WWW.LAB-Z.COM

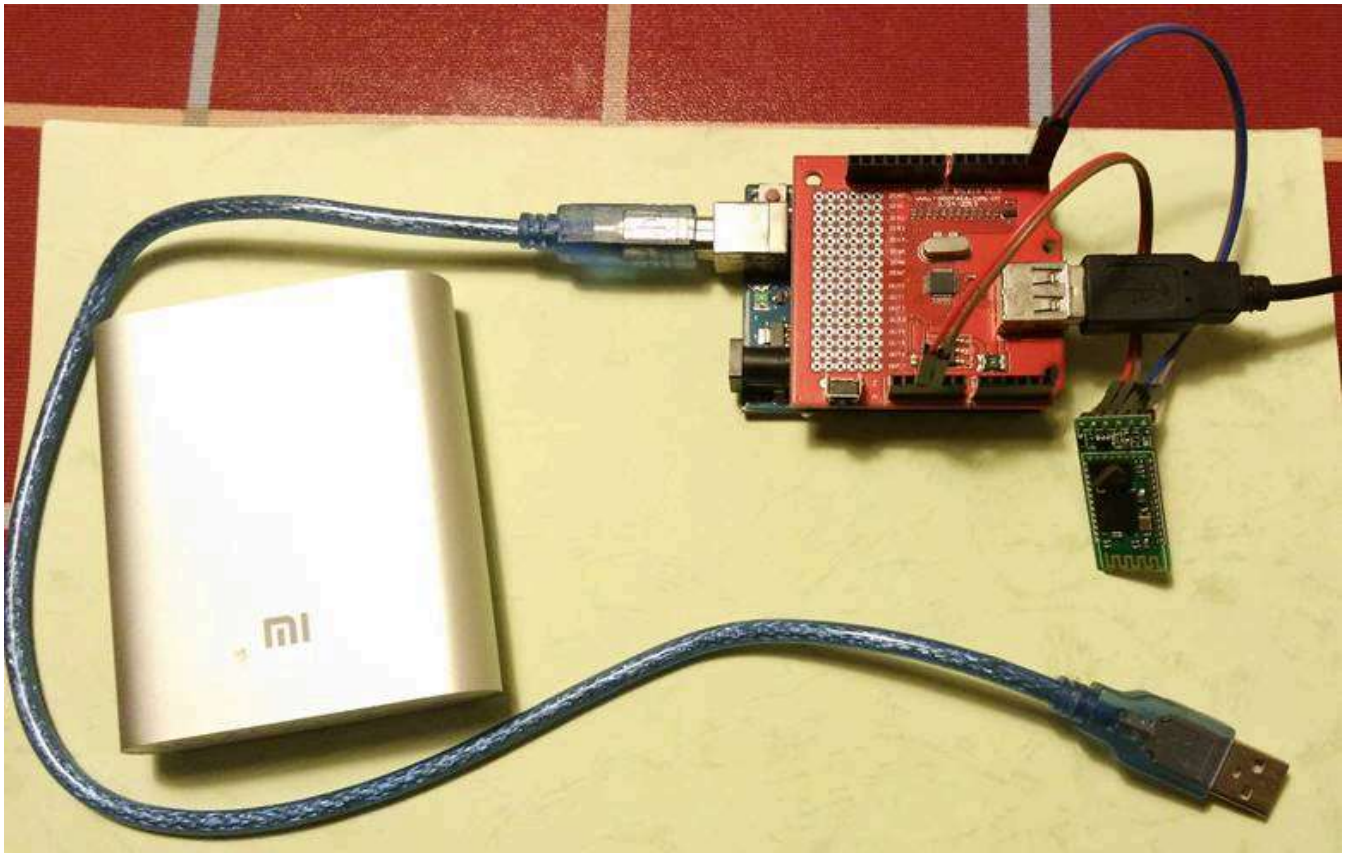
Arduino打造USB蓝牙键盘转接器

本文介绍如何使用 Arduino 打造一个设备，能够将你的USB键盘转化为蓝牙键盘。

键盘可以算作PC上最古老的设备了，他的出现使得人类可以用非常简单的方法与电脑进行交互。同样的，由于各种历史原因，键盘也是PC上最复杂，兼容性问题最多的设备之一（类似的还有硬盘，不过从IDE到SATA的进化过程中，标准明确，兼容性问题少多了）。

