ps2 wireless controller control

1. Introduction

Please read the "Introduction to Motors and Usage" in the four-way motor driver board information first to understand the motor parameters, wiring methods, and power supply voltage you are currently using. To avoid burning the motherboard or motor.

2. Experimental preparation

Wheeled car chassis V1 four-wheel drive version, 4*310 motors, 7.4V lithium battery, PS2 wireless controller, MSPM0 robot expansion board (optional), MSPM0G3507 core board (Yahboom).

The relationship between the 4 motor interfaces and the car is as follows:

M1 -> upper left motor (left front wheel of the car)

M2 -> lower left motor (left rear wheel of the car)

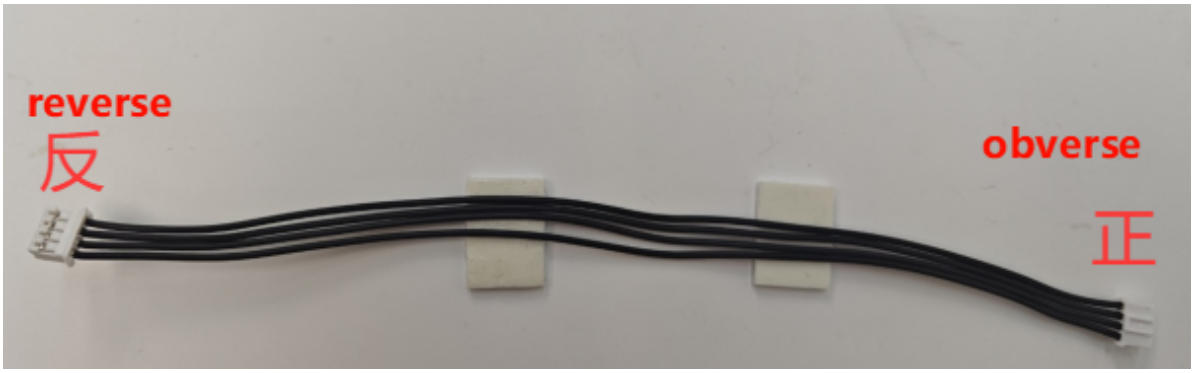
M3 -> upper right motor (right front wheel of the car)

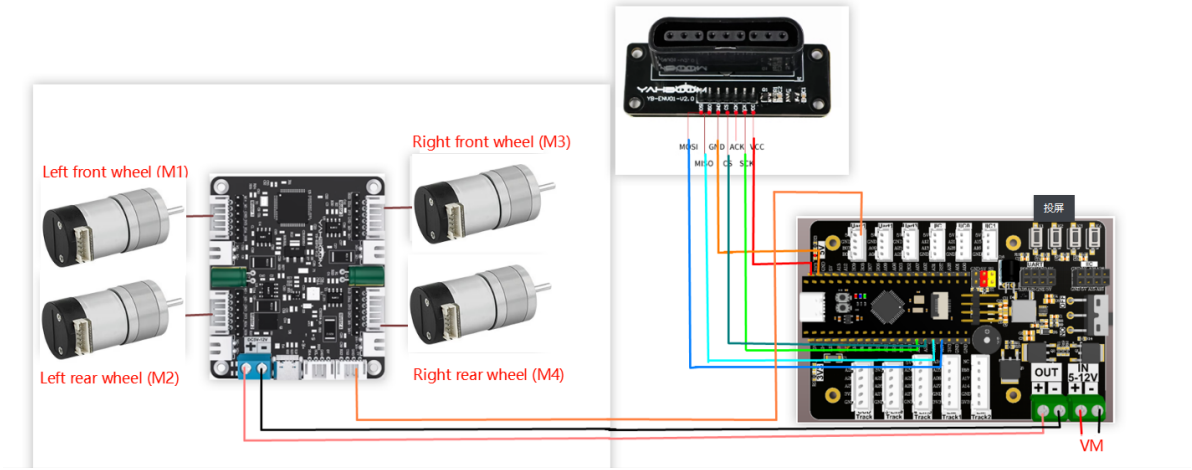
M4 -> lower right motor (right rear wheel of the car)

Hardware wiring:

