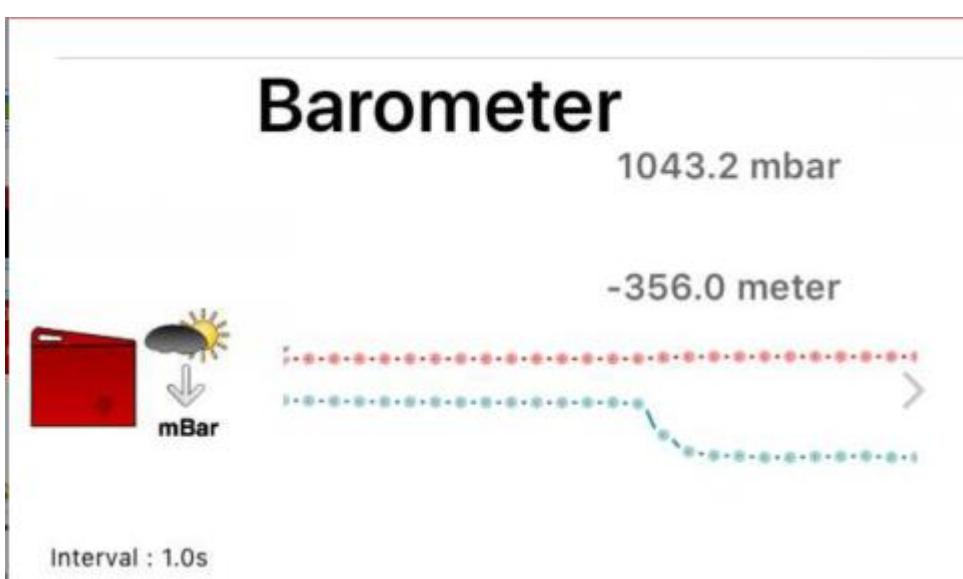
可是，拿着小 T 举过头顶，放到脚下，确实看到高度变化了，不太精确的变化，2 米以内，怎么证明小 T 的高度值是通过气压换算出来的呢？得借助工具，不是撒高大尚的工具，



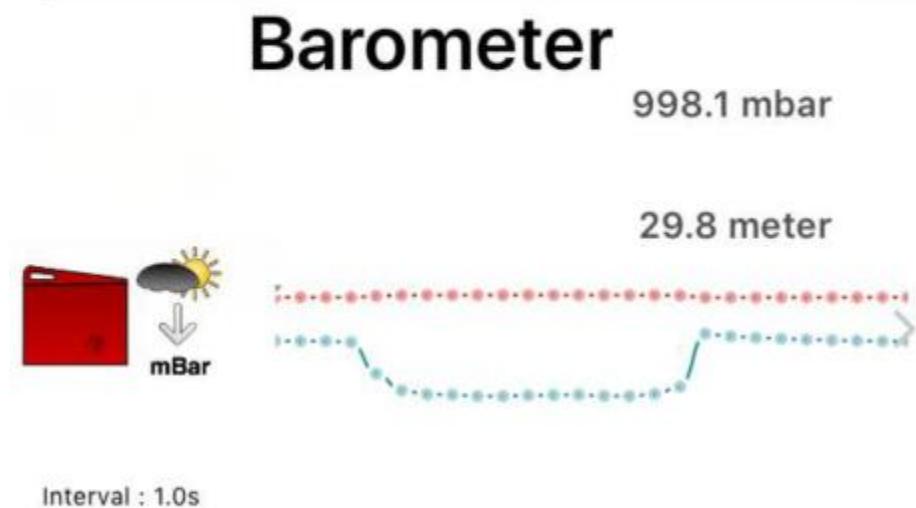
一个爆米花桶即可，所以，轻易别乱丢东西，谁知道啥时候用上，没有用高压锅是因为蓝牙信号穿不透金属。下边，开始，捏，



看看值，好了，真相大白了，高度值就是换算过来的，捏的时候，桶内气压增高，高度显示为-356米，



松开手，因为桶的气密性不好刚才漏气了，桶恢复到原来形状后空气不够了，气压降低了，于是，这个在二楼的小 T，被认为在 29.8 米处



一般地，当高度低于 1,500 米时，大气压几乎呈线性降低，每 100 米大约降低 11.2 mbar，即每 10 米大约降低 1.1 mbar。有兴趣的童鞋可以在网上查查相关资料。气压和高度看完了，接下来看看温度这块这个是温度，有两个，一个是普通的环境温度，还有一个是红外温度传感器。

Ambient Temperature



IR Temperature





开发板试用平台 <http://bbs.elecfans.com/try.html>

普通的环境温度传感器大体分两种，热电阻式和热电偶式。热电阻式的比较好理解，金属温度升高电阻率变大，温度降低，电阻率变小，中学物理学过的；热电偶式，比如双金属片（杆）式，两种不同金属热膨胀系数不同，温度变化会造成双金属片的弯曲，将弯曲的曲率输出成信号用来测量温度。

上边的两种都属于接触式传感器，测量时与被测物体充分接触，红外传感器属于非接触式的，不难想象，如果被测物体温度过高，比如太阳表面，火箭穿过大气层时外壳温度，用接触式传感器恐怕非挂掉不可，别说测温了。于是有人想出各种办法，解决这类问题。红外温度计就是一种，非接触的优点不仅可以测高温物体，比如在一些特殊场合，医院测病人体温，流行病高发时车站红外测体温，也安全避免和潜在的传染源接触。非典时期，这种温度计就用的比较多。

红外温度计使用时，应该和被测物保持一个合适的距离，合适的角度正对最好，不要隔着玻璃，我们这个小 T 用的是 TMP007 红外热电堆温度传感器，小巧低功耗，每次测量仅消耗 675uJ 的能量。如果童鞋们感兴趣可以网搜下这些非接触测温原理性的东。

我打开煤气灶，烧热了锅，测了下，



可见环境温度还是 30.1 度，锅温度已经到了 49.7 度，因为怕把小 T 烤坏，所以没再往高测。

Ambient Temperature



IR Temperature





开发板试用平台 <http://bbs.elecfans.com/try.html>

TI SimpleLink SensorTag 上的 9 轴运动跟踪器件

作者 : @ [birdinskydzfsy](#)

地址 : http://bbs.elecfans.com/jishu_576690_1_1.html

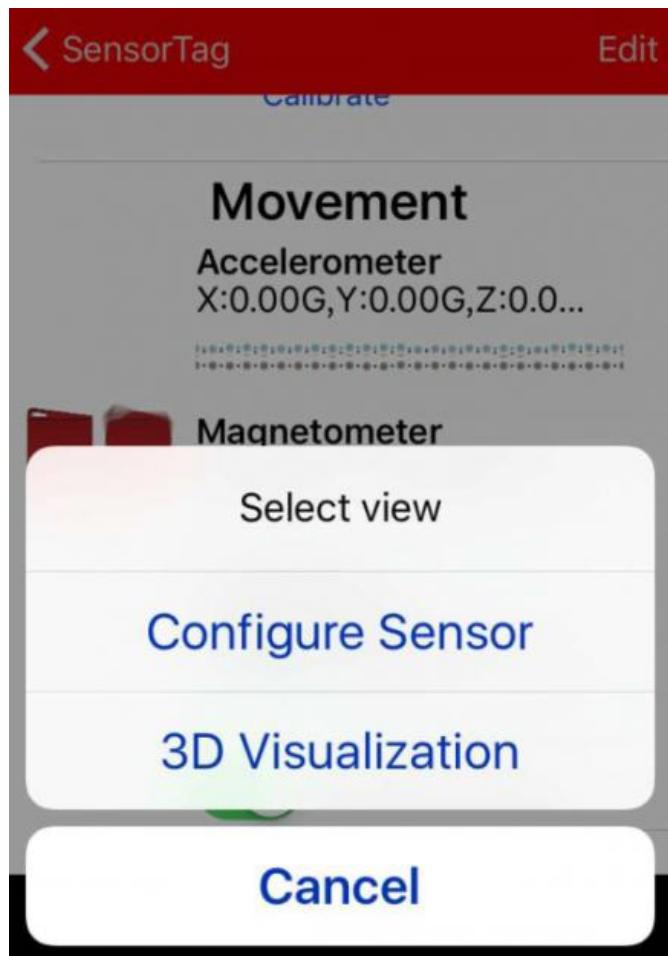
看下 9 轴传感器，很多厂商都做过运动传感器方面的 app, 感觉 TI 做的这东很好玩，分享下，下边这个图之前看过了，最下边的开关是设置“抖动唤醒”的，关闭时，运动参数会不停地传送，打开时，如果小 T 静止几秒后参数就不再传递了，除非晃一下唤醒。



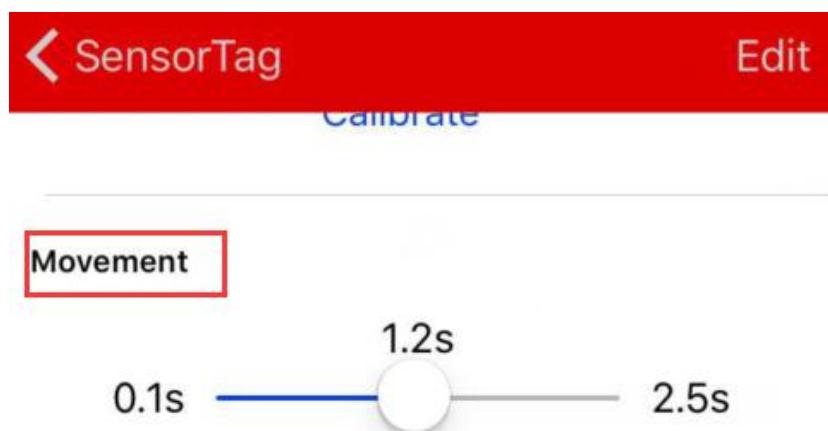
接着看好玩儿的，点中间右边的 “>” 会出来菜单，



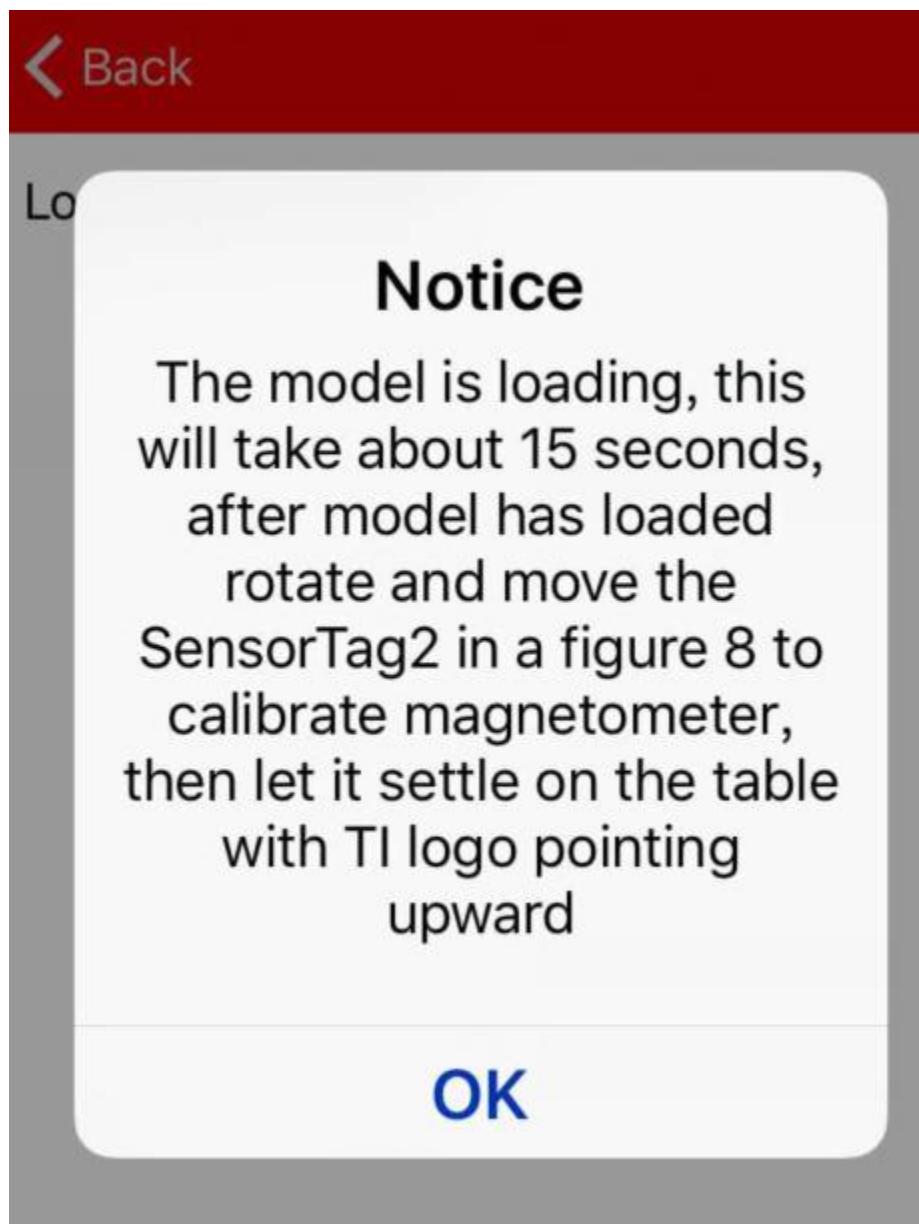
开发板试用平台 <http://bbs.elecfans.com/try.html>



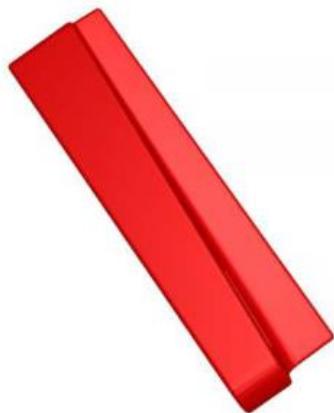
选择“Configure Sensor”,这里配置数据刷新频率，0.1s 到 2.5 秒，配置完点“Movement”返回



再点“>”，出来刚才那个菜单，选“3D Visualization”，3D 数据可视化工具，先出来的是这个，意思是等待 15 秒左右，有个模块正在加载中云云。。。



当当当当，于是出来个文件夹，我们的小 T，闪亮登场，接下来的任务就是尽情地摇摆吧 !!!



你会发现，屏幕上的小 T 也随你的摇摆而摇摆，而且，画面上的小 T 是高清 3D 的，板子上电路高清可见，而且不同角度上光影变化做得很逼真，连外壳遮挡电路板的效果都有，转到不同角度可以看见不同的元件

 Texas Instruments SensorTag 2





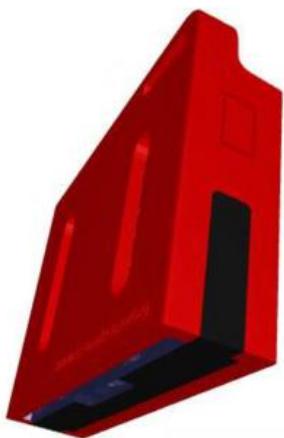
开发板试用平台 <http://bbs.elecfans.com/try.html>

外壳上 TI 的 logo 清晰可见





Texas Instruments SensorTag 2



当小 T 立在桌子上时，画面上的小 T 会顺时针转动，不知道是不是地球自转造成的，嘎嘎嘎，真是高大上啊 !!!



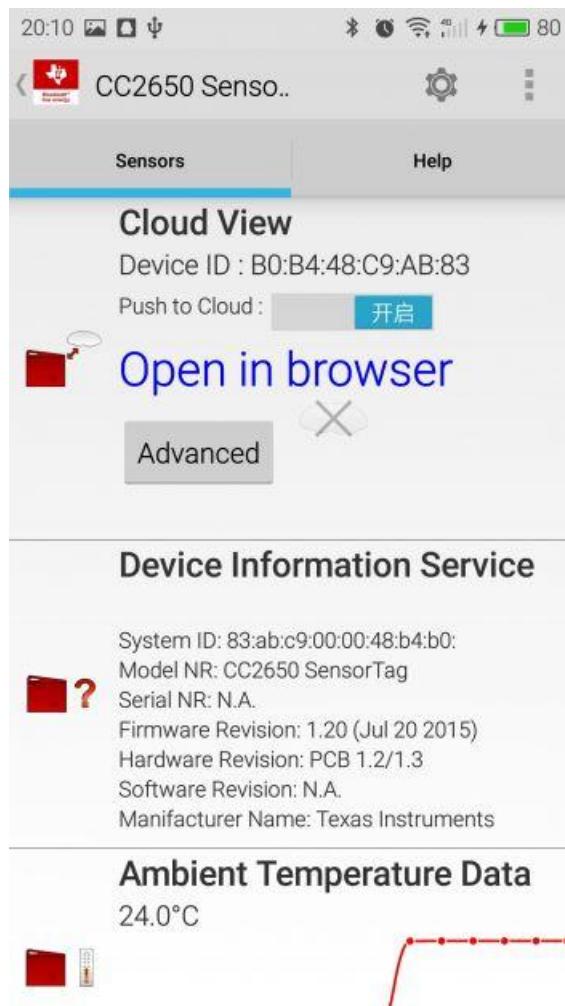
开发板试用平台 <http://bbs.elecfans.com/try.html>

云端体验篇

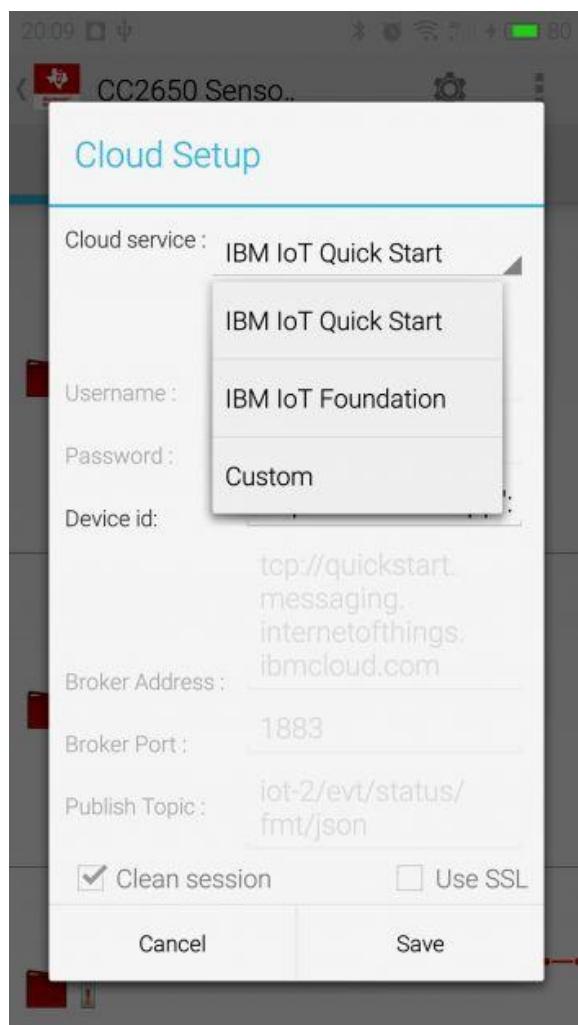
作者：[@zjhhello](#)

