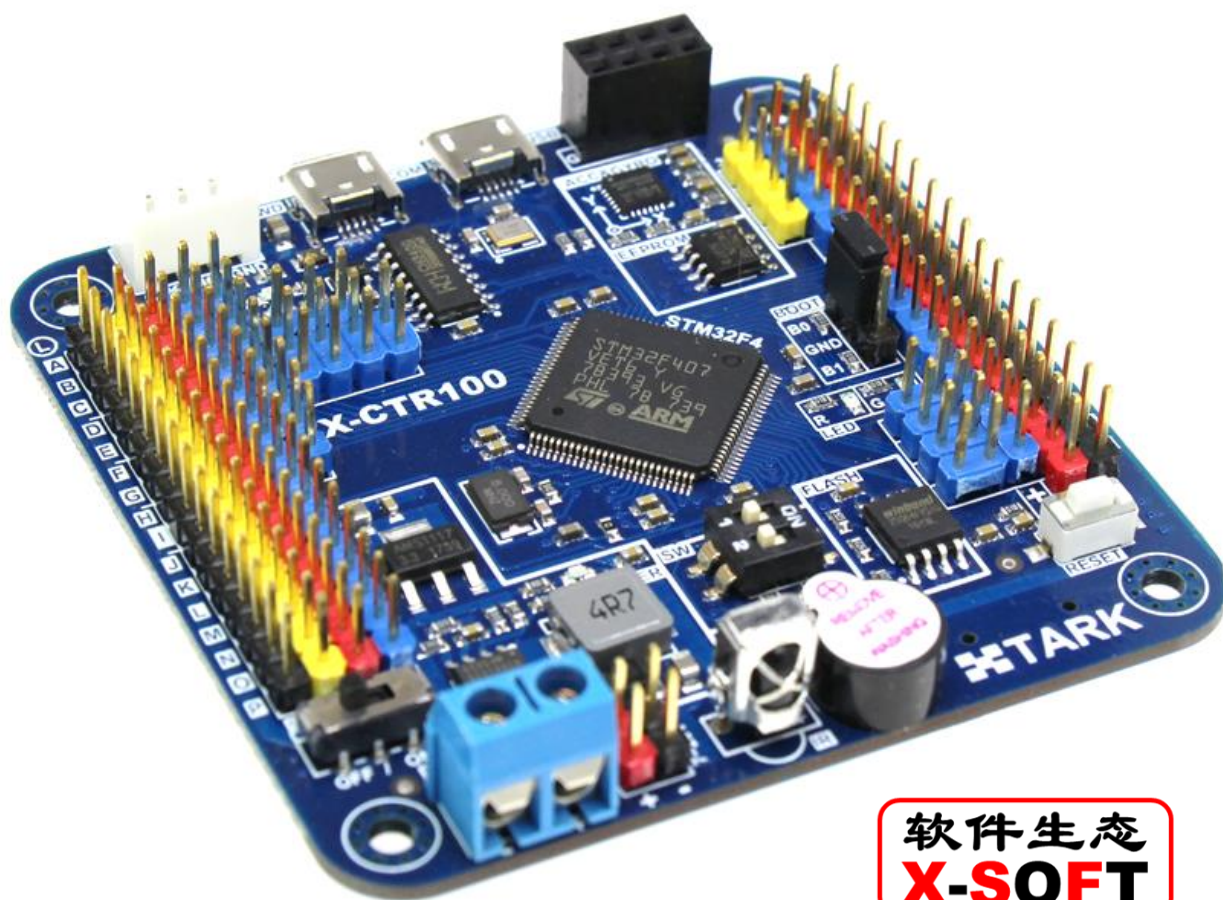


XTARK 塔克创新 X-CTR100 控制器

【玩转 X-CTR100】系列教程

PS2 无线手柄



软件生态
X-SOFT

修订记录

版本	日期	修改记录
V1.0	2018-1-1	创建文档

版权申明

本教程版权归属塔克创新所有，并保留一切权力。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。

此教程不断更新中，建议您登陆塔克创新社区或博客了解下载最新版本。

塔克淘宝：

<https://shop246676508.taobao.com/>

塔克官网社区：

www.xtark.cn

塔克博客：

<http://www.cnblogs.com/xtark/>



SONY 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，本文使用 X-CTR100 控制器对 PS2 进行解码，并显示键值，方便用户使用。