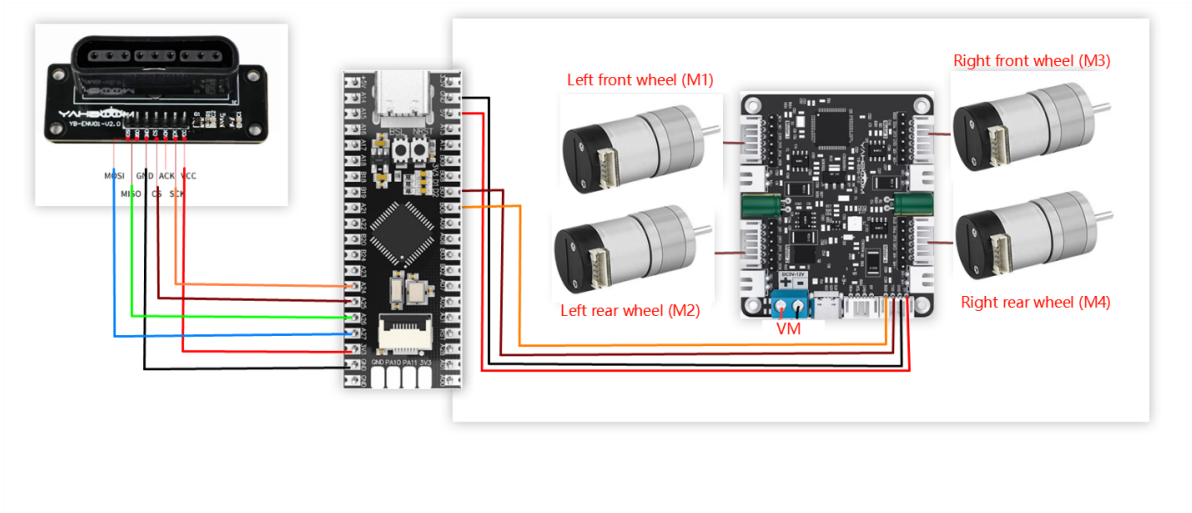
Wiring using MSPM0 robot expansion board

Note: The wire used to connect the MSPM0 robot expansion board to the four-way motor drive module is: XPH2.0-4pin cable, double-ended all black, reverse (200mm), the direction of the reverse cable holder is shown in the figure below





Wiring using MSPM0G3507 core board (Yahboom)



Wiring pins

Four-way motor driver board	MSPM0G3507 core board (Yahboom)
RX2	PB6
TX2	PB7
GND	GND
5V	5V

Take M1 motor as an example below, and other motors are similar

Motor	Four-way motor driver board (Motor)
M2	M1-
VCC	3V3
A	H1A
B	H1B
GND	GND
M1	M1+

PS2 wireless controller	MSPM0G3507 core board (Yahboom)
VCC	3V3
SCK	PA24
ACK	No connection
CS	PA25
GND	GND
MISO	PA26
MOSI	PA27

