Ultrasonic Servo Gimbal Obstacle Avoidance

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

National Race Chassis V2 Four-wheel Drive Version, 4*L520 Motors, 12V Lithium Battery, Servo Gimbal, Ultrasonic Distance Measurement Module (HC-SR04), 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)

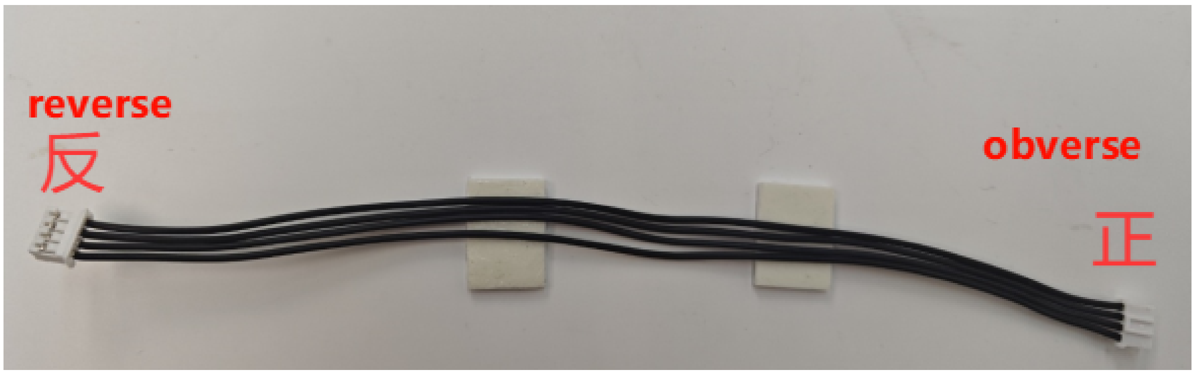
M3 -> upper right motor (right front wheel of the car)

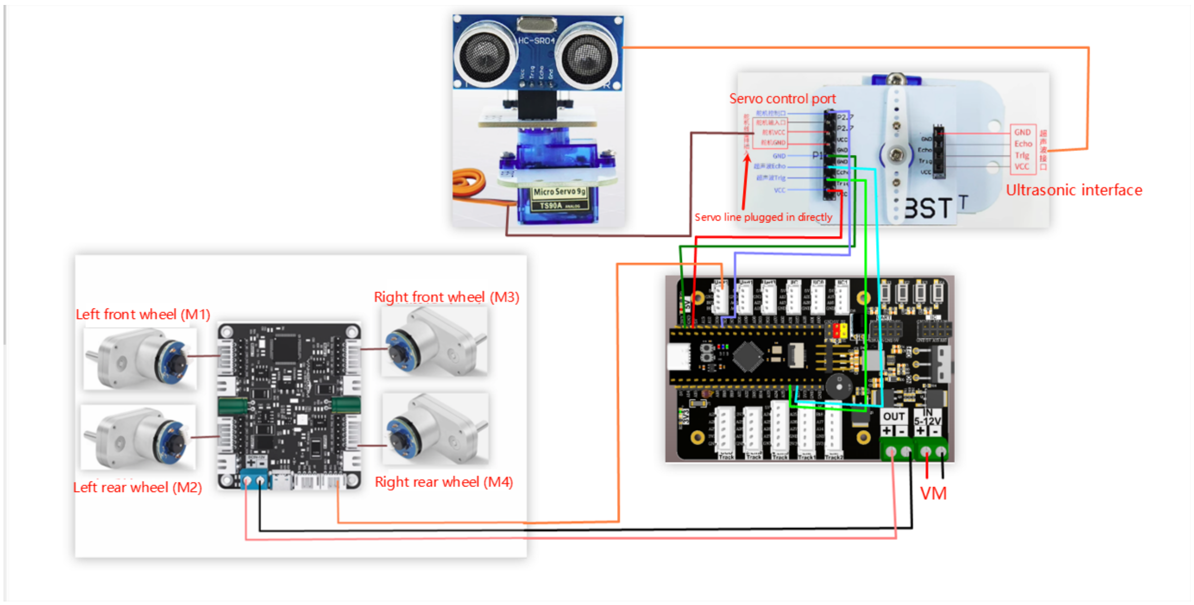
M4 -> lower right motor (right rear wheel of the car)

