蓝牙控制

1. 开篇说明

请先阅读四路电机驱动板资料中的《电机介绍以及用法》，了解清楚自己现使用的电机参数、接线方式、供电电压。以免造成烧坏主板或者电机的后果。

2.实验准备

国赛底盘V2四驱版本、4*L520电机、12V锂电池、蓝牙5.0模块（亚博）、四路电机驱动模块、MSPM0机器人扩展板（选配）、MSPM0G3507核心板（亚博）。

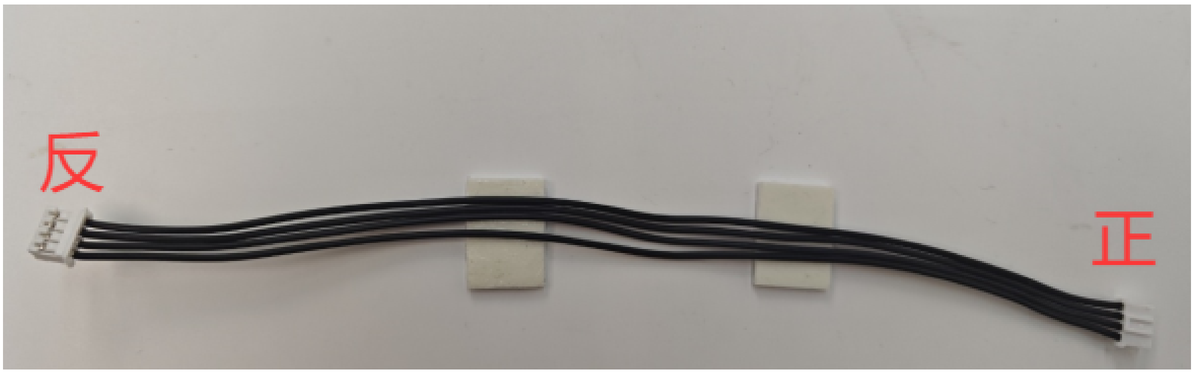
4个电机接口对应小车的关系如下：

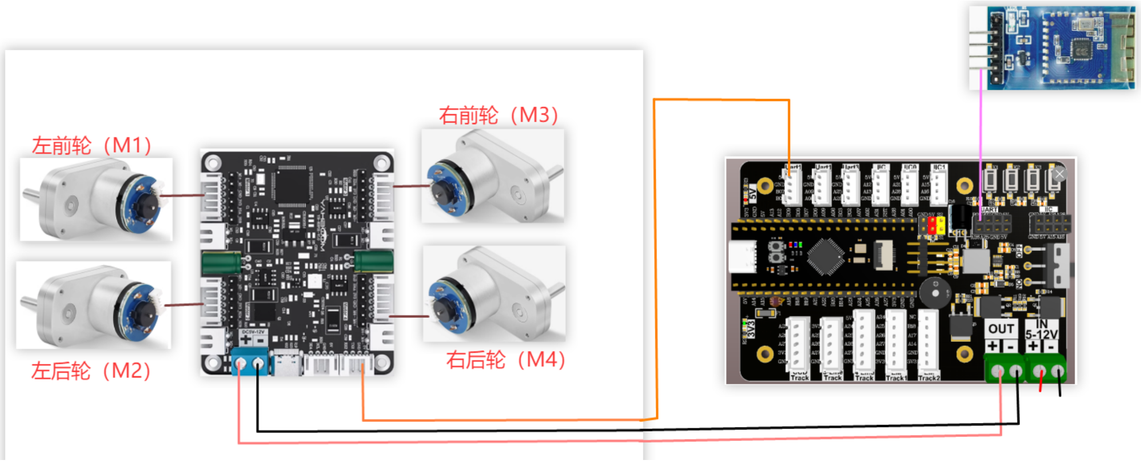
M1 -> 左上电机(小车的左前轮)
M2 -> 左下电机(小车的左后轮)
M3 -> 右上电机(小车的右前轮)
M4 -> 右下电机(小车的右后轮)

