



PS2 无线遥控手柄

用户手册



关注塔克创新微信公众号，获取更新资讯

烟台塔克电子科技有限公司

版权申明

本手册版权归属塔克创新所有，并保留一切权力，受法律保护。非经(书面形式)同意，任何单位及个人不得擅自摘录或修改本手册部分或全部内容，违者我们将追究其法律责任。

版本说明

版本	日期	内容说明
V1.0	2024/6/18	第一次发布

塔克媒体

塔克官网	www.xtark.cn
淘宝店铺	https://xtark.taobao.com
塔克哔哩	https://space.bilibili.com/511052131
销售邮箱	sales@xtark.cn

本手册塔克 PS2 无线遥控手柄用户使用手册。

1. 手柄说明

索尼的 PS2 无线手柄是索尼游戏机的遥控手柄，控制协议已被全球创客破解，可以方便用于机器人控制器，作为遥控输入设备。这款手柄价格便宜，性价比极高。



1.1. 硬件介绍

手柄按键定义如下图，具有 2 个 360°摇杆和 17 个物理按键，摇杆也可以作为按键使用。



手柄通过 2 节 7 号电池供电，使用前请安装电池。并将下图所示开关打开，拨动到 ON 一侧为打开。



1.2. 配对说明

手柄电源打开后，在未搜索到接收器的状况下，手柄的红灯或绿灯会快速闪烁，进入等待配对状态。

接收器仅供电，不连接控制信号线，手柄未开启状态下，接收器电源灯红灯常亮，信号灯绿灯快速闪烁，进入等待配对状态。此时开启手柄开关，手柄和接收器完成配对，手柄两个指示灯常亮，接收器的红灯常亮，绿灯进入慢闪状态。

如果手柄接收器已经连接了单片机控制信号，并运行手柄程序，手柄和接收器配对成功后，手柄和接收器的两个指示灯会常亮显示。

1.3. 休眠说明

无配对状态下，30s 后启动休眠模式；开启并配对完成后，5 分钟无按键按下，自动启动休眠模式。按下“START”即可唤醒手柄。

1.4. 接收器

手柄无线接收器引脚定义。



1-DATA: 信号流向从手柄到主机。此信号是一个 8 bit 的串行数据，同步传送于时钟下降沿 (输入输出信号在时钟信号由高 到低时变化，所有信号的读取在时钟前沿到电平变化之前完成。)

2-COMMAND : 信号流向从主机到手柄。此信号和 DATA 相对，同样是一个 8 bit 的串行数据，同步传送于时钟下降沿。

3-N/C (9 Volts unused)

4-GND

5-VCC : 电源电压从 5V 到 3V 原装的索尼手柄都可以工作。主机主板上装有表面安装的 750mA 保险丝，用于防止外设过载 (750mA 是包括左右手柄和记忆卡)。

6-ATT : 用于提供手柄触发信号。信号在通信期间处于低电平。又有人将此针脚叫做 CS。

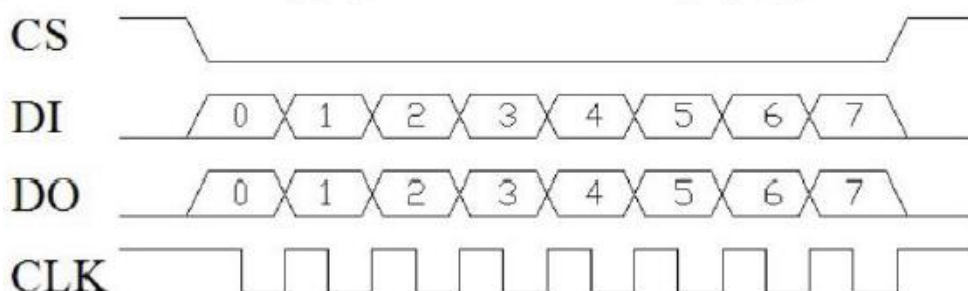
7-CLOCK : 信号流向从主机到手柄。用于保持数据同步。

8-N/C

9-ACK : 从手柄到主机的应答信号。此信号在每个 8 bits 数据发送之后的最后一个时钟周期变低，并且 ATT 一直保低电平。如果 ACK 信号不变低约 60 微秒 PS 主机会试另一个外设。