地址：http://bbs.elecfans.com/jishu_583904_1_1.html

前面体验了 Tag 的 APP 的功能，为了实现联网，远程查看数据，Tag 提供了展示，可以很方便地把数据上传到云端 APP 中，打开“push to cloud”开关。



在 Advanced 下面可以设置远端，它提供了三种方式，第一种是很方便的快速接入，无需注册；第三种是可以自己定义服务器连接



除了前面说的，可以用[手机](#)上的 APP 进行查看，还可以远程使用浏览器查看云端数据，现以第一种为例，透过包含 IBM BluemixIoT 基础架构的 TI IoT 云端生态系统，可在 3 分钟内链接至云端网址：
<https://quickstart.internetofthings.ibmcloud.com/#/>无需注册，设备 ID 号即为 tag 的 MAC 地址。



开发板试用平台 <http://bbs.elecfans.com/try.html>

Quickstart

无需注册，便可了解如何轻松地将您的设备连接到 IoT Foundation 并查看实时传感器数据

设备标识，例如，580b0c07ac

前往

-  获取您的设备（或模拟设备）
-  遵循诀窍以建立连接
-  查看您的设备的实时数据

已有物理设备？

我们拥有一项针对 IoT 的伙伴计划以及一组经过验证的指示信息或“诀窍”，以用于连接设备、传感器和网关。

[查看诀窍](#)

没有设备？

您无需物理设备来查看正在运行的 Quickstart。请使用模拟器尝试一下。

[了解更多信息](#)

无需注册，便可了解如何轻松地将您的设备连接到 IoT Foundation 并查看实时传感器数据

b0b448c9ab83

前往

● 在下午8:13:48 收到最后一条消息

status.light



数据点

值

接收时间

gyro_x

0.62

2016年
4月24
日 下午
8:13:48

除此之外，如果没有 tag 设备，点击右下角，可以提供模拟器来体验，
默认模拟了 3 个器件



开发板试用平台 <http://bbs.elecfans.com/try.html>

Quickstart

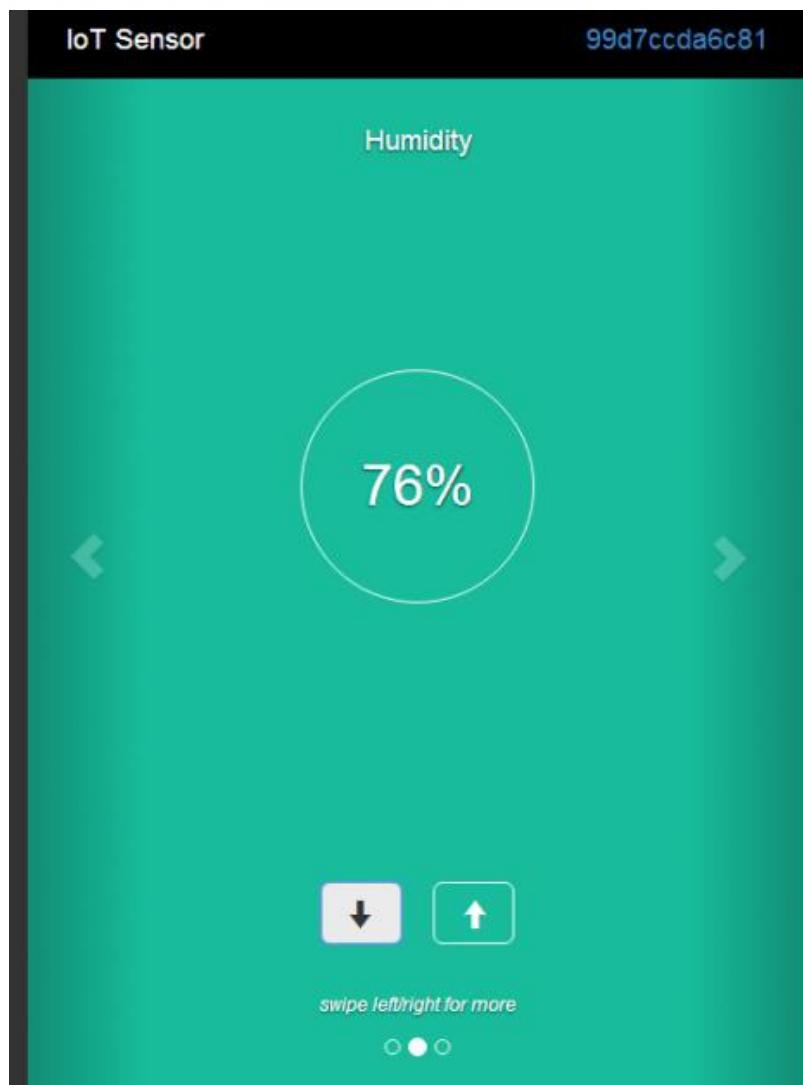
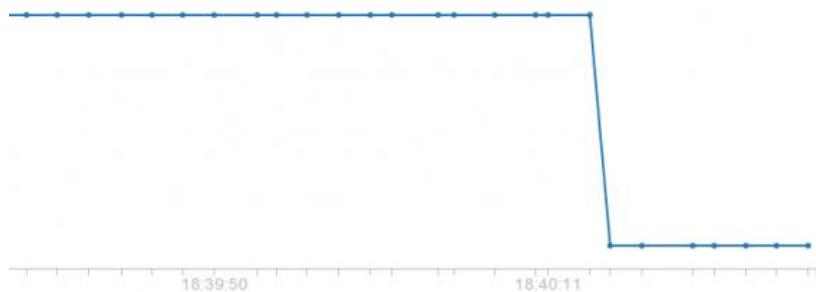
无需注册，便可了解如何轻松地将您的设备连接到 IoT Foundation 并查看实时传感器数据

99d7ccda6c81

前往

● 在下午6:40:28 收到最后一条消息

iotsensor.temp





这里可以看到 IBM BluemixIoT 基础架构的 TI IoT 云端生态系统
的数据接入文档简介 — IBM IOT Foundation 1.0 文档

https://docs.internetofthings.ibmcloud.com/zh_CN/

要连接到 Quickstart 服务，用户必须能够与 MQTT 交互。MQTT 3.1 是所需的最低级别，但是 V3.1.1 提供了更强的功能。同时数据传输采用 Json 格式。

消息格式

所有消息都通过 MQTT 以 JSON 格式进行发送。有两种类型的消息。

1. 请求。

请求的格式如下：

```
{  
    "d": {...},  
    "reqId": ...  
}
```

这些消息必须包含以下两个 JSON 元素：“d”和“reqId”。“d”字段中存放与请求相关的数据，“reqId”字段是请求的标识并且必须复制到响应中。如果不需要响应，那么应省略此字段。

2. 响应。

响应的格式如下：

```
{  
    "rc": ...,  
    "message": ...,  
    "d": {...},  
    "reqId": ...  
}
```

这些消息可以包含以下四个 JSON 元素：“rc”、“message”、“d”和“reqId”。“rc”字段是原始请求的结果代码，“message”字段是包含响应代码文本描述的可选字段，“d”是响应随附的数据字段（如果需要），而“reqId”是原始请求的请求标识。“reqId”字段用于将响应与请求联系在一起，因此设备需要确保所有请求标识都是唯一的。

响应 Internet of Things Foundation 请求时，必须在响应中发送正确的“reqId”值。

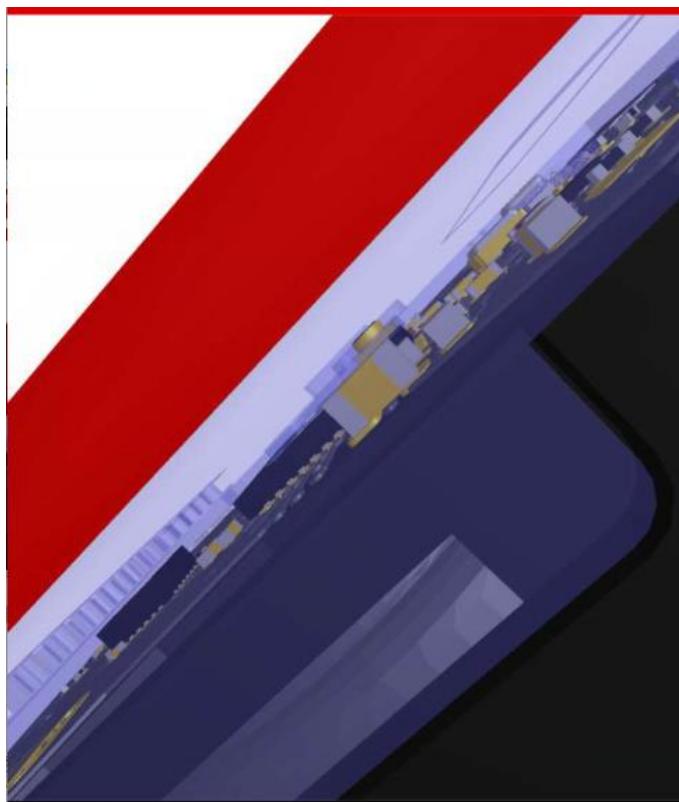
同时平台上提供了嵌入式 C、java、Node.js、C#的接入编程指南，可以快速进行编程。

固件升级 OTA

作者 : [@birdinskydzfsy](#)

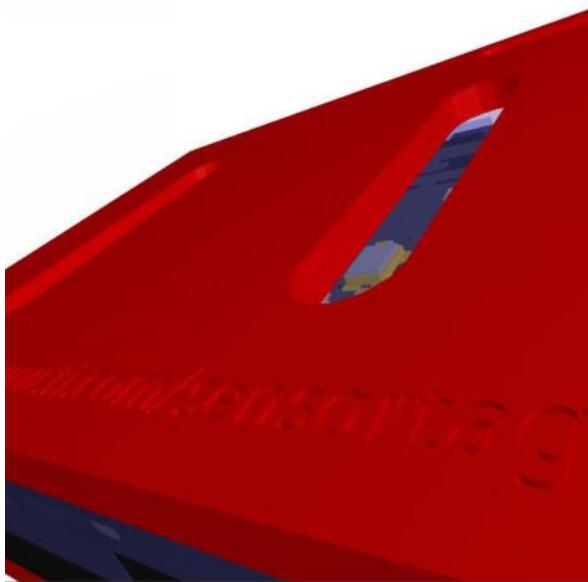
地址 : http://bbs.elecfans.com/jishu_577134_1_1.html

小 T 提供了一种比较高大尚的固件升级方式 , OTA , 也叫 OAD , oad 的话就是 Over the Air Download , OTA 的话就是 Over The Air download , 看你愿意神马样的姿势了 , 在说这个 OTA 之前 , 再说点 9 轴传感器的 偶然发现 , 上图小 T 的 [电源](#) 按钮在 9 轴 3D 图像里是图像放大按钮 , 用户按钮是缩小按钮。



放大后 , 外壳上立体感很强的 TI 网址

< Texas Instruments SensorTag 2

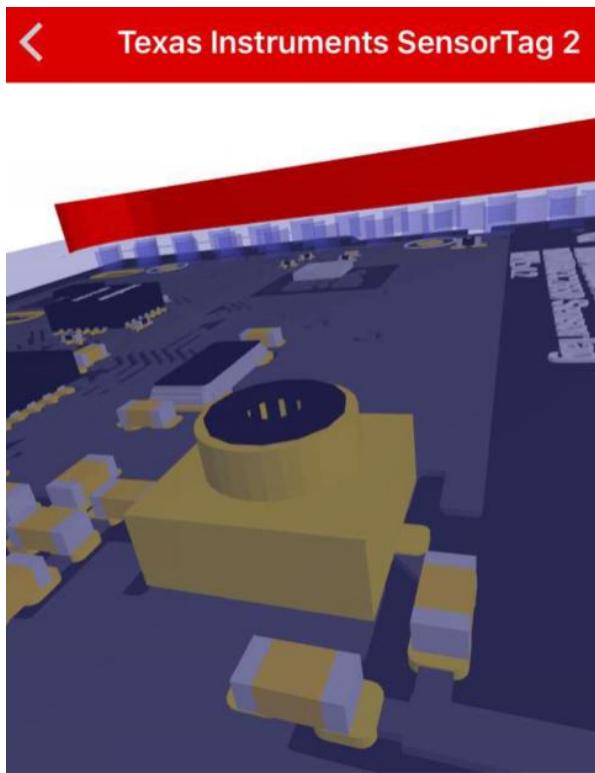


看，放大后从给传感器留的缝隙里可以窥见立体感很强的文字，那些
个洞洞也是立体的

< Texas Instruments SensorTag 2



这是内部的元件，高低错落，好像城市里的建筑



这是放大了很多倍的，好像是元件的管脚吧，是不是很神奇 !!!





开发板试用平台 <http://bbs.elecfans.com/try.html>

看完了，咱接着说 OTA，固件下载，小 T 支持无线模式下载固件，童鞋们都知道单片机烧写程序需要下载器烧写固件，但如果是开发产品都会面临一个比较头疼的问题，如果产品已经投放市场，又需要增加新的功能，或者修改 Bug，总不能把所有的产品全部召回，一个一个用 JTAG 重新烧吧？所以让用户能直接更新固件就非常重要了，傻瓜式的操作更为重要，不要期望用户们都具备程序猿的素质，21 世纪不会写代码就是文盲只是调侃，就算是会写代码 也不见得懂得烧写，这也不是丢人的事，术业有专攻，都社会化大分工了，分的越细说明社会越进步。

实现固件更新的基本原理就是单片机内置一个 Bootloader，可以提供 Flash 等内部存储的更新和引导，这样就可以通过外部接口（USB、蓝牙、web 界面等）加载新固件。

小 T 的“心”是 TI CC2650，我们可以把它理解为一个带有蓝牙协议栈的 51 单片机。TI 在 SenorTag 上实现的 OTA 功能，就是通过一个 51 的 bootloader 来控制固件的更新和加载。我们来看一下 OTA 升级固件的过程。

来到这个项，点“>”，进入固件下载，请童鞋们注意，现在版本是 1.20
(jul 20 2015)



开发板试用平台 <http://bbs.elecfans.com/try.html>



FW Download



Current FW rev. : 1.20 (Jul 20 2015)



进入下载设置页面，上半部显示的就是当前固件版本信息，中间有三个模式选择，“安全模式”“高速模式”“高级”，小心使得万年船，我们用安全模式，感觉“安全”的意思应该是传输速率比较低，数据验证比较严格吧，这样传输的固件数据可能比较 安全，然后点中间的“Select FW File”,选择固件文件



SensorTag 2.0 Current FW

Version : **1.20 (Jul 28 2015)**

Image Type :

Safe mode High Speed Advanced

Select FW File

0.0%

Block : ----- / ----- Bytes : ----- / -----

Speed : ---- B/s Connection interval : 22.50ms

Use notifications Yes

Timer delay 10 ms **- +**

Packets per timer / notif. 18 **- +**



开发板试用平台 <http://bbs.elecfans.com/try.html>

选上边这个，xxxxxx (BLE) 的，下边那个 xxxxxx (Zigbee) 的需要 zigbee 模块支持，即便是小 T 支持，我的 4s 也不支持，所以我 只能在 xxxxxx (BLE) 里选个最新版本的，这里有很多打勾的，这是说这个固件镜像和当前的小 T 是兼容的，“Compatible version”兼容版，所以为了安全起见，不兼容的就别搞了，搞出事来没人收拾。



点击然后就是酱紫了，下边是传输数据的状态信息，神马块信息，字节数，速率。。。等了四五分钟的样子，



SensorTag 2.0 Current FW

Version : 1.20 (Jul 20 2015)

Image Type :

Safe mode

High Speed

Advanced

Cancel

4.4%

Block 0360/8192 Bytes : 005760/1310...

Speed : 352 B/s Connection interval : 22.50ms

Use notifications

Yes



Timer delay

10 ms



Packets per timer / notif.

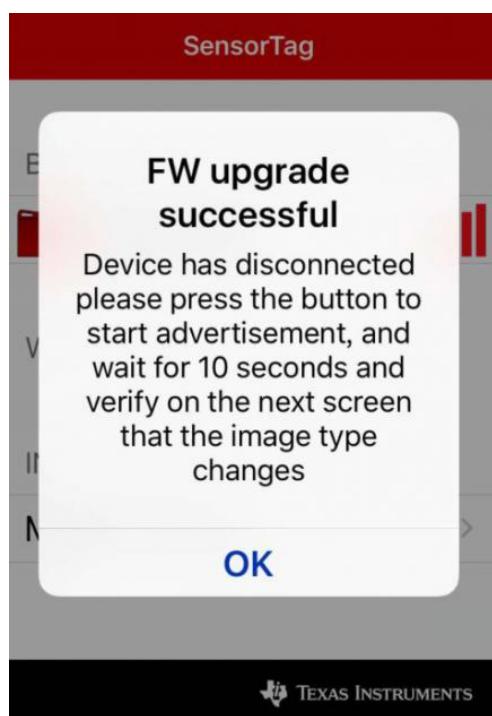
1



CC254xx

CC26xx

出来提示 ,“固件升级成功” ,说设备断开了 ,请按下按钮来启动设备 ,
等待 10 秒可以从屏幕上核对固件镜像版本的变化 ,大概是这个意思。



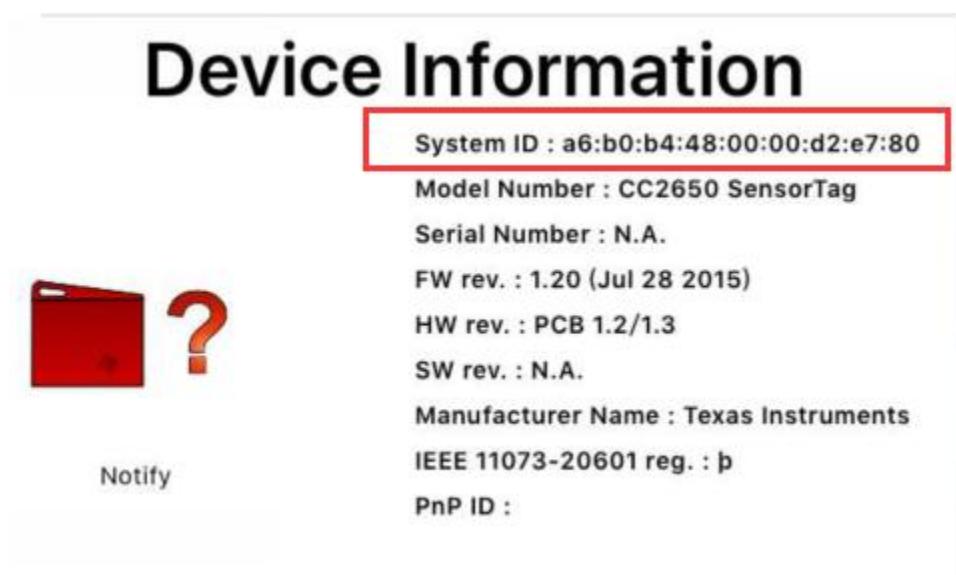


开发板试用平台 <http://bbs.elecfans.com/try.html>

重新连接后，看下版本信息，变成 1.20(jul 28 2015)，之前是 20



看下设备信息，system id 变化了，第一个字节变化了，后边没变，



之前是，显然，后边的 8 字节是设备号，前边的第一个字节和版本有关。



开发板试用平台 <http://bbs.elecfans.com/try.html>

< SensorTag

Edit



/20.00ms

Device Information

System ID : ff:b0:b4:48:00:00:d2:e7:80

Model Number : CC2650 SensorTag

Serial Number : N.A.