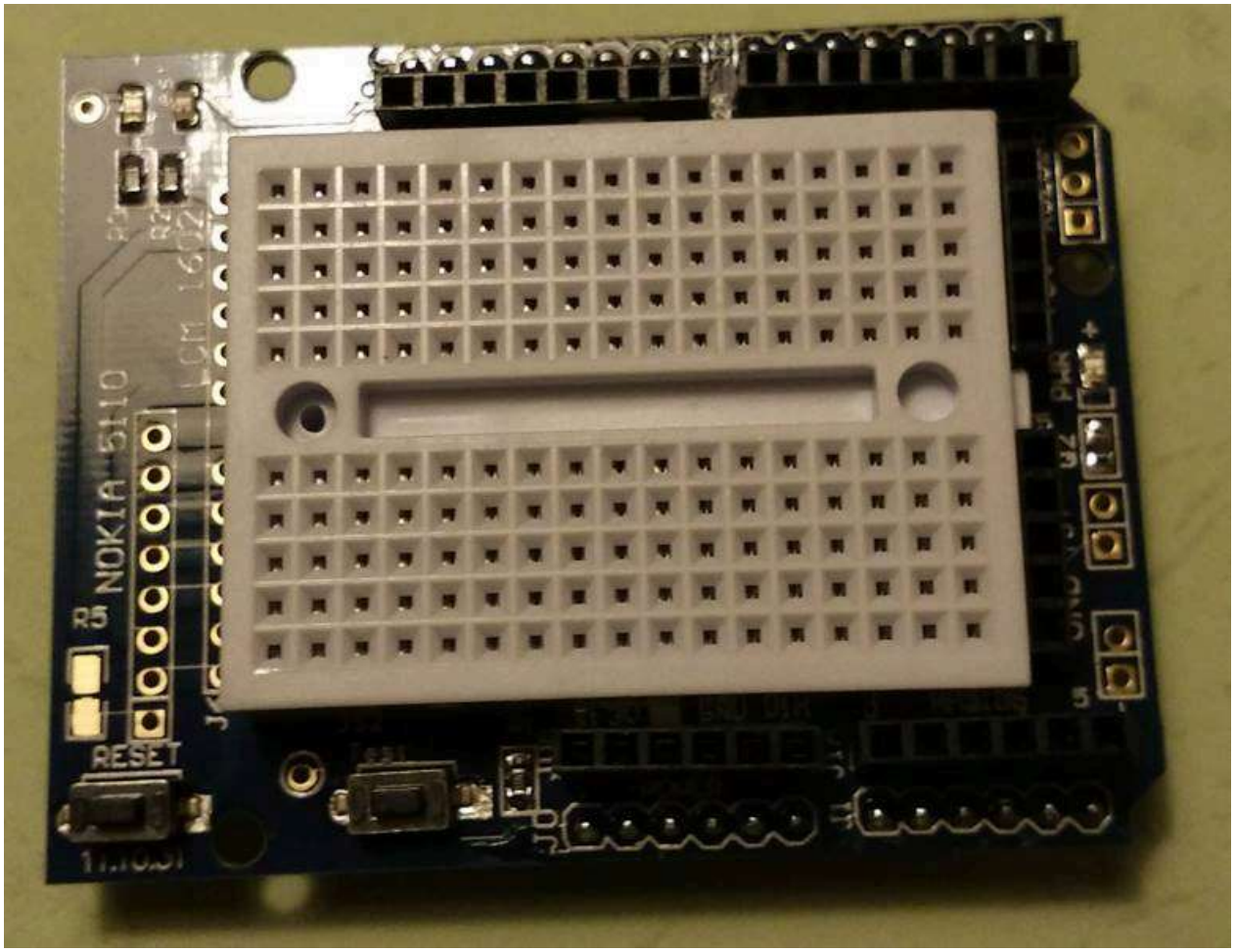
网上流传着一篇DIY USB键盘转换为无线的文章，非常不幸的是，那篇文章是错误的，很明显的错误是作者认为键盘是单向传输，而实际上传输是双向的。比如，USB每次通讯都需要HOST和SLAVE的参与，即便是PS2键盘的通讯也同样如此。此外，大小写键之类切换是主机端进行控制的。

硬件部分Arduino UNO , USB Host Shield 和 HID 蓝牙芯片。强调一下这里使用的是 HID 蓝牙芯片，并非普通的蓝牙串口透传芯片【特别注意是“蓝牙键盘芯片”也不是“蓝牙条码模块”】。关于这个模块可以参考我在【参考1】中的实验。
硬件连接很简单，USB HOST Shield插在 Arduino上，然后VCC/GND/TX/RX将Arduino 和HID蓝牙模块连接在一起。