3. Key code analysis

- pstwo.c

```
//判断是否为红灯模式 Judge whether it is red light mode
//返回值：0，红灯模式    Return value: 0, red light mode
//其他，其他模式       Others, other modes
u8 PS2_RedLight(void)
{
    CS_L;
    PS2_Cmd(Comd[0]);  //开始命令   Start Command
    PS2_Cmd(Comd[1]);  //请求数据   Request data
    CS_H;
    if( Data[1] == 0X73)   return 0 ;
    else return 1;

}
...
void Get_PS2data(void)
{
    PS2_LX=PS2_AnologData(PSS_LX);
    PS2_LY=PS2_AnologData(PSS_LY);
    PS2_RX=PS2_AnologData(PSS_RX);
    PS2_RY=PS2_AnologData(PSS_RY);
    PS2_KEY=PS2_DataKey();

//  printf("PS2_LX=%d    ",PS2_LX);
```

```
//  printf("PS2_LY=%d    ",PS2_LY);
//  printf("PS2_RX=%d    ",PS2_RX);
//  printf("PS2_RY=%d    ",PS2_RY);
//  printf("PS2_KEY=%d    \r\n",PS2_KEY);

}
...

void PS2_CarControl(void)
{
    //----摇杆直向控制---//
    //----Joystick vertical control---//
    if(PS2_LX == 128 && PS2_LY == 127)
    {
        Contrl_Pwm(0,0,0,0);
    }

    if(PS2_LY>=0&&PS2_LY<70)//直走    Go straight
    {
        Contrl_Pwm(1500,1500,1500,1500);
    }

    if(PS2_LY<=255 && PS2_LY>200 &&PS2_LX == 128)//后退   Back
    {
        Contrl_Pwm(-1500,-1500,-1500,-1500);
    }

    if(PS2_LX>=0&&PS2_LX<70)//左旋    Rotate Left
    {
        Contrl_Pwm(-1500,-1500,1500,1500);
    }

    if(PS2_LX<=255 && PS2_LX>200 && PS2_LY == 127)//右旋  Rotate Right
    {
        Contrl_Pwm(1500,1500,-1500,-1500);
    }

    //----摇杆斜向控制---//
    //----Joystick tilt control---//
    if(PS2_LX >= 0&&PS2_LX <= 90 && PS2_LY >= 0&&PS2_LY <= 70)//左前旋转    Left
front rotation
    {
        Contrl_Pwm(0,0,1500,1500);
    }
    if(PS2_LX >= 165&&PS2_LX <= 255 && PS2_LY >= 0&&PS2_LY <= 70)//右前旋转 Right
front rotation
    {
        Contrl_Pwm(1500,1500,0,0);
    }
    if(PS2_LX >= 0&&PS2_LX <= 90 && PS2_LY >= 165&&PS2_LY <= 255)//左后旋转 Left
rear rotation
    {
        Contrl_Pwm(0,0,-1500,-1500);
    }
    if(PS2_LX >= 165&&PS2_LX <= 255 && PS2_LY >= 165&&PS2_LY <= 255)//右后旋转
Right rear rotation
    {
        Contrl_Pwm(-1500,-1500,0,0);
```

```
    }

    //---方向按键控制---//
    //---Direction button control---//
    switch(PS2_KEY){
        case KEY_FORWARD:
            Contrl_Pwm(1500,1500,1500,1500);
        break;

        case KEY_BACK:
            Contrl_Pwm(-1500,-1500,-1500,-1500);
        break;

        case KEY_SPINRIGHT:
            Contrl_Pwm(1500,1500,-1500,-1500);
        break;

        case KEY_SPINLEFT:
            Contrl_Pwm(-1500,-1500,1500,1500);
        break;

        defalut:
        break;
    }
}
```

PS2_RedLight: Determine whether it is in red light mode; return value: 0, red light mode; return value: other, other modes.

Get_PS2data: Get the key value of the handle

PS2_CarControl: The key value of the handle button is bound to the car's motion status

- app_motor_usart.c

```
//控制pwm Control PWM
void Contrl_Pwm(int16_t M1_pwm,int16_t M2_pwm,int16_t M3_pwm,int16_t M4_pwm)
{
    sprintf((char*)send_buff,"$pwm:%d,%d,%d,%d#",M1_pwm,M2_pwm,M3_pwm,M4_pwm);
    Send_Motor_ArrayU8(send_buff, strlen((char*)send_buff));
}
```

Controlling 4 motors via PWM

- empty.c

```
int main(void)
{
    USART_Init();
    while(1)
    {
        Get_PS2data();
        if(!PS2_RedLight())//切换为模拟量输入模式时，开始控制小车 When switching to
analog input mode, start controlling the car
        {
            PS2_CarControl();
        }
        delay_ms(2);
    }
}
```

USART_Init: Initialize the serial port for communication with the four-way motor driver board

Get_PS2data: Get the key value of the handle button

PS2_RedLight: Red light mode judgment

PS2_CarControl: Get the key value of the handle button and then control the car

4. Experimental phenomenon

After connecting the car, burning the program to MSPM0, put the car on the ground and connect the car power supply. At this time, the red light of the receiver is always on and the green light is flashing. It is waiting for pairing. Then turn on the handle switch. At this time, when the red and green lights on the receiver are always on, it means the matching is successful. If the matching is not successful, please check the wiring again. Press and hold the handle mode key to switch the mode. The mode in which the red and green lights on the top of the handle are always on is the analog value mode. At this time, you can press the direction key on the left side of the handle or push the left joystick; the direction key up and down controls the car to move forward and backward, and the direction key left and right controls the car to rotate left and right; the control effect of the left joystick is similar to the direction key.