

Car line patrol based on eight-way line patrol module

Car line patrol based on eight-way line patrol module

1. Introduction

2. Experimental preparation

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

Hardware wiring:

Wiring using MSPM0G3507 core board (Yahboom)

Wiring pins

3. Key code analysis

4. Experimental phenomenon

1. Introduction

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.

2. Experimental preparation

Wheeled car chassis V1 four-wheel drive version, 4*310 motors, 7.4V lithium battery, eight-way line patrol module, 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)

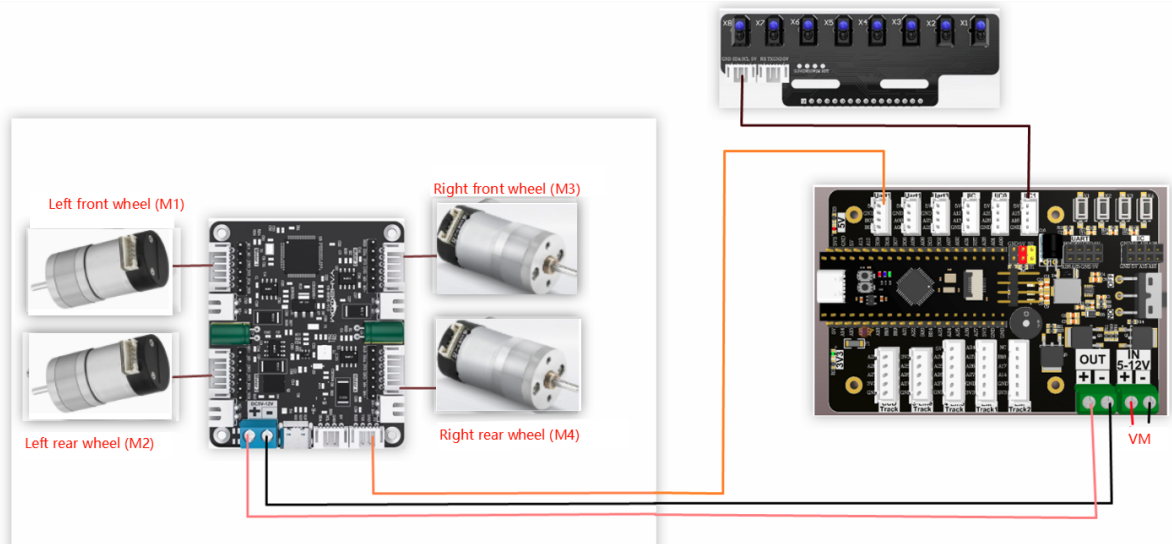
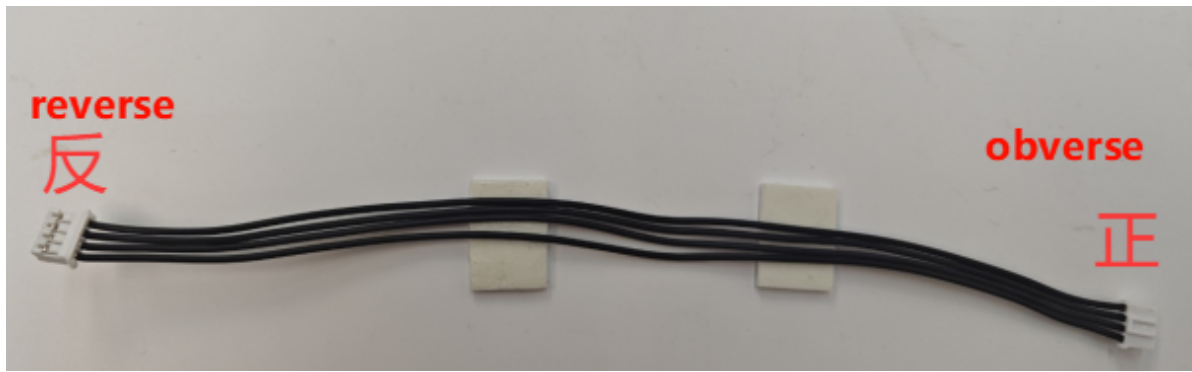
M2 -> lower left motor (left rear wheel of the car)