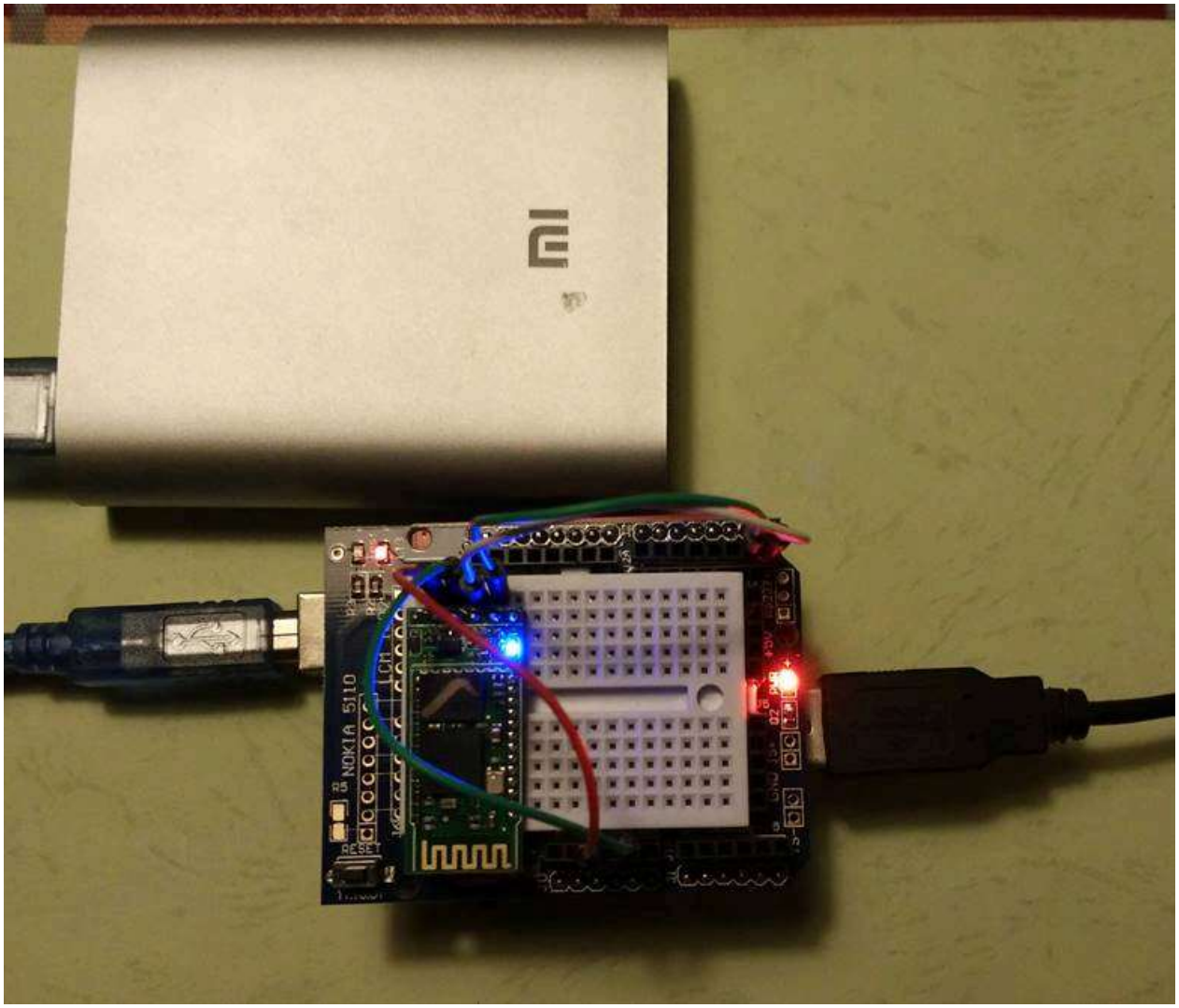


原理：首先，为了通用性和编程简单，我们用USB HOST发送命令把键盘切换到 Boot Protocol 模式下。这样即使不同的键盘，每次发出来的数据也都是统一的格式。然后，我们直接读取缓冲数据就可以解析出按键信息了。最后，将取下来的按键信息（Scan Code）按照HID蓝牙模块的格式要求通过串口送到模块上，主机端就收到了。

