

PS2无线手柄快速上手教程

使用前准备

程序分析

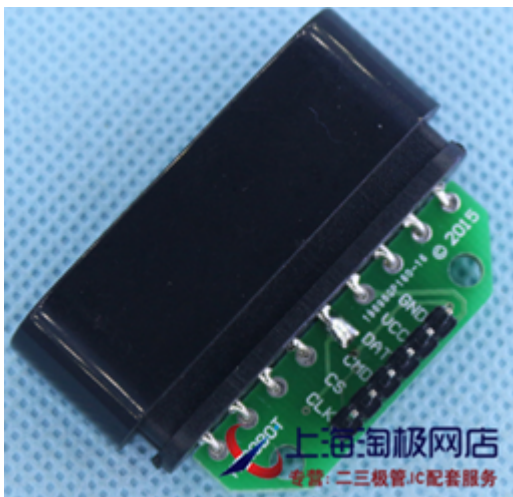
使用前准备

- 将接收器连接到arduino上（共用到6个引脚，程序中使用的是8、9、10、11号引脚，接收器跟arduino的连线可根据实际情况修改引脚。如果没有接收器转接板，请查看接收器引脚说明）。
- 下载例程（使用的是arduino uno r3），打开串口监视器，按下遥控按键查看效果（具体使用见下面程序分析）。
- 程序源码链接：

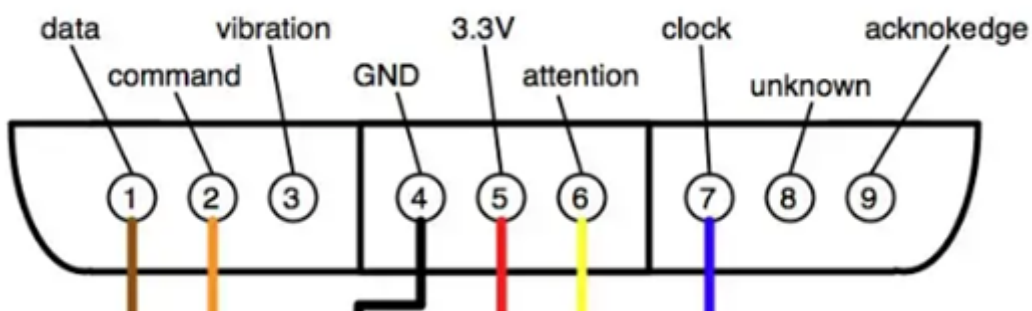
<https://pan.baidu.com/s/11EbPFIEPorOjgwS98srUjA> 提取码: j6vj

```
9  #define PS2_DAT      8
10 #define PS2_CMD      9
11 #define PS2_CS       10
12 #define PS2_CLK      11
```

引脚需要与程序相对应



接收器转接板



接收器引脚说明

程序分析

1. 首先将文件PS2X_lib.cpp, PS2X_lib.h拷贝到项目文件夹下。

此电脑 > LENOVO (F:) > 桌面 > example > PS2X_Example				
名称	修改日期	类型	大小	
PS2X_Example.ino	2018-09-26 20:50	Arduino file	4 KB	
PS2X_lib.cpp	2013-07-07 10:18	C++ Source File	13 KB	
PS2X_lib.h	2013-07-07 10:18	H 文件	8 KB	

2. 添加头文件。

```
1 #include "PS2X_lib.h" //for v1.6
```

3. 遥控接收器引脚声明（需要跟实际连线相对应）。

```
9 #define PS2_DAT      8
10 #define PS2_CMD      9
11 #define PS2_CS       10
12 #define PS2_CLK      11
13
```

4. 实例化一个PS2对象

```
23 PS2X ps2x; // create PS2 Controller Class
```

5. arduino初始化

```
25 void setup() {
26     Serial.begin(57600);
27
28     /*right now, the library does NOT support hot pluggable controllers, meaning
29     you must always either restart your Arduino after you connect the controller,
30     or call config_gamepad(pins) again after connecting the controller.*/
31     //setup pins and settings: GamePad(clock, command, attention, data, Pressures?, Rumble?)
32     int error = 0;
33     do {
34         error = ps2x.config_gamepad(P2_CLK, P2_CMD, P2_CS, P2_DAT, pressures, rumble);
35         if (error == 0) {
36             break;
37         } else {
38             delay(100);
39         }
40     } while (1);
41 }
```

ps2x.config_gamepad(P2_CLK, P2_CMD, P2_CS, P2_DAT, pressures, rumble);
PS2引脚初始化，返回值为0说明遥控连接成功。

6. 在循环中读取遥控数据

```
ps2x.read_gamepad(false, 0);
```

ps2x.read_gamepad(false, 0);

You must Read Gamepad to get new values and set vibration values
ps2x.read_gamepad(small motor on/off, larger motor strenght from 0-255), if
you don't enable the rumble, use ps2x.read_gamepad(); with no values. You
should call this at least once a second.