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

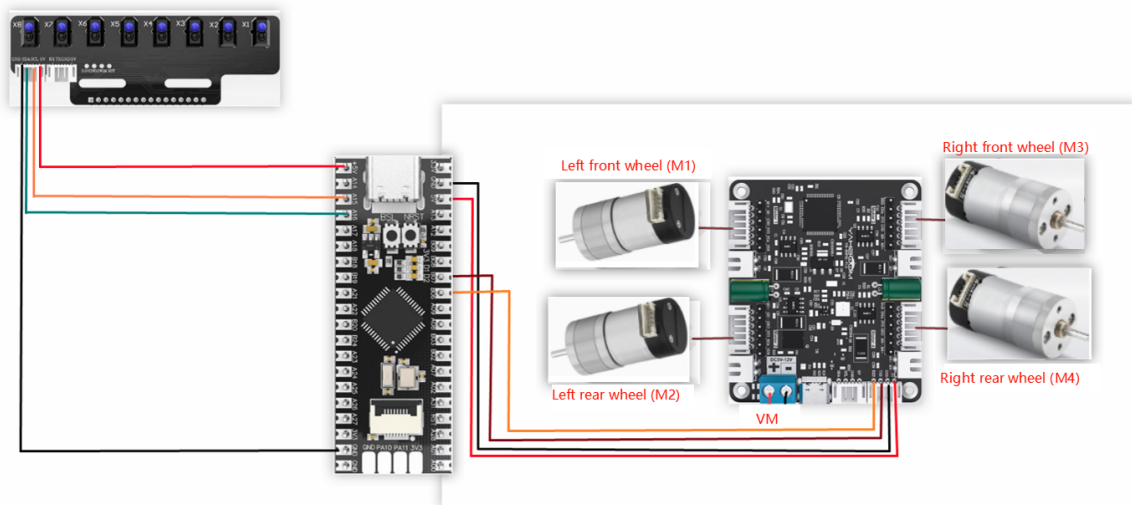
M4 -> lower right motor (right rear wheel of the car)

Hardware wiring:

Note: The wire used for the eight-way patrol module and the wire used to connect the MSPM0 robot expansion board and the four-way motor drive module are: PH2.0-4pin cable, double-ended all black, reverse (200mm), the direction of the reverse cable holder is shown in the following figure



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 the 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+ |

| Eight-way line patrol module | MSPM0G3507 core board (Yahboom) |
|------------------------------|---------------------------------|
| 5V | 5V |
| SCL | PA15 |
| SDA | PA16 |
| GND | GND |

3. Key code analysis

- app_irtracking.c

```
#define IRTrack_Trun_KP (500)
#define IRTrack_Trun_KI (0)
#define IRTrack_Trun_KD (0)

int pid_output_IRR = 0;
u8 trun_flag = 0;

#define IRR_SPEED 300 //巡线速度 Patrol speed

float PID_IR_Calc(int8_t actual_value)
{
    float IRTrackTurn = 0;
    int8_t error;
    static int8_t error_last=0;
    static float IRTrack_Integral;

    error=actual_value;

    IRTrack_Integral +=error;

    //位置式pid Positional pid
    IRTrackTurn=error*IRTrack_Trun_KP
                +IRTrack_Trun_KI*IRTrack_Integral
                +(error - error_last)*IRTrack_Trun_KD;

    return IRTrackTurn;
}
```

```

void deal_IRdata(u8 *x1,u8 *x2,u8 *x3,u8 *x4,u8 *x5,u8 *x6,u8 *x7,u8 *x8)
{
    u8 IRbuf = 0xFF;
    IRbuf = IRI2C_ReadByte(0x30);

    *x1 = (IRbuf>>7)&0x01;
    *x2 = (IRbuf>>6)&0x01;
    *x3 = (IRbuf>>5)&0x01;
    *x4 = (IRbuf>>4)&0x01;
    *x5 = (IRbuf>>3)&0x01;
    *x6 = (IRbuf>>2)&0x01;
    *x7 = (IRbuf>>1)&0x01;
    *x8 = (IRbuf>>0)&0x01;
}

void Linewalking(void)
{
    static int8_t err = 0;
    static u8 x1,x2,x3,x4,x5,x6,x7,x8;

    deal_IRdata(&x1,&x2,&x3,&x4,&x5,&x6,&x7,&x8);

    ///L1为x1, 白底灭灯时为1, 黑线亮时为0      L1 is x1, 1 when the white background
    is off, 0 when the black line is on///

    //优先判断是否到直角或锐角  Prioritize whether to right angles or acute angles
    if(x1 == 0 && x2 == 0 && x3 == 0&& x4 == 0 && x5 == 0 && x6 == 1 && x7 ==
    1 && x8 == 1) // 0000 0111
    {
        err = -15;
        delay_ms(100);
    }
    else if(x1 == 1 && x2 == 1 && x3 == 1&& x4 == 0 && x5 == 0 && x6 == 0 &&
    x7 == 0 && x8 == 0) // 1110 0000
    {
        err = 15;
        delay_ms(100);
    }

    .....

    //剩下的就保持上一个状态      The rest will stay the same.
    pid_output_IRR = (int)(PID_Calc(err));

    Motion_Car_Control(IRR_SPEED, 0, pid_output_IRR);
}

