Bluetooth 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, Bluetooth 5.0 module (Yahboom), 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)

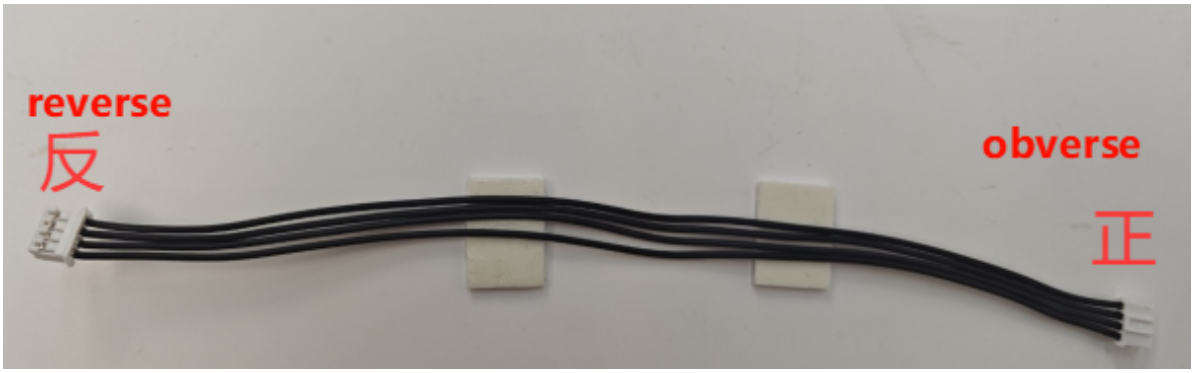
M3 -> upper right motor (right front wheel of the car)

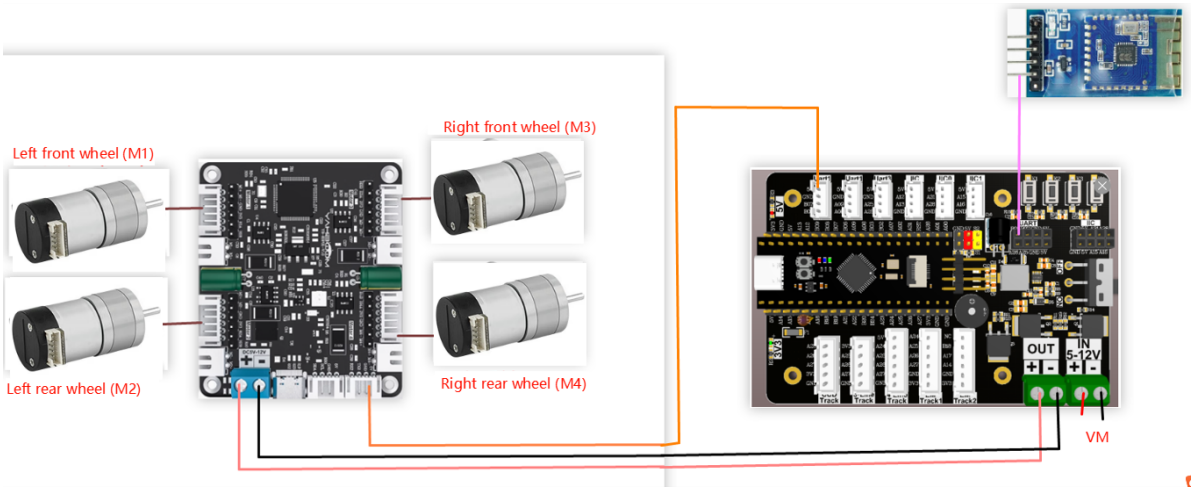
M4 -> lower right motor (right rear wheel of the car)

Hardware wiring:

