

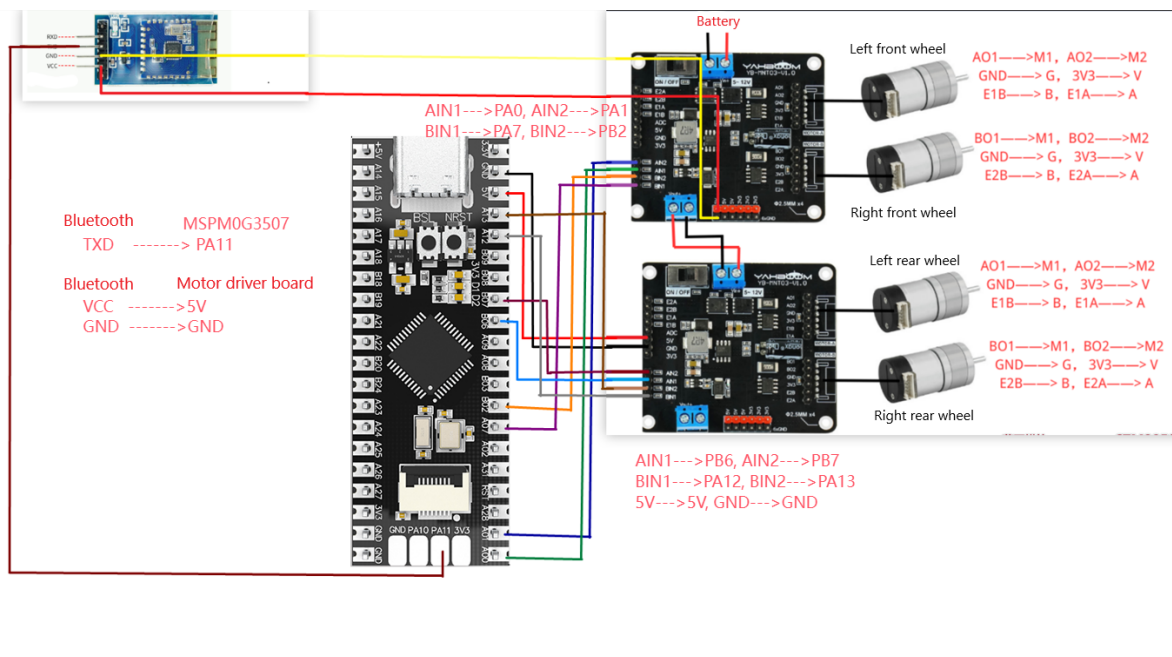
# Bluetooth control

## 1. Learning Objectives

Use the yahboomrobot APP to control the robot's motion status through the Bluetooth 5.0 module

## 2. Hardware Connection

Wiring of Bluetooth 5.0 module, MSPM0G3507 and dual-channel motor driver board. The motor used is 310 motor.



Lower drive plate (left front wheel, right front wheel)	MSPM0G3507		Upper drive plate (left rear wheel, right rear wheel)	MSPM0G3507
AIN1	PA0		AIN1	PB6
AIN2	PA1		AIN2	PB7
BIN1	PA7		BIN1	PA12
BIN2	PB2		BIN2	PA13
			5V	5V
			GND	GND
Bluetooth 5.0	MSPM0G3507		Bluetooth 5.0	Motor driver board
TXD	PA11		VCC	5V
			GND	GND

### 3. Program Description

- usart.c

```
void USART_Init(void)
{
    // SYSCFG初始化
    // SYSCFG initialization
    SYSCFG_DL_init();
    //清除串口中断标志
    //Clear the serial port interrupt flag
    NVIC_ClearPendingIRQ(UART_0_INST_INT_IRQN);
    //使能串口中断
    //Enable serial port interrupt
    NVIC_EnableIRQ(UART_0_INST_INT_IRQN);
}

//串口的中断服务函数
//Serial port interrupt service function
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;
    }
}

```

- USART\_Init: System and serial port initialization function.
- UART\_0\_INST\_IRQHandler: Serial port receiving interrupt function, used to receive data sent from the Bluetooth module.