```

- PID_IR_Calc: Position PID calculation, the result is used to control the movement of the car. If the line patrol effect is not good, you can set KP and KD to 0 first, then slowly increase KP, and finally try to increase the value of KD
- deal_IRdata: Get the data of the eight-way line patrol module
- LineWalking: Control the car to adjust different movement states according to the status of the eight-way line patrol module

- app_motor.c

```
void Set_Motor(int MOTOR_TYPE)
{
    if(MOTOR_TYPE == 1)
    {
        send_motor_type(1);//配置电机类型 Configure motor type
        delay_ms(100);
        send_pulse_phase(30);//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);
        send_pulse_line(11);//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);
        send_wheel_diameter(67.00);//配置轮子直径,测量得出 Configure the
wheel diameter and measure it
        delay_ms(100);
        send_motor_deadzone(1900);//配置电机死区,实验得出 Configure the motor dead
zone, and the experiment shows
        delay_ms(100);
    }

    else if(MOTOR_TYPE == 2)
    {
        send_motor_type(2);
        delay_ms(100);
        send_pulse_phase(20);
        delay_ms(100);
        send_pulse_line(13);
        delay_ms(100);
        send_wheel_diameter(48.00);
        delay_ms(100);
        send_motor_deadzone(1600);
        delay_ms(100);
    }

    else if(MOTOR_TYPE == 3)
    {
        send_motor_type(3);
        delay_ms(100);
        send_pulse_phase(45);
        delay_ms(100);
        send_pulse_line(13);
        delay_ms(100);
        send_wheel_diameter(68.00);
        delay_ms(100);
        send_motor_deadzone(1600);
        delay_ms(100);
    }

    else if(MOTOR_TYPE == 4)
    {
        send_motor_type(4);
        delay_ms(100);
        send_pulse_phase(48);
    }
}
```

```

        delay_ms(100);
        send_motor_deadzone(1000);
        delay_ms(100);
    }

    else if(MOTOR_TYPE == 5)
    {
        send_motor_type(1);
        delay_ms(100);
        send_pulse_phase(40);
        delay_ms(100);
        send_pulse_line(11);
        delay_ms(100);
        send_wheel_diameter(67.00);
        delay_ms(100);
        send_motor_deadzone(1900);
        delay_ms(100);
    }
}

void Motion_Car_Control(int16_t V_x, int16_t V_y, int16_t V_z)
{
    float robot_APB = Motion_Get_APB();
    speed_lr = 0;
    speed_fb = V_x;
    speed_spin = (V_z / 1000.0f) * robot_APB;
    if (V_x == 0 && V_y == 0 && V_z == 0)
    {
        Contrl_Speed(0,0,0,0);
        return;
    }

    speed_L1_setup = speed_fb + speed_spin;
    speed_L2_setup = speed_fb + speed_spin;
    speed_R1_setup = speed_fb - speed_spin;
    speed_R2_setup = speed_fb - speed_spin;

    if (speed_L1_setup > 1000) speed_L1_setup = 1000;
    if (speed_L1_setup < -1000) speed_L1_setup = -1000;
    if (speed_L2_setup > 1000) speed_L2_setup = 1000;
    if (speed_L2_setup < -1000) speed_L2_setup = -1000;
    if (speed_R1_setup > 1000) speed_R1_setup = 1000;
    if (speed_R1_setup < -1000) speed_R1_setup = -1000;
    if (speed_R2_setup > 1000) speed_R2_setup = 1000;
    if (speed_R2_setup < -1000) speed_R2_setup = -1000;

    //printf("%d\t,%d\t,%d\t,%d\r\n",speed_L1_setup,speed_L2_setup,speed_R1_setup,s
    peed_R2_setup);

    Contrl_Speed(speed_L1_setup, speed_L2_setup, speed_R1_setup,
    speed_R2_setup);
}

