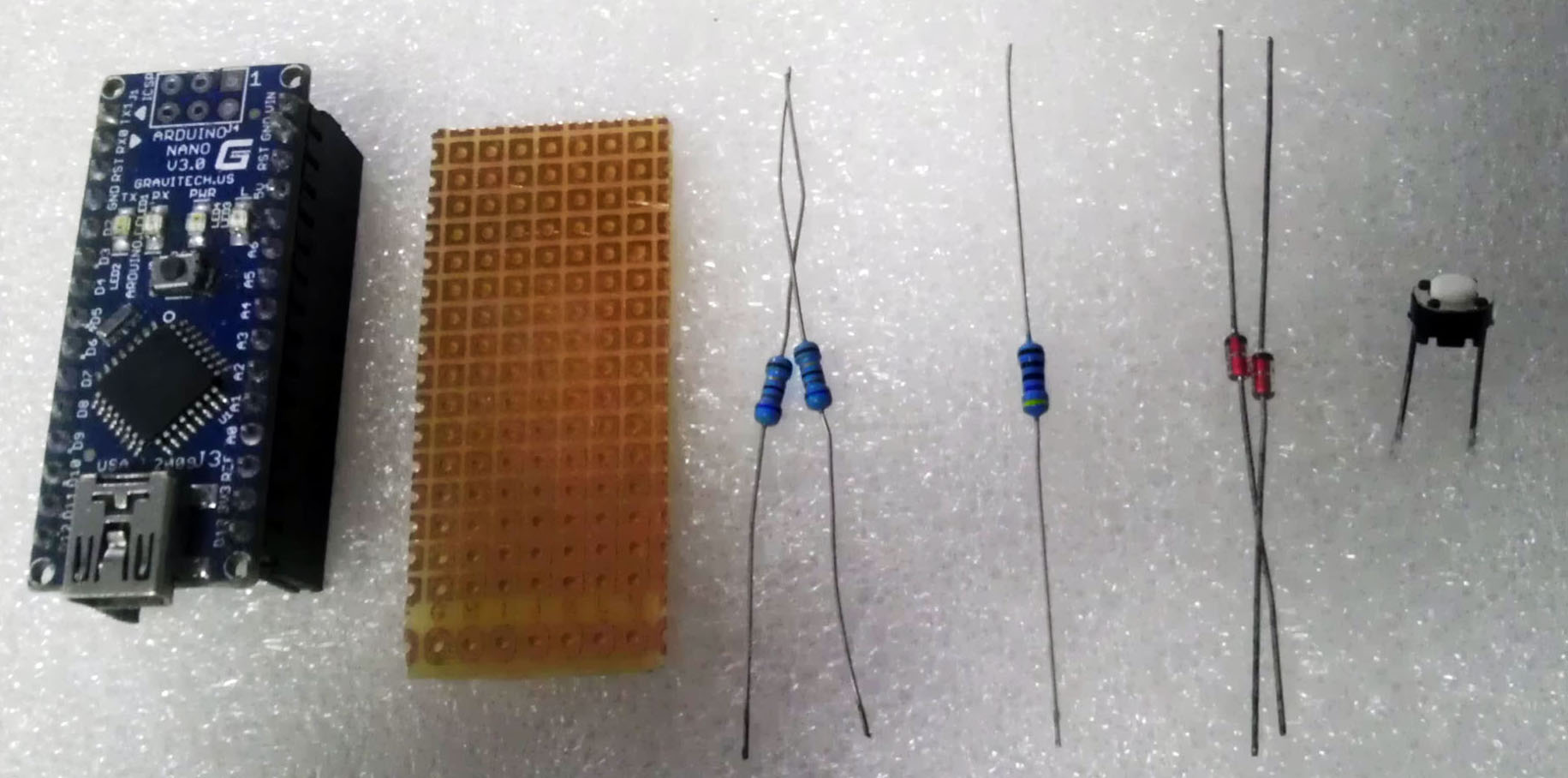
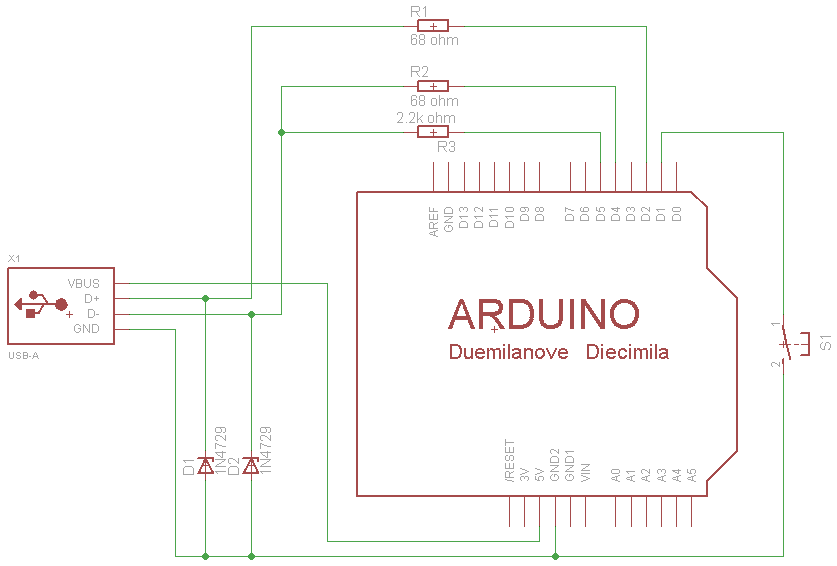
Arduino模拟键盘

