WWW.LAB-Z.COM

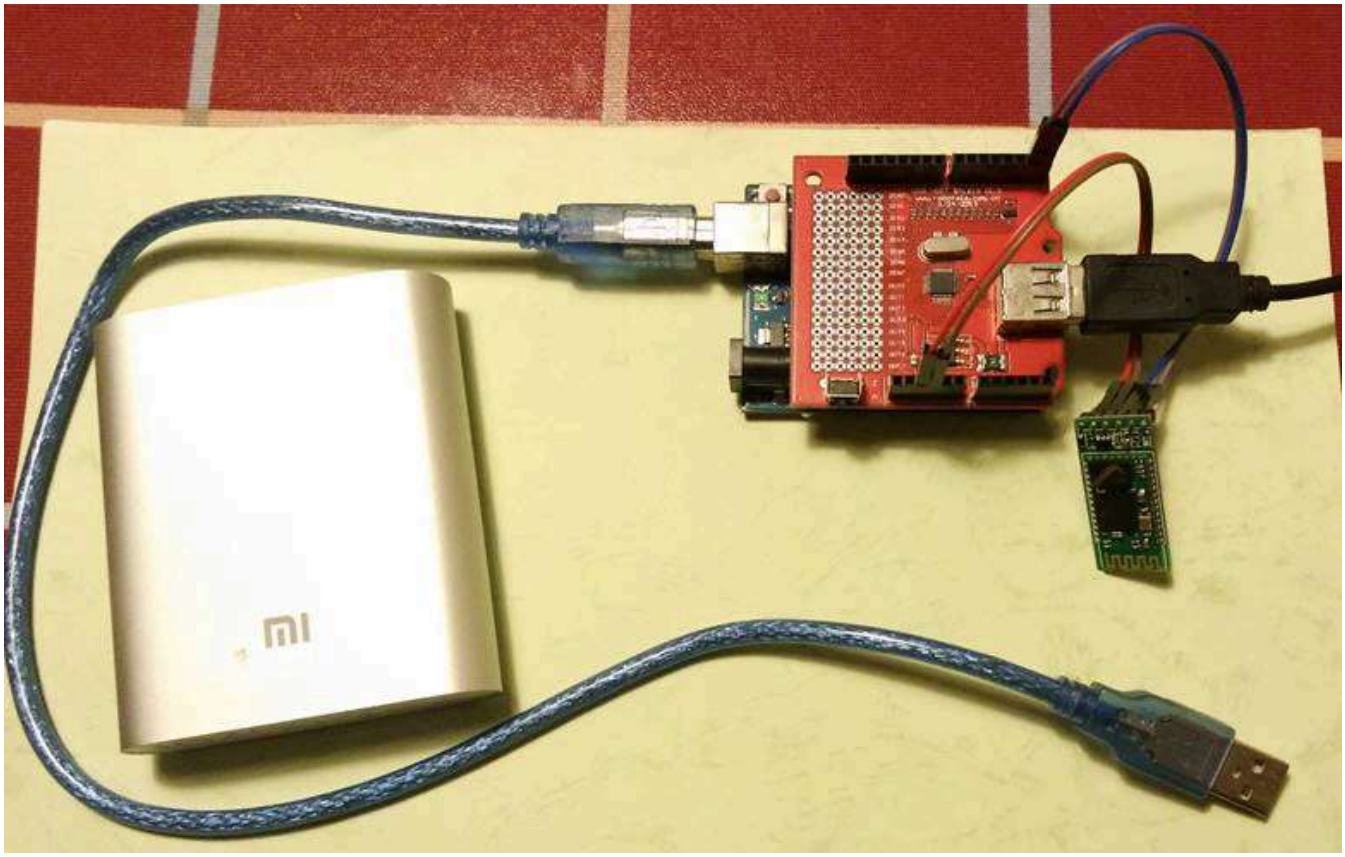
Arduino makes a USB Bluetooth keyboard adapter

This article shows you how to use Arduino to build a device that can turn your USB keyboard into a Bluetooth keyboard.

The keyboard can be regarded as the oldest device on the PC. Its appearance allows humans to interact with computers in a very simple way. Similarly, due to various historical reasons, the keyboard is also one of the most complex devices on the PC with the most compatibility issues (similar to the hard disk, but in the evolution from IDE to SATA, the standards are clear and the compatibility issues are much less).