```

- Set_Motor: This function is used to store the parameters of the motors sold in our store. By modifying the MOTOR_TYPE parameter at the beginning of empty.c, one-click configuration can be achieved. Under normal circumstances, do not modify the code here when using the motors of our store. If you use your own motors, you can select any motor and change the parameters to the parameters of the motor you use. If you do not understand the meaning of these parameters, you can check the document "4-way motor driver board control instructions" to understand the specific meaning of each instruction.
- Motion_Car_Control: Control the movement of the car
- app_motor_usart.c

```
//发送电机类型      Transmitter motor type
void send_motor_type(motor_type_t data)
{
    sprintf((char*)send_buff, "$mtype:%d#", data);
    Send_Motor_ArrayU8(send_buff, strlen((char*)send_buff));
}

//发送电机死区      Send motor dead zone
void send_motor_deadzone(uint16_t data)
{
    sprintf((char*)send_buff, "$deadzone:%d#", data);
    Send_Motor_ArrayU8(send_buff, strlen((char*)send_buff));
}

//发送电机磁环脉冲  Send motor magnetic ring pulse
void send_pulse_line(uint16_t data)
{
    sprintf((char*)send_buff, "$mline:%d#", data);
    Send_Motor_ArrayU8(send_buff, strlen((char*)send_buff));
}

//发送电机减速比    Transmitting motor reduction ratio
void send_pulse_phase(uint16_t data)
{
    sprintf((char*)send_buff, "$mphase:%d#", data);
    Send_Motor_ArrayU8(send_buff, strlen((char*)send_buff));
}

//发送轮子直径      Send wheel diameter
void send_wheel_diameter(float data)
{
    sprintf((char*)send_buff, "$wdiameter:%.3f#", data);
    Send_Motor_ArrayU8(send_buff, strlen((char*)send_buff));
}
```

Send the motor parameters to the four-way motor driver board through the serial port

- empty.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)
{
    USART_Init();//打印串口初始化 Print serial port initialization
    printf("please wait...\r\n");

    //使能DMA通道 Enable DMA Channel
    NVIC_ClearPendingIRQ(UART_1_INST_INT_IRQN);
    DL_DMA_enableChannel(DMA, DMA_CH0_CHAN_ID);
    NVIC_EnableIRQ(UART_1_INST_INT_IRQN);

    //设置电机类型 Set motor type
    Set_Motor(MOTOR_TYPE);

    //修改电机PID, 这里的参数是为四驱310底盘配置的, 其他底盘需要自己测试修改
    //Modify the motor PID, the parameters here are configured for the 4WD 310
chassis, other chassis need to test and modify their own!
    send_motor_PID(1.9,0.2,0.8);

    printf("Initialization Succeed\r\n");

    while(1)
    {
        Linewalking();//开始八路巡线 Starting eight-way patrols.
    }
}

```

- MOTOR_TYPE: Change to the motor type you are using
- USART_Init: Initialize the serial port for communicating with the four-way motor driver board
- Set_Motor: Set the motor type
- send_motor_PID: Set the motor PID
- LineWalking: Control the car to patrol the line

4. Experimental phenomenon

Connect the car, burn the program to MSPM0, and put the car on the map with white background and black lines. The map adapted for this project is the high-difficulty map sold in our store, which is adapted to the chassis of the four-wheel drive 310. If the patrol line effect of other chassis is not good, you need to modify the motor PID and patrol line PID in the code. Plug in the power supply. (It is recommended to calibrate the module before patrolling the line), and the car will patrol the black line.