PS2遥控内部是有两个震动马达的，这里我们不使用。

调用此函数读取遥控器状态，每个循环调用一次，但不要太频繁。

按键状态读取有三个函数

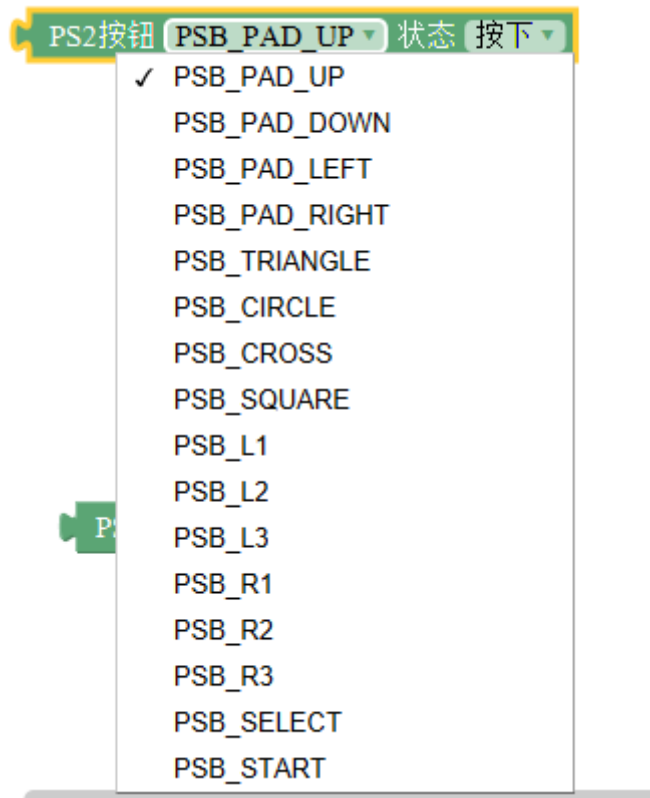
```
46 if (ps2x.Button(P2_START)) //will be TRUE as long as button is pressed
47     Serial.println("Start is being held");
48 if (ps2x.ButtonPressed(P2_SELECT))
49     Serial.println("Select is being held");
50
```

```
53 } else if (ps2x.ButtonReleased(P2_PAD_UP)) {
54     Serial.println("Up Button Released!");
55 }
```

ps2x.Button(x),只要按键按下会一直触发，按下时串口一直打印数据。

ps2x.ButtonPressed(x), 按键按下时触发一次，按下时串口只打印一次数据。

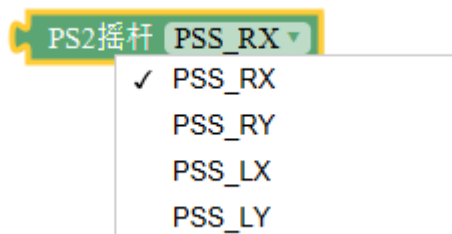
ps2x.ButtonReleased(x), 按键释放时触发一次, 松开时串口只打印一次数据。
x表示按键的宏定义, 具体如下图:
具体效果请自行测试。



摇杆数据读取

```
82     if (ps2x.Button(PSS_L1) || ps2x.Button(PSS_R1)) { //print stick values if either is TRUE
83         Serial.print("Stick Values:");
84         Serial.print(ps2x.Analog(PSS_LY), DEC); //Left stick, Y axis. Other options: LX, RY
85         Serial.print(",");
86         Serial.print(ps2x.Analog(PSS_LX), DEC);
87         Serial.print(",");
88         Serial.print(ps2x.Analog(PSS_RY), DEC);
89         Serial.print(",");
90         Serial.println(ps2x.Analog(PSS_RX), DEC);
91     }
92     delay(30);
```

ps2x.Analog(x), x表示摇杆的宏定义, 具体如下图:



遥控读取数据的教程就到这里了, 按键的具体功能就需要结合自己的机器人使用了。