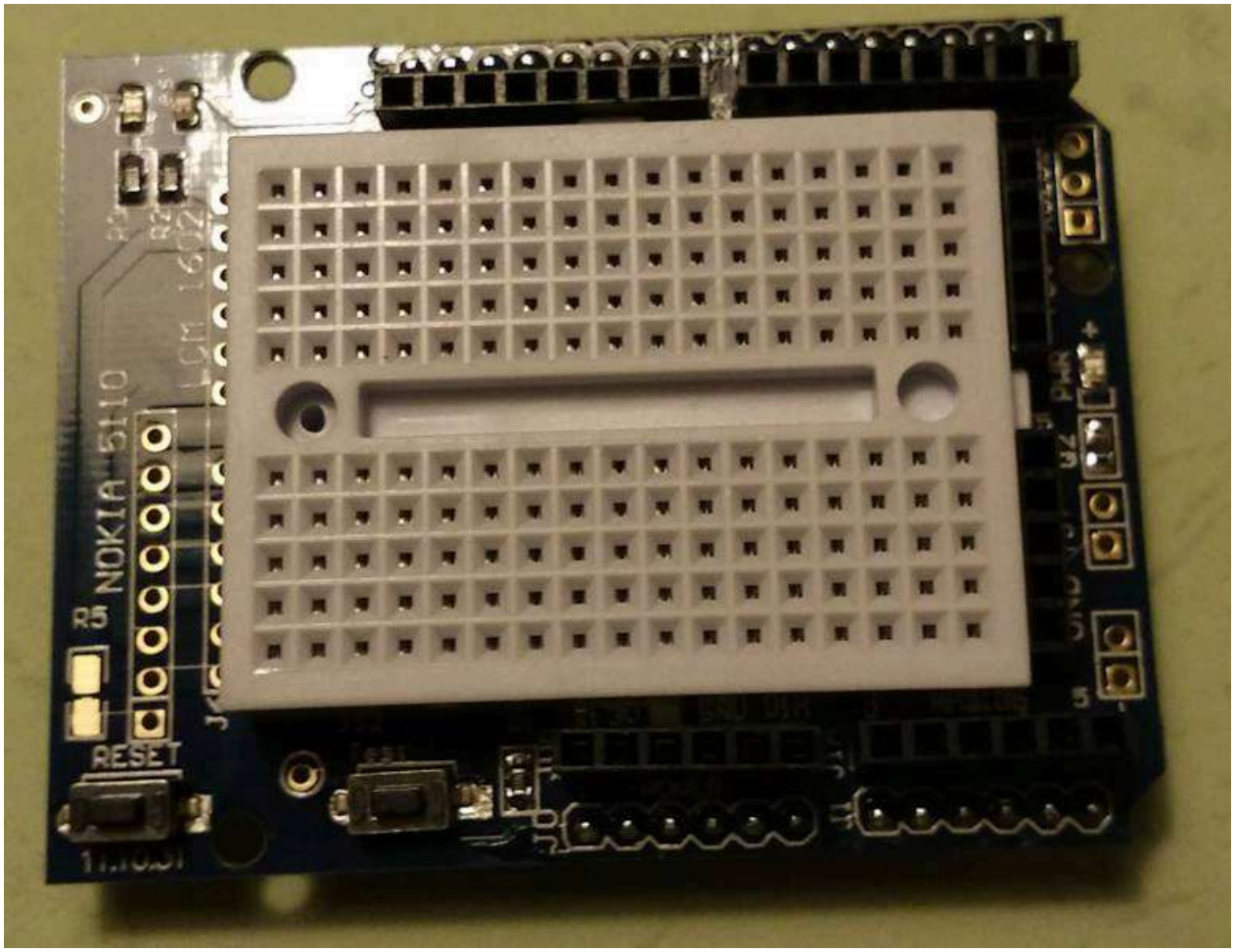
There is an article about DIY USB keyboard conversion to wireless on the Internet. Unfortunately, the article is wrong. The obvious mistake is that the author thinks that the keyboard is a one-way transmission, but in fact the transmission is two-way. For example, every USB communication requires the participation of the host and the slave, even the communication of the PS2 keyboard is the same. In addition, the switching of the uppercase and lowercase keys is controlled by the host.

Hardware: Arduino UNO, USB Host Shield and HID Bluetooth chip. I would like to emphasize that the HID Bluetooth chip used here is not an ordinary Bluetooth serial port transparent chip [specially note that it is "Bluetooth keyboard chip" or "Bluetooth barcode module"]. For this module, you can refer to my experiment in [Reference 1]. The hardware connection is very simple. The USB HOST Shield is plugged into the Arduino, and then VCC/GND/TX/RX connects the Arduino and the HID Bluetooth module together.