引脚	定义	引脚说明
①	DI/DAT	信号从手柄到手机，8bit串行数据，在时钟的下降沿进行传送。读取信号需要在时钟由高到低的变化过程中进行。
②	DO/CMD	信号从主机到手柄，8bit串行数据，也是在时钟的下降沿进行传送。此信号与DI/DAT信号相对应。
④	GND	手柄电源地信号。
⑤	VDD	接收器工作电源，供电电压范围3~5V。
⑥	CS/SEL	手柄工作使能信号，在通信期间，设置为低电平。
⑦	CLK	时钟信号，由主机提供，用于保持数据同步。

PS 手柄信号，所有通讯都是 8 bit 串行数据最低有效位先行。在 PS 手柄总线的所有时码在时钟下降沿都是同步的。传送一个字节的情况如下所示。数据线的逻辑电平在时钟下降沿驱动下触发改变。数据的接收读取在时钟的前沿 (在记号*处) 到电平变化之前完成。在被选手柄接收每个 COMMAND 信号之后，手柄需拉低 ACK 电平在最后一个时钟。如果被选手柄没 ACK 应答主机将假定没手柄接入。



当 PS 主机想读一个手柄的数据时，将会拉低 ATT 线电平并发出一个开始命令 (0x01)。手柄将会回复它的 ID (0x41=数字, 0x23=NegCon, 0x73=工作模式)。在手柄发送 ID 字节的同时主机将传送 0x42 请求数据。随后命令线将空闲和手柄送出 0x5A 意思说：“数据来了”。

字节定义如下。左右摇杆模拟值为无效，推到极限时，对应发送 UP、RIGHT、DOWN、LEFT、△、○、X、□，按键 L3、R3 无效。

BYTE	CMND	DATA
01	0x01	idle
02	0x42	0x73
03	idle	0x5A Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7
04	idle	data SLCT JOYR JOYL STRT UP RIGHT DOWN LEFT
05	idle	data L2 R2 L1 R1 △ ○ X □
06	idle	data Right Joy 0x00 = Left 0xFF = Right
07	idle	data Right Joy 0x00 = Up 0xFF = Down
08	idle	data Left Joy 0x00 = Left 0xFF = Right
09	idle	data Left Joy 0x00 = Up 0xFF = Down

1.5. 转接板

为了方便使用，我们设计了多种款式转接板。



直插转接板(排针)



直插转接板(XH2.54)



横插转接板(直排针)



横插转接板(弯排针)



横插转接板(XH2.54)

市面上 PS2 手柄工作电压均为 3.3V，虽然使用 5V 供电也能工作，一旦连接舵机之类感性负载，手柄必定烧坏。本转接板做了专门保护设计。

- 带有 3.3V 稳压电路，无论您连接 5V 还是 3.3V 电源，手柄供电均为 3.3V，供电稳定不烧手柄。
- 数据接口均加有保护电阻，不管您是 3.3V 还是 5V 单片机，都兼容。
- 电源输入带有反接保护电路，有效保护您的手柄、控制器不被损坏。



背面带稳压、防反接电路
全面防护，耐用，防烧

接口处有清晰的引脚定义丝印标识，方便用户接线，无线手柄通讯一共需要用到 6 个引脚。



转接板尺寸规格如下，固定孔位 M3，可兼容 M2.5 螺丝。



1.6. 锁定款使用

普通款 PS2 手柄开机后自动搜寻无线接收器并配对连接，这种连接方式优点是比较灵活，可以支持一个手柄和多个接收器配合使用。缺点是在多个手柄同时使用的比赛场景，配对比较混乱，容易造成交叉配对。

锁定款 PS2 手柄很好的解决了上述问题，手柄与接收器支持锁定配对，锁定后手柄只与该接收器连接，在多个手柄同时使用场景不会配对混乱情况，特别适合机器人、智能小车竞赛场景。

锁定款手柄和普通款主要区别是接收器多了一个隐藏的按键，可以通过顶针或牙签进行按键操作，用于解除配对锁定。



锁定款手柄发货默认已配对锁定。如果手柄和接收器上电后都闪烁，按一下锁定按键，即可重新配对锁定。

接收器如需绑定其它手柄，可关闭当前手柄电源，打开另一个待绑定的手柄，此时接收器和手柄均会闪烁，按一下锁定按键即可重新配对锁定。

2. STM32 例程说明

本小节讲解通过 STM32 连接手柄的使用，提供 STM32F103RCT6 和 STM32F407VET6 两个版本程序。两个程序相似，这里以 STM32F103 为例讲解。