硬件接线：

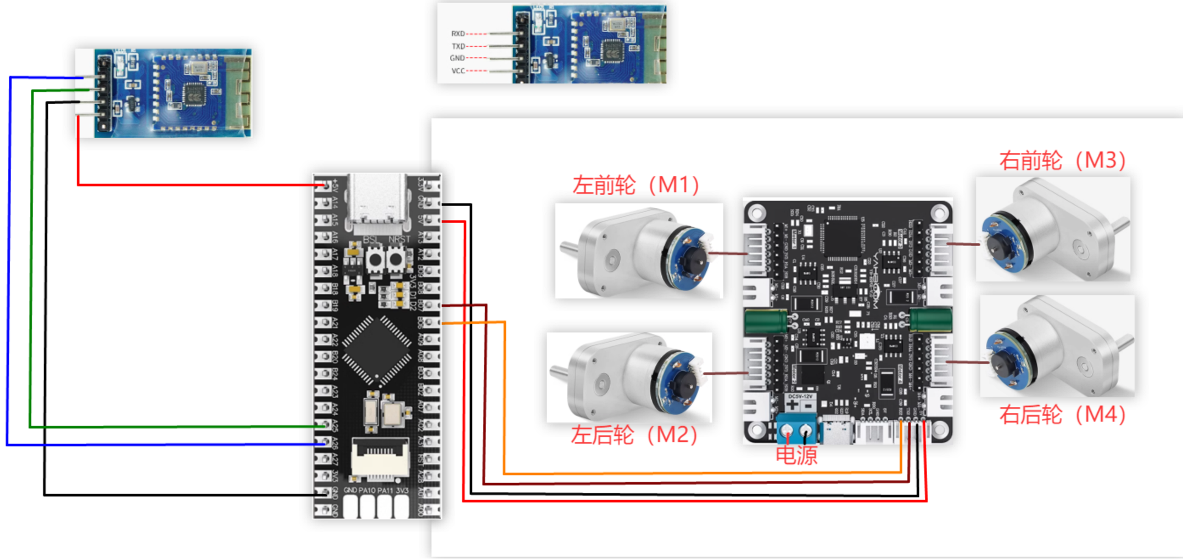
使用MSPM0机器人扩展板接线

注意：MSPM0机器人扩展板与四路电机驱动模块连接所用的线材为：XPH2.0-4pin排线 双头全黑 反向(200mm)，反向排线座子方向如下图所示





使用MSPM0G3507核心板（亚博）接线



接线引脚

四路电机驱动板	MSPM0G3507核心板（亚博）
RX2	PB6
TX2	PB7
GND	GND
5V	5V

下面以M1电机为例，其他电机依此类推

电机	四路电机驱动板(Motor)
M+	M1+
M-	M1-
GND	GND
VCC	3V3
B	H1A
A	H1B

蓝牙5.0模块（亚博）	MSPM0G3507核心板（亚博）
5V	5V
GND	GND
TXD	PA25
RXD	PA26

3.关键代码解析

- usart.c

```
void UART_0_INST_IRQHandler(void)
{
    uint8_t receivedData = 0;
    static uint8_t rec_state = 0;

    //如果产生了串口中断
    //If a serial port interrupt occurs
    switch(DL_UART_getPendingInterrupt(UART_0_INST) )
    {
        case DL_UART_IIDX_RX://如果是接收中断  If it is a receive interrupt

            // 接收发送过来的数据保存  Receive and save the data sent
            receivedData = DL_UART_Main_receiveData(UART_0_INST);
            switch(rec_state)
            {
                case 0:
                    if((receivedData == '$') && (!recv0_flag))
                    {
                        rec_state = 1;
                        recv0_length = 0;
                    }
                    else
                    {
                        rec_state = 0;
                    }
                    break;
                case 1:
                    if(receivedData == '#')
                    {
                        recv0_flag = 1;
                        rec_state = 0;
                    }
                    else
                    {
                        recv0_buff[recv0_length++] = receivedData;
                    }
                    break;
            }
            default://其他的串口中断  Other serial port interrupts
            break;
    }
}
```