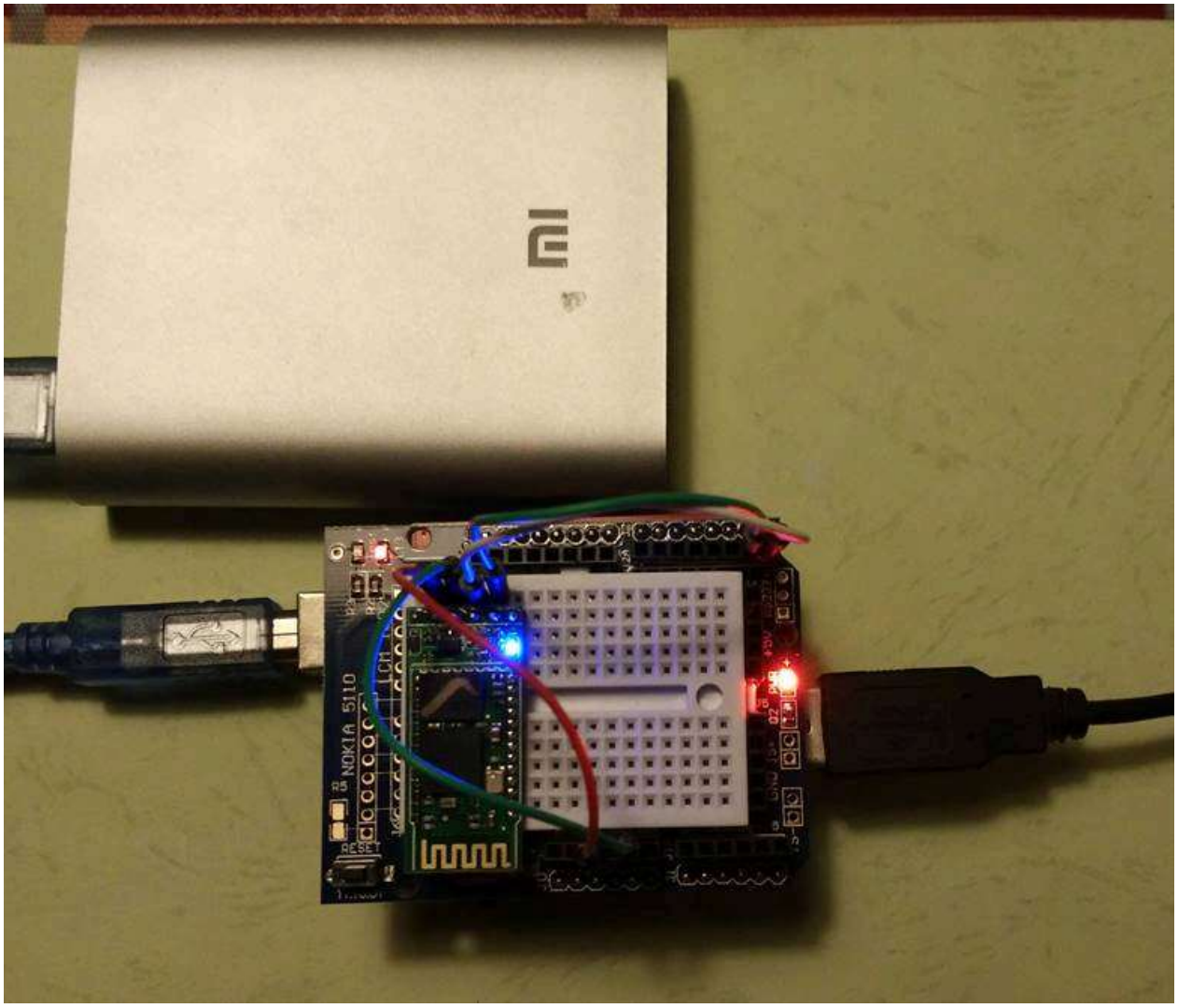


Principle: First, for universality and simple programming, we use USB HOST to send commands to switch the keyboard to Boot Protocol mode. In this way, even if the keyboard is different, the data sent each time is in the same format. Then, we can directly read the buffer data to parse the key information. Finally, the key information (Scan Code) is sent to the module through the serial port according to the format requirements of the HID Bluetooth module, and the host side receives it.

The above connection can work normally, but for the sake of aesthetics and reliability, I found a breadboard shield I bought before.



After plugging it in,



Specific code:

```
/* 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 characters (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.

==========Updated on May 11, 2017==========

Some friends said that Win, Shift, and Alt don't work. I happened to have some time to study it today. It turned out that there was something wrong with my code. In the parsed USB keyboard key information, Buf[0] is the flag bit of these keys. I didn't pass it to the module correctly. Therefore, I corrected the following two codes, one for sending and the other for saving the current key status each time:

```
        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 ];
    }
```

Modified code that works fine:

bkc2com1.1

Reference:

1. http://www.lab-z.com/btkeyboard/ Bluetooth keyboard module experiment

ziv2013 / September 20, 2015 / Funny

There are 44 ideas for "Arduino USB Bluetooth Keyboard Adapter"

Will

November 2, 2015 at 2:39 PM

Hello:
I am a beginner enthusiast who develops web programs. I want to use the Bluno nano development board to complete a function similar to the one in your article. 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
But after studying it for a while, I found that I didn't understand USB and Bluetooth well enough. The Bluno nano module is actually an Arduino + The Bluetooth module itself supports transparent transmission and HID. At first, I discussed the possibility of doing this function with their technical support. I planned to use the built-in USB to serial data (a function built into this board) to directly obtain the data inside the chip and then transmit it to Bluetooth for transmission. But after actually doing it, I found that it didn't work. After asking the technical support again, I found that USB and Bluetooth are actually a serial signal. It couldn't meet my requirements, so they recommended that I buy another USB to serial chip and connect the signal to a set of virtual serial pins connected by the chip.
There is an example (http://www.dfrobot.com.cn/community/forum.php?mod=viewthread&tid=2685) They recommended that I refer to your article, but after

downloading the original code, I found that it was difficult for me to fully understand it and design the logic according to the current hardware processing method.
So here I would like to ask you to take a look at how to modify the program to process it according to my method? At least give some ideas.
Thank you very much!

---

ziv2013

November 2, 2015 at 5:01 PM

1. The interface of a general Bluetooth module is a serial port, which means that if Arduino wants to send data, it needs to use the serial port output. If Arduino wants to receive data, it also needs to use the serial port. Therefore, it can be seen that there is a problem here: the serial port on ATmega328 is not enough.

2. Software Serial may have problems. For example, when I tested it, I found that for baud rates above 9600, this thing would have inexplicable problems.

3. I don't understand why you need to buy another USB to serial port? Do you plan to view the debug data? If you really want to do this, then remember that the serial port must be GND (common ground) except RX TX

4. If you want to make one with the same functions as mine, I suggest you buy the things I mentioned above. The price of the whole set will not be higher than that of a single Bluno nano... I specifically confirmed it and it is indeed the case.

5. When developing, especially when you don't understand something, it is recommended that you buy a large, powerful board. Once you understand it and are sure that the function can be realized, then find a way to shrink it. Accomplishing everything in one go is only possible under ideal circumstances with a small probability, and in reality, small probability events in a single experiment usually do not occur.

6. I have also used DFRobot's products, and the quality of their hardware far exceeds the quality of their services. Really.

So, the conclusion is: go buy a regular Uno and the Bluetooth module I recommended, the total cost is about 70~

Porn Proud Loli

November 17, 2015 at 6:35 AM

Hello, after reading your article, I found that you went through USB keyboard -> USB HOST -> serial port -> Bluetooth module. That is to say, the program written in the Bluetooth module is originally brought by the buyer, so how is the combination key in this HID Bluetooth module controlled? I started the development of your project a long time ago and also considered your solution. However, there are two reasons why I have not started to act so far. One is that the size is too large and the power consumption is too high. The other is that the HID Bluetooth module program is unknown and there are always many bugs, such as the combination key problem. So I am considering directly purchasing a Bluetooth module with a USB interface, and then writing the USB HOST program in the Bluetooth module. In this way, a single chip can meet all the needs directly and it is very power-saving, but I have not been able to complete this USB HOST so far.

---

ziv2013

November 17, 2015 at 9:18 AM

The key scan code in this HID Bluetooth module comes from the USB HID protocol, so the USB Host can just send it out directly after capturing it. I'm not sure if there is a problem with the key combination, but there is no problem with typing in general applications. Theoretically, it is impossible to have a Bluetooth chip with a USB HOST, because these two things are quite complex protocols, and it doesn't make much sense to integrate them together. The size of the USB host and the Bluetooth chip is not large, and there is a chance to put them on a very small board.

Porn Proud Loli

November 17, 2015 at 7:25 PM

It is precisely because there is no such thing. In fact, many CSR Bluetooth chips are equipped with USB interfaces. So we need to write HOST programs into these chips. If something does not exist in the world, we have to create it ourselves.

---

ziv2013

November 18, 2015 at 12:52 PM

Well, it's good, and you have some ideas. If you have the chance, read more books and learn English well. Come on!

---

batsing

June 28, 2021 at 10:45 AM

There are many Bluetooth chips with USB slave devices, such as the classic CC2450, but I haven't found one with USB HOST yet.

---

ziv2013

June 29, 2021 at 9:08 AM

How many WCH families can

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

Thanks to @ziv2013, I have successfully completed the experiment according to this tutorial, and made an upgraded version with a small OLED screen. You can check out the code on my Github at the following address: https://github.com/pengfaliu/lfp-arduino-studio/tree/master/sketchs/bluetooth-keyborad-transverter.

The hardware part is manually soldered into the small box, you can also refer to it.

---

ziv2013

March 8, 2022 at 5:04 pm

That's good. I'm waiting for ESP32 S3, and then there will be a simpler solution.

Minions Geek

March 14, 2022 at 10:11 am

I am an outsider. I am just a software programmer. My level is not as high as yours. I will learn from you when your ESP32-S3 comes out. Haha