2.1. 硬件连接

例程硬件连接关系如下。

GND——GND

VCC——3.3V 或 5V

DATA——PA4

CMD——PA5

CS——PA6

CLK——PA7

2.2. 程序说明

PS2 手柄驱动程序主要在 ax_ps2.c 和 ax_ps2.h 文件中，程序按照 X-SOFT 规范设计，接口简洁，只有两个接口函数。

```
void AX_PS2_Init(void); //手柄初始化
void AX_PS2_ScanKey(JOYSTICK_TypeDef* JoystickStruct); //PS2 获取按键及摇杆数值
```

关于 PS2 键值，这里定义了结构体数据结构，定义内容如下。

```
//PS2手柄键值数据结构体
typedef struct
{
    uint8_t mode;           /* 手柄的工作模式 */
    uint8_t btn1;           /* B0:SLCT B1:JR B0:JL B3:STRT B4:UP B5:R B6:DOWN B7:L */
    uint8_t btn2;           /* B0:L2 B1:R2 B2:L1 B3:R1 B4:Y B5:B B6:A B7:X */
    uint8_t RJoy_LR;        /* 右边摇杆 0x00 = 左 0xff = 右 */
    uint8_t RJoy_UD;        /* 右边摇杆 0x00 = 上 0xff = 下 */
    uint8_t LJoy_LR;        /* 左边摇杆 0x00 = 左 0xff = 右 */
    uint8_t LJoy_UD;        /* 左边摇杆 0x00 = 上 0xff = 下 */
}JOYSTICK_TypeDef;
```

主程序也比较简洁，完成 PS2 初始化，即可获取手柄数据。通过串口输出手柄获取的数据值。

```
//延时函数初始化
AX_DELAY_Init();

//LED 初始化
AX_LED_Init();

//调试串口初始化
AX_UART1_Init(115200);

printf("***OpenCTR PS2 无线手柄例程***\r\n\r\n");

//功能配置
AX_PS2_Init(); //PS2 初始化

while (1)
{
    //手柄键值扫描
    AX_PS2_ScanKey(&my_joystick);

    //打印手柄键值
    printf("MODE:%2x BTN1:%2x BTN2:%2x RJOY_LR:%2x RJOY_UD:%2x LJOY_LR:%2x LJOY_UD:%2x\r\n",
        my_joystick.mode, my_joystick.btn1, my_joystick.btn2,
        my_joystick.RJoy_LR, my_joystick.RJoy_UD, my_joystick.LJoy_LR,
        my_joystick.LJoy_UD);

    AX_Delays(30);
}
```