上述连接就可以正常工作了，但是为了美观和提高可靠性，我找到之前买的一个面包板 Shield。



插好之后就是这样



具体代码：

```
/* MAX3421E USB Host controller LCD/keyboard demonstration */
//#include <Spi.h>
#include "Max3421e.h"
#include "Usb.h"

/* keyboard data taken from configuration descriptor */
#define KBD_ADDR        1
#define KBD_EP          1
#define KBD_IF          0
#define EP_MAXPKTSIZE   8
#define EP_POLL         0x0a
/**/
//**************************************************************************
****
//  macros to identify special charaters(other than Digits and Alphabets)
```

```
//**************************************************************************
****
#define BANG        (0x1E)
#define AT          (0x1F)
#define POUND       (0x20)
#define DOLLAR      (0x21)
#define PERCENT     (0x22)
#define CAP         (0x23)
#define AND         (0x24)
#define STAR        (0x25)
#define OPENBKT     (0x26)
#define CLOSEBKT    (0x27)

#define RETURN      (0x28)
#define ESCAPE      (0x29)
#define BACKSPACE   (0x2A)
#define TAB         (0x2B)
#define SPACE       (0x2C)
#define HYPHEN      (0x2D)
#define EQUAL       (0x2E)
#define SQBKTOPEN   (0x2F)
#define SQBKTCLOSE  (0x30)
#define BACKSLASH   (0x31)
#define SEMICOLON   (0x33)
#define INVCOMMA    (0x34)
#define TILDE       (0x35)
#define COMMA       (0x36)
#define PERIOD      (0x37)
#define FRONTSLASH  (0x38)
#define DELETE      (0x4c)
/**/
/* Modifier masks. One for both modifiers */
#define SHIFT       0x22
#define CTRL        0x11
#define ALT         0x44
#define GUI         0x88
/**/
/* "Sticky keys */
#define CAPSLOCK    (0x39)
#define NUMLOCK     (0x53)
#define SCROLLLOCK  (0x47)
/* Sticky keys output report bitmasks */
#define bmNUMLOCK       0x01
#define bmCAPSLOCK      0x02
#define bmSCROLLLOCK    0x04
/**/
```

```
EP_RECORD ep_record[ 2 ];  //endpoint record structure for the keyboard

char buf[ 8 ] = { 0 };      //keyboard buffer
char old_buf[ 8 ] = { 0 };  //last poll
/* Sticky key state */
bool numLock = false;
bool capsLock = false;
bool scrollLock = false;
bool line = false;

void setup();
void loop();

MAX3421E Max;
USB Usb;

void setup() {
  Serial.begin( 9600 );
  Serial.println("Start");
  Max.powerOn();
  delay( 200 );
}

void loop() {
    Max.Task();
    Usb.Task();
    if( Usb.getUsbTaskState() == USB_STATE_CONFIGURING ) {  //wait for
addressing state
        kbd_init();
        Usb.setUsbTaskState( USB_STATE_RUNNING );
    }
    if( Usb.getUsbTaskState() == USB_STATE_RUNNING ) {  //poll the keyboard
        kbd_poll();
    }
}
/* Initialize keyboard */
void kbd_init( void )
{
 byte rcode = 0;  //return code
/**/
    /* Initialize data structures */
    ep_record[ 0 ] = *( Usb.getDevTableEntry( 0,0 ));  //copy endpoint 0
parameters
    ep_record[ 1 ].MaxPktSize = EP_MAXPKTSIZE;
    ep_record[ 1 ].Interval  = EP_POLL;
    ep_record[ 1 ].sndToggle = bmSNDTOG0;
```

```
    ep_record[ 1 ].rcvToggle = bmRCVTOG0;
    Usb.setDevTableEntry( 1, ep_record );              //plug kbd.endpoint
parameters to devtable
    /* Configure device */
    rcode = Usb.setConf( KBD_ADDR, 0, 1 );
    if( rcode ) {
        Serial.print("Error attempting to configure keyboard. Return code
:");
        Serial.println( rcode, HEX );
        while(1);  //stop
    }
    /* Set boot protocol */
    rcode = Usb.setProto( KBD_ADDR, 0, 0, 0 );
    if( rcode ) {
        Serial.print("Error attempting to configure boot protocol. Return
code :");
        Serial.println( rcode, HEX );
        while( 1 );  //stop
    }
    delay(2000);
    Serial.println("Keyboard initialized");
}

/* Poll keyboard and print result */
/* buffer starts at position 2, 0 is modifier key state and 1 is irrelevant
*/
void kbd_poll( void )
{
 char i;
 boolean samemark=true;
 static char leds = 0;
 byte rcode = 0;     //return code