原理

PS2 无线手柄按键比较多，例如在一款游戏中的设定：



PS 手柄无线接收器，针脚输出端(面对插头)



PIN 1->| o o o | o o o | o o o |

针脚 从左到右分别是：

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 Select, DTR 和 Command

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。

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。

|BIT 0|BIT 1|BIT 2|BIT 3|BIT 4|BIT 5|BIT 6|BIT 7|

CLOCK ---- - - - - -

DATA ----00000011111222223333344444555556666677777-----

* * * * *

CMND -----000000111112222233333444445555566666777777-----

ACK -----

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=模拟红灯, 0x53=模拟绿灯)。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”。

本文只针对“模拟手柄红灯模式”进行破解，字节定义如下。

模拟手柄红灯模式

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 RGHT DOWN LEFT
05	idle	data	L2 R2 L1 R1 ^ O 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

例程

X-CTR100 控制器解码 PS2 无线手柄，并通过串口输出显示键值。

硬件说明

硬件资源：

- 串口 UART1
- LC 扩展接口（非串口，作为普通 IO 口使用）
- PS2 无线手柄（需自备）

连接关系如下，

DATA——LC 接口-D6

COMM——LC 接口-D5

GND——LC 接口-GND

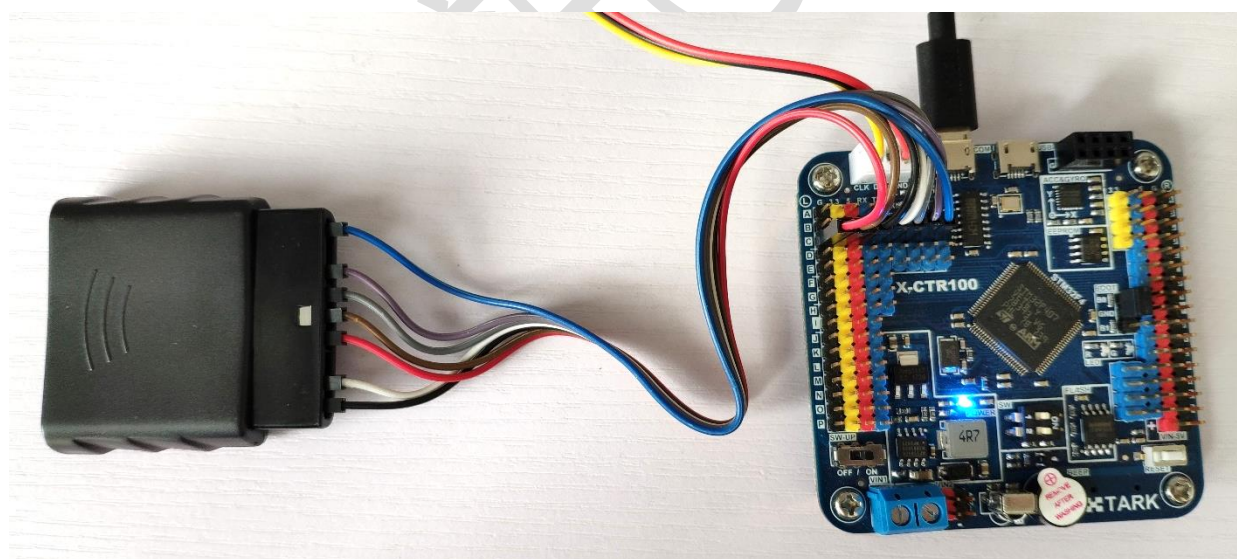
VCC——LC 接口-3.3V

ATT——LC 接口-D7

CLOCK——LC 接口-D4

ACK——LC 接口-D3

连接实物图



软件生态

X-SOFT 软件生态，X-API 扩展文件如下。

ax_ps2.c——PS2 无线手柄操作源文件

ax_ps2.h——PS2 无线手柄头文件

接口函数

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