UART_0_INST_IRQHandler：处理app发过来的协议数据

- Motor.c

```
void Data_Analyse(void)//解析串口中断串口接收的数据  Parse the data received by the serial port interrupt
{
        if(recv0_flag == 1)
        {
            switch(recv0_buff[0])
            {
                case '1':
                    printf("Forward!\n");
                    Car_state = 1;
                    break;
                case '2':
                    printf("Backward!\n");
                    Car_state = 2;
                    break;
                case '3':
                    printf("Left!\n");
                    Car_state = 3;
                    break;
                case '4':
                    printf("Right!\n");
                    Car_state = 4;
                    break;
                case '0':
                    printf("Stop!\n");
                    Car_state = 0;
                    break;
            }
            switch(recv0_buff[2])
            {
                case '1':
                    printf("SpinLeft!\n");
                    Car_state = 5;
                    break;
                case '2':
                    printf("SpinRight!\n");
                    Car_state = 6;
                    break;
            }
            recv0_flag = 0;
        }
}
void Car_Function(unsigned int Car_state)//控制小车不同状态 Control the car in different states
{
    switch(Car_state)
    {
            case 0:
                    printf("Stop\n");
                    Stop();
                    break;
            case 1:
                    printf("Forward\n");
                    Forward(2300);
                    break;
            case 2:
                    printf("Backward\n");
                    Backward(2300);
                    break;
            case 3:
                    printf("Turnleft\n");
                    Turnleft(2500);
                    break;
            case 4:
                    printf("Turnright\n");
                    Turnright(2500);
                    break;
            case 5:
                    printf("SpinLeft\n");
                    SpinLeft(2500);
                    break;
            case 6:
                    printf("SpinRight\n");
                    SpinRight(2500);
                    break;
    }
}
void Forward(int Speed)
{
    Contrl_Pwm(1000,1000,1000,1000);
}
void Backward(int Speed)
{
    Contrl_Pwm(-1000,-1000,-1000,-1000);
}
void Turnleft(int Speed)
{
    Contrl_Pwm(0,0,1200,1200);
}
void Turnright(int Speed)
{
    Contrl_Pwm(1200,1200,0,0);
}
void Stop(void)
{
    Contrl_Pwm(0,0,0,0);
}
void SpinLeft(int Speed)
{
    Contrl_Pwm(-700,-700,700,700);
}
void SpinRight(int Speed)
{
    Contrl_Pwm(700,700,-700,-700);
}
```

Data_Analyse：将中断接收的数据解析

Car_Function：根据解析的数据控制小车运动。

Contrl_Pwm：通过PWM控制4个电机，从而控制小车的运动状态

- 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));
}
```

通过PWM控制四路电机。

- empty.c