    /* poll keyboard */
    rcode = Usb.inTransfer( KBD_ADDR, KBD_EP, 8, buf );
    if( rcode != 0 ) {
        return;
    }//if ( rcode..

    for( i = 2; i < 8; i++ ) {
     if( buf[ i ] == 0 ) {  //end of non-empty space
        break;
     }
      if( buf_compare( buf[ i ] ) == false ) {   //if new key
        switch( buf[ i ] ) {
          case CAPSLOCK:
            capsLock =! capsLock;
```

```
            leds = ( capsLock ) ? leds |= bmCAPSLOCK : leds &= ~bmCAPSLOCK;
// set or clear bit 1 of LED report byte
            break;
          case NUMLOCK:
            numLock =! numLock;
            leds = ( numLock ) ? leds |= bmNUMLOCK : leds &= ~bmNUMLOCK;
// set or clear bit 0 of LED report byte
            break;
          case SCROLLLOCK:
            scrollLock =! scrollLock;
            leds = ( scrollLock ) ? leds |= bmSCROLLLOCK : leds &=
~bmSCROLLLOCK;   // set or clear bit 2 of LED report byte

          Serial.write(0x0c);  //BYTE1
          Serial.write(0x00);  //BYTE2
          Serial.write(0xA1);  //BYTE3
          Serial.write(0x01);  //BYTE4
          Serial.write(00);  //BYTE5
          Serial.write(0x00);  //BYTE6
          Serial.write(0x1e);  //BYTE7
          Serial.write(0);  //BYTE8
          Serial.write(0);  //BYTE9
          Serial.write(0);  //BYTE10
          Serial.write(0);  //BYTE11
          Serial.write(0);  //BYTE12
          delay(500);
          Serial.write(0x0c);  //BYTE1
          Serial.write(0x00);  //BYTE2
          Serial.write(0xA1);  //BYTE3
          Serial.write(0x00);  //BYTE4
          Serial.write(0);  //BYTE5
          Serial.write(0x00);  //BYTE6
          Serial.write(0);  //BYTE7
          Serial.write(0);  //BYTE8
          Serial.write(0);  //BYTE9
          Serial.write(0);  //BYTE10
          Serial.write(0);  //BYTE11
          Serial.write(0);  //BYTE12

            break;
          case DELETE:
            line = false;
            break;
          case RETURN:
            line =! line;
            break;
```

```
            //default:
              //Serial.print(HIDtoA( buf[ i ], buf[ 0 ] ));
            //  break;
          }//switch( buf[ i ...

          rcode = Usb.setReport( KBD_ADDR, 0, 1, KBD_IF, 0x02, 0, &leds );
          if( rcode ) {
            Serial.print("Set report error: ");
            Serial.println( rcode, HEX );
          }//if( rcode ...
       }//if( buf_compare( buf[ i ] ) == false ...
      }//for( i = 2...

      i=0;
      while (i<8)
        {
          if (old_buf[i]!=buf[i]) { i=12; }
          i++;
        }
      if (i==13) {
       // for (i=0;i<8;i++) {
            //  Serial.print(buf[ i ],HEX);
            //  Serial.print(']');
       //  }
       // Serial.println(' ');

            Serial.write(0x0c);  //BYTE1
            Serial.write(0x00);  //BYTE2
            Serial.write(0xA1);  //BYTE3
            Serial.write(0x01);  //BYTE4
            //Labz_Debug Serial.write(buf[1]);  //BYTE5
            Serial.write(buf[0]);  //BYTE5     //Labz_Debug
            Serial.write(0x00);  //BYTE6
            Serial.write(buf[2]);  //BYTE7
            Serial.write(buf[3]);  //BYTE8
            Serial.write(buf[4]);  //BYTE9
            Serial.write(buf[5]);  //BYTE10
            Serial.write(buf[6]);  //BYTE11
            Serial.write(buf[7]);  //BYTE12
      }

      //Labz_Debug for( i = 2; i < 8; i++ ) {                    //copy new
buffer to old
       for( i = 0; i < 8; i++ ) {                    //copy new buffer to old
//Labz_Debug
        old_buf[ i ] = buf[ i ];
```

```
    }
  }
  /* compare byte against bytes in old buffer */
  bool buf_compare( byte data )
  {
   char i;
   for( i = 2; i < 8; i++ ) {
     if( old_buf[ i ] == data ) {
       return( true );
     }
   }
   return( false );
  }
```