- motor.h

```

#define PWMA_IN1(value)
DL_TimerG_setCaptureCompareValue(PWM_0_INST,value,GPIO_PWM_0_C0_IDX);
#define PWMA_IN2(value)
DL_TimerG_setCaptureCompareValue(PWM_0_INST,value,GPIO_PWM_0_C1_IDX);
#define PWMB_IN1(value)
DL_TimerG_setCaptureCompareValue(PWM_0_INST,value,GPIO_PWM_0_C2_IDX);
#define PWMB_IN2(value)
DL_TimerG_setCaptureCompareValue(PWM_0_INST,value,GPIO_PWM_0_C3_IDX);

#define PWMC_IN1(value)
DL_TimerG_setCaptureCompareValue(PWM_1_INST,value,GPIO_PWM_1_C0_IDX);
#define PWMC_IN2(value)
DL_TimerG_setCaptureCompareValue(PWM_1_INST,value,GPIO_PWM_1_C1_IDX);
#define PWMD_IN1(value)
DL_TimerG_setCaptureCompareValue(PWM_2_INST,value,GPIO_PWM_2_C0_IDX);
#define PWMD_IN2(value)
DL_TimerG_setCaptureCompareValue(PWM_2_INST,value,GPIO_PWM_2_C1_IDX);

```

Define the PWM duty cycle setting function for the 4 motors.

- 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;
    }
}

```

- Data\_Analyse: Analyze the data received from the serial port and return the control status value of the car according to the communication protocol.
- Car\_Function: Drive the four motors according to the control status value of the car. Use the Forward and Backward functions to move forward and backward; the Turnleft and Turnright functions to turn left and right; the SpinLeft and SpinRight functions to rotate left and right; and the Stop function to stop; the values can be modified to change the speed.

**If you are using the motor driver board and motor mentioned in this tutorial, and the car is running in the wrong direction, you need to check the wiring again. If you are using a motor driver board and motor not mentioned in this tutorial, you need to modify the forward and reverse rotation of each tire according to your own situation.**

- 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
    }
}
```

After the program is running, it will always be in the state of receiving data. After receiving the data, it will process it and control the state of the car.

**Note: The project source code must be placed in the SDK path for compilation,**

**For example, the path:D:\TI\M0\_SDK\mspm0\_sdk\_1\_30\_00\_03\1.TB6612**

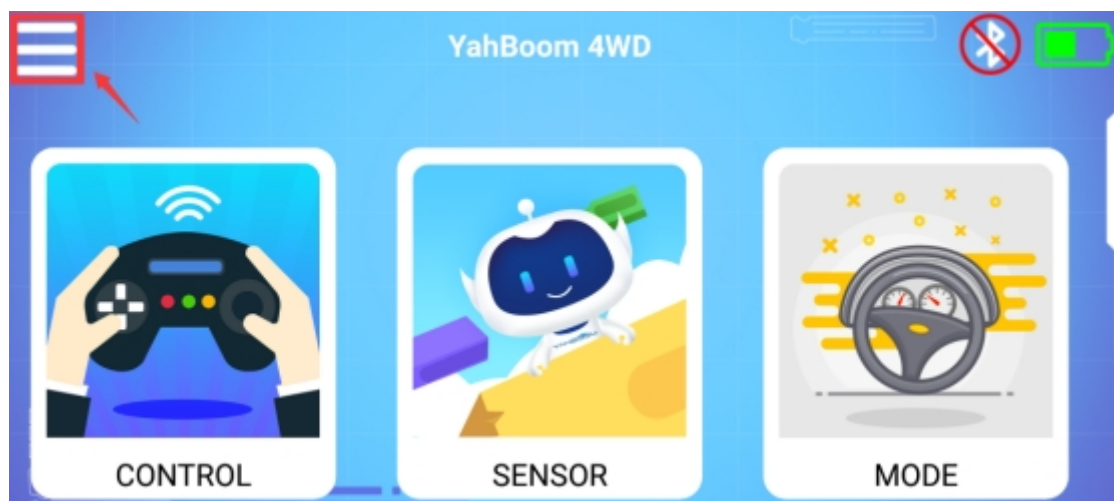
新加卷 (D:) > TI > M0_SDK > mspm0_sdk_1_30_00_03				
名称	修改日期	类型	大小	
1.TB6612	2024/7/22 18:59	文件夹		
2.AT8236	2024/7/22 19:47	文件夹		
3.Encoder	2024/7/23 10:36	文件夹		
4.Servo	2024/7/23 11:13	文件夹		
docs	2024/7/23 10:33	文件夹		
examples	2024/7/23 10:34	文件夹		
kernel	2024/7/23 10:37	文件夹		
source	2024/7/23 10:33	文件夹		
tools	2024/7/23 10:33	文件夹		
imports.mak	2024/1/25 11:45	MAK 文件	2 KB	
known_issues_FAQ.html	2024/1/25 11:42	Microsoft Edge ...	67 KB	
license_mspm0_sdk_1_30_00_03.txt	2024/1/25 11:42	文本文档	33 KB	
manifest_mspm0_sdk_1_30_00_03.html	2024/1/25 11:42	Microsoft Edge ...	113 KB	
mspm0sdk_1_30_00_03.log	2024/7/23 10:42	文本文档	5,237 KB	
release_notes_mspm0_sdk_1_30_00_0...	2024/1/25 11:42	Microsoft Edge ...	108 KB	
uninstall.dat	2024/7/23 10:39	DAT 文件	344 KB	
uninstall.exe	2024/7/23 10:39	应用程序	6,048 KB	