定义 PS2 键值结构体 JOYSTICK_TypeDef，通过结构体进行键值和摇杆数据传输。

```
//手柄键值数据结构体
typedef struct
{
    uint8_t mode;                /* 手柄的工作模式 */

    uint8_t btn1;                /* Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7
                               SLCT          STRT UP   R    DOWN L  */

    uint8_t btn2;                /* Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7
                               L2   R2   L1   R1   /\   O    X    口  */

    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;             /*右边摇杆 0x00 = 左    0xff = 右    */
    uint8_t LJoy_UD;             /*右边摇杆 0x00 = 上    0xff = 下    */
}JOYSTICK_TypeDef;
```

软件说明

主程序通过键值扫描函数，获取按键状态，并通过串口显示，主程序代码如下。

```
int main(void)
{
    //X-CTR100 初始化
    AX_Init(115200);
    printf("***X-CTR100 PS2 无线手柄例程***\r\n\r\n");

    //功能配置
    AX_PS2_Init(); //PS2 初始化

    while (1)
    {
        AX_PS2_ScanKey(&my_joystick);

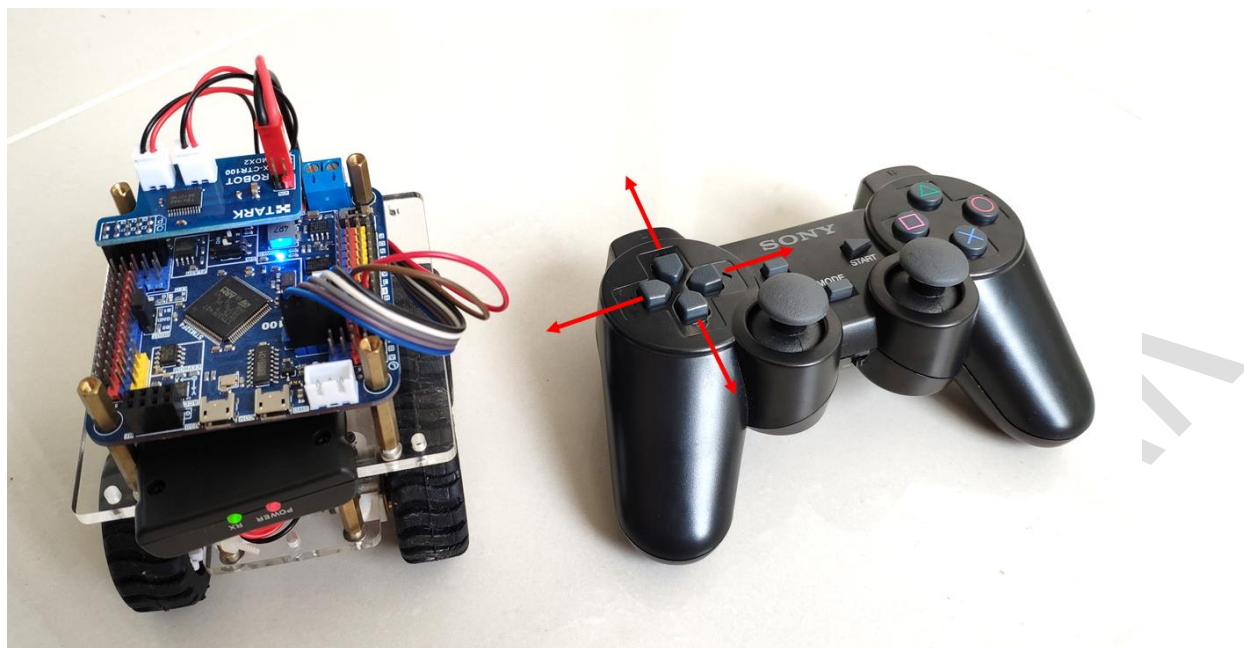
        //打印手柄键值
        printf("MODE:%2x BTN1:%2x BTN2:%2x RJOY_LR:%2x RJOY_UD:%2x LJOY_LR:%2x
LJOY_UD:%2x\r\n",
            my_joystick.mode, my_joystick.btn1, my_joystick.btn2,
            my_joystick.RJoy_LR, my_joystick.RJoy_UD, my_joystick.LJoy_LR,
            my_joystick.LJoy_UD);

        AX_Delays(30);
    }
}
```