FW rev. : 1.20 (Jul 20 2015)

HW rev. : PCB 1.2/1.3

SW rev. : N.A.

Manufacturer Name : Texas Instruments

IEEE 11073-20601 reg. : þ

PnP ID :



Notify

其他一切正常。

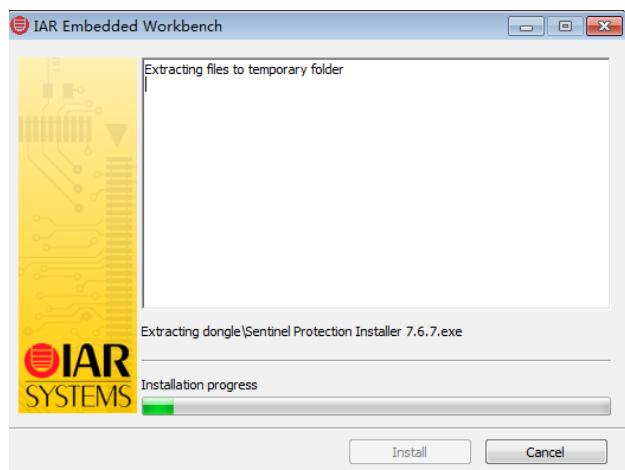
这回主要说了小 T 的 OTA 固件升级方法，本想再看看有线升级，好像寄过来的只有小 T 没有烧写用线，只好作罢。

开发环境搭建

作者：[@anger0925](#)

地址：http://bbs.elecfans.com/jishu_578241_1_1.html

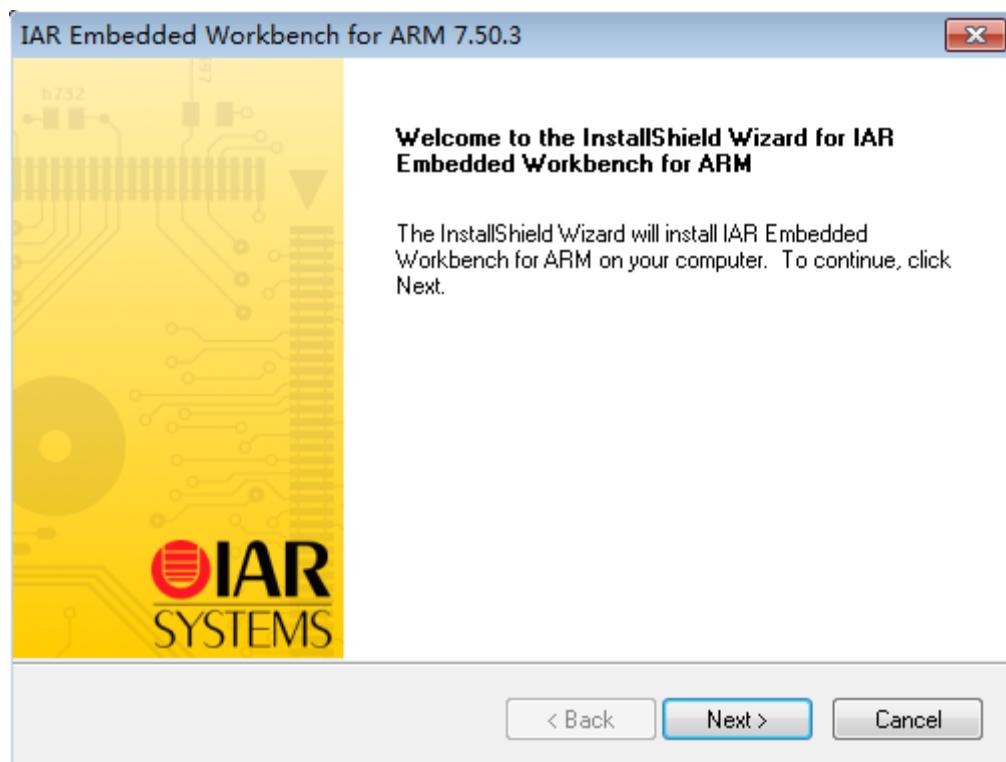
我这里安装 IAR FOR [ARM](#) 7.5 版本，目前是最新的版本，安装包在 IAR 官网上下载。



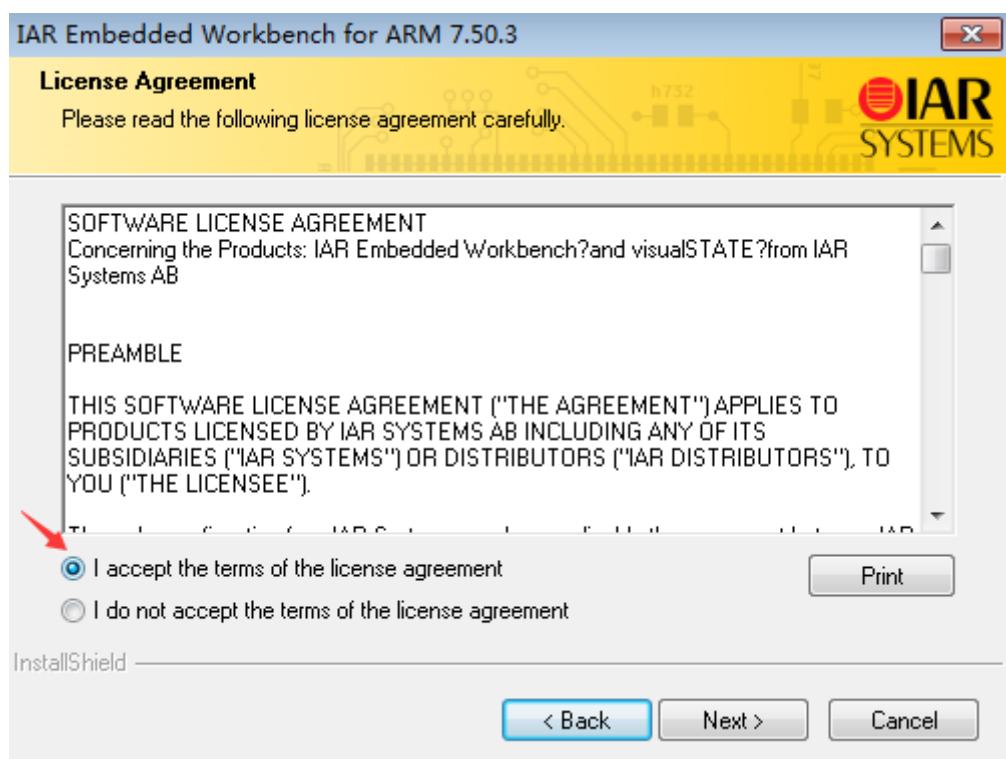
解压完成后，出现下图.



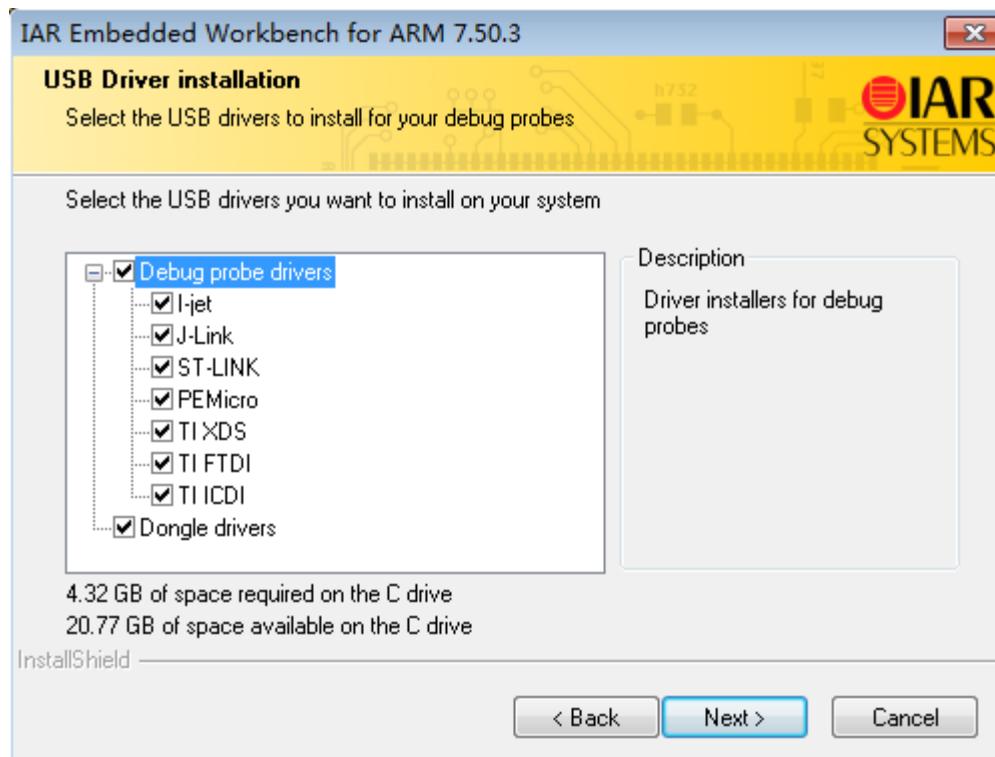
点击 Install IAR Embedded Workbench 进行安装。



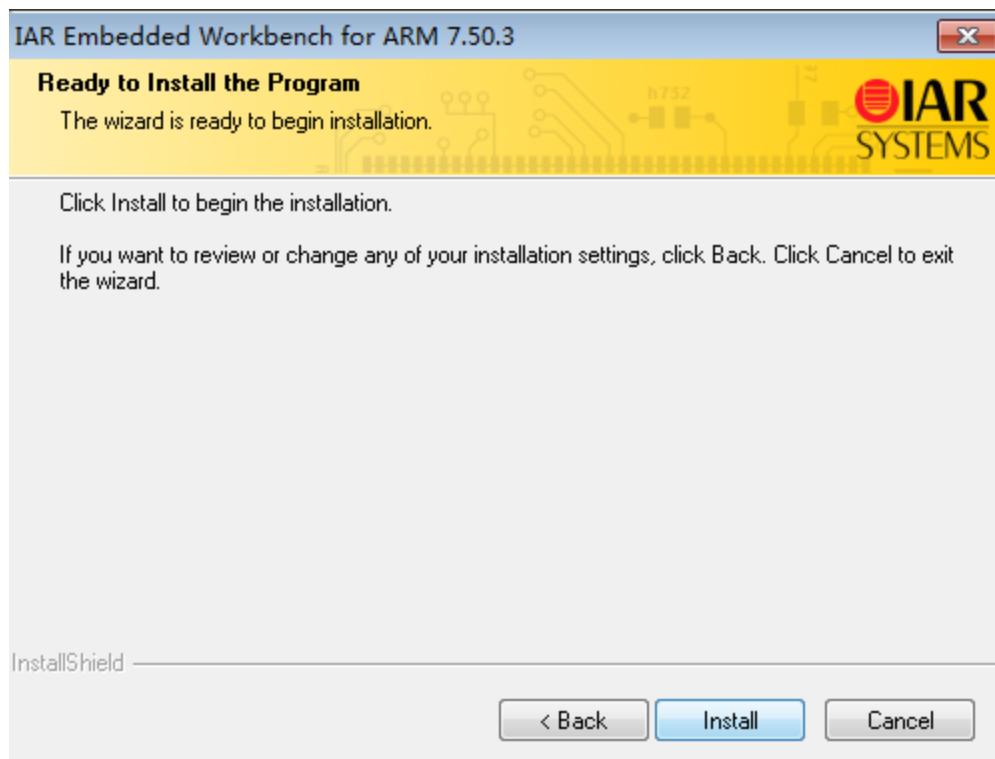
点击下一步，



选择同意。



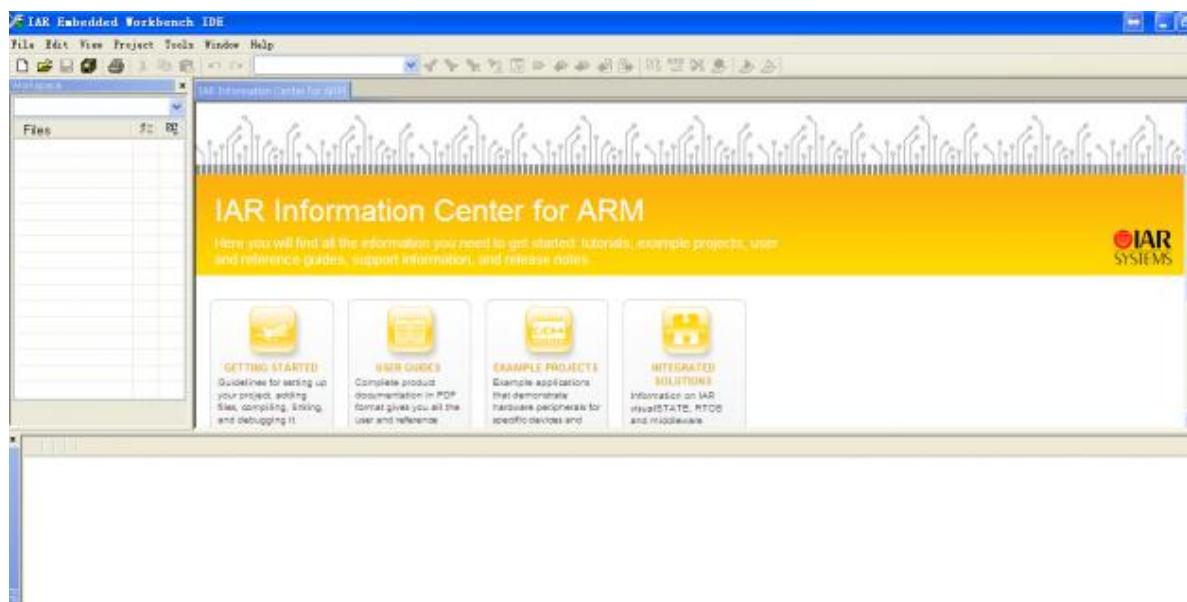
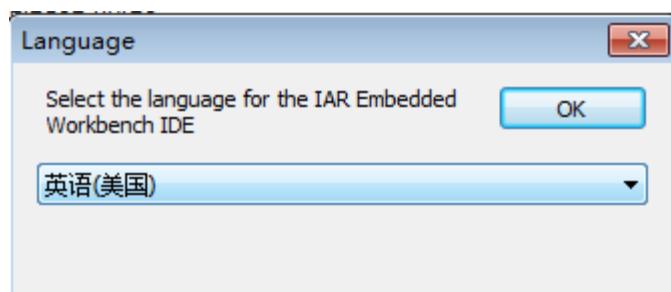
其他界面直接点击下一步，直到安装完成。





开发板试用平台 <http://bbs.elecfans.com/try.html>

等待安装完成。第一次启动会出现语言选择。我们选择英语。



安装完成之后 ,还需要破解 ,不然会有限制。破解文件我放在附件里 ,
按照附件破解就可使用。

除了使用 IAR 集成环境外 ,还可以使用 CCS ,可以在 TI 官网上下载 ,
ccs 是全免费的。



开发板试用平台 <http://bbs.elecfans.com/try.html>

BLE 协议栈的安装

作者 : @anger0925

地址 : http://bbs.elecfans.com/jishu_580082_1_1.html

1 , 到官网上下周协议栈安装包 , 及相关资料。

http://www.ti.com.cn/ww/wireless_connectivity/sensortag2015/index.html

The screenshot shows the TI SensorTag website interface. In the center, there is a section titled "多标准 SensorTag" (Multi-standard SensorTag) which includes a "设计文件 (BOM, 原理图, 3D 设计文件)" (Design files (BOM, Schematic, 3D Design)) button. To the right, under the "iOS 应用程序" (iOS Application) heading, there is a "BLE-STACK for SensorTag 固件" (BLE-STACK for SensorTag Firmware) button, which is highlighted with a red arrow. Below this, there are other options like "Z-STACK HOME for SensorTag 固件" and "6LoWPAN SensorTag 固件". The top navigation bar includes tabs for "SensorTag 展示" (SensorTag Showcase), "入门" (Getting Started), "解析" (Analysis), "DevPack" (Development Pack), "应用程序与合作伙伴" (Applications & Partnerships), and "社区与声音" (Community & Voice).

TI 主页 > 半导体 > 微控制器 (MCU) > Bluetooth 低能耗软件协议栈

China (CN)

Bluetooth 低能耗软件协议栈

(正在供货) BLE-STACK

描述/特性

技术文档

支持和社区

立即订购

| 器件型号 | 从德州仪器 (TI) 或第三方购买 | 通知我 | 状态 |
|--|----------------------|---------------------|--------|
| BLE-STACK: BLE-STACK (Support for CC2540/CC2541) | 获得软件 | 通知我 | ACTIVE |
| BLE-STACK-2-1-1: BLE-STACK V2.1.1 (Support for CC2640/CC2650) | 获得软件 | 通知我 | ACTIVE |

下载完成后的安装包是

ble_cc26xx_setupwin32_2_01_01_44627.exe。

2 , 安装协议栈



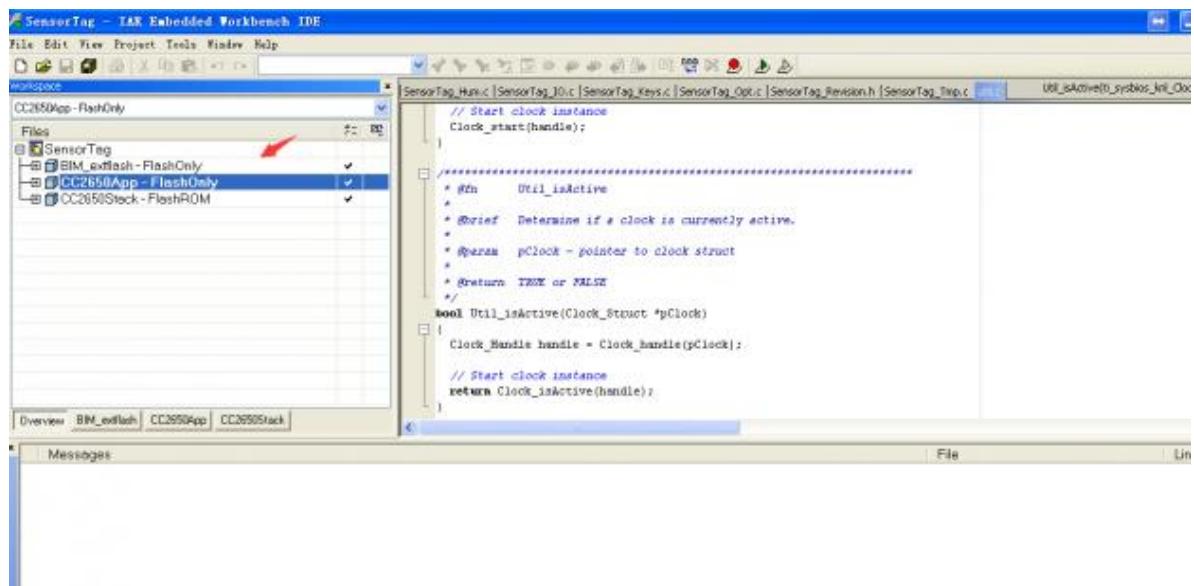
开发板试用平台 <http://bbs.elecfans.com/try.html>