Hardware wiring:

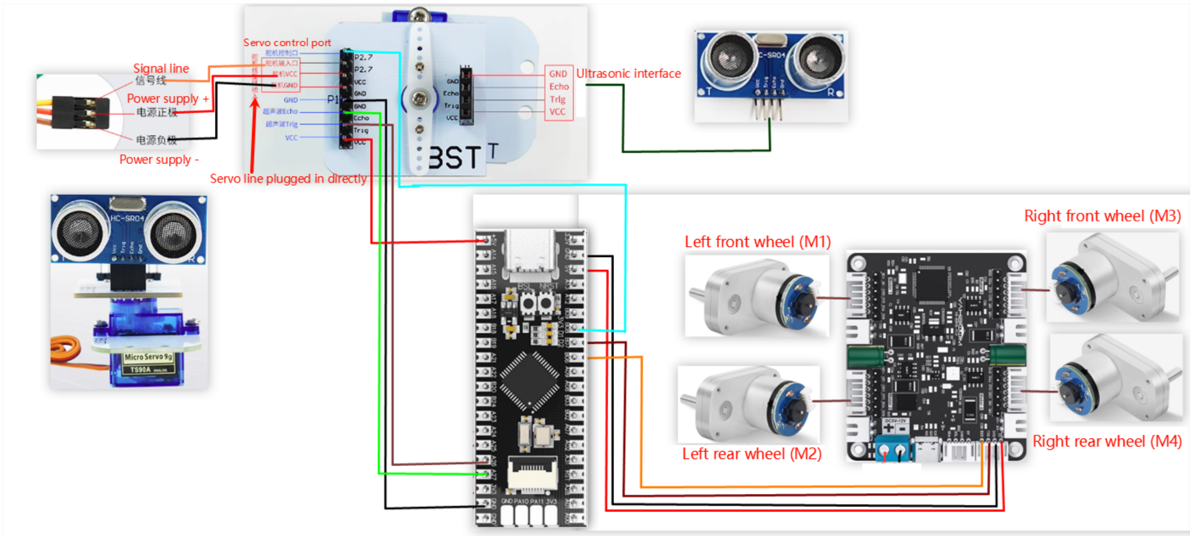
Wiring using MSPM0 robot expansion board

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)
M+	M1+
M-	M1-
GND	GND
VCC	3V3
B	H1A
A	H1B

Ultrasonic ranging module (HC-SR04)	MSPM0G3507 core board (Yahboom)
GND	GND
Echo	PA27
Trig	PA26
VCC	5V