我在处理SCROLLLOCK 键的地方插入了一个测试代码，理论上按下这个键的时候，主机还会收到 1 这个字符，这样是为了测试工作是否正常。

我在 x86 台式机上实测过，工作正常；小米4手机上实测过，工作正常； iPad 上是测过，工作也正常。

在iPad上工作的视频在下面：

完整代码下载

BKC2COM

特别注意：

1. 因为我们使用的是最简单的Boot Protocol，所以如果你的键盘上有音量键之类的有可能失效；

2. 我不确定是否所有的键盘都会支持 Boot Protocol ，从之前玩USB鼠标的经验来看，确实有可能；

3. 供电部分没有经过优化，不知道电力消耗如何，不确定一个充电宝能够工作的时间；

最后讲一个小故事：有一次我去实验室，发现他们在折腾键盘。那是一款带着音量控制功能的键盘。系统测试的时候发现，按一下键盘音量键之后，屏幕上显示的音量会跳2格。从原理上说，按下那个键之后，键盘发出特定的Scan Code，系统中还有个专门响应这个Scan Code的程序然后在屏幕上绘制音量指示方块。蛮有意思的一件事情是：很多人认为大公司有操控供应商的能力，供应商在大厂面前会唯唯诺诺，这也是高层会有的想法，问题是底层人员未必吃这一套。每次想起这个事情，我都要想起敏感字关于矛盾的辩证法的论证。这个事情就是双方的下层在不停的扯，更准确的说，是键盘厂商，软件开发商和我们在一起纠缠，键盘厂商说同样的键盘在其他人家用起来没问题，软件开发商说我的软件在之前的机型上一直用，我们的人说，少扯淡，赶紧解决，前后一个多月都没有搞定.....那时候，组里刚买了一个usb逻辑分析仪，我用着感觉很好玩。于是，我就用逻辑分析仪测试了一下键盘，测试的结果是，键盘发出来的Scan Code没有问题，每次按键都是一个Press一个Release，所以真相肯定是写上位机程序的软件厂商搞错了什么。截图附带着数据包一起丢给三方。这是最底层的传输，如果依然嘴硬，那只能落下笑柄而已。然后很快软件厂商就服软自己去修改了。只是说说我经历的事情，如果非要说出一些道理的话这个故事是为了说明：USB逻辑分析仪很有用.....

就是这样.

==========2017年5月11日更新==========

有朋友说 Win ，Shift，Alt 都不工作，今天正好有空研究了一下，是我的代码有问题。解析出来的USB 键盘按键信息中 Buf[0] 是这些Key 的标志位，我没有正确Pass给模块。因此，修正下面2处代码，一个是发送，一个是每次保存当前的按键状态的位置：

```
            Serial.write(0x0c);  //BYTE1
            Serial.write(0x00);  //BYTE2
            Serial.write(0xA1);  //BYTE3
            Serial.write(0x01);  //BYTE4
```

```
            Serial.write(buf[0]);  //BYTE5
            Serial.write(0x00);  //BYTE6
            Serial.write(buf[2]);  //BYTE7
            Serial.write(buf[3]);  //BYTE8
            Serial.write(buf[4]);  //BYTE9
            Serial.write(buf[5]);  //BYTE10
            Serial.write(buf[6]);  //BYTE11
            Serial.write(buf[7]);  //BYTE12

        }

        for( i = 0; i < 8; i++ ) {                    //copy new buffer to old
          old_buf[ i ] = buf[ i ];
        }
```

修改后可以正常工作的代码：

bkc2com1.1

参考：

1. http://www.lab-z.com/btkeyboard/ 蓝牙键盘模块的实验

ziv2013 / 2015年9月20日 / Funny

《Arduino打造USB蓝牙键盘转接器》有44个想法

Will

2015年11月2日 下午 2:39

您好：

我是个刚入门的爱好者,做Web程序的,想使用Bluno nano 开发板完成一个和您文章中一样的功能.

http://wiki.dfrobot.com.cn/index.php/(SKU:DFR0296)Bluno_nano_%E8%93%9D%E7%89%994.0%E6%8E%A7%E5%88%B6%E5%99%A8_%E5%85%BC%E5%AE%B9Arduino_nano
但是入手研究一段时间后,发现自己对usb和蓝牙的理解很不到位,Bluno nano模块实际是一个Arduino + 蓝牙的模块,本身支持透传和HID.最开始我同他们技术支持讨论下做这个功能的可能性,是打算通过自带的USB转串行数据 (这块板自带的功能),直接在芯片内部获取后传入蓝牙发送.但实际做好后发现行不通,再次找技术确认后,发现usb和蓝牙实际是一个串行信号.没法做到我的要求,他们推荐我又买了一个usb转串行芯片,将信号接入一组由芯片接入的虚拟串行引脚.
有个例程(http://www.dfrobot.com.cn/community/forum.php?mod=viewthread&tid=2685)他们推荐我参考下您的文章,但我下载原码后发现自己水平难以完全理解并按照目前的硬件处理方法设计逻辑.
所以这里想请你给看下,按照我这个方式,需要如何修改程序来处理?至少给一点思路点拨一下.
万分感激!

---

ziv2013

2015年11月2日 下午 5:01

1. 一般的蓝牙模块的接口是串口，意思是：Arduino 如果要发送那么需要走串口输出，如果Arduino想接收数据，那么也需要走串口，因此这里可以看出有一个问题：ATmega328 上面串口不够用。

2.Software Serial可能会有问题，比如，我试验的时候发现对于超过 9600的波特率，这个东西会出现莫名其妙的问题

3.不明白为什么你还要再买一个USB转串口？你是打算要看调试数据么？如果你真要这样做，那么一定记得转出来的串口除了 RX TX 一定要 GND (共地)

4.如果你想做一个和我一样功能的，建议你就按照我上面的东西买，整套的价格不会高于单独一个 Bluno nano ..... 我特地确认了一下，确实是这样的。

5.研发的时候，特别是不懂的时候，建议你买大的，功能强的板子，等你搞明白，确定能实现功能再想办法缩小，一蹴而就只在小概率理想的情况下，而现实中通常单次实验小概率事件不会发生。