直接双击安装就可以，但是必须安装在默认路径，如果修改了路径，会导致编译协议栈时候出错。具体后面讲解。

安装完成后在 simplelink\ble_cc26xx_2_01_01_44627\Projects\ble 目录下，有很多 TI 产品对应 ble 的项目。我们这里的是 SensorTag。所以进入 SensorTag。可以看到他有使用 CCS 和 IAR 建立的工程。我这里使用 IAR。上面章节已经说明 IAR FOR ARM 7.5 的安装。我就使用 7.5 的版本打开工程。

工程项目路径：

C:\ti\simplelink\ble_cc26xx_2_01_01_44627\Projects\ble\SensorTag\CC26xx\IAR



打开后，可以看到工程下面有：

CC2650App-FlashOnly

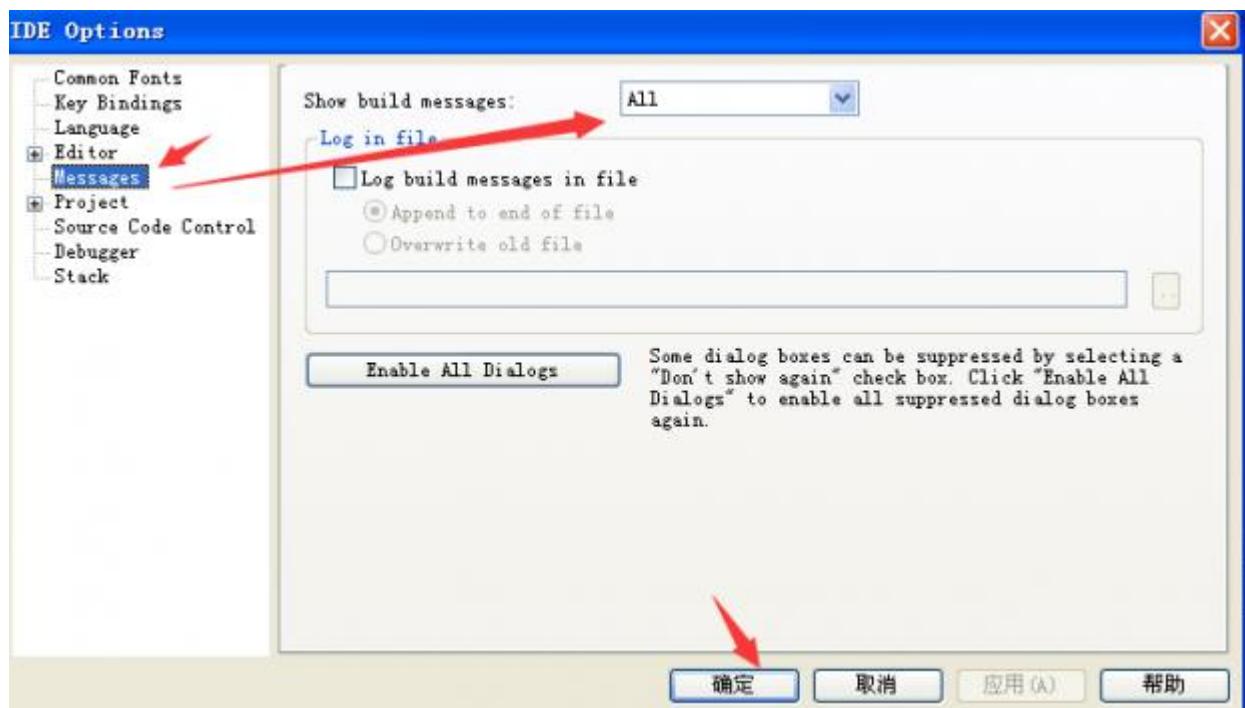
CC2650Stack-FlashROM

BIM_extflash-FlashOnly

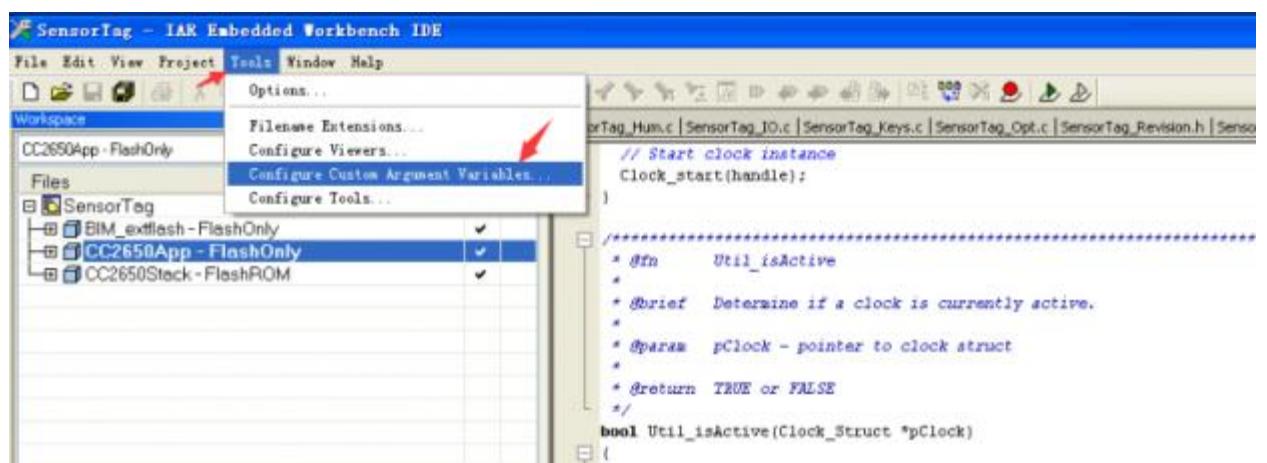
从结构上来看，TI 把协议栈与应用程序是分开有的，因为协议栈基本上不需要改动。我们需要实现的是在应用程序中修改，即 CC2650App-FlashOnly。打开之后进行编译。

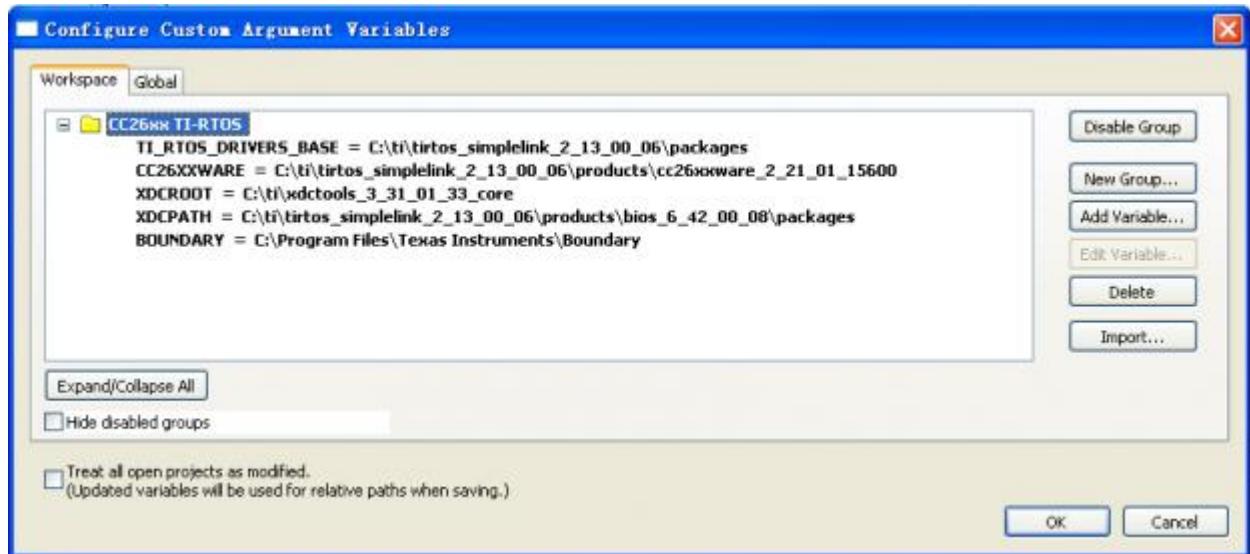
编译前设置一下：

一是打开编译所有输出信息，tools-options。



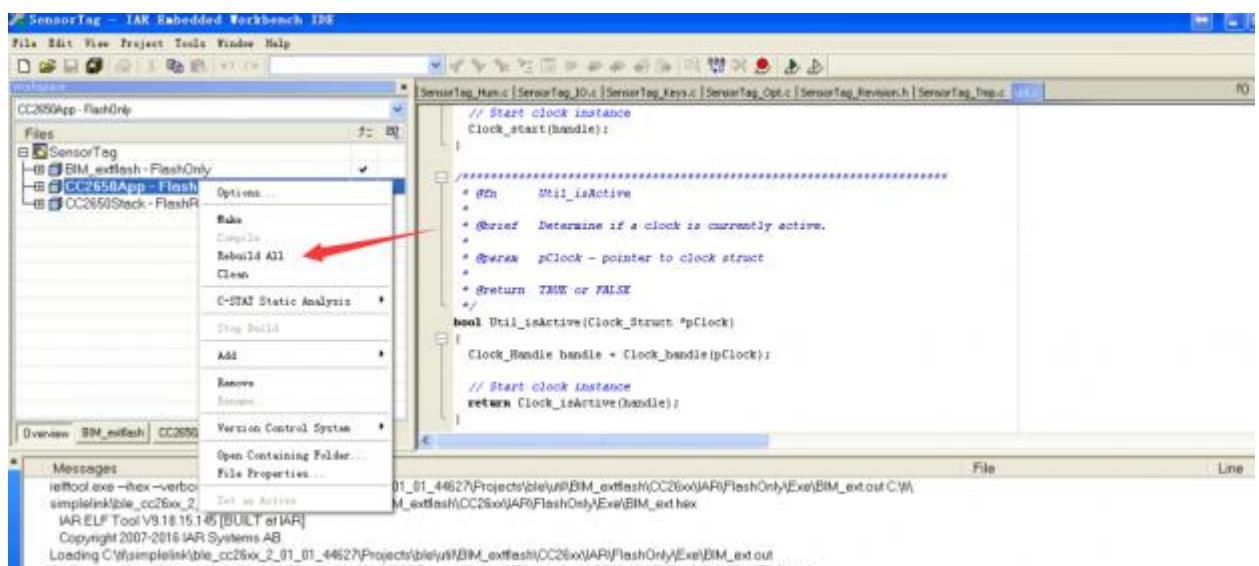
而是设置编译环境变量，





3 , 编译

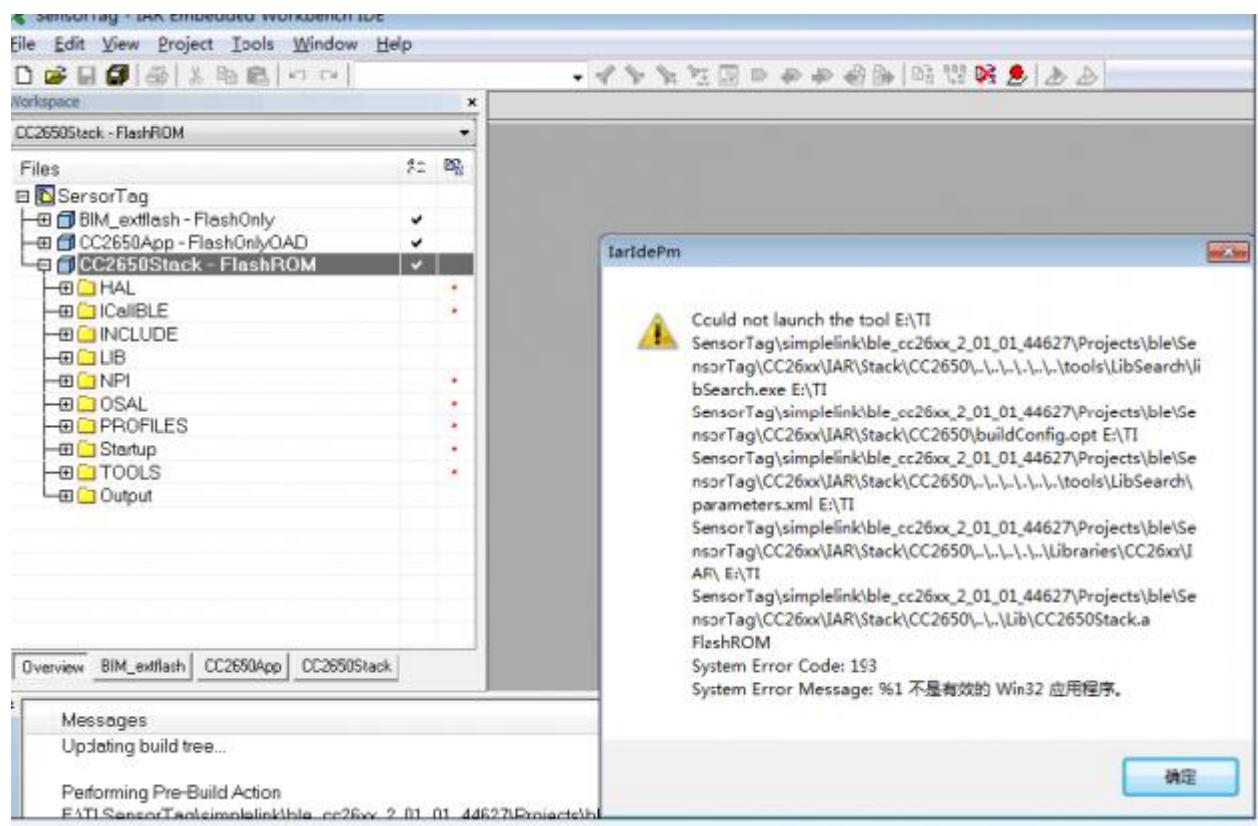
分别对每个子工程进行编译，右键我们选择 Rebuild All。



如果不安装在默认路径，编译 CC2650App-FlashOnly 和 BIM

extflash-FlashOnly 都没有问题。但是编译 CC2650Stack-FlashROM

时会出现错误。



所以必须安装在默认路径。也行修改路径后需要设置一些东西就可以解决。我没有去研究。

4，调试和下载

由于没有合适的仿真器进行调试和下载，所以我这里简单的说一下。

CC2650 的例子项目中，是把 APP 与 stack 分开的，正常情况下 stack 是不需要修改的，下载一次就可以了。用户一般就在 app 上修改和调试。TI 这里想得非常周到，这样使用升级，程序也会小很多。

只有把 CC2650Stack-FlashROM 和 CC2650App -FlashOnly 下载后，系统才能正常工作。之后，我们如果只修改 CC2650App -FlashOnly，我们就下载这个就可以了。



开发板试用平台 <http://bbs.elecfans.com/try.html>

服务浏览 (1)

作者 : [@birdinskydzfsy](#)

地址 : http://bbs.elecfans.com/jishu_577361_1_1.html

上一阶段较为详细的在 app 上看了又看传感器采集的数值，还有 3D 效果的，接下来看看纯字符的，Service Explorer,之前略略溜了一眼，就是这 11 个服务，有童鞋说了，这是 9 个，我这 4s 一屏截不全。。。

| < Service found on SensorTag... | |
|---------------------------------|---|
| 0 - Device Information... | > |
| UUID : 180a | |
| 1 - TI SensorTag Temper... | > |
| UUID : f000aa00-0451-4000-... | |
| 2 - TI SensorTag Humid... | > |
| UUID : f000aa20-0451-4000-... | |
| 3 - TI SensorTag Barom... | > |
| UUID : f000aa40-0451-4000-... | |
| 4 - TI SensorTag 2.0 M... | > |
| UUID : f000aa80-0451-4000-... | |
| 5 - TI SensorTag 2.0 Lig... | > |
| UUID : f000aa70-0451-4000-b... | |
| 6 - TI Simple Keys Servi... | > |
| UUID : ffe0 | |
| 7 - TI SensorTag 2.0 I/... | > |
| UUID : f000aa64-0451-4000-... | |
| 8 - TI SensorTag 2.0 Re... | > |

咱来看看这服务，这里也有好玩的，先看第一个，设备信息，还是点进去，凡是右边有“>”的，都可以点进去，好像是 dos 的 style



开发板试用平台 <http://bbs.elecfans.com/try.html>

| < Characteristics for Device Inf... | |
|-------------------------------------|---------|
| 0 - System ID | R W N I |
| UUID : 2a23 | |
| 1 - Model Number | R W N I |
| UUID : 2a24 | |
| 2 - Serial Number | R W N I |
| UUID : 2a25 | |
| 3 - Firmware Revision | R W N I |
| UUID : 2a26 | |
| 4 - Hardware Revision | R W N I |
| UUID : 2a27 | |
| 5 - Software Revision | R W N I |
| UUID : 2a28 | |
| 6 - Manufacturer Name | R W N I |
| UUID : 2a29 | |
| 7 - IEEE11073 | R W N I |
| UUID : 2a2a | |
| 8 - PnP ID | R W N I |

设备信息里有这么 9 个值 还是点进去第一个 发现没 这是 systemID,
只不过值的顺序是反的

< Back

SYSTEM ID

Read characteristic

Value : <80e7d200 0048b4b0>

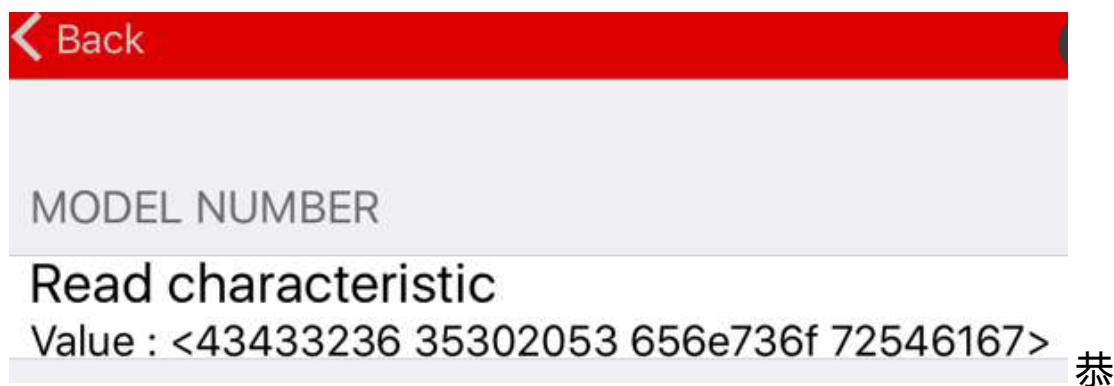
还记得上一篇升级完固件查看的设备信息，对照下 system id,后 8 个字节是设备号

Device Information

System ID : a6:b0:b4:48:00:00:d2:e7:80

看第二项，Model Number这个比较长，得横着看，这是撒？？？？

直觉告诉我，16进制表示的字符串



喜我，答对了!!!





开发板试用平台 <http://bbs.elecfans.com/try.html>

接下来，第三个，还是 16 进制的字符串 !!!

The screenshot shows a mobile application interface. At the top is a red header bar with a back arrow and the text "Back". Below the header, the text "SERIAL NUMBER" is displayed. Underneath it, the text "Read characteristic" is followed by "Value : <4e2e412e>".

恭喜我，都会抢答了 !!!