实现效果

BTN1、BTN2 为按键值，后面为摇杆数据，摇杆中位为 0x7F 或 0x80。

例程-PS2 无线手柄-4WD 智能小车 (方向模式)



使用 4 个方向按键，实现智能小车前进、后退、原地左转、原地右转四个运动操作。

硬件说明

硬件资源：

- 串口 UART1
- LC 扩展接口 (非串口，作为普通 IO 口使用)
- RO、RP 接口，控制电机
- X-CTR100-MDX2 电机驱动模块或自备其它直流电机驱动模块 (自备)
- PS2 无线手柄 (需自备)

PS2 手柄连接关系如下，

DATA——LC 接口-D6

COMM——LC 接口-D5

GND——LC 接口-GND

VCC——LC 接口-3.3V

ATT——LC 接口-D7

CLOCK——LC 接口-D4

ACK——LC 接口-D3

特别说明，注意左右电机连接顺序和电机正反连接顺序，如果不能按照预期动作，请自行调整，或修改程序代码。

软件说明

主程序代码如下。

```
int main(void)
{
    //X-CTR100 初始化
    AX_Init(115200);
    printf("***X-CTR100 PS2 无线手柄例程***\r\n\r\n");

    //功能配置
    AX_MOTOR_TIM1_MNOP_Init(20); //初始化电机控制端口，PWM 频率为 20KHz
    AX_PS2_Init(); //PS2 初始化
    AX_Delays(10);

    while (1)
    {
        AX_PS2_ScanKey(&my_joystick);

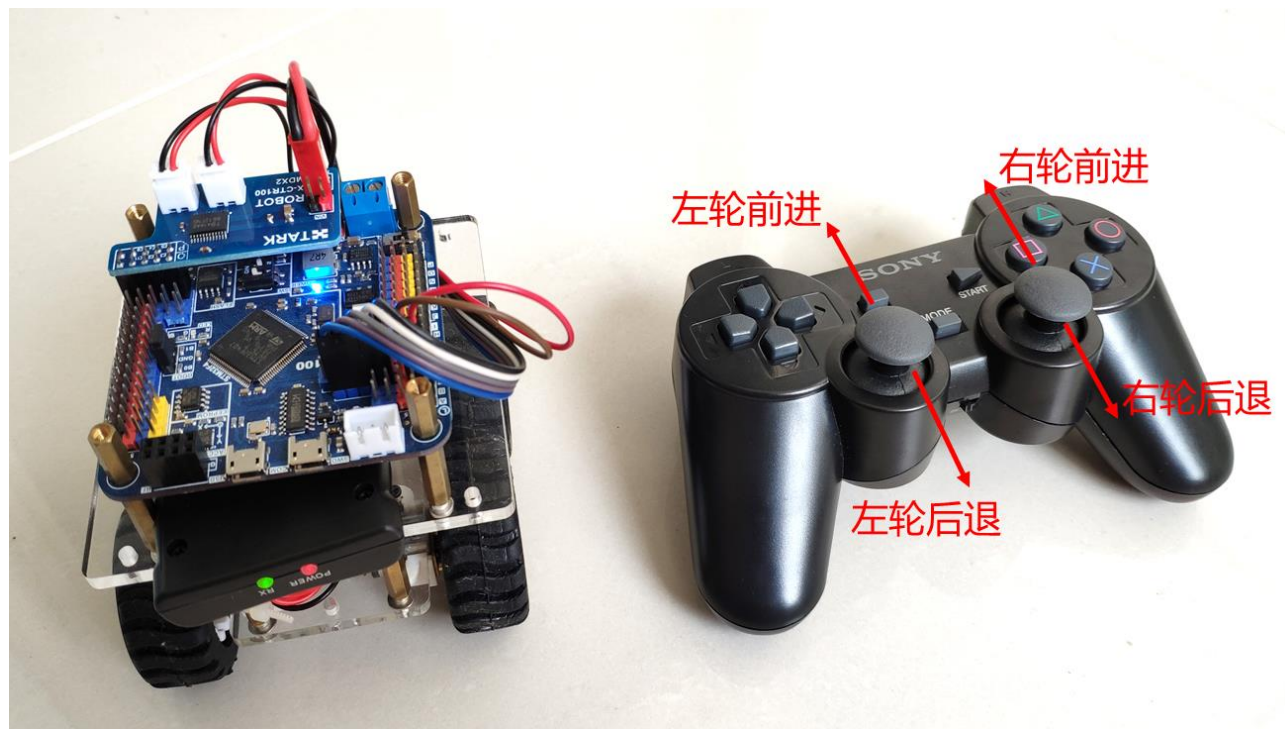
        //LR 按键控制方式(塔克模式)
        if((my_joystick.btn1|0xEF) == 0xEF) //前
        {
            AX_MOTOR_TIM1_O_SetSpeed(60);
            AX_MOTOR_TIM1_P_SetSpeed(60);
        }
        else if((my_joystick.btn1|0xBF) == 0xBF) //后
        {
            AX_MOTOR_TIM1_O_SetSpeed(-60);
            AX_MOTOR_TIM1_P_SetSpeed(-60);
        }
        else if((my_joystick.btn1|0xDF) == 0xDF) //右
        {
            AX_MOTOR_TIM1_O_SetSpeed(-60);
            AX_MOTOR_TIM1_P_SetSpeed(60);
        }
        else if((my_joystick.btn1|0x7F) == 0x7F) //左
        {
            AX_MOTOR_TIM1_O_SetSpeed(60);
            AX_MOTOR_TIM1_P_SetSpeed(-60);
        }
        else //停止
        {
            AX_MOTOR_TIM1_O_SetSpeed(0);
            AX_MOTOR_TIM1_P_SetSpeed(0);
        }

        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);
    }
}
```