6.DFRobot的东西我也用过，硬件质量远远超过他们服务的质量。真的。

所以，结论是：去买普通的 Uno 和我推荐的蓝牙模块吧，加到一起差不多才 70～

啪啪啪傲娇萝莉

2015年11月17日 上午 6:35

您好，我在看完您的文章之后发现您是通过USB键盘–>USB HOST–>串口–>蓝牙模块。也就是说蓝牙模块里写的程序是买家本来就自带的，那么这个HID蓝牙模块里的组合键是怎么控制的。我很久以前就开始了您的这个项目的开发，也考虑过您的这个方案。但是至今没有开始行动有两个原因，一是体积过于庞大耗电过高，二是HID蓝牙模块程序未知总是有很多BUG比如说组合键的问题，于是我在考虑直接购买带有USB接口的蓝牙模块，然后再蓝牙模块内写入USB HOST的程序 这样直接单片就能完成所有的需求而且非常省电，但是至今没能完成这个USB HOST。

---

ziv2013

2015年11月17日 上午 9:18

这个HID蓝牙模块里按键 Scan code 之类是来自 USB HID协议的，所以USB Host抓下来之后直接送出去即可。组合键是否有问题我不确定，不过一般的应用打字是完全没有问题的。理论上不可能有带有USB HOST的蓝牙芯片，因为这两个东西都是蛮复杂的协议，整合在一起意义不大。usb host和蓝牙芯片的尺寸都不大，有机会放在很小的一个板子上。

啪啪啪傲娇萝莉

2015年11月17日 下午 7:25

正是因为没有啊 实际上很多CSR的蓝牙芯片都是带USB接口的 所以要往这些芯片里面写HOST的程序啊世界上没有的东西就自己创造

---

ziv2013

2015年11月18日 下午 12:52

恩 不错，有想法。有机会多读点书，学好英文哈。加油！

---

batsing

2021年6月28日 上午 10:45

蓝牙芯片带USB从设备的很多，经典如CC2450，带USB HOST我还没找到。

---

ziv2013

2021年6月29日 上午 9:08

WCH 家有几个可以的

http://www.wch.cn/products/category/63.html

有耐心你可以试试

Pingback： expander | Kevins Bobo

回到**b**不再度**v**

2016年4月14日 下午 12:19

博主对电源最后是怎么处理的？在下想做个蓝牙键盘但是电源不知道怎么办。

---

ziv2013

2016年4月15日 上午 9:24

我最后是直接上的充电宝，我没有考虑省电的问题。

哦哦哦

2016年6月2日 下午 9:01

请问楼主我的背光键盘接上usbhost背光没反应是不是键盘带不起来呢，插上小的蓝牙模块有反应

---

ziv2013

2016年6月3日 下午 12:38

确实 有可能供电不足唉

建议你再找一个普通键盘试试看？

马迎新

2016年6月7日 下午 8:49

楼主我的键盘现在能亮了，没有问题但是蓝牙连上没反应，我用一个串口调试助手接受数据，发现他显示 Start
Keyboard initialized
Data packet error: FESet report error: FE
请问这是什么原因呢 ？ 我用的是非标准UNO,用的不是16u2那块芯片，跟这个有关系么

马迎新

2016年6月8日 下午 2:01

楼主我接上蓝牙后电脑上显示一个端口COM5，然后我用串口调试助手打开，看到数据是这样的Start
Keyboard initialized
0C 00 A1 01 00 00 04 00 00 00 00 00 0C 00 A1 01 00 00 00 00 00 00 00 00

我知道这意味着数据已经是正确的了，可是关闭端口他不能直接当做蓝牙键盘用，按键没有反应，请问这是怎么回事呢，好像只有打开蓝牙串口才能生效啊

---

ziv2013

2016年6月8日 下午 6:52

应该和你的非标准的uno没关系，你用的什么蓝牙芯片？

马迎新

2016年6月8日 下午 11:41

楼主我今天试了别的键盘成功了就是只要是组合键都无法生效，还有win键也没反应，用的是同学的罗技薄膜键盘，但是用我的阿米洛全键无冲键盘就不行，提示Start
Keyboard initialized
Data packet error: FESet report error: FE；请问这是什么原因呢？

---

ziv2013

2016年6月9日 下午 9:53

有可能是你键盘输出格式特殊导致的。

---

ziv2013

2016年6月9日 下午 9:57

我手上没有你的那种全程键盘，没有办法调试唉

马迎新

2016年6月9日 下午 10:40

我跟那个全键无冲厂商联系了，他说可以给我弄一个六键无冲的固件，如果六键无冲的话，是不是可以实现了。

---

ziv2013

2016年6月10日 上午 11:05

主要是要支持 Boot Protocol 。我这个转接器是要把键盘切换到 Boot protocol 然后才做解析的。

---

athson

2016年6月10日 下午 11:10

Error attempting to configure keyboard. Return code ： 5

你好啊 我按你的教程 我已经弄出来了 。 可以接普通的usb键盘 也可以接机械键盘 但是我介绍hhkb 就显示上面的错误。 usb那块 有没有见到的例子 让我看看如何接受数据和现实数据 我看了下资料 官网没有 USB Host Shield 找到了另一个库 但是和你的不同 例子也有些看不明白 毕竟刚接触。。 我只需要个见到的usb入门例子就可以了 如果你有时间的啊 希望能帮下我..谢谢 我想看看是不是板子不支持hhkb还是什么原因

