

蓝牙遥控

一、实验目的

在电机驱动例程的基础上，使用YahboomRobot APP通过蓝牙5.0模块控制小车运动状态。
可通过APP控制的状态如下：

停止	前进	后退
左转	右转	左旋
右旋		

二、实验原理

蓝牙模块

蓝牙模块作为从机，手机APP作为主机进行数据传输。
蓝牙模块将手机APP传输的数据通过TXD发送给GD32F103C8T6，单片机再将接收到的数据进行解析，获取对应的指令数据，从而控制小车运动状态。
GD32F103C8T6接收的数据格式（可参考亚博智能小程序通讯协议）：

数据头 + 数 据 + 数据尾
\$ + 数据 + #

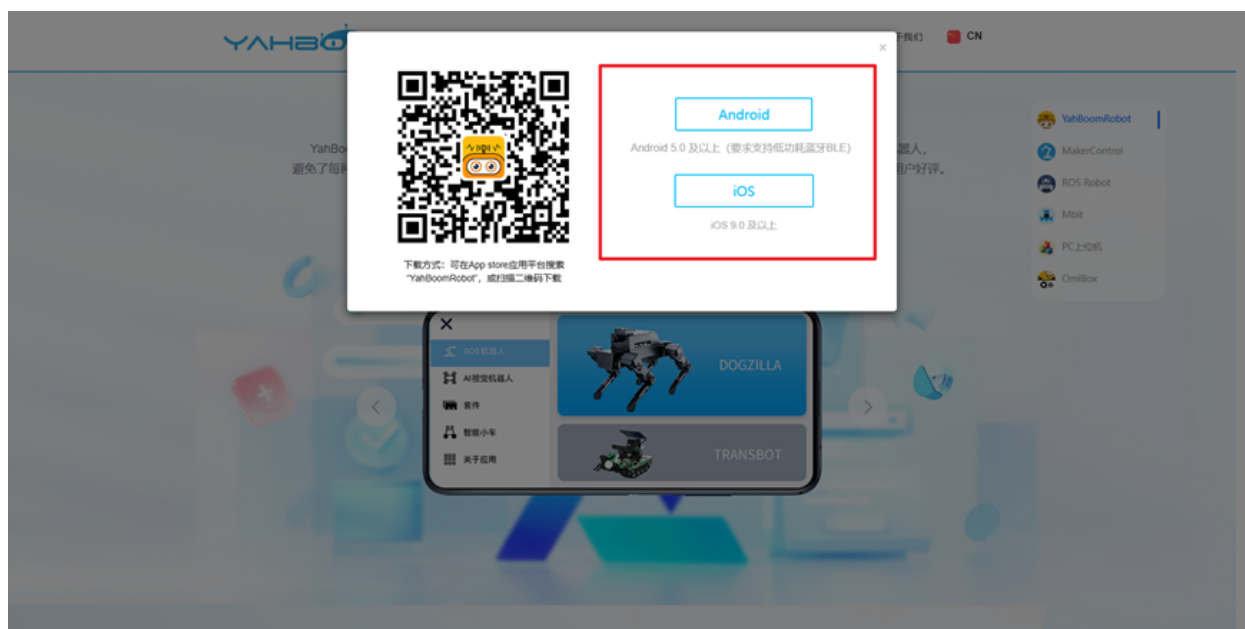
电机控制

采用两个双路电机驱动板驱动四个电机，具体驱动原理可以参考电机驱动例程。

三、硬件连接

- 开发板：**GD32F103C8T6
- 模块：**双路电机驱动板（AT8236）、310电机（4个）、蓝牙5.0模块
- 连接：**杜邦线（若干）
- 电源：**8V左右

硬件接线图



2.YahboomRobot APP具体操作



五、主要代码

1.代码文件

蓝牙控制在电机驱动的基础上实现蓝牙控制，所使用电机驱动例程的文件：

Motor.h、Motor.c、Pwm.h、Pwm.c

Motor.c文件内增加两个函数。

2.实现原理

使用串口中断获取数据，将正确的数据存入数组中。

数组中的第一个元素：控制小车停止、前进、后退、左转、右转

数据中的第二个元素：控制小车左旋、右旋

3.主要代码

1.main.c

```
#include "stm32f10x.h"
#include "SysTick.h"
#include "Uart.h"
#include "Pwm.h"
#include "Motor.h"

int main(void)
{
    SysTick_Configuration();    //配置定时器
    Uart1_Configuration();      //配置串口
    Uart1_NVIC_Configuration(); //配置串口中断
    PWM_Int(7199,0, 7199,0);    //PWM初始化

    while(1)
    {
        Data_Analyse();          //解析串口中断接收的数据
        Car_Function(Car_state); //控制小车不同状态
    }
}
```

2.Uart.c

```
void USART1_IRQHandler(void)
{
    uint8_t recv_dat = 0;
    static uint8_t rec_state = 0;
    while(USART_GetFlagStatus(USART1, USART_FLAG_RXNE) == SET)
    {
        recv_dat = USART_ReceiveData(USART1);

        switch(rec_state)
        {
            case 0:
                if((recv_dat == '$') && (!rx_flag))
                {
                    rec_state = 1;
                    rx_index = 0;
                }
                else
                {
                    rec_state = 0;
                }
                break;
            case 1:
                if(recv_dat == '#')
                {
                    rx_flag = 1;
                }
            }
        }
    }
}
```

```

        rec_state = 0;
    }
    else
    {
        rxd_buf[rxd_index++] = recv_dat;
    }
    break;
}
}
}

```

3.Motor.c

```

void Data_Analyse(void)           //解析串口接收并存储在数组的数据
{
    if(rxd_flag == 1)
    {
        switch(rxd_buf[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(rxd_buf[2])
        {
            case '1':
                printf("SpinLeft!\n");
                Car_state = 5;
                break;
            case '2':
                printf("SpinRight!\n");
                Car_state = 6;
                break;
        }
        rxd_flag = 0;
    }
}

```

```

    }
}

void Car_Function(unsigned int Car_state)    //控制小车状态
{
    switch(Car_state)
    {
        case 0:
            printf("Stop\n");
            Stop();
            break;
        case 1:
            printf("Forward\n");
            Forward(6000);
            break;
        case 2:
            printf("Backward\n");
            Backward(6000);
            break;
        case 3:
            printf("Turnleft\n");
            Turnleft(6500);
            break;
        case 4:
            printf("Turnright\n");
            Turnright(6500);
            break;
        case 5:
            printf("SpinLeft\n");
            SpinLeft(7000);
            break;
        case 6:
            printf("SpinRight\n");
            SpinRight(7000);
            break;
    }
}

```

六、实验现象

使用APP内蓝牙连接控制小车，串口会连续打印小车当前状态信息