The screenshot shows a mobile application interface with a green header bar containing "操作", "工具", and "关于" buttons. Below the header is a menu bar with three items: "进制转换" (highlighted in blue), "字符转换", and "子网划分辅助". A text input field contains the hex value "4e2e412e". Below the input field is a status bar with the text "N.A."

接下来是固件版本信息，显示不全的也是 16 进制的字符串 !!!

The screenshot shows a mobile application interface. At the top is a red header bar with a back arrow and the text "Back". Below the header, the text "FIRMWARE REVISION" is displayed. Underneath it, the text "Read characteristic" is followed by "Value : <312e3230 20284a75 6c203238 20323031 3...>".

恭喜我，没显示全都会猜啦 !!



开发板试用平台 <http://bbs.elecfans.com/try.html>



好了，这种 16 进制字符串表示的东东咱就不一一看了，看多了就睡着了。

接下来咱们看第二个服务环境温度，就是这个，这个我们仔细点说，貌似其他很大一片服务都是如此。

< SensorTag Service found on SensorTag 2.0

0 - Device Information Service

UUID : 180a

1 - TI SensorTag Temperature Service

UUID : f000aa00-0451-4000-b000-000000000000

2 - TI SensorTag Humidity Service

UUID : f000aa20-0451-4000-b000-000000000000

3 - TI SensorTag Barometer Service

UUID : f000aa40-0451-4000-b000-000000000000

4 - TI SensorTag 2.0 Movement Service



开发板试用平台 <http://bbs.elecfans.com/try.html>

点进去，是酱紫，从上到下，是温度数据（只读看右边），温度设置

（读写），温度周期就是采样周期（读写）

Characteristics for TI SensorTag Temperature Se

0 - Temperature Data

R W N I

UUID : f000aa01-0451-4000-b000-000000000000

1 - Temperature Configuration

R W N I

UUID : f000aa02-0451-4000-b000-000000000000

2 - Temperature Period

R W N I

UUID : f000aa03-0451-4000-b000-000000000000

看温度数据先，进去酱紫，有两个项目，上边数据下边是开关，数据是空值，这是因为没读还，需要点一下“Set notify state...”，

Back

TEMPERATURE DATA

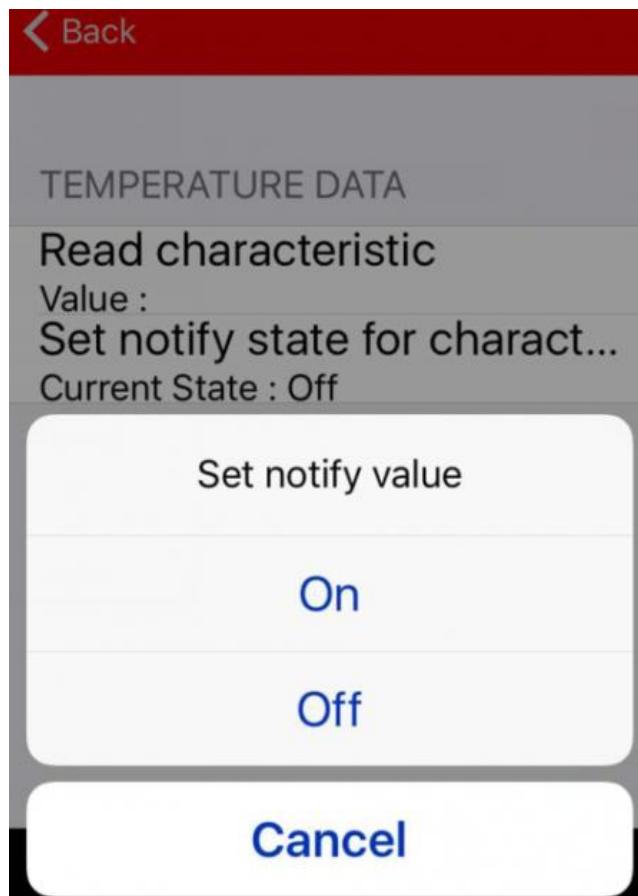
Read characteristic

Value :

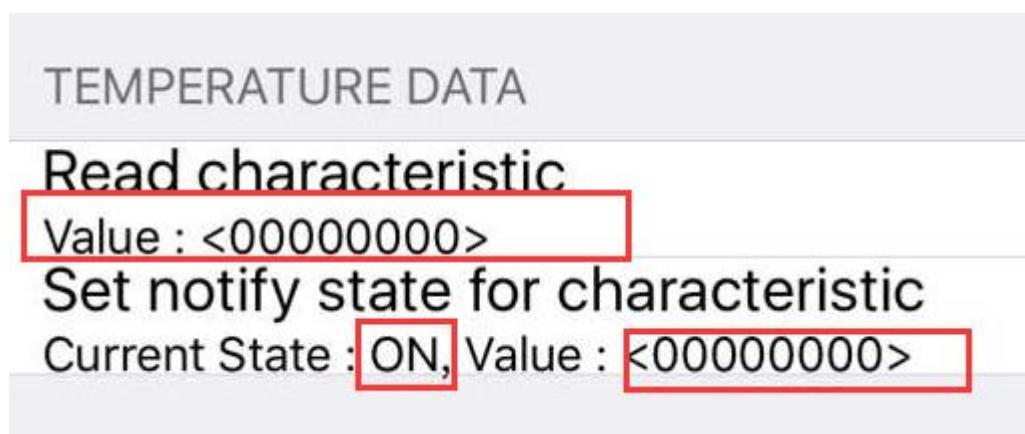
Set notify state for characteristic

Current State : Off

这里出来菜单，戳“On”，打开显示值

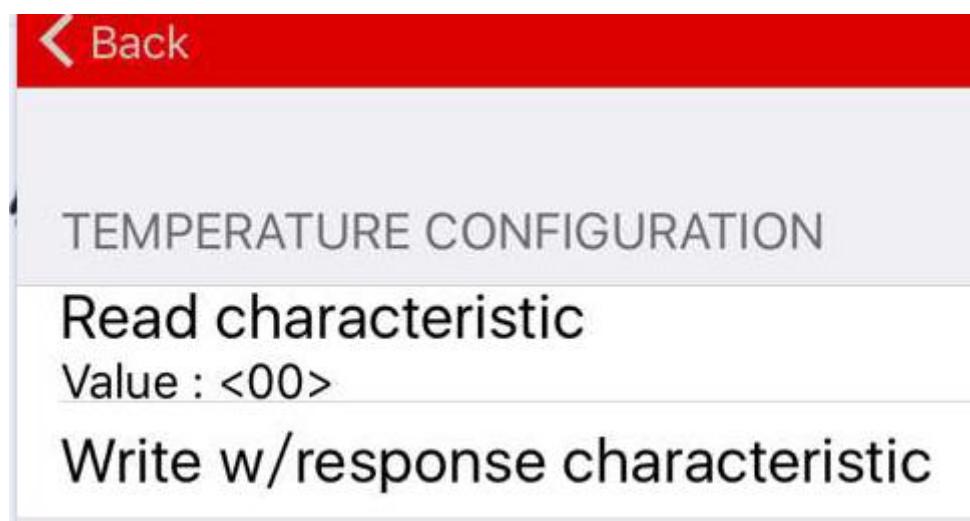


值出来了，全是 00000，为啥呢，因为这块可能读取的是温度寄存器里的值吧，温度还没采集呢，没值

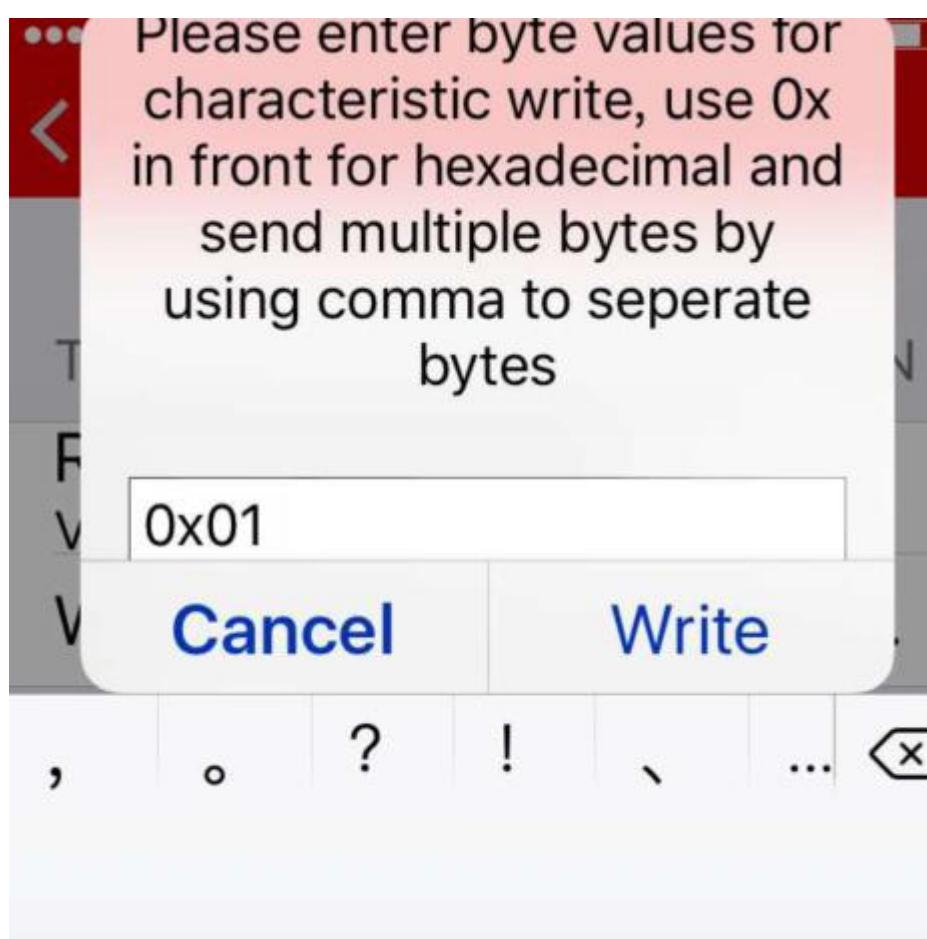


要想有值看第二个温度设置，戳进去。

这里也有两个选项，上边是读值，下边是写值，这个值很简单，这应该就是类似特殊功能寄存器，把某位置 1 就是打开，置 0 就是关闭



我们写 1，注意，这鸟语是说，你写的值前边要加上那个 0x 前缀，





< Back

TEMPERATURE CONFIGURATION

Read characteristic

Value : <01>

Write w/response charact...

然后回头看看温度值有了没有，可以看到能显示值了，隔一秒刷新一次，<8c0bb80b>，最后的 0b 应该是个特定字符表示值传输完毕吧，前边的值经过变换应该就转换成温度值

TEMPERATURE DATA

Read characteristic

Value : <ac08880b>

Set notify state for characteristic

Current State : ON, Value : <ac08880b>

TEMPERATURE DATA

Read characteristic

Value : <8c0bb80b>

Set notify state for characteristic

Current State : ON, Value : <8c0bb80b>



开发板试用平台 <http://bbs.elecfans.com/try.html>

还差一个，采样周期，看第三项，

戳进去，酱紫，上边值，下边写值，初始值 0x64，貌似这个是 1 秒一刷，

TEMPERATURE PERIOD

Read characteristic

Value : <64>

Write w/response charact...

改成 0xC8，貌似两秒一刷，改成 0x32 貌似一秒两刷，

TEMPERATURE PERIOD

Read characteristic

Value : <c8>

Write w/response charact...

OK，其他这类读取传感器值的服务都是类似的，TI 这个 App 做的还挺细致的，嘎嘎嘎，此篇完结。



开发板试用平台 <http://bbs.elecfans.com/try.html>

服务浏览 (2)

作者 : @birdinskydzfsy

地址 : http://bbs.elecfans.com/jishu_577746_1_1.html

接着说 , 再看一个 I/O Service , 这个是双向控制的 , 尽管其他的“xxxx Config”里也都是配置寄存器的值 , 这个不同的就是控制小 T 上的两个灯和蜂鸣器。

< SensorTag Service found on SensorTag 2.0

- 4 - TI SensorTag 2.0 Movement Service
UUID : f000aa80-0451-4000-b000-000000000000
- 5 - TI SensorTag 2.0 Light Sensor Servi...
UUID : f000aa70-0451-4000-b000-000000000000
- 6 - TI Simple Keys Service
UUID : ffe0
- 7 - TI SensorTag 2.0 I/O Service
UUID : f000aa64-0451-4000-b000-000000000000
- 8 - TI SensorTag 2.0 Registe Service
UUID : f000ac00-0451-4000-b000-000000000000
- 9 - TI Connection Control Service

戳进

去 , 有两项 , 上边是数据下边是控制 , 下边这个就是相当于个总开关 ,
先戳进去看看 ,

< Characteristics for TI SensorTag 2.0 I/O Service.. X

- 0 - I/O Data R W N I
UUID : f000aa65-0451-4000-b000-000000000000
- 1 - I/O Config R W N I
UUID : f000aa66-0451-4000-b000-000000000000



开发板试用平台 <http://bbs.elecfans.com/try.html>

酱紫的，跟以前一样的操作，把值设置成<01>，这就是打开总开关了

< Back

I/O CONFIG

Read characteristic

Value : <01>

Write w/response characteristic

接下来，返回到“I/O Data”看看，这里边设置值就能控制俩 led 和蜂鸣器，这个<70>，设置对了灯就亮了，

< Back

I/O DATA

Read characteristic

Value : <70>

Write w/response characteristic

经过测试，发现，“7”没用，寄存器的后三位控制依次控制蜂鸣器、绿 led、红 led 也就是，7 照发，0000B，后三位是开关，其他值无所谓，至少对我们的 led 和蜂鸣器无所谓比如让蜂鸣器响，那就是 0100B，数字 0x74 发出去，蜂鸣器就会嗡嗡作响，红绿灯同时亮且蜂鸣器不响，就是 0011B，发 0x73，如此等等不再举例子。



开发板试用平台 <http://bbs.elecfans.com/try.html>

既然这样，那么凡是具备满足这种操作的 App 都能用，所以我们大胆试一下。其实这样的 app 也蛮多的。这是 Punch Through 开发的一款 App，去年研究 ble 下载了这东，近期又有更新，要不是试用小 T，这东又被尘封了。



打开 App，先是





开发板试用平台 <http://bbs.elecfans.com/try.html>

然后，我们发现了小 T，不同的是，我们的小 T 叫 irr，这里没有，估计不是保存在 SensorTag 的 App 里了就是写在某个不知道的位置了，按照惯例截一下，

The screenshot shows the LightBlue Explorer application interface. At the top, there are tabs for 'Info' and 'LightBlue Explorer +'. Below that, a section titled 'Peripherals Nearby' lists a single entry: 'SensorTag 2.0' with a signal strength of '-49' and '1 service'. Under 'Virtual Peripherals', there is an entry for 'Alert Notification' with '1 service'. At the bottom of the screen, there are 'Log' and 'PunchThrough' buttons.

接着，连接中.....。





开发板试用平台 <http://bbs.elecfans.com/try.html>

出来了，看着有点乱，但是仔细看看，不乱，还是刚才的东西，这是设备信息，只是显示形式换了，

The screenshot shows a mobile application interface for a SensorTag 2.0 device. At the top, there are three buttons: 'Back' with a left arrow icon, 'Peripheral' (highlighted in blue), and 'Clone'. Below this is the device name 'SensorTag 2.0'. Underneath the name is the UUID: 'UUID: FF67F043-7235-D1A9-B305-350879AF0253'. The status 'Connected' is displayed. A section titled 'ADVERTISEMENT DATA' has a 'Show' button next to it. The main content area is titled 'Device Information' and lists several items with disclosure arrows: 'System ID <80e7d200 0048b4b0>', 'Model Number String CC2650 SensorTag', and 'Serial Number String'. At the bottom of the screen are two buttons: 'Log' and 'PunchThrough' with a blue fist icon.

我们往下翻翻看看有没有一眼能看懂的，出来了，显然这是按键的，于是知道了，这些个服务一拉溜都列出来了，只显示了 UUID，SensorTag 的 App 里整理了一下，看着比较整齐而已。



开发板试用平台 <http://bbs.elecfans.com/try.html>

< LightBlue Peripheral Clone

0xF000AA72-0451-4000-B000-000000000000 >
Properties: Read Write

0xF000AA73-0451-4000-B000-000000000000 >
Properties: Read Write

UUID: FFE0

Key Press State
Properties: Notify
UUID: FFE1

UUID: F000AA64-0451-4000-B000-000000000000

0xF000AA65-0451-4000-B000-000000000000 >
Properties: Read Write

0xF000AA66-0451-4000-B000-000000000000 >

Log

二话不说戳进去，看了看，果断点了下中下偏右部的 label 按钮，开始监听，然后按了小 T 上的用户键，又按了电源键，于是中下部出现了十六进制值和时间，貌似用户键按下是 0x01, 电源键是 0x02, 同时按下是 0x03，干簧管是 0x04，不管神马键抬起都有一个 0x00，

< SensorTag 2.0 0xFFE1 Hex

SensorTag 2.0

Key Press State
UUID: FFE1
Connected

NOTIFIED VALUES 监听点这个

Listen for...otifications

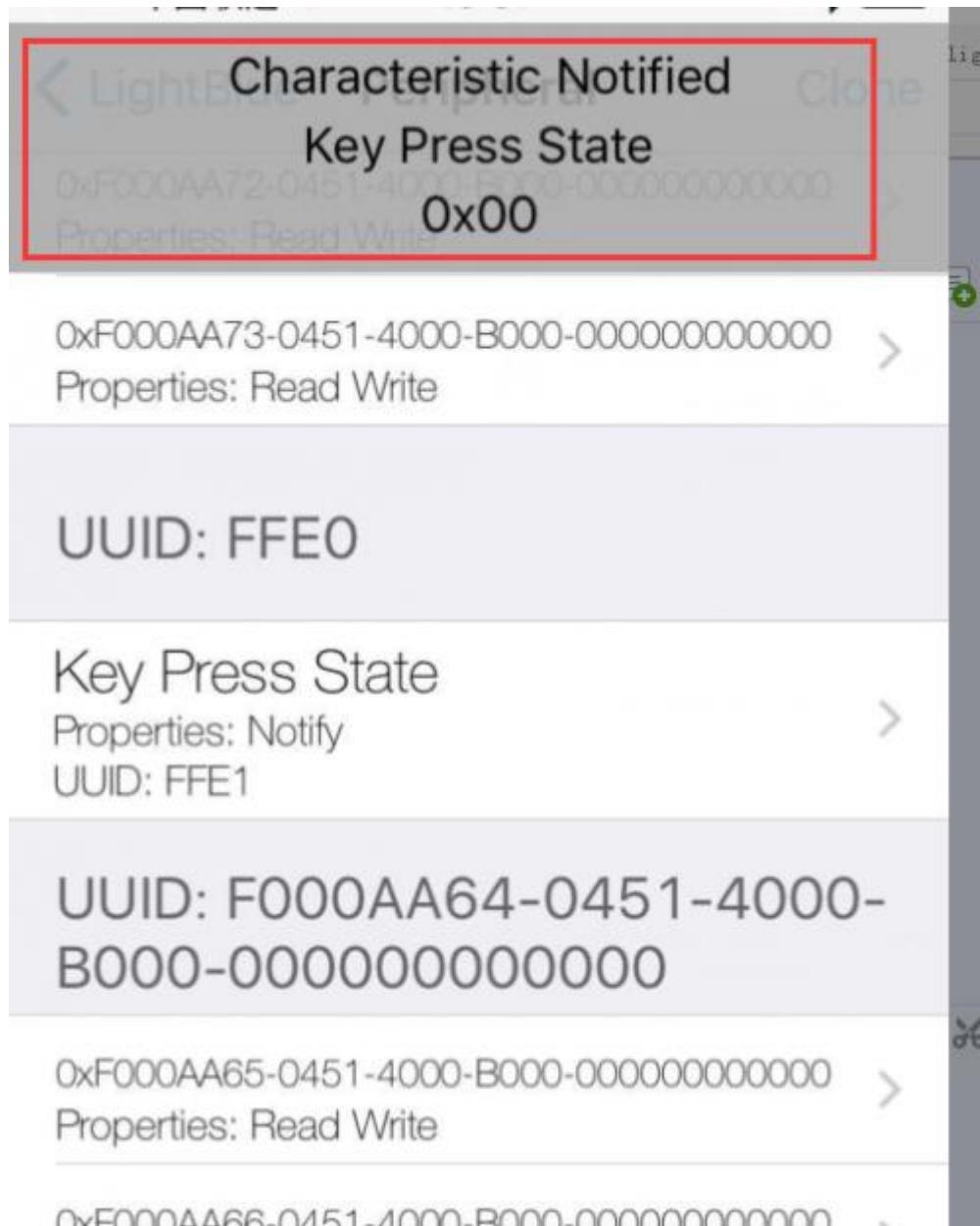
0x00
16:29:40.359

0x02
16:29:40.209

0x00
16:29:38.649

Log

在这个地方按，会出现漂浮的提示，这东东做的也挺不错的，



我们再看看其他的，我们找那个 IO 的看看，根据 UUID 即可，前边 SensorTag 的 a p p 里看一眼 UUID，就看前边几个就行了，f000aa64-XXXXXXXXXX，原来就是刚才试的 Key Press State 下边那个，UUID 为 f000aa**64**-XXXX...、f000aa**65**-XXXX...、f000aa**66**-XXXX...，这三