测试结果如下图，可以按到模式，按键值，及摇杆数据变化情况。



3. Arduino 例程说明

本小节讲解通过 Arduino 连接手柄的使用，以 Arduino UNO 为例，采用 PS2X_lib 库。

3.1. 硬件连接

例程硬件连接关系如下。

GND——GND

VCC——5V

DATA——13

CMD——11

CS——10

CLK——12



3.2. 程序说明

具体程序源码，请参考源码文件。PS2X 库的使用网上有非常多教程，这里不再赘述。

```
PS2X_Example | Arduino 1.8.19
文件 编辑 项目 工具 帮助

PS2X_Example
#include <PS2X_lib.h>

#include <PS2X_lib.h>

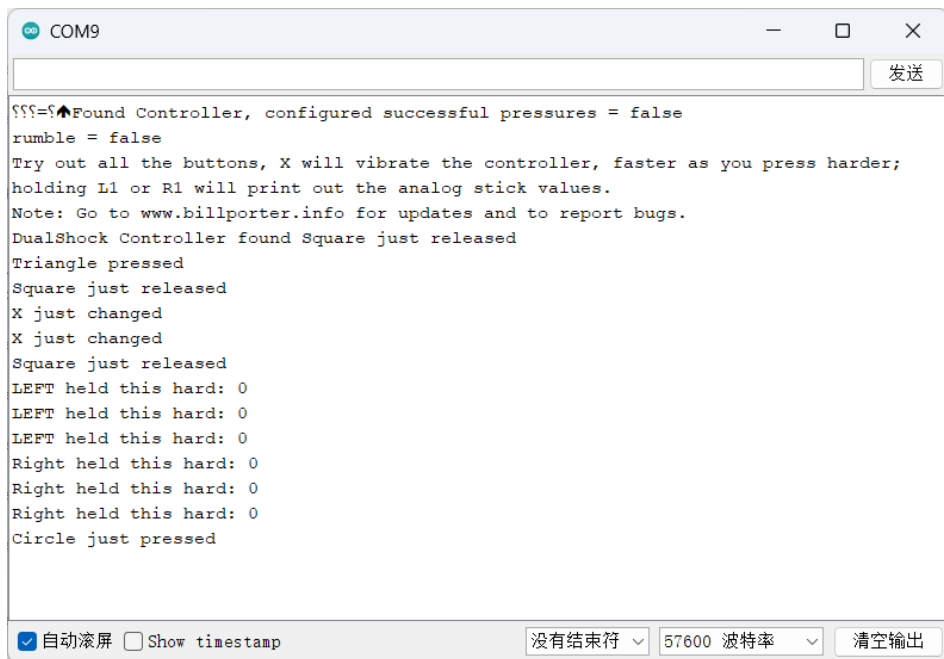
#include <PS2X_lib.h> //for v1.6

/*****
 * set pins connected to PS2 controller:
 * - 1e column: original
 * - 2e column: Stef?
 * replace pin numbers by the ones you use
 *****/

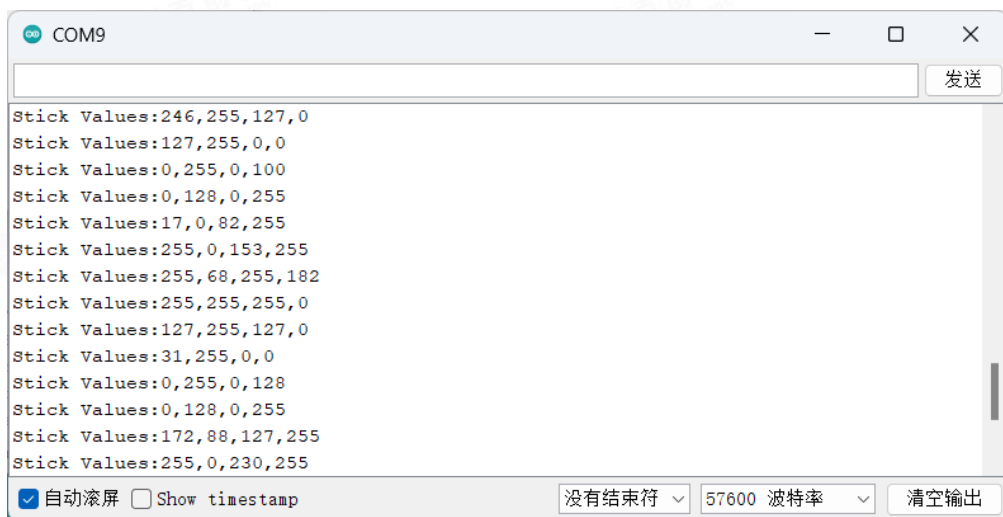
#define PS2_DAT      13 //14
#define PS2_CMD      11 //15
#define PS2_SEL      10 //16
#define PS2_CLK      12 //17

/*****
 * select modes of PS2 controller:
 * - pressures = analog reading of push-buttons
 * - rumble     = motor rumbling
 * uncomment 1 of the lines for each mode selection
 *****/
```

程序下载到 Arduino 后，打开对应串口，就可以参考 PS2 手柄数据的输出，按键就会有对应键值显示。



注意使用摇杆，需要按住 R1 或 L1 按键。波动摇杆，可以观察到数据变化。



4. 51 单片机例程说明

详见程序源码

5. FAQ 问题

如何判断 PS2 手柄与接收器连接成功?

给 PS2 和接收器供电，当你的 PS2 的指示灯，不再频闪开始常亮，说明你的 PS2 手柄与接收器连接成功。

关于 PS2 手柄红绿灯模式切换?

PS2 手柄在 STM32 例程中，通过 MODE 按键可以调节模式，这里建议调节到红灯和绿灯都亮起的红绿灯模式再使用。Arduino 例程 MODE 按键无效。