## 4. Experimental Phenomena

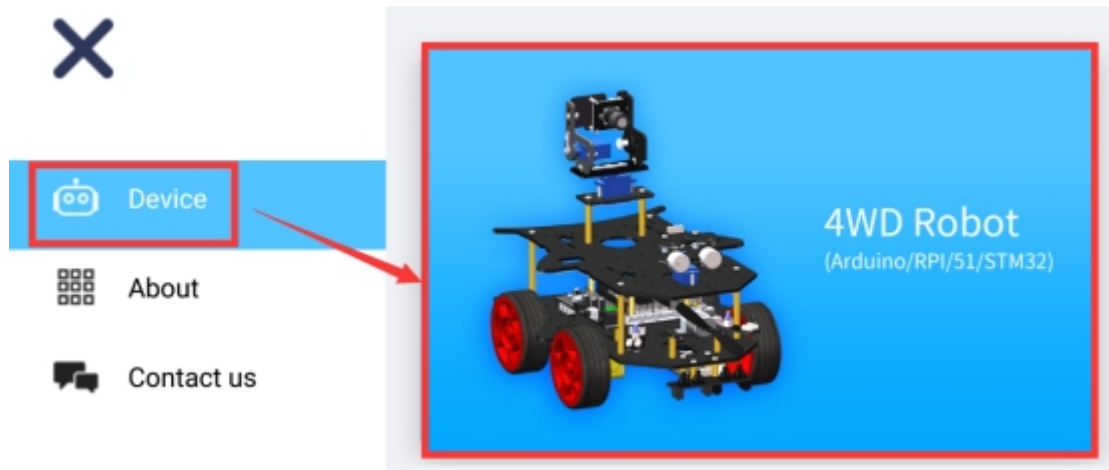
Burn the Bluetooth control program to MSPM0G3507. Connect the wires patiently according to the wiring diagram and check carefully. If the wires are connected incorrectly, the car may not be controlled at best, or the board may be burned. After confirming that everything is correct, turn on the upper and lower drive board switches. After turning on the switches, open the yahboomrobot APP. Select the 4wd model, and after connecting the Bluetooth, you can control the car to move forward and backward, turn left and right, and rotate left and right through the APP.