```
int main(void)
{
    USART_Init();
    while (1)
    {
        Data_Analyse();//解析串口中断接收的数据    Parsing data received by serial port interrupt
        Car_Function(Car_state);//控制小车不同状态  Control the car in different states
    }
}
```

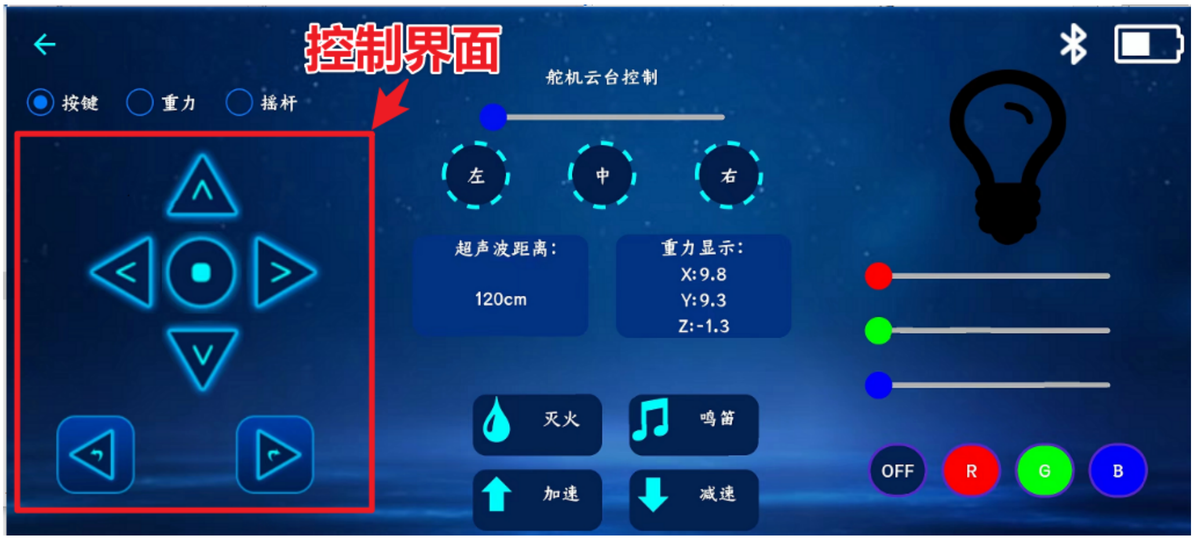
USART_Init: 初始化与四路电机驱动板通信的串口和蓝牙模块的串口

Data_Analyse：解析app发送来的数据

Car_Function：控制小车进行不同的状态的切换

4.实验现象

将小车接好线，给MSPM0烧录程序后，把小车放在地上，插上电源。打开app，选择4WD车型，连接蓝牙后就可以通过app控制小车前进后退、左转右转、左旋右旋。



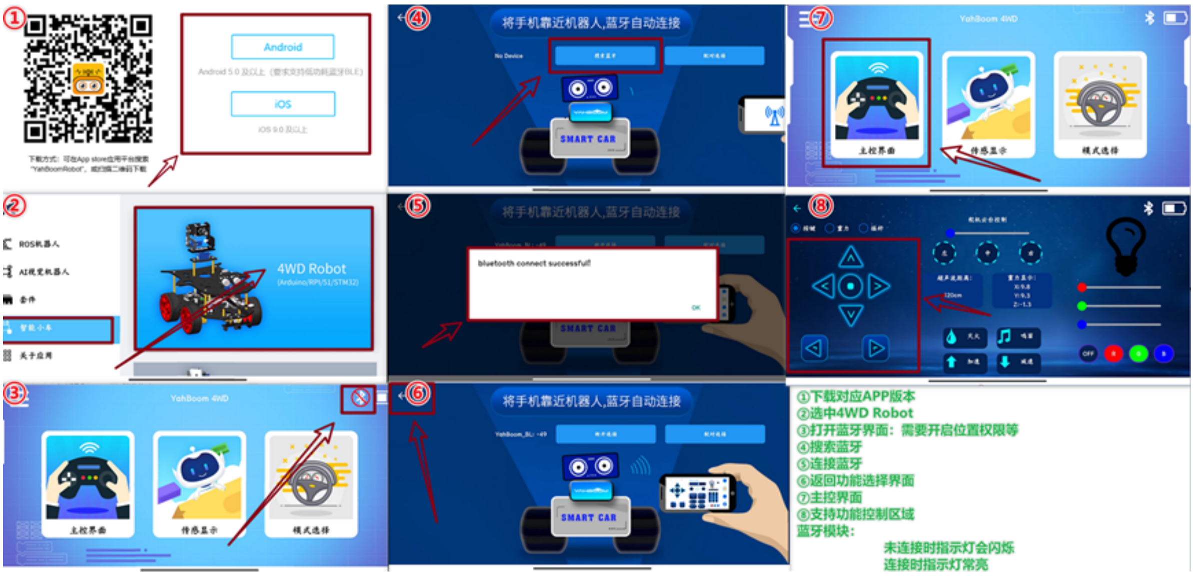
超声波距离：
120cm

重力显示：
X:9.8
Y:9.3
Z:-1.3

附录：

APP下载：https://www.yahboom.com/download_app

YahboomRobot APP具体操作



①下载对应APP版本
②选中4WD Robot
③打开蓝牙界面：需要开启位置权限等
④搜索蓝牙
⑤连接蓝牙
⑥返回功能选择界面
⑦主控界面
⑧支持功能控制区域
蓝牙模块：
未连接时指示灯会闪烁
连接时指示灯常亮