3-Channel Tracking

1. Opening instructions

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

Motor: The case and code take the 310 motor of our store as an example.

2. Experimental preparation

National race chassis V2 four-wheel drive version, 4*310 motor, 7.4V lithium battery, 3-channel patrol module, STM32F103C8T6 core board.

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

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

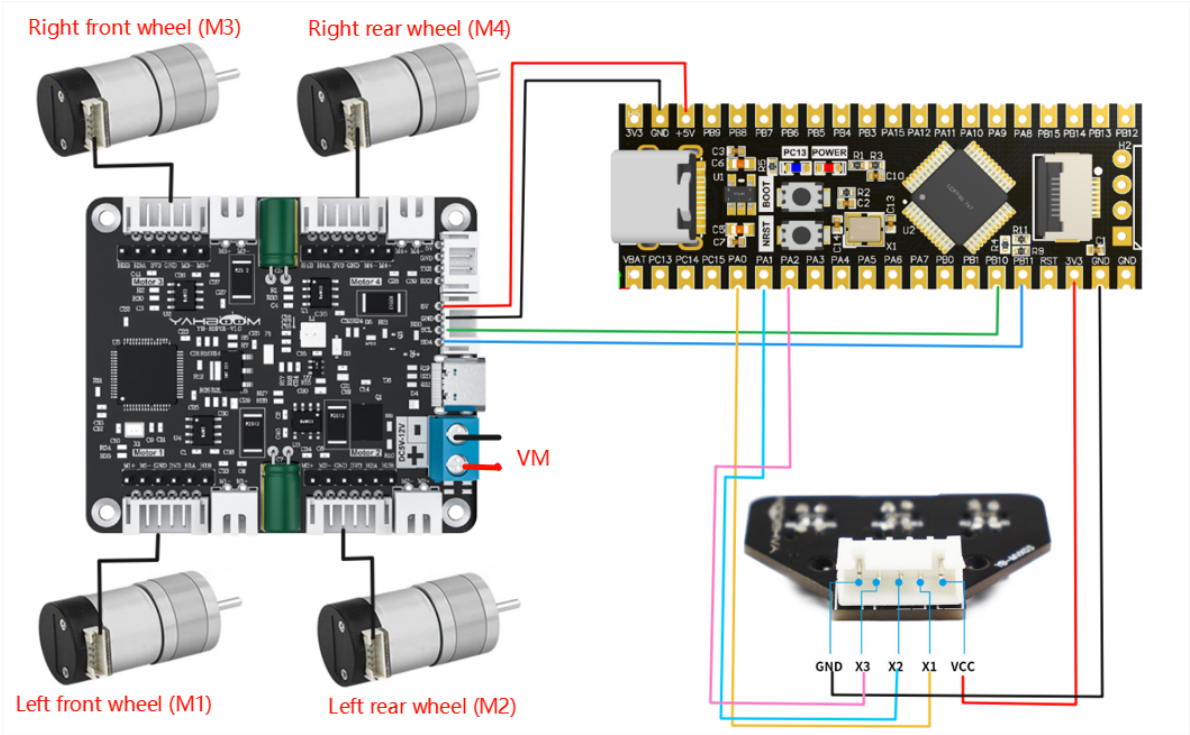
M2 -> lower left motor (left rear wheel of the car)

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

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

Hardware wiring:

Overall wiring



Wiring pins

Four-way motor driver board	STM32C8T6
5V	5V
GND	GND
SCL	PB10
SDA	PB11

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+

3-channel patrol module	STM32C8T6
VCC	3V3
GND	GND
X1	PA0
X2	PA1
X3	PA2

3. Key code analysis

- bsp_motor_iic.c

```
//配置电机  Configure the motor
void Set_motor_type(uint8_t data)
{
    i2cWrite(Motor_model_ADDR,MOTOR_TYPE_REG,2,&data);
}

//配置死区  Configuring Dead Zone
void Set_motor_deadzone(uint16_t data)
{
    static uint8_t buf_tempzone[2];

    buf_tempzone[0] = (data>>8)&0xff;
    buf_tempzone[1] = data;

    i2cWrite(Motor_model_ADDR,MOTOR_DeadZONE_REG,2,buf_tempzone);
}

//配置磁环线 Configuring magnetic loop
void Set_Pluse_line(uint16_t data)
{
    static uint8_t buf_templine[2];

    buf_templine[0] = (data>>8)&0xff;
    buf_templine[1] = data;

    i2cWrite(Motor_model_ADDR,MOTOR_PluseLine_REG,2,buf_templine);
}

//配置减速比 Configure the reduction ratio
void Set_Pluse_Phase(uint16_t data)
{
    static uint8_t buf_tempPhase[2];

    buf_tempPhase[0] = (data>>8)&0xff;
    buf_tempPhase[1] = data;

    i2cWrite(Motor_model_ADDR,MOTOR_PlusePhase_REG,2,buf_tempPhase);
}


//配置直径  Configuration Diameter
```

```
void Set_Wheel_dis(float data)
{
    static uint8_t bytes[4];

    float_to_bytes(data,bytes);

    i2cWrite(Motor_model_ADDR,WHEEL_DIA_REG,4,bytes);
}
//只能控制带编码器类型的电机 Can only control motors with encoders
//传入参数:4个电机的速度      Input parameters: speed of 4 motors
void control_speed(int16_t m1,int16_t m2 ,int16_t m3,int16_t m4)
{
    static uint8_t speed[8];

    speed[0] = (m1>>8)&0xff;
    speed[1] = (m1)&0xff;

    speed[2] = (m2>>8)&0xff;
    speed[3] = (m2)&0xff;

    speed[4] = (m3>>8)&0xff;
    speed[5] = (m3)&0xff;

    speed[6] = (m4>>8)&0xff;
    speed[7] = (m4)&0xff;

    i2cWrite(Motor_model_ADDR,SPEED_Control_REG,8,speed);

}
```

Define the function of writing configuration parameters to the four-way motor driver board and the motor control function, which are used to set key parameters such as motor type, dead zone, number of magnetic ring lines, reduction ratio and wheel diameter, and control the speed of the four motors.