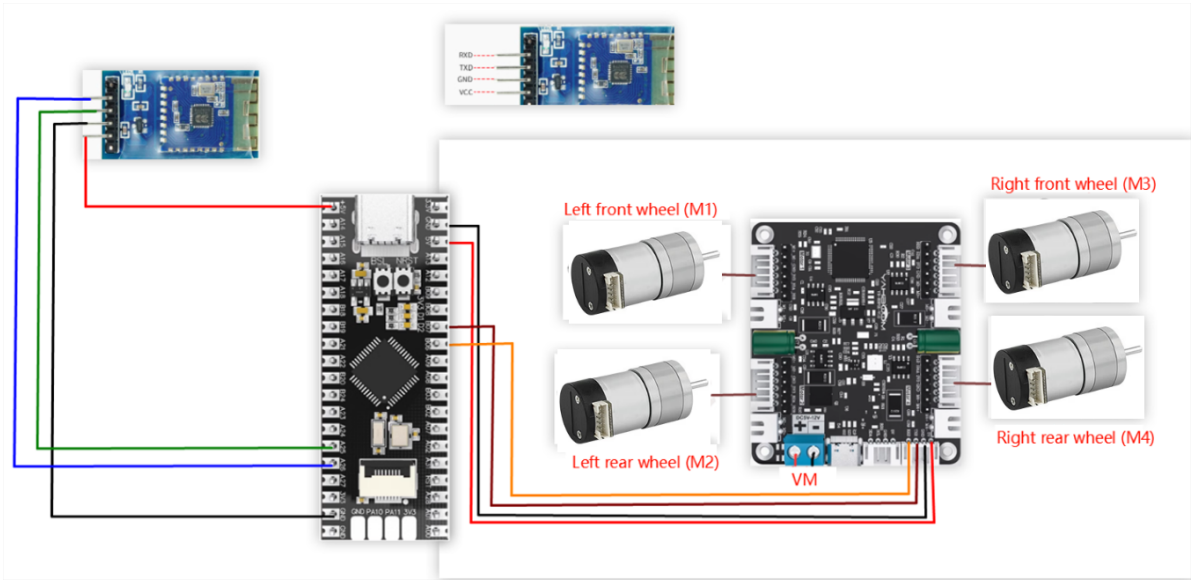
Wiring using MSPM0 robot expansion board

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+

Bluetooth 5.0 module (Yabo)	MSPM0G3507 core board (Yahboom)
5V	5V
GND	GND
TXD	PA25
RXD	PA26

3. Key code analysis

- 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: Processes the protocol data sent by the app

- 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: Analyze the data received by the interruption

Car_Function: Control the movement of the car according to the analyzed data.

Contrl_Pwm: Control the 4 motors through PWM to control the movement state of the car