9G Servo	MSPM0G3507 Core Board (Yahboom)
SIG	PB8
VCC	5V
GND	GND

3. Key code analysis

- ultrasonic.c

```
void Ultrasonic_Init(void)
{

    SYSCFG_DL_init();
    //清除定时器中断标志
    //Clear the timer interrupt flag
    NVIC_ClearPendingIRQ(TIMER_0_INST_INT_IRQN);
    //使能定时器中断
    //Enable timer interrupt
    NVIC_EnableIRQ(TIMER_0_INST_INT_IRQN);

}

float Hcsr04GetLength(void)
{
    uint8_t valid_samples = 0;  // 有效测量次数计数器    Valid measurement count
counter
    /*测5次数据计算一次平均值*/
    /*Measure 5 times and calculate the average value*/
    volatile float length = 0;
    t = 0;
    volatile float sum = 0;
    volatile unsigned int  i = 0;

    Close_Timer();

    while(i != 5)
    {
        SR04_TRIG(1);//TRIG引脚拉高信号，发出高电平 TRIG pin pulls up the signal and 
sends a high level

        delay_us(20);//TRIG引脚发出高电平信号10us以上   TRIG pin sends a high level 
signal for more than 10us

        SR04_TRIG(0);//TRIG引脚拉低信号，发出低电平 TRIG pin pulls down the signal 
and sends a low level

        /*Echo发出信号 等待回响信号*/
        /*输入方波后，模块会自动发射8个40KHz的声波，与此同时回波引脚（echo）端的电平会由0变为
1;
        （此时应该启动定时器计时）；当超声波返回被模块接收到时，回波引脚端的电平会由1变为0;
        （此时应该停止定时器计数），定时器记下的这个时间即为超声波由发射到返回的总时长；*/

        /*Echo sends a signal and waits for the echo signal*/
        /*After inputting the square wave, the module will automatically emit 8 
40KHz sound waves. At the same time, the level of the echo pin (echo) will change 
from 0 to 1;
        (the timer should be started at this time); when the ultrasonic wave is 
returned and received by the module, the level of the echo pin will change from 1 
to 0;
        (the timer should be stopped at this time), and the time recorded by the 
timer is the total duration of the ultrasonic wave from emission to return;*/

        while(SR04_ECHO() == 0);//echo等待回响  echo Wait for response

        Open_Timer();   //打开定时器    Turn on the timer

        i++;

        while(SR04_ECHO() > 0);

        Close_Timer();   // 关闭定时器  Turn off the timer

        t = (float)Get_TIMER_Count();   // 获取时间,分辨率为1us     Acquisition 
time, resolution is 1us
        length = (float)t / 58.0f;   // cm
        // 过滤异常值（范围：2cm~400cm）  Filter outliers (range: 2cm~400cm)
        if (length >= 2.0 && distance <= 400.0) {
            sum += length;
            valid_samples++;
        }

        delay_ms(5);// 测量间隔，防止信号干扰  Measurement interval to prevent 
signal interference

    }
    // 计算平均值    Calculate the average
    if (valid_samples > 0) {
        length = sum / valid_samples;  // 取有效样本的平均值 取有效样本的平均值 Take 
the average of the valid samples
    } else {
        length = 0;  // 无有效数据时返回0   Returns 0 if there is no valid data
    }
    return length;
}

void Set_Servo_Angle(unsigned int angle)
{
      if(angle > 180)
      {
            angle = 180; // 限制角度在0到180度之间   Limit the angle to between 0 
and 180 degrees
      }

      Servo_Angle = angle;

      // 计算PWM占空比             Calculate PWM duty cycle
      // 0.5ms对应的计数 = 10    Count corresponding to 0.5ms = 10
      // 2.5ms对应的计数 = 50    Count corresponding to 2.5ms = 50
      float min_count = 10.0f;
      float max_count = 50.0f;
      float range = max_count - min_count;
      float ServoAngle = min_count + (((float)angle / 180.0f) * range);

      DL_TimerG_setCaptureCompareValue(PWM_0_INST, (unsigned int)(ServoAngle + 
0.5f), GPIO_PWM_0_C0_IDX);
}

void Ultrasonic_servo_mode(void)
{
    int Len = 0;
    int LeftDistance = 0, RightDistance = 0;

    Len = (u16)Hcsr04GetLength();

//    printf("Len=%d",Len);
    if(Len <= 15)//当遇到障碍物时  When encountering obstacles
    {
        Contrl_Pwm(0,0,0,0);//停下来做测距  Stop to measure distance
        Set_Servo_Angle(180);       // 左边   Left
        delay_ms(500); //等待舵机到位 Waiting for the servo to arrive
        Len = Hcsr04GetLength();
        LeftDistance = Len;
//        printf("L:%d\r\n", LeftDistance);

        Set_Servo_Angle(0);      // 右边   Right
        delay_ms(500); //等待舵机到位 Waiting for the servo to arrive
        Len = Hcsr04GetLength();
        RightDistance = Len;
//        printf("R:%d\r\n",RightDistance);


        Set_Servo_Angle(90);        //