---

ziv2013

2016年6月11日 下午 3:33

呃 你这个键盘我没实验过。我用的库也是网上的（这个应该只有一个就是做 USB Host的那家）。没办法，只能你自己慢慢琢磨了。

马迎新

2016年6月12日 下午 5:22

你的键盘是不是全键无冲，我的是全键无冲，就无法完成，每次都是抓包错误。阿米洛的VA87M

---

ziv2013

2016年6月12日 下午 9:50

你的键盘支持 Boot Protocol吗？

---

mzy

2017年3月28日 下午 10:37

楼主，我将你给的代码输入到arduino中，编译的时候总是报错是什么情况，库文件下了还是报错

---

ziv2013

2017年3月30日 下午 3:48

所有的代码都在附件中。现在 arduino ide中带的是 2.0的USB Host库，和我代码不兼容。另外，不用担心，已经有很多朋友根据我的文章制作出来他们的转接器了。

ps: 现在taobao 有卖这种转接器的了，如果你是想使用，可以直接买。

李继鹏

2017年5月11日 下午 5:48

你好，我将您的代码下载下来试了一遍，Shif、Ctrl、Alt、Win键按了之后是没有反应的。代码中是不是没有作处理（我没有找到相应代码，是不需要像shift这样的键单独处理吗？）？还是说是我键盘的原因？如果是因为代码中本来就没有相应处理，那么我现在想要实现这四个键也能正常使用的话，具体应当使用Serial.write()函数发送什么样的数据呢？望赐教。

ziv2013

2017年5月12日 下午 10:49

昨天看了一下，确实是代码的问题，具体的修改写在上面了，请实验。

李继鹏

2017年5月13日 上午 12:57

感谢！下载试了一下完美运行。得查一下协议，好好看一下代码的实现。

---

suoma

2017年7月15日 下午 11:44

没有usb扩展板，用leonardo和蓝牙可以实现相同功能吗？我想通过手机发送指令给leonardo，然后判断字符控制键鼠动作

---

ziv2013

2017年7月16日 上午 9:48

不懂你要做什么.

---

suoma

2017年7月16日 下午 12:03

类似通过手机玩电脑游戏，手机通过蓝牙和leonardo，然后通过手机的蓝牙串口助手发送命令{或者预先定义键盘模式各命令}，命令到leonardo，然后看着键鼠动作，按哪个键。
感觉你的这个和这个挺相近的

suoma

2017年7月16日 下午 12:04

是控制键鼠动作

---

suoma

2017年7月15日 下午 11:52

常见的HC-05可以替换吗？

---

ziv2013

2017年7月16日 上午 9:47

不可以.必须是专门的蓝牙 HID

CharlieJiang

2017年9月16日 下午 11:24

其实可以，只是要刷上RN-42模块的固件.....很麻烦，油管上面有这个的教程

三孔布

2018年2月13日 下午 2:18

这个方案耗电量高么？体积有点大，有没有可能缩减大小？

---

ziv2013

2018年2月14日 下午 1:45

耗电应该挺大的，毕竟没有考虑过功耗的问题。如果想具体优化，可以自己做板子。Leonrado 主要就是一个 32U4 加晶振。蓝牙的话，是一个邮票板。

---

jim

2018年11月18日 下午 1:29

你好，请教一下，能否使用软串口 SoftwareSerial?

---

ziv2013

2018年11月19日 上午 9:33

可以没问题，这里的蓝牙模块对arduino 来说只是接收。

小黄人**Geek**

2022年3月8日 下午 2:43

感谢@ziv2013， 已经根据这个教程成功实验完成，并且做了升级版本，加了一个小小的OLED屏幕，大家可以看看我Github上的代码，地址如下：https://github.com/pengfaliu/lfp-arduino-studio/tree/master/sketchs/bluetooth-keyborad-transverter。

硬件部分，手工焊接到小盒子里了，大家也可以参考。

---

ziv2013

2022年3月8日 下午 5:04

不错啊，我这边在等 ESP32 S3 ，到时候会有更简单的方案。

小黄人**Geek**

2022年3月14日 上午 10:11

我这属于外行，本身只是写软件的，没有你那么高的水准啦，等你ESP32-S3出来再跟着你学学，哈哈