- three_linewalking.c

```
//获取X1X2X3的引脚电平 Get the pin levels of X1X2X3
void three_GetLineWalking(int *p_iL, int *p_iM, int *p_iR)
{
    *p_iL = GPIO_ReadInputDataBit(GPIOA, LineWalk_L_PIN);
    *p_iM = GPIO_ReadInputDataBit(GPIOA, LineWalk_M_PIN);
    *p_iR = GPIO_ReadInputDataBit(GPIOA, LineWalk_R_PIN);

}

void three_LineWalking(void)
{
    int LineL = 0, LineM = 0, LineR = 0;

    three_GetLineWalking(&LineL, &LineM, &LineR);   //获取黑线检测状态  Get black
line detection status

  if( LineL == HIGH &&  LineM == HIGH) //直角锐角   Right angle acute angle
    {
        control_speed(400,400,400,400); //继续直行  Continue straight
```

```
    }
    else if ( LineM == HIGH && LineR == HIGH) //直角锐角    Right angle acute
angle
    {
        control_speed(400,400,400,400); //继续直行  Continue straight

    }
    else if (LineL == HIGH ) //左最外侧检测微调车左转       Left outermost detection
fine-tuning vehicle turns left
    {
        control_speed(-300,-300,500,500); //左急转弯    Sharp left turn
    }
    else if ( LineR == HIGH) //右最外侧检测微调车右转  Right outermost detection
fine-tuning vehicle right turn
    {
        control_speed(500,500,-300,-300); //右急转弯    Sharp right turn
    }
    else if(LineM == HIGH ) //黑色，加速前进  Black, Speed Up
    {
        control_speed(400,400,400,400);
    }

}
```

The `three_GetLineWalking` function gets the level (high or low) of the corresponding pins of the three infrared sensors (X1, X2, X3) through `GPIO_ReadInputDataBit`. It stores the read level in the passed parameters `p_iL`, `p_iM` and `p_iR`, which represent the status of the left, middle and right sensors respectively.

The `three_LineWalking` function determines the movement of the car based on the status of the sensors. If the left and middle sensors detect a black line (`LineL == HIGH && LineM == HIGH`) or the middle and right sensors detect a black line (`LineM == HIGH && LineR == HIGH`), the car continues to go straight; if only the left sensor (`LineL == HIGH`) detects a black line, the car makes a sharp left turn; if only the right sensor (`LineR == HIGH`) detects a black line, the car makes a sharp right turn; if the middle sensor detects a black line (`LineM == HIGH`), it accelerates forward.

- main.c

```
#define MOTOR_TYPE 2   //1:520电机 2:310电机 3:测速码盘TT电机 4:TT直流减速电机 5:L型
520电机
                       //1:520 motor 2:310 motor 3:speed code disc TT motor 4:TT
DC reduction motor 5:L type 520 motor

int main(void)
{
    bsp_init();

    TIM3_Init();
    three_LineWalking_GPIO_Init();
    //电机模块iic通信初始化  Motor module iic communication initialization
    IIC_Motor_Init();

    #if MOTOR_TYPE == 1
    ...
```

```
    #elif MOTOR_TYPE == 2
    Set_motor_type(2);//配置电机类型  Configure motor type
    delay_ms(100);
    Set_Pluse_Phase(20);//配置减速比 查电机手册得出 Configure the reduction ratio.
Check the motor manual to find out
    delay_ms(100);
    Set_Pluse_line(13);//配置磁环线 查电机手册得出  Configure the magnetic ring wire.
Check the motor manual to get the result.
    delay_ms(100);
    Set_Wheel_dis(48.00);//配置轮子直径,测量得出      Configure the wheel diameter
and measure it
    delay_ms(100);
    Set_motor_deadzone(1300);//配置电机死区,实验得出  Configure the motor dead zone,
and the experiment shows
    delay_ms(100);

    ...

    #endif

    while(1)
    {
        three_LineWalking();
    }

}
```

`MOTOR_TYPE`: used to set the type of motor used. Modify the corresponding number according to the comments based on the motor you are currently using.

Initialize the underlying hardware through `bsp_init()`, `TIM3_Init()` initializes timer 3, `three_LineWalking_GPIO_Init()` configures the line patrol sensor pin, `IIC_Motor_Init()` initializes the I2C communication of the motor, and uses `Set_motor_type()` and other functions to set the motor type and parameters according to `MOTOR_TYPE`. In the main loop, call the `three_LineWalking()` function to track the black line, continuously monitor the status of the black line sensor, and control the movement of the motor.

4. Experimental phenomenon

After connecting the car and burning the program to the STM32, put the car on the map with a white background and black lines, let the middle probe of the three-way line patrol module be on the black line, connect the power supply, and the car patrols the black line.

Note: **In order to avoid the influence of sunlight on the infrared line patrol sensor, this experiment must be carried out indoors.**