开发板试用平台 <http://bbs.elecfans.com/try.html>

个是 IO Data Service , 65 是小开关 , 66 是总开关我猜 , 戳 65 看看 ,

果然 , 看见了 0x7F 比较熟悉 , 那得打开总开关先 ,

< 0xF000AA65-0451-400... Hex

UUID: F000AA65-0451-4000-B000-000000000000

Connected

READ VALUES

Read again

0x7F
16:44:01.105

WRITTEN VALUES

Write new value

现在是 66 , 总开关 , 这个 App 做的有优点 , 写过的几个值有保存 ,

如果还用 , 可以直接戳 ,

< 0xF000AA66-0451-400... Hex

0x01
16:46:14.334

0x00
16:46:11.155

0x01
16:43:47.875

0x00
16:43:27.652

WRITTEN VALUES

Write new value 点这个写新的值

0x00
16:46:18.228 如果不需要新值直接戳 ⓘ

0x01
16:46:14.215 ⓘ

The note '如果不需要新值直接戳' is highlighted with a red box.



开发板试用平台 <http://bbs.elecfans.com/try.html>

值的编辑页面也好，直接输入值，这是 65 小开关，分别控制两个 led 和蜂鸣器的，十六进制的有 abcdef,

< Back Edit Value

Hex

| | | |
|----|------|---|
| 73 | | |
| D | E | F |
| A | B | C |
| 7 | 8 | 9 |
| 4 | 5 | 6 |
| 1 | 2 | 3 |
| 0 | Done | |

要是用二进制的就 0 和 1

< 0xF000AA65-0451-400... Bin

SensorTag 2.0 这里选择显示什么数制

0xF000AA65-0451-4...

UUID: F000AA65-0451-4000-B000-000000000000

Connected

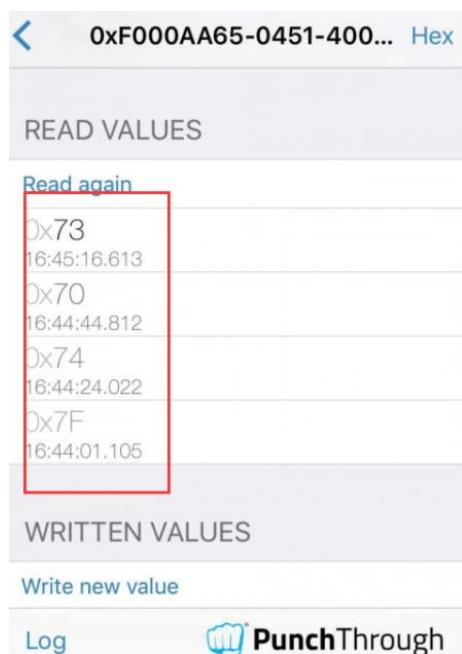
READ VALUES

Read again

0b01111111
17:52:54.381



这是输过的值，控制 led 和蜂鸣器只需点下这些值就行了，貌似方便点，要是换成二进制显示，很方便猜出哪位控制啥元件。



其他的就不看了，都类似，OTA 的会有个警告信息，提醒你别整错了，大概就这样子。

这篇我们又对这 Service 应用部分加深了理解，换了个 app 测了一下，收获大大滴。

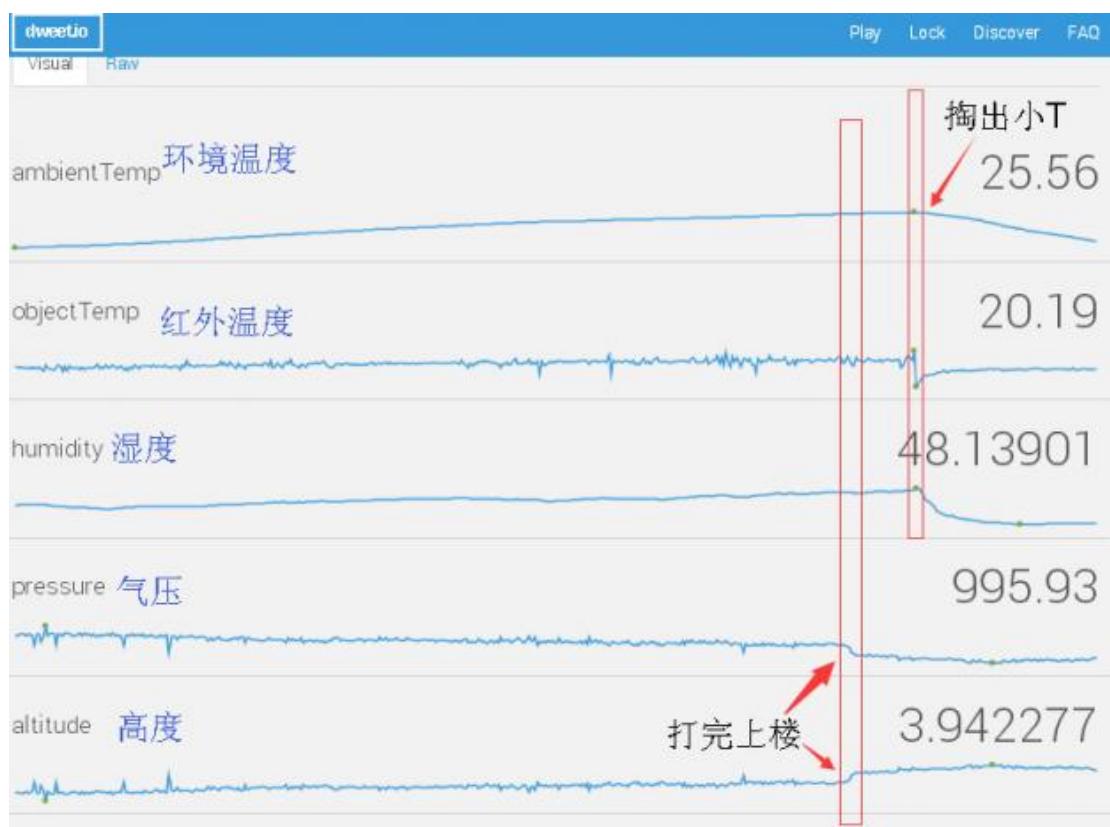
运动生理监测：羽毛球

作者：[@birdinskydzfsy](#)

地址：http://bbs.elecfans.com/jishu_580194_1_1.html

今天打了半小时左右的羽毛球，打之前想起了小 T，小 T 眼中打场羽毛球是什么样子呢？于是装备上阵，打开蓝牙和 app，打开云，设置 3 秒钟传送一个数据包，下去打球。

回来了，于是，上图，云太好了，



我们看看这上边这个图，记录了半小时左右环境的变化，我把小 T 装到兜里了，可以看出：

环境温度逐渐上升，因为运动产生了热量，随着散热，身体周围在不断升温，小 T 发现了这一趋势；



开发板试用平台 <http://bbs.elecfans.com/try.html>

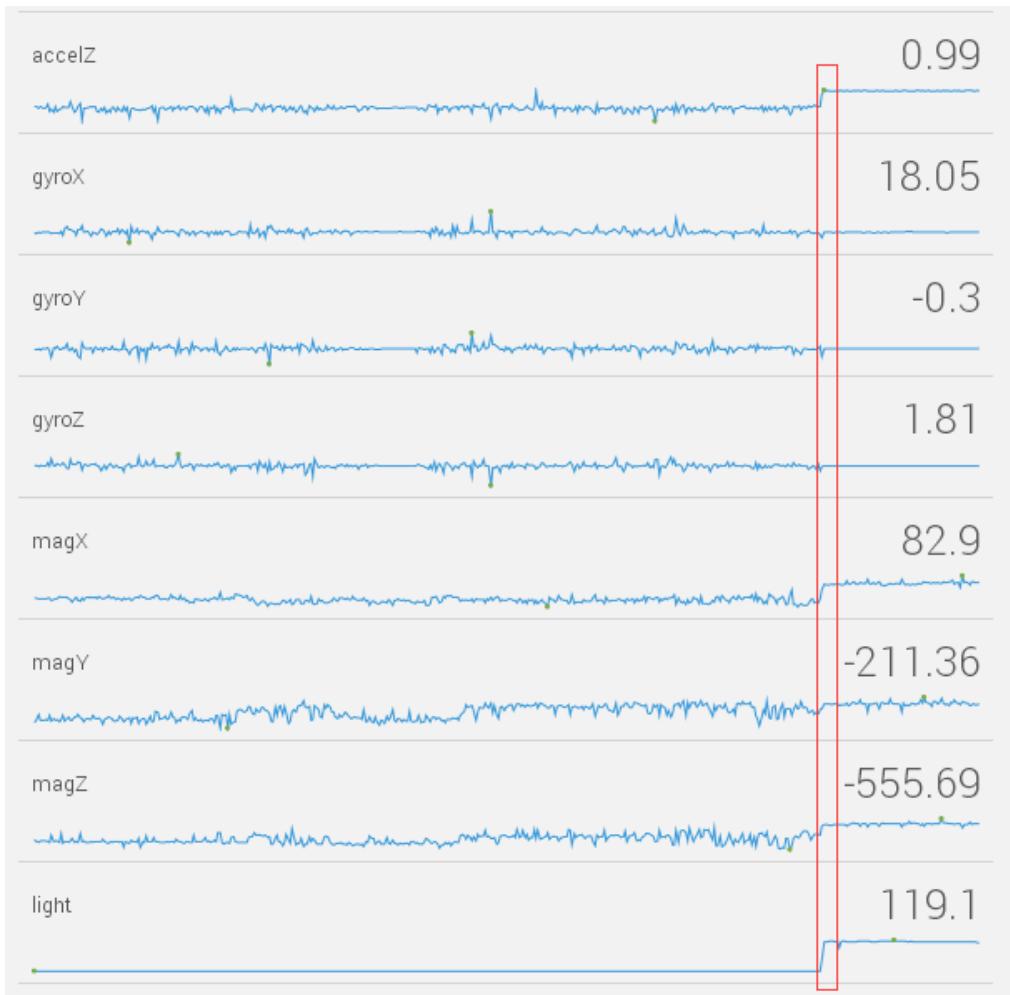
红外温度显示趋势比较平稳，但有很多忽高忽低的波动，我们也许可以理解为运动中的动作使小 T 离身体忽远忽近，以红外测温显示，身体散热良好，体温恒定，散热系统工作良好；

湿度开始表现了一个小的下降趋势，说明运动中空气流动带走了水分，湿度降低，后来又随体温上升而湿度增加，说明身体散热机制开始只动用了辐射形式，后来启动了排汗形式加速散热，于是湿度升高；

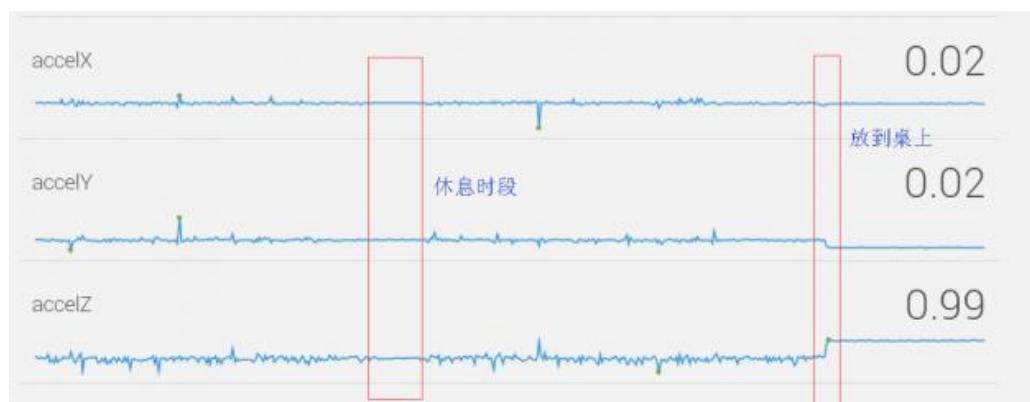
运动过程中，室外的环境有所变化，气压逐渐变低了，由气压推算出的高度也随之变化表现为高度变高；

打完球上楼，气压和高度有比较剧烈的变化，从兜里掏出小 T，温度和湿度也明显变化，因为环境发生了变化从紧贴身体到放到桌子上。

在看下图的几个参数，最后一个光照，掏出小 T 后光照陡然上升，放到桌上后，三轴上的重力加速度基本上不变化了，三轴上的磁场在被掏出后有个比较明显的变化，然后趋向平稳，因为小 T 的姿态由立在兜里变成了平躺在桌上，在运动过程中，除了光照为 0 其他磁场和重力三轴都一直在变化，磁场的值静置小 T 时，各轴上都会有 2uT 左右的变化，也许是空间电磁波扰动的吧。



下边是三轴加速度，因为中场休息了一段时间，表现为中间都有一段比较平滑的线，后来放到桌上又继续平滑了，上图的重力三轴中间也有一段平滑区。



小 T 可以用各种方式来陪你玩耍，看你有撒想法了。



看源码

作者：[@birdinskydzfsy](#)

地址：http://bbs.elecfans.com/jishu_581025_1_1.html

前边写了好些个帖子，基本上都是从 Customer 的角度来写的，也就是说，我买了产品，我要用，我们用起来确实是挺高大上的，200 多大洋的话其实性价比挺高的。

然而，作为 Coder，做为有追求的程序猿，没有见到代码是不死心的，为代码而生，把 bug 灭绝，始终是我们不变的目标，我们冰清玉洁般的小心灵始终朝向构筑优雅代码的大道。

好吧，由于没有下载器，没有线，只有小 T，巧男难为无米之炊，我们只能看看代码了。其实有下载器也好无下载器也好，烧写也只是整个链条上的一个环而已，我们可以跳过这个环节，看看更多的其他内容。

从官网下载源码，<http://www.ti.com.cn/ww/wireless...g2015/tearDown.html>，这里有文档、源码和开发工具，因为 ccs6 之前已经装了，只下载源码，源码分 app 的和小 T 的，我们看小 T 的，安卓的 app 给出了源码也，有时间我们看看。



开发板试用平台 <http://bbs.elecfans.com/try.html>



小 T 源码 ,

<http://www.ti.com/tool/cn/ble-stack?DCMP=wbu-blestack&HQS=ble-stack> , 选下边的 , CC2650 , TI 下载貌似要注册 , 问你干啥用 , 商用还是军用 , 你随便填就行了。

(正在供货) BLE-STACK

描述/特性 技术文档 支持和社区

立即订购

| 器件型号 | 从德州仪器 (TI) 或第三方购买 | 通知我 | 状态 |
|--|-------------------|-----|--------|
| BLE-STACK: BLE-STACK (Support for CC2540/CC2541) | 获得软件 | 通知我 | ACTIVE |
| BLE-STACK-2-1-1: BLE-STACK V2.1.1 (Support for CC2640/CC2650) | 获得软件 | 通知我 | ACTIVE |



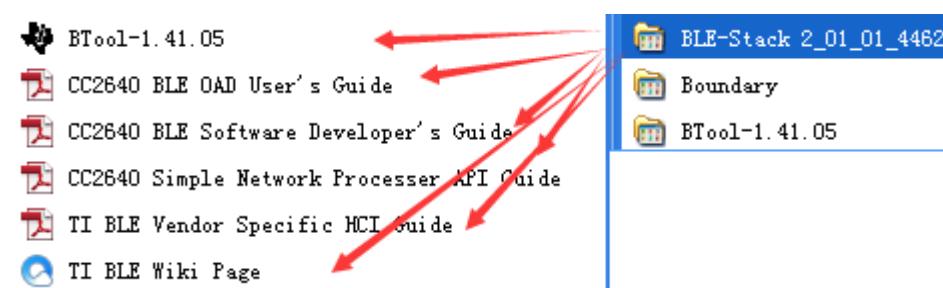
下载完成 , 装 ! 其实我比较抵触装 C 盘 , 但好些个带 demo 的开发套

装没办法就默认 c 盘，唉，你要是选别的盘吧也行，你得熟悉 ide 里边配置路径的东，没准哪儿就找不着路径了，玩耍过 makefile 的童鞋都知道，里边有变量啊、通配符啊，不知道就指到哪儿去了，所以别改了，装 c 盘吧，别因为改个路径耽误半天，我们是非专业人员也不常用。