### APP Instructions:

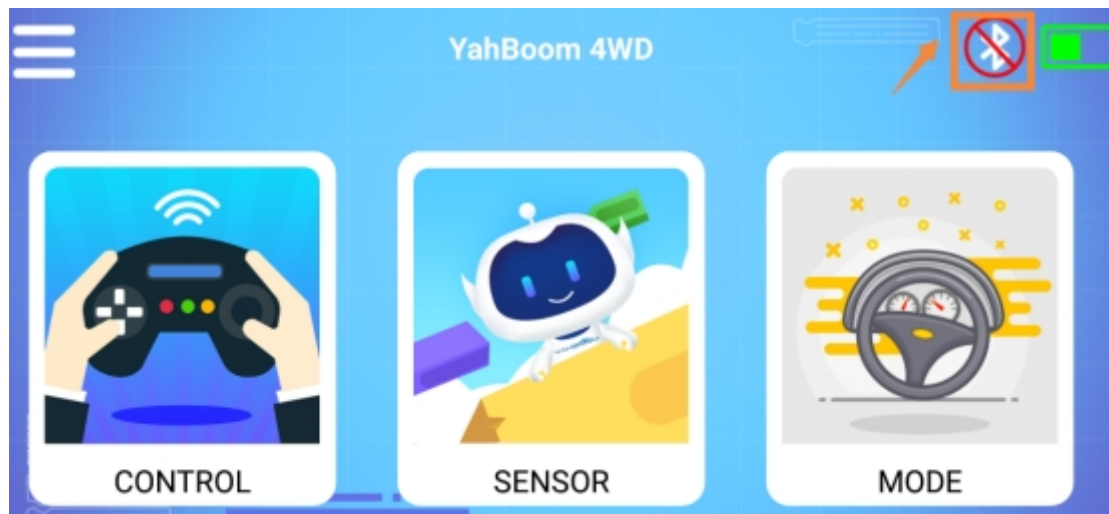
Step 1: Open the **YahboomRobot** APK. You will see the APK interface and we need to click on the top left corner of the APK to select the device as shown below.



Step 2: Select 【4WD Robot】 to enter the remote control interface, as shown below:



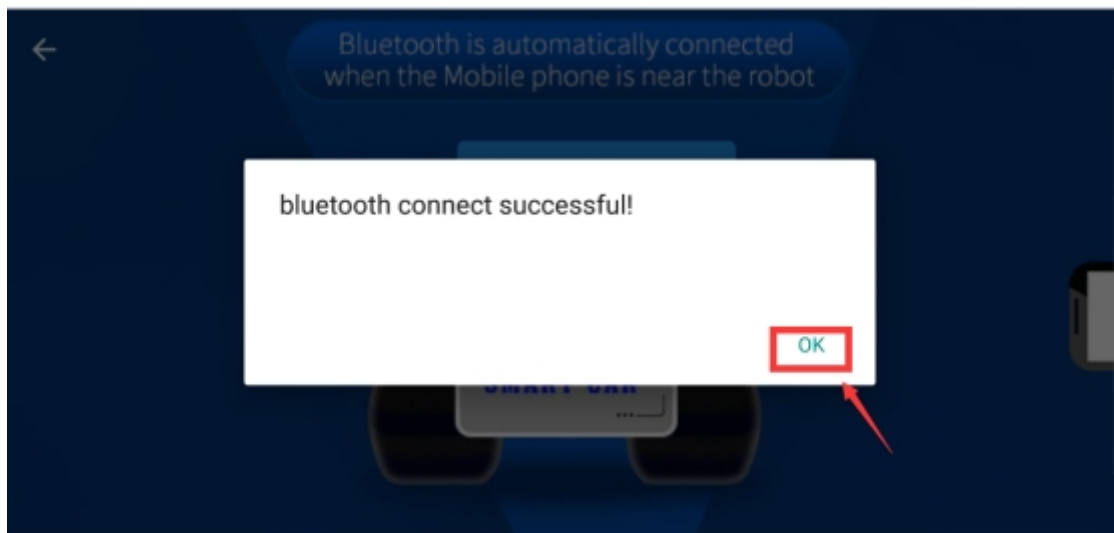
Step 3: You will this interface as shown below. Click on the top right corner of the APK to connect bluetooth.