转速可根据自己的 4WD 转速调试，支持-200~200 转速，负为倒转，0 为停止。

例程-PS2 无线手柄-4WD 智能小车 (坦克模式)



像坦克车一样使用左右摇杆控制小车，左摇杆控制左轮前进后退，可调速，右摇杆控制右轮。

硬件说明

硬件资源：

- 串口 UART1
- LC 扩展接口 (非串口，作为普通 IO 口使用)
- RO、RP 接口，控制电机
- X-CTR100-MDX2 电机驱动模块或自备其它直流电机驱动模块 (自备)
- PS2 无线手柄 (需自备)

PS2 手柄连接关系如下，

DATA——LC 接口-D6

COMM——LC 接口-D5

GND——LC 接口-GND

VCC——LC 接口-3.3V

ATT——LC 接口-D7

CLOCK——LC 接口-D4

ACK——LC 接口-D3

特别说明，注意左右电机连接顺序和电机正反连接顺序，如果不能按照预期动作，请自行调整，或修改程序代码。

软件说明

主程序代码如下。

```
int main(void)
{
    //X-CTR100 初始化
    AX_Init(115200);
    printf("***X-CTR100 PS2 无线手柄例程***\r\n\r\n");

    //功能配置
    AX_MOTOR_TIM1_MNOP_Init(20); //初始化电机控制端口，PWM 频率为 20KHz
    AX_PS2_Init(); //PS2 初始化
    AX_Delays(1);

    while (1)
    {
        AX_PS2_ScanKey(&my_joystick);

        //双摇杆控制方式(塔克模式)
        //左摇杆控制左轮前进后退，右摇杆控制右轮
        AX_MOTOR_TIM1_O_SetSpeed((0x7F-my_joystick.RJoy_UD)/2);
        AX_MOTOR_TIM1_P_SetSpeed((0x7F-my_joystick.LJoy_UD)/2);

        //打印手柄键值
        printf("MODE:%2x BTN1:%2x BTN2:%2x RJOY_LR:%2x RJOY_UD:%2x LJOY_LR:%2x\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);
    }
}
```