安装过程中要点几次下一步啊确定啥的，默认即可。装完了，我们慢慢看下都折腾了撒，下边的仨是新装的，之前装过 Monitor 也没用，没有线



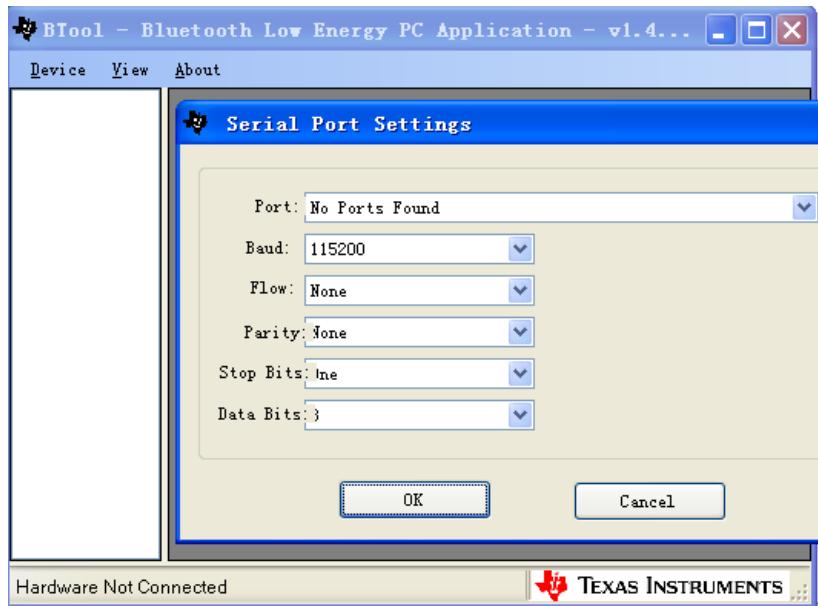
BLE-Stack 这个里有文档，还有个工具，



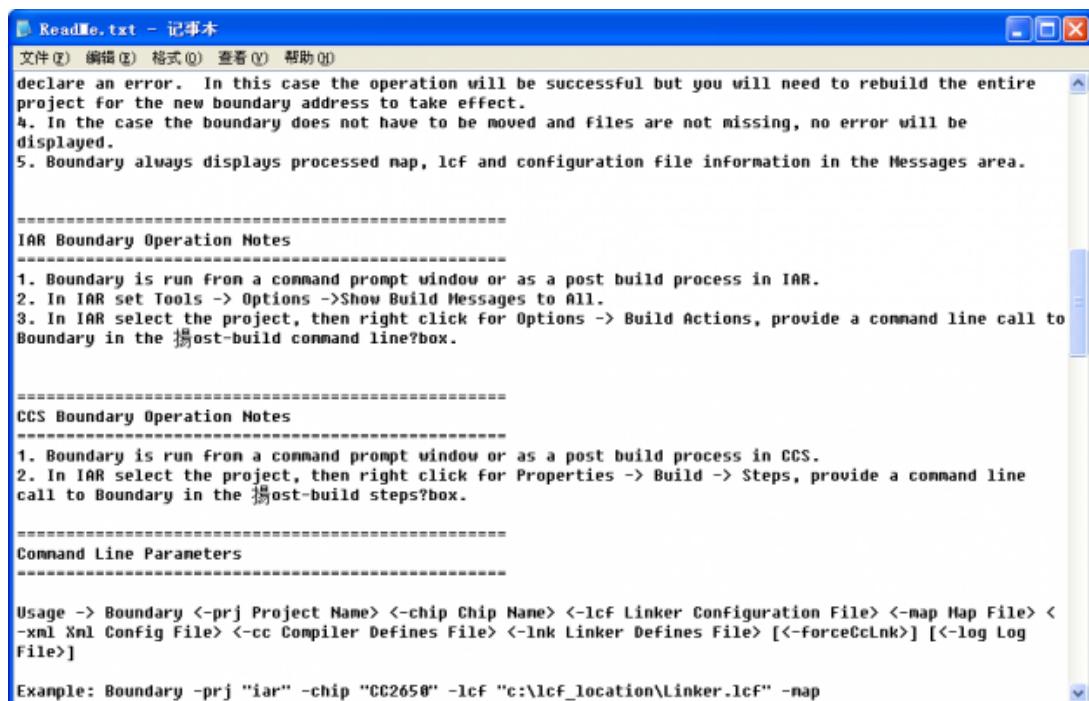
这个工具八成得需要下载器才可以用，



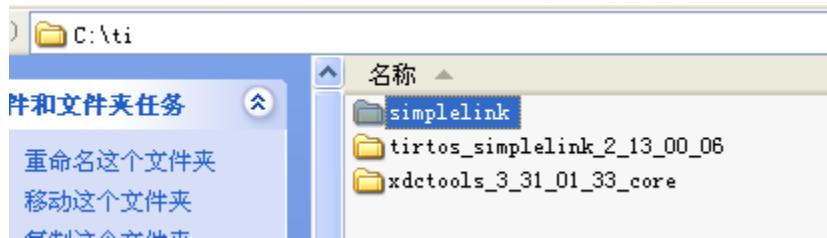
开发板试用平台 <http://bbs.elecfans.com/try.html>



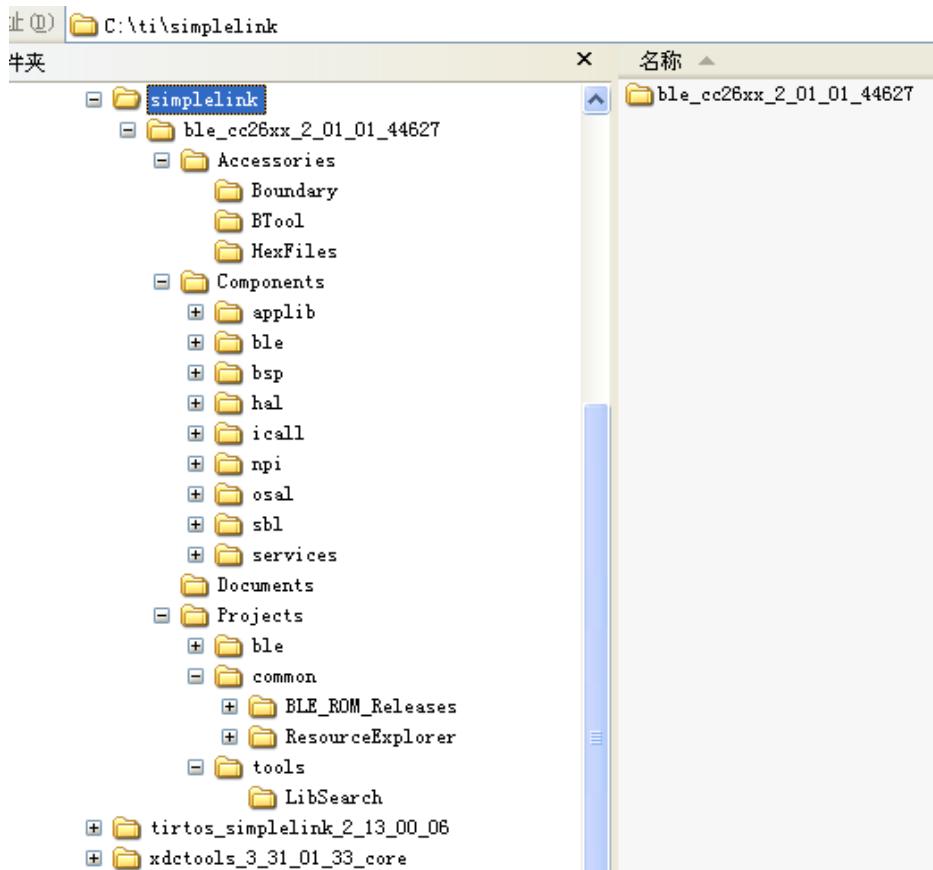
Boundary 里边只有一个 readme , 没找着我们需要的源码啊 ? 看看文件夹吧 ,



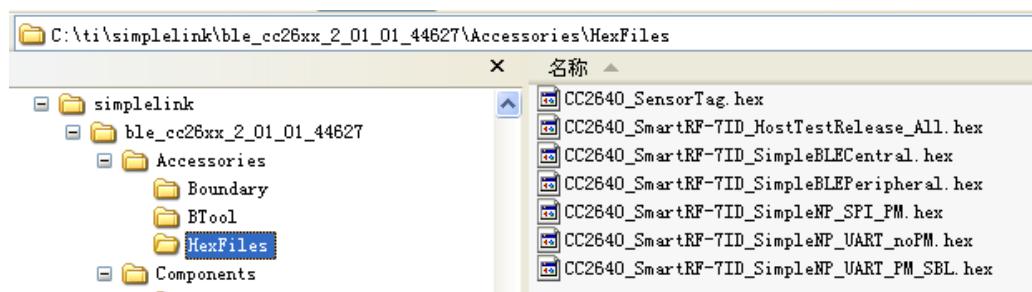
这回有点意思了 , 我们发现了 simplelink , 小 T 不就是 Simplink SensorTag 麽 ,



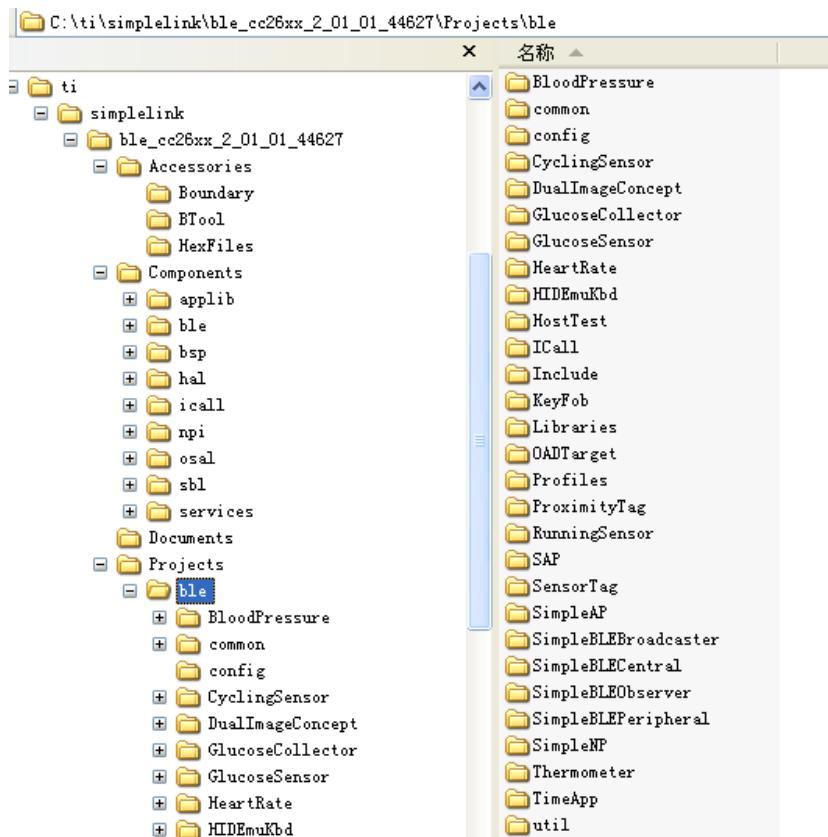
果然找到了，这里边有编译好的文件，也有源码



HEX 文件，



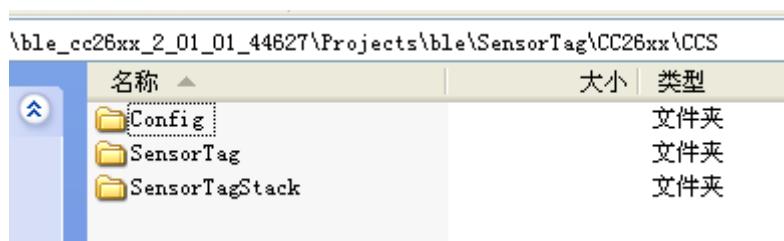
各种 Demo 源码，



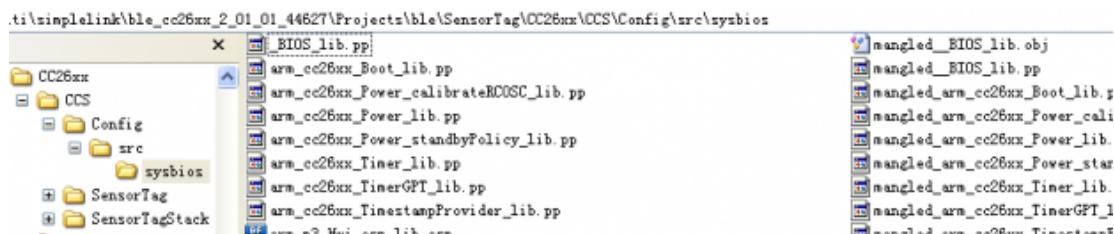
这下有的看了，眼都飘了，最后聚焦到 SensorTag 上，看看这个吧，
里边有 IAR 的和 CCS 的，其实这两 IDE 我都没咋玩过，之前装了 ccs，
就看看 ccs 的吧，打开



打开文件夹，发现三个工程，配置、应用程序和协议栈，我这么推测
啊，

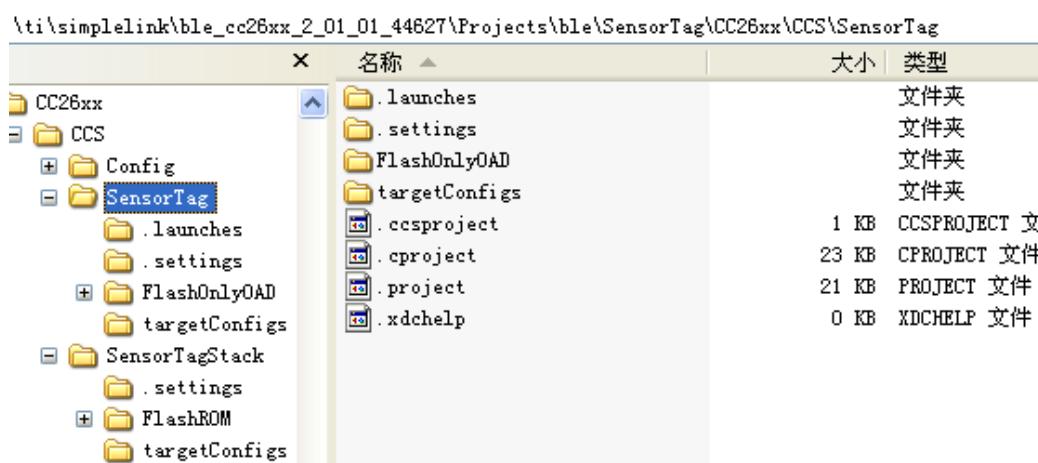


配置里的东，我看了半天没看懂，



应用程序和协议栈推测可以用ccs打开,因为有后缀是project的文件,

根据我的经验,有这种类型的文件出现肯定能用某种IDE打开,



预言应验了,两个工程都能打开,有点像eclipse的意思,随便翻翻

源码,不禁想起以前试用的fireBLE,低功耗蓝牙都是这个风格吧,代

码组织有序,书写规范,注释比代码都多,嘎嘎,



开发板试用平台 <http://bbs.elecfans.com/try.html>

This screenshot shows the Code Composer Studio interface with the SensorTagStack/HAL/include/hal_key.h file open. The code defines several external functions for handling keys:

```
126 */
127 * Exit sleep mode, restore values
128 */
129 extern uint8 HalKeyExitSleep ( void );
130
131 */
132 * This is for internal used by hal_driver_
133 */
134 extern void HalKeyPull ( void );
135
136 */
137 * This is for internal used by hal_sleep
138 */
139 extern hocl HalKeyPressed( void );
140
141 extern uint8 hal_key_keys(void);
142
143 extern uint8 hal_key_int_keys(void);
144
145 ****
146 ****
147
148 #ifdef __plmplus
149
150#endif
151
152#endif
153
```

二话不说，先编译下，SensorTag

This screenshot shows the Code Composer Studio interface with the SensorTag project open. A 'Build Project' dialog box is displayed, indicating the build command is being executed:

Building project...
Invoking Command: E:\ti\ccs6\bin\tools\bin\psoc6 -b all
Run in background

The background shows the project structure and a terminal window displaying the build logs.

还不小呢，i5 的处理器，编了 10 来分钟



开发板试用平台 <http://bbs.elecfans.com/try.html>

The screenshot shows the CCS (Code Composer Studio) IDE interface. On the left is a file browser with a tree view of project files under 'SensorTag'. The main workspace shows a C code snippet for 'SensorTag_Task.c' with several ASSERT statements. To the right of the code is a 'Windows Task Manager' window titled 'Windows 任务管理器' which displays multiple sensor data plots (ECG, Heart Rate, etc.) and system resource usage statistics.

```
// Check manufacturer ID
ST_ASSERT(sensorReadReg(REG_MANUFACTURER_ID, (uint8_t *)val) == MANUFACTURER_ID);

// Check device ID
ST_ASSERT(sensorReadReg(REG_DEVICE_ID, (uint8_t *)val) == DEVICE_ID);

ERRORTYPE sensorOpt3DToIntConvert
{
    /* Brief   Convert raw data to object and ambient temperature
     * @param  data - raw data from sensor
     * @param  lux - converted value (lux)
     * @return  NONE
    */
}

***** Build Finished ****
```

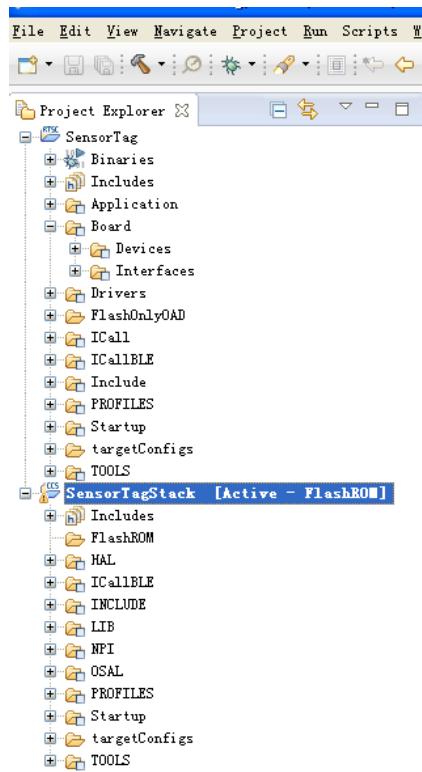
出来咯，一切顺利

| 名称 | 大小 | 类型 |
|------------------------|----------|--------------------|
| Application | | 文件夹 |
| Board | | 文件夹 |
| configPkg | | 文件夹 |
| Drivers | | 文件夹 |
| ICall | | 文件夹 |
| ICallBLE | | 文件夹 |
| PROFILES | | 文件夹 |
| Startup | | 文件夹 |
| TOOLS | | 文件夹 |
| ccsObjs.opt | 2 KB | OPT 文件 |
| makefile | 11 KB | 文件 |
| objects.mk | 1 KB | Makefile |
| SensorTag.hex | 116 KB | HEX 文件 |
| SensorTag.map | 307 KB | Linker Address Map |
| SensorTag.out | 1,150 KB | OUT 文件 |
| SensorTag_linkInfo.xml | 2,486 KB | XML 文档 |
| sources.mk | 3 KB | Makefile |