键盘作为经典的输入设备，使用在很多互动中都有特别的优势，比如我们可以通过键盘直接给flash传递按键事件。而无需通过串口之类的特殊接口，虽然我们可以拆一个传统的键盘，然后将里面的按键引出来，但是这样有一个缺点，就是键值不能动态改变并且不能一次多键。使用模拟键盘的话，我们就可以随意在程序设置按键的时间和键值。比如本文的例子就是按下一个按键，模拟键盘就在电脑输入“HELLO WORLD”。  
  
硬件部分：  
材料清单：  
Arduino x1  
68Ω电阻 x2 (没有68Ω的话，用50~100Ω电阻也行)  
2.2kΩ电阻 x1 (没有2.2kΩ的话，用1.5k~2.2k电阻也行)  
USB连接线（一端是USB口，另一端是电线） x1  
3.6v 稳压管（建议选用功耗0.25~0.5w的）x2  
   
  
电路原理图：  
   
电路图解说：  
1、两个68Ω的电阻起到限流和保护作用，防止在意外情况下损坏计算机的USB端口或单片机的端口。  
2、2.2kΩ电阻是上拉电阻，用于分辨总线状态。如果上拉电阻接于D+和+5v端则是高速USB设备，接于D-与+5v端，则是低速设备。此处键盘传输速率不高，接到D-作为低速设备即可。  
3、D+和D-上的3.6V稳压二极管D1和D2起到限制数据线上的电平的作用。因为在USB规范中规定数据线D+和D-上的电平范围是3.0V至3.6V，而AVR单片机的输出电平是Vcc。如果单片机的Vcc是5V，在没有D1和D2的情况下将造成电平不匹配，会造成在很多计算机中无法正确识别出USB设备。如果用户系统的Vcc在3.0V至3.6V之间，就可以省略这两个稳压二极管。从这里也可以看出用户系统的Vcc必须高于3V。  
4、由于低速AVRUSB所需要的是1.5MHz时钟，而单片机每8条指令就能精确完成一个数据位的采集。所以AVRUSB最小单片机时钟频率是12MHz。并且可以使用的时钟频率有12MHz、12.8MHz、15MHz、16MHz、16.5MHz、20MHz，其他的不支持。所以如果使用最小系统制作此模拟键盘的话8MHz的话，ATMega8L不能用。  
————————  
软件部分：  
  
Arduino支持库文件：  
[http://soft1.wmzhe.com/download/ ... ino/UsbKeyboard.zip](http://soft1.wmzhe.com/download/Ansifa2012/Arduino/UsbKeyboard.zip)  
文件下载后解压到arduino编译器的libraries文件夹下面。