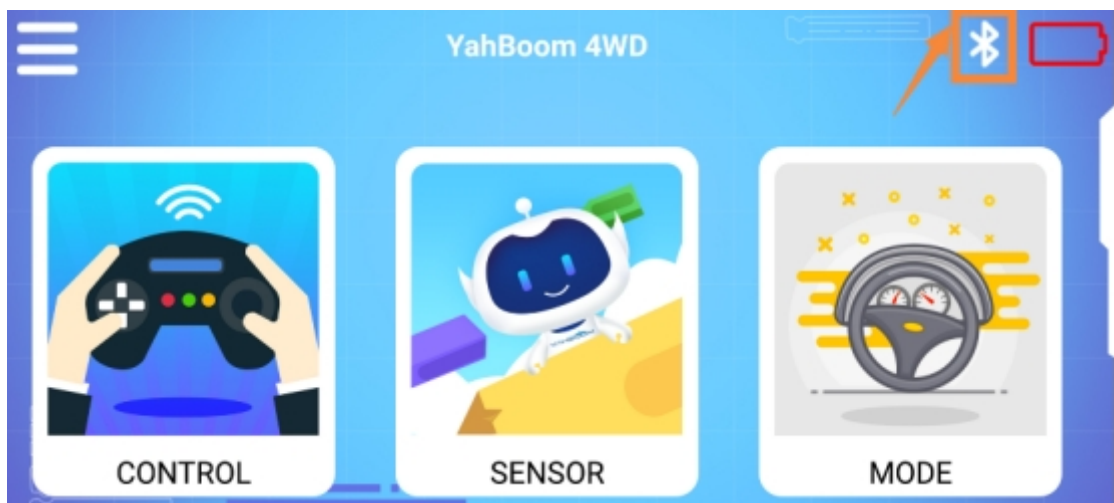
Step 4: You can see bluetooth signal. Wait patiently, the phone will automatically connect to the Bluetooth near the Car.



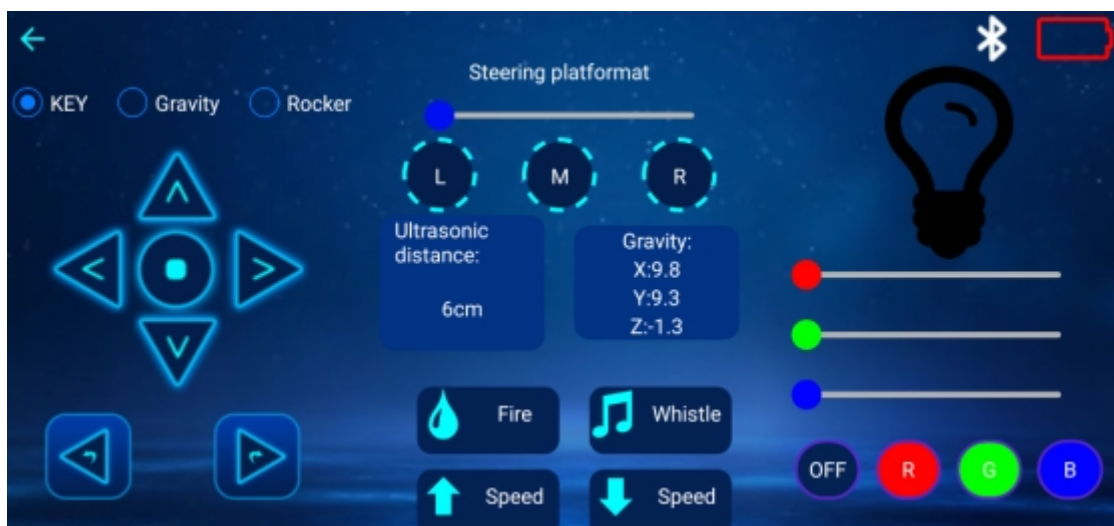
Step 5: Bluetooth can be successfully connected, and the APP will enter the interface as shown below. At the same time, the red indicator of the Bluetooth module will be keep on. You need to click"OK".



You will enter the interface as shown below.



Step 6: Click "CONTROL" to enter interface as shown below. Wait for the ultrasonic data to change, it prove that Bluetooth starts to transmit data normally. You can start to control the car.



## Appendix:

Android users scan the following QR code by browser to download APP;

iOS users scan the following QR code by camera or search "YahboomRobot" in App Store to download APP.



As shown in figure below.



**Note:** During installation, If you find any prompts on your phone (for example: location permissions of your phone). You must select "Yes".

After the APP is installed, open the Bluetooth of the your phone, open the power switch of the Car the red indicator of the Bluetooth module keeps flashing.

For the communication protocol, see the **Bluetooth communication protocol** file in the same folder.