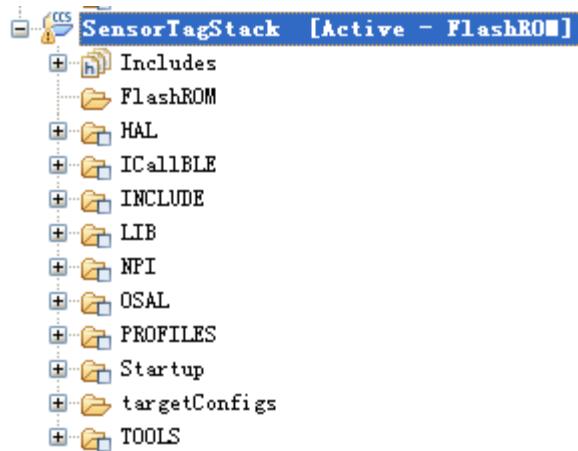
下边是 Stack ,

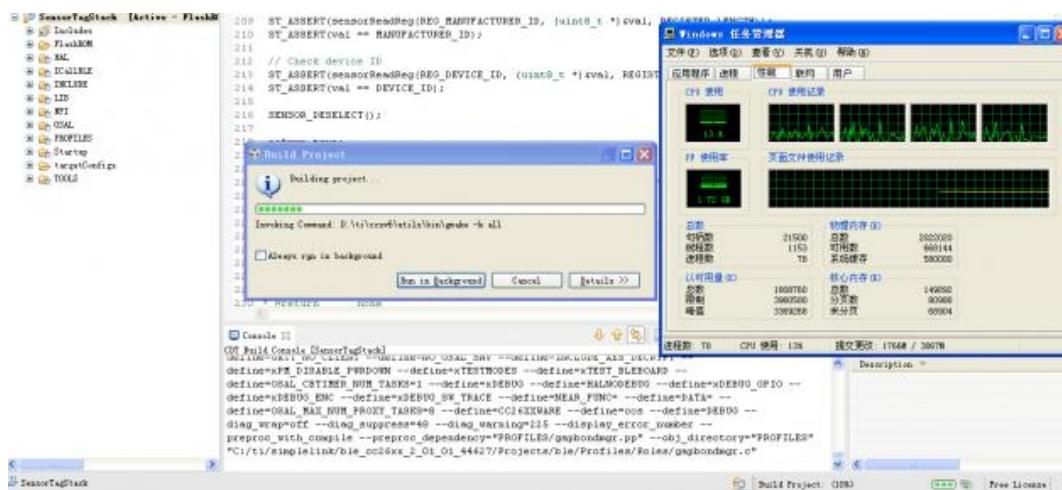


开发板试用平台 <http://bbs.elecfans.com/try.html>

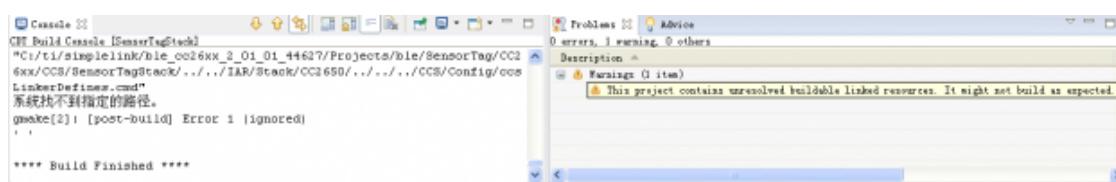


也是 10 来分钟 ,

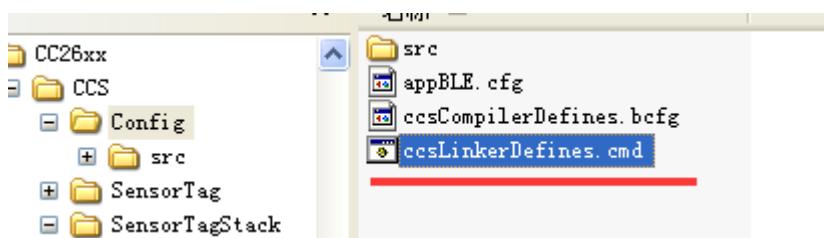




这里有个警告，说找不着路径，找不着一个叫 ccsLinkerDefines.cmd 的文件，这个文件在 Config 文件夹里是有的，



失联的文件，这不是错误，只是个警告，过吧，究竟怎么回事先不纠结。



如此，算是草草过了一遍小 T 上的代码，然而还不死心，没有见到兔子，我们仔细再看看代码，找些蛛丝马迹，证明这确实是小 T 上的代码，我们看应用程序的 Application/SensorTag.c，直接找到了证据，这和之前我们看的小 T 的设备名、设备信息、led 和蜂鸣器 的初始化状态是吻合的。



开发板试用平台 <http://bbs.elecfans.com/try.html>

Project Explorer [Active - FlashOnly0A] SensorTag_Bar.c SensorTag.c SensorTag_Imm.c

```
271 #define TI_ST_KEY_DATA_ID, // Key state
272 0x00
273 #if
274
275 // GAP GATT Attributes
276 static const uint8_t setDeviceName[GAP_DEVICE_NAME_LEN] = "SensorTag 2.0";
277
278 #if FEATURE_OAD
279 // Remote data from OAD profile.
280 static Queue_Handle oadQ;
281 static Queue_Handle hoadQ;
282 #endif //FEATURE_OAD
283
284 // Device information parameters
285 static const uint8_t devInfoModelNumber[] = "CC2650 SensorTag";
286 static const uint8_t devInfoName[] = "M_A";
287 static const uint8_t devInfoFirmwareRev[] = "FU_VERSION_STB";
288 static const uint8_t devInfofirmwareName[] = "Texas Instruments";
289 static const uint8_t devInfoHardwareRev[] = "PCE 1.2/1.3";
290
291 // Pins that are actively used by the application
292 static PDI_Config SensorTagAppPinTable[] =
293 {
294     Board_LED1 | PIN_GPIO_OUTPUT_EM | PIN_PUSHPULL | PIN_DRVSTR_MAX,
295     Board_LED2 | PIN_GPIO_OUTPUT_EM | PIN_PUSHPULL | PIN_DRVSTR_MAX,
296     Board_KEY_LEFT | PIN_INPUT_EM | PIN_PULLUP | PIN_INQ_BOTHEDGES | PIN_HYSERESIS,
297     Board_KEY_RIGHT | PIN_INPUT_EM | PIN_PULLUP | PIN_INQ_BOTHEDGES | PIN_HYSERESIS,
298     Board_RELAY | PIN_INPUT_EM | PIN_PULLDOWN | PIN_INQ_BOTHEDGES | PIN_HYSERESIS,
299     Board_BTN1 | PIN_GPIO_OUTPUT_EM | PIN_PUSHPULL | PIN_DRVSTR_MAX,
300 };
301 /* LED initially off
302 * LED initially off
303 * Button is active low
304 * Button is active low
305 * Relay is active high
306 * Button initially off
307 */
```

再看看这个文件夹下的其他代码，看看温度的，引用的头文件有红外传感器 tmp_007 的，和小 T 吻合，

Project Explorer [Active - FlashOnly0A] SensorTag_Bar.c SensorTag.c SensorTag_Imm.c

```
40
41 ****
42 * INCLUDES
43 */
44 #include "gatt.h"
45 #include "gattservapp.h"
46
47 #include "irtempservice.h"
48 #include "SensorTag_Tmp.h"
49 #include "sensor_tmp007.h" 温度计型号吻合
50 #include "sensor.h"
51 #include "Board.h"
52
53 #include "string.h"
54 #include <ti/sysbios/knl/Semaphore.h>
55 #include <ti/sysbios/knl/Task.h>
56
57 ****
58 * MACROS
59 */
60
61 ****
```

MPU-92509 轴运动传感器，头文件吻合，又看了几个，这些头文件里提示的设备型号和小 T 上的都是一致的，看来不用瞎猜了，这就是小 T 的源码，嘎嘎嘎，可惜，没有下载器，不然，至少可以改下设备信息，编译完烧上试试。



The screenshot shows the Eclipse IDE interface with the 'SensorTag' project selected in the Project Explorer. The code editor displays the 'SensorTag_Bar.c' file. A red box highlights the line '#include "sensor_mpu9250.h"'. A red arrow points from this highlighted line to the text '9轴运动传感器吻合' (9-axis motion sensor compatibility) located in the bottom right corner of the code editor.

```
40 ****
41
42 /* INCLUDES
43 */
44 */
45 #include <ti/sysbios/knl/Semaphore.h>
46 #include <ti/sysbios/knl/Queue.h>
47
48 #include "gatt.h"
49 #include "gattservapp.h"
50
51 #include "Board.h"
52 #include "movementservice.h"
53 #include "SensorTag_Mov.h"
54 #include "sensor_mpu9250.h" // 9轴运动传感器吻合
55 #include "sensor.h"
56 #include "util.h"
57 #include "string.h"
58
59 */
60 /* MACROS
```

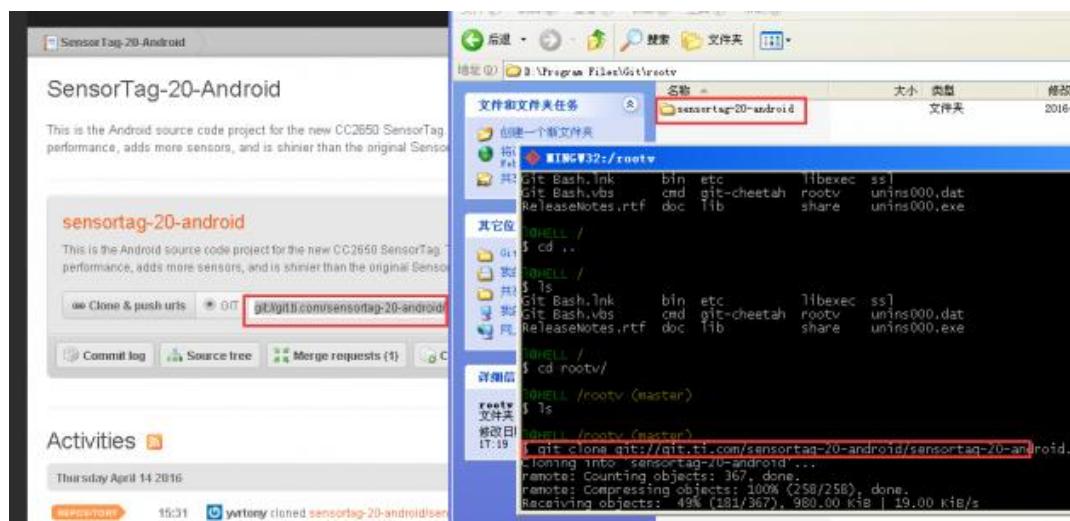
安卓 app 代码一观

作者：[@birdinskydzfsy](#)

地址：http://bbs.elecfans.com/jishu_581520_1_1.html

尽管我不是写 Java 的，还是看一下安卓的 app 代码吧，本来想看看 ios 的，不知道哪里有。

还是通过 TI 官网，找到 android 源码，得用 git 拿下来，我以前装过，也不知道试用撒了，好像也是 BLE 吧，下了 10 好几分钟，主要是这条线路速度不给力。我用命令行工具下的，因为 UI 看了半天没动静，撒也没下来。



我从 git 根目录建了个文件夹，存下载的源码，

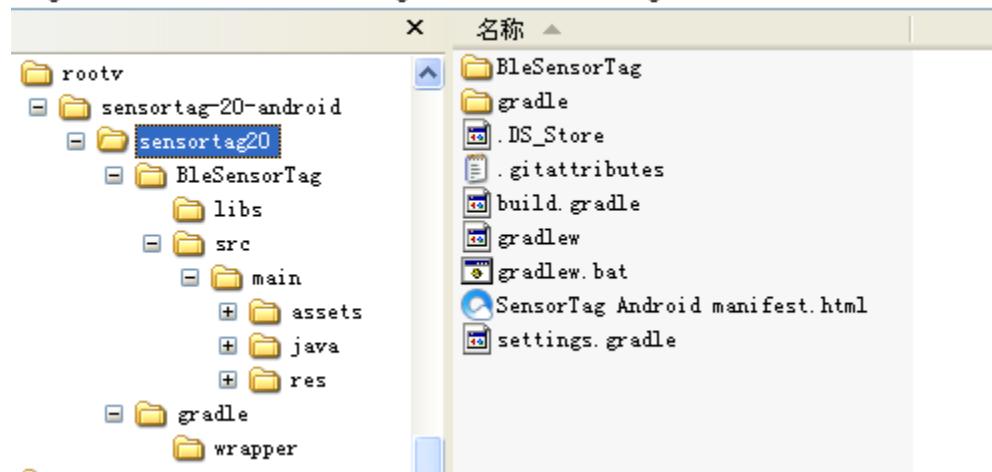




开发板试用平台 <http://bbs.elecfans.com/try.html>

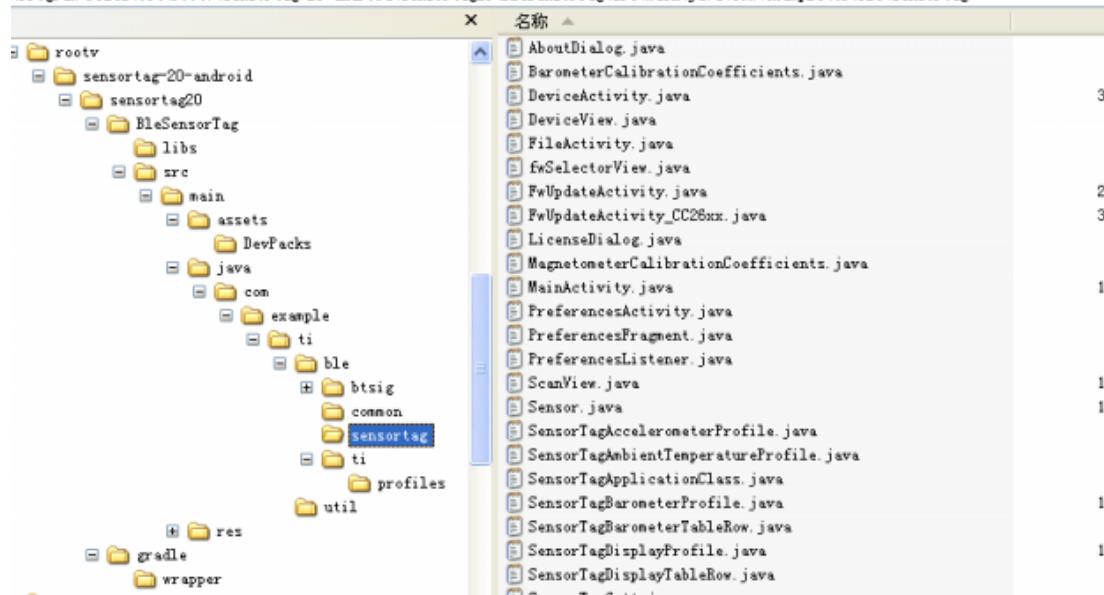
这是散编的，eclipse+gradle 麽？

Program Files\Git\rootv\sensortag-20-android\sensortag20



看看有散源码我认识的，隐藏得真深啊，总算找到 Java 文件了，

\Program Files\Git\rootv\sensortag-20-android\sensortag20\BleSensorTag\src\main\java\com\example\ti\ble\sensortag



随便翻了翻代码，我也没有装 eclipse，这写应该是安卓的源码了，看这个，这应该是连接 IBM 云的代码，有句代码表示有 url 和设备号的变量。



开发板试用平台 <http://bbs.elecfans.com/try.html>

```

    if (config_service == CloudProfileConfigurationDialogFragment.DEF_CLOUD_IBMQICKSTART_CLOUD_SERVICE) {
        ((IBMIoCloudTableRow) this.tRow).cloudURL.setText("Open in browser");
        ((IBMIoCloudTableRow) this.tRow).cloudURL.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                v.getContext().startActivity(new Intent(Intent.ACTION_VIEW, Uri.parse(
                    "https://quickstart.internetofthings.ibmcloud.com/#/device/" + addrShort + "/sensor/")));
            }
        });
    } else {
    }
}

```

设备号

参照下我的 iosApp 里的连接，是吻合的

下图们，浏览器里看到的推送的数据，前文提到过，在设备信息里有一个9字节的System ID，其后8个字节就是云数据里的设备标识，这是我的设备云的位置，

<https://quickstart.internetofthings.ibmcloud.com/#/device/b0b448d2e780/sensor/>

貌似是ibm的云服务器，下边是信息推送的时间点，也许我试用时你可以偷窥数据呢

IBM Internet of Things Foundation

Quickstart

无需注册，便可了解如何轻松地将您的设备连接到 IoT Foundation 并查看实时传感器数据

b0b448d2e780 前往

■ 设备断开连接时间：上午10:40:31

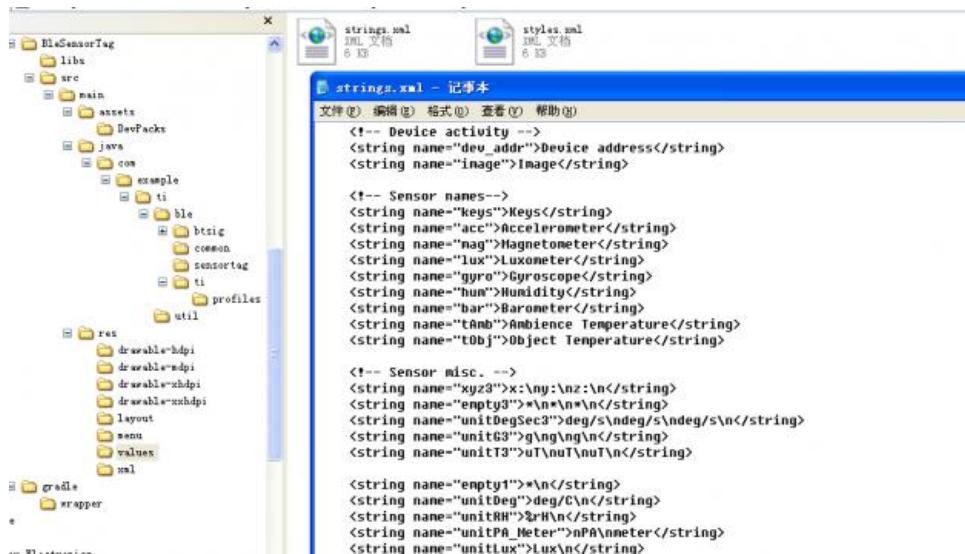
这个文件夹里是图片资源



其他的还有一些字符串常量啊，界面相关的设置等等



开发板试用平台 <http://bbs.elecfans.com/try.html>



不看了，有兴趣的童鞋可以装个 git，clone 下来代码自己研究



我用的这个版本的，也不是最新的，可以到官网下最新的





总结

TI 做的传感器已经不再仅仅是传感器，增加了艺术品的属性，不但能用而且好看，放在那是装饰，用起来是工具。现在大部分传感器走向数字化，不用程序猿翻着 datasheet 进行 AD 转换，程序猿做的工作更像是在写代码了，而不是苦逼的翻译机。

各种传感器基本上都是利用物质的理化特性，将力热声光磁等能量形式转化成电的形式放大读取，最后可能被封装成 统一的数字化模式，方便了读写配置。程序猿把更多的注意力转移到处理程序的逻辑上，而不再是繁杂重复的 AD 翻译，很有可能一个函数便能处理一片传感器的读 写配置，或者只需进行简单的修改。反映了人类对物质文明和精神文明不断的提炼升华，只有这样，文明才会加速发展，终有一天在与宇宙中其他文明交流时不会被 看成虫子，好吧说得太大了。

小 T 上基本包含了常见的所有类型的传感器，方寸之间尽显人类智慧，可以说这阵子玩得不亦乐乎，经常揣在兜里，测量个环境气压、温湿度、磁场及变化曲线啥 的，从另一个角度感知这个世界，比如感觉手比较干燥出气鼻子干，用小 T 会发现湿度只有百分之五的样子，提示注意补水，等等。活跃下脑洞，小 T 会成为你最好的小秘书。

最后，向打造小 T 的 TI 人致敬！



开发板试用平台 <http://bbs.elecfans.com/try.html>

了解更多开发板试用



扫一扫，直接在手机上打开