注意：  
  
1、UsbKeyboard库中，usbconfig.h里面可以更改USB接线的引脚定义，下面给出一个大概解释：

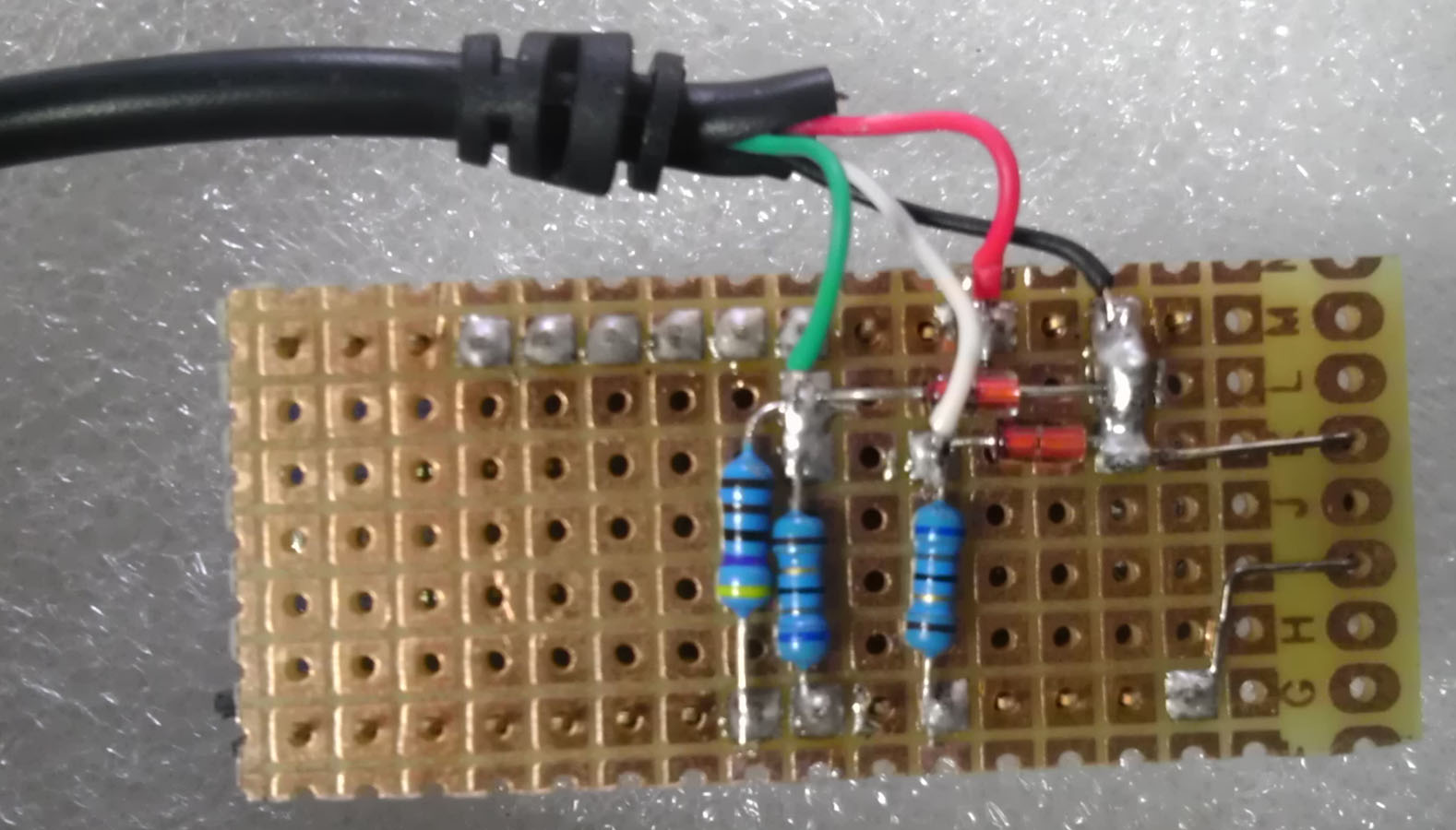
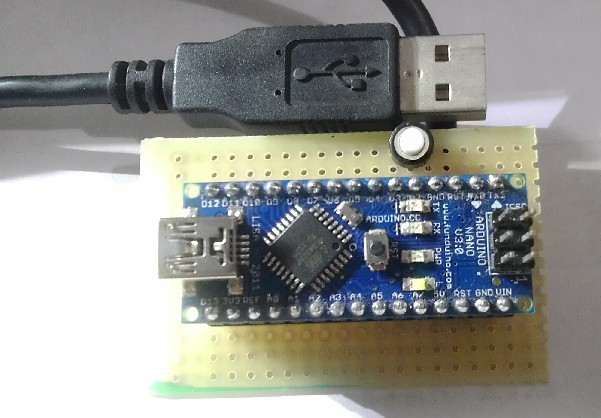
|  |  |
| --- | --- |
| #define USB\_CFG\_IOPORTNAME D | USB输入输出引脚使用AVR单片机的PORTD，如果改成B就是使用PORTB |
| #define USB\_CFG\_DMINUS\_BIT 4 | USB的D-接PORTD的第四位PD4，对应Arduino D4 |
| #define USB\_CFG\_DPLUS\_BIT  2 | USB的D+接PORTD的第二位PD2，对应Arduino D2 |
| #define USB\_CFG\_PULLUP\_IOPORTNAME D | USB上拉引脚使用AVR单片机的PORTD，如果改成B就是使用PORTB |
| #define USB\_CFG\_PULLUP\_BIT  5 | USB的上拉电阻接PORTD的第五位PD5，对应Arduino D5 |