- 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));
}
```

Four motors are controlled via PWM.

- 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: Initialize the serial port for communicating with the four-way motor driver board and the serial port of the Bluetooth module

Data_Analyse: Analyze the data sent by the app

Car_Function: Control the car to switch between different states

4. Experimental phenomenon

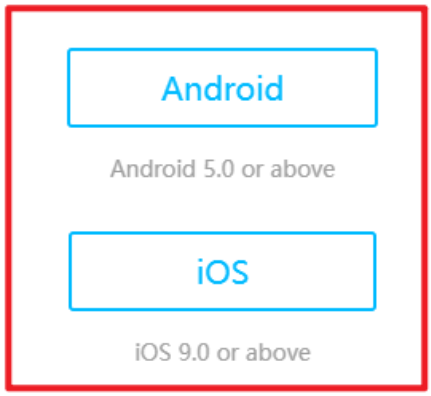
Connect the car, burn the program to MSPM0, put the car on the ground and plug in the power supply. Open the app, select the 4WD model, and after connecting to Bluetooth, you can control the car to move forward and backward, turn left and right, and rotate left and right through the app.

APP download: http://www.yahboom.net/software/app

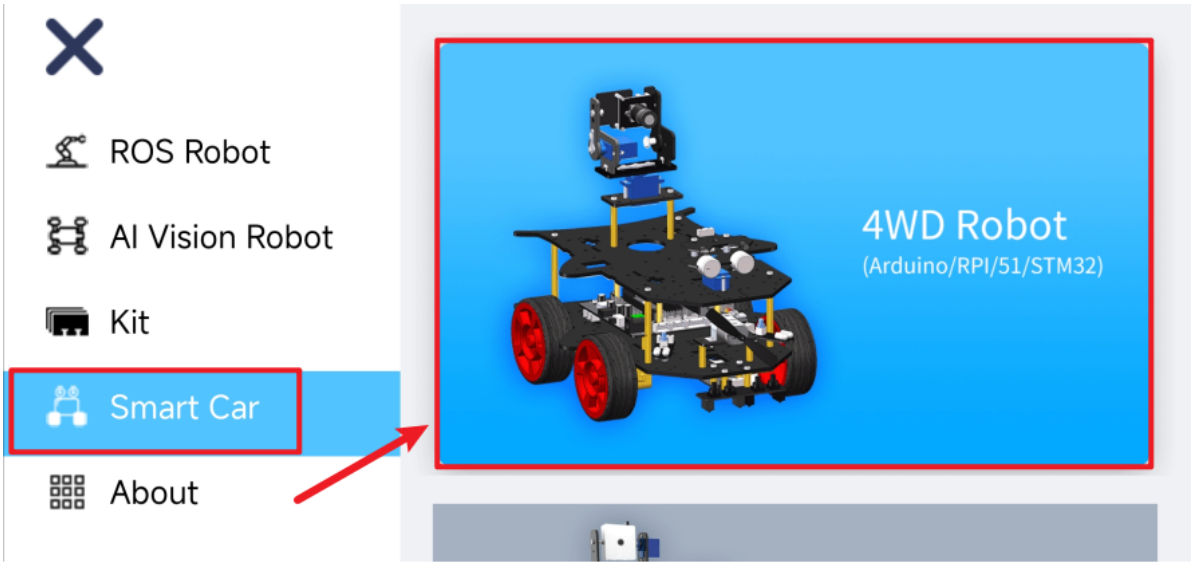
Specific operation steps of YahboomRobot APP

① Download the corresponding App

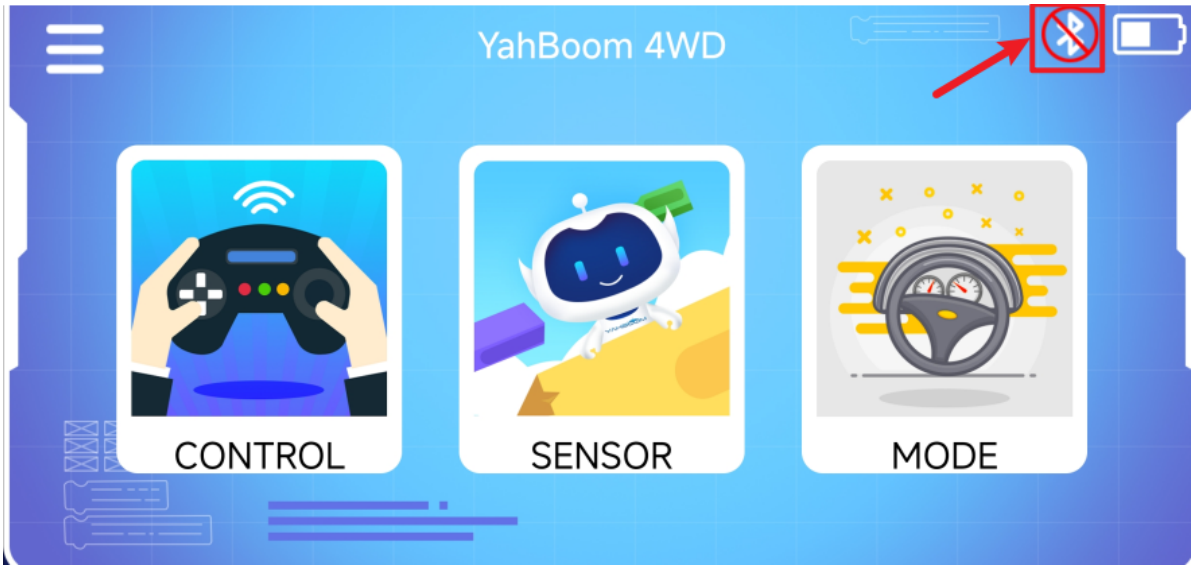
Android users scan the QR code by browser to download APP.
iOS users scan the QR code by browser or camera to download APP. Or search "YahBoomRobot" in App Store to download APP.



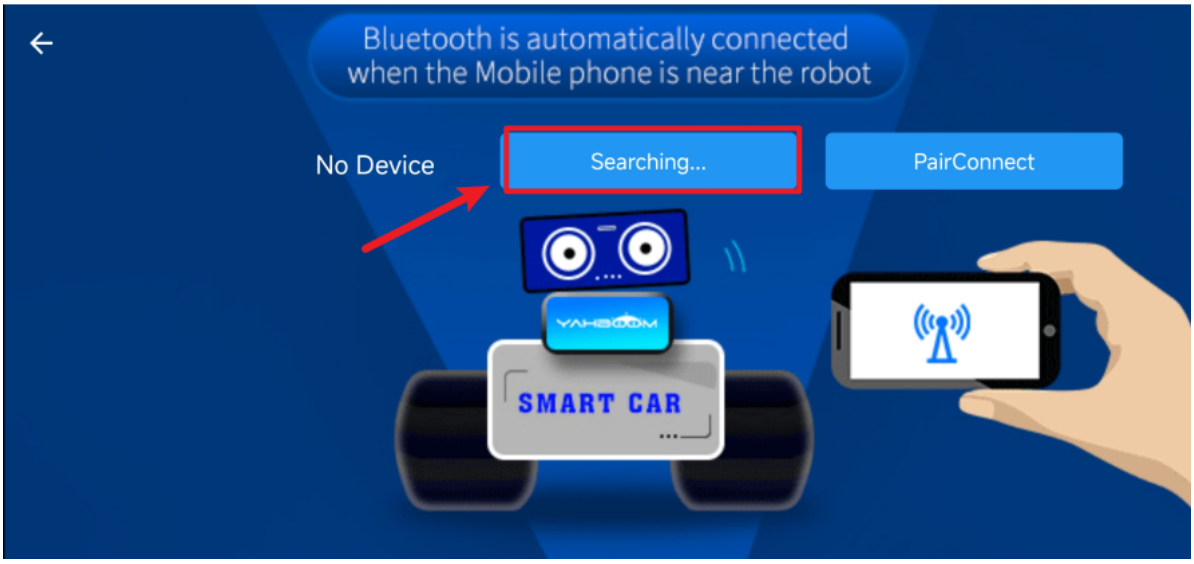
② Open the App and select 4WD Robot



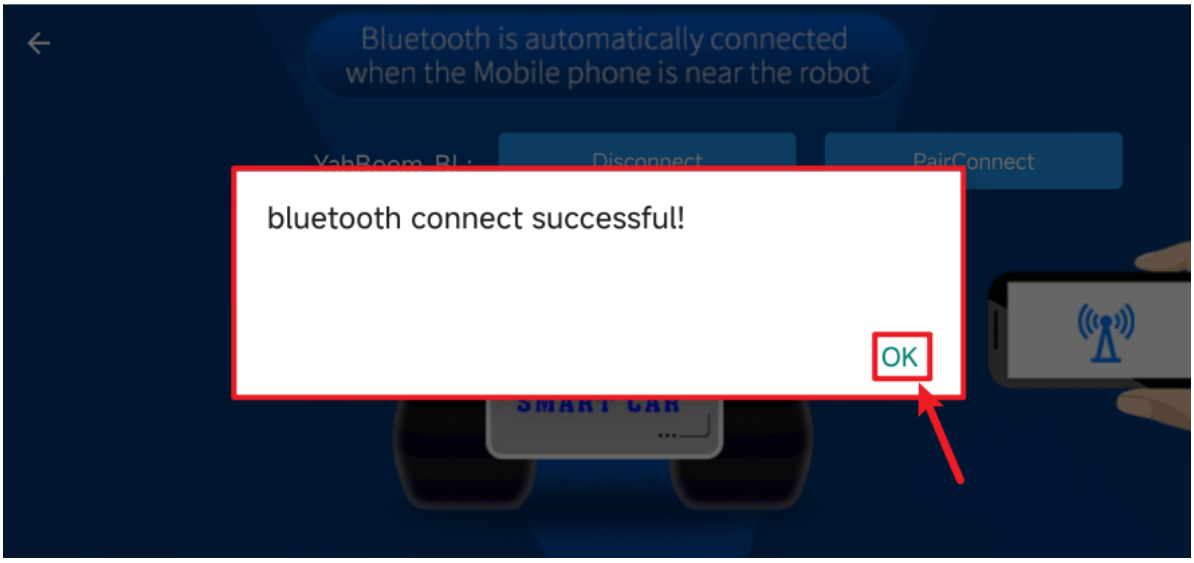
③ Open the Bluetooth interface and enable location permissions, etc.



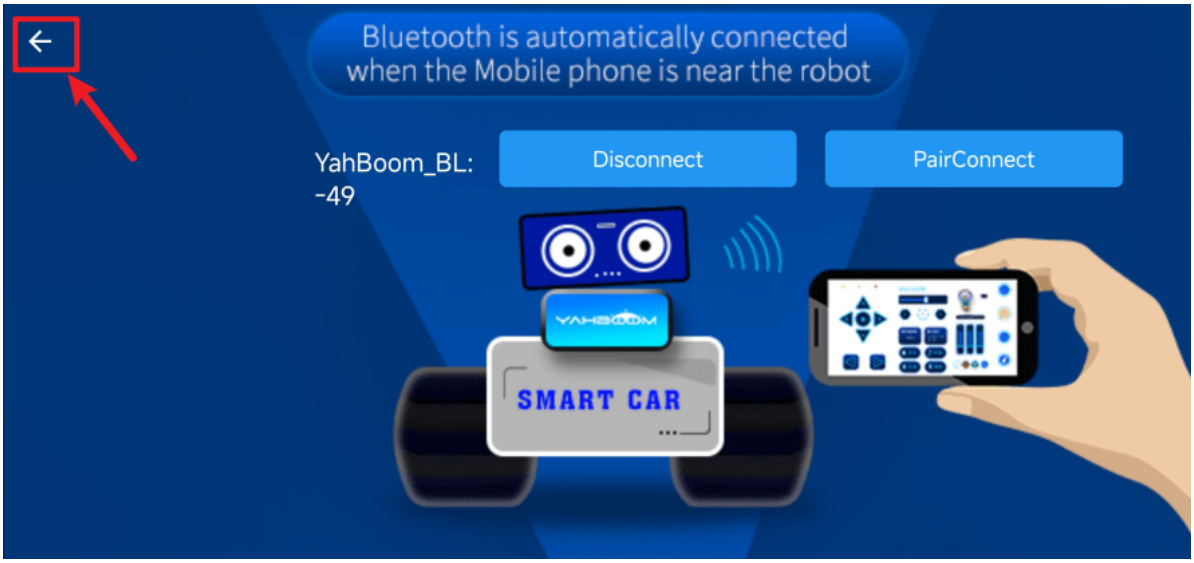
④Search for Bluetooth



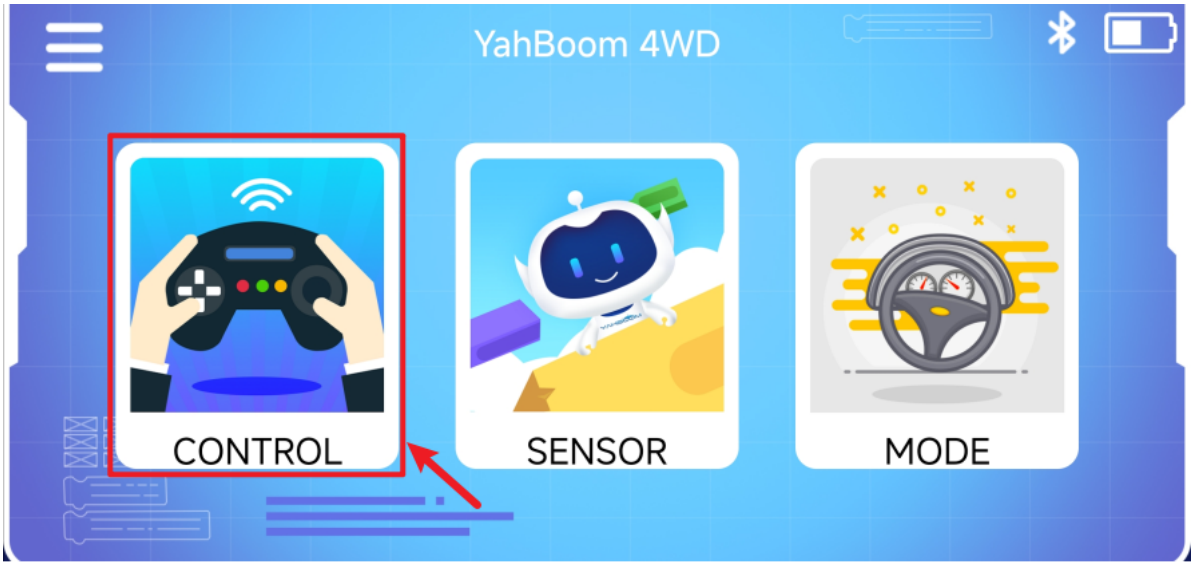
⑤Connect Bluetooth



⑥Return to function selection page



⑦Control interface



⑧Areas supporting function control