2、在UsbKeyboard库的UsbKeyboard.h里面，有关于模拟键值的表  
#define KEY\_A       4  
#define KEY\_B       5  
#define KEY\_C       6  
#define KEY\_D       7  
#define KEY\_E       8  
等。但不齐全。经过测试，其实这个键盘基本可以模拟几乎所有键值（Power，Sleep，Pause似乎不能）。  
比如方向键右左下上分别对应79,80,81,82数字。即写成  
UsbKeyboard.sendKeyStroke(79);  
UsbKeyboard.sendKeyStroke(81);  
等。由于整理比较麻烦，大家可以自己下载个KeyboardTest软件测试不同数字下面的键值。

程序示例：  
下面的例子演示了用Arduino虚拟键盘的应用例子。打开记事本，然后将Arduino的D12引脚和GND连起来，就会打印HELLO WORLD字样。

ARDUINO CODECOPY

1. /\*
2. Arduino模拟键盘 by Ansifa
3. 2012.6.8
5. 功能描述：插上此模拟键盘，打开记事本，然后按下按钮，即可在记事本打印出HELLO WORLD字样
7. 接线方法：
8. <img src="http://www.geek-workshop.com/forum.php?mod=image&aid=5359&size=300x300&key=e9fa5559e6d5724d51f770bc6c55b941&nocache=yes&type=fixnone" border="0" aid="attachimg\_5359" alt="">
9. Arduino D2接68Ω电阻后，接USB线D+
10. Arduino D4接68Ω电阻后，接USB线D-
11. Arduino D5接2.2kΩ电阻后，接USB线D-
12. Arduino D2接3.6v稳压管到GND
13. Arduino D4接3.6v稳压管到GND
14. +5v接USB线VCC
15. GND接USB线GND
16. Arduino D1接一个开关到GND
18. 附：USB线序颜色（由于各生产厂不同，不一定准确，仅供参考）
19. \*USB键鼠：      |        \*USB接口
20. 白<->VCC        |        红<->VCC
21. 橙<->D-         |        白<->D-
22. 绿<->D+         |        绿<->D+
23. 蓝<->GND        |        黑<->GND
25. \*/
27. #include "UsbKeyboard.h"
28. **int** KEYPIN = 1;                //按键接在D1引脚，也可以改成任何引脚
29. **void** **setup**()
30. {
31. TIMSK0 &= !(1 << TOIE0);        //
32. **pinMode**(KEYPIN, INPUT);
33. **digitalWrite**(KEYPIN, HIGH);
34. }
35. **void** **loop**()
36. {
37. UsbKeyboard.update();
38. **if**(**digitalRead**(KEYPIN) == HIGH)
39. {
40. **delay**(100);
41. **if**(**digitalRead**(KEYPIN) == LOW)
42. {
43. UsbKeyboard.sendKeyStroke(KEY\_H);
44. UsbKeyboard.sendKeyStroke(KEY\_E);
45. UsbKeyboard.sendKeyStroke(KEY\_L);
46. UsbKeyboard.sendKeyStroke(KEY\_L);
47. UsbKeyboard.sendKeyStroke(KEY\_O);
48. UsbKeyboard.sendKeyStroke(KEY\_SPACE);
49. UsbKeyboard.sendKeyStroke(KEY\_W);
50. UsbKeyboard.sendKeyStroke(KEY\_O);
51. UsbKeyboard.sendKeyStroke(KEY\_R);
52. UsbKeyboard.sendKeyStroke(KEY\_L);
53. UsbKeyboard.sendKeyStroke(KEY\_D);
54. UsbKeyboard.sendKeyStroke(KEY\_ENTER);
55. }
56. }
57. }

注意，先插上Arduino数据线，将程序写入Arduino。然后拔掉数据线，将模拟键盘USB线接到电脑，即可使用。  
  
下面随便点实物图  
  
刚焊好的正反面  
   
  
插数据线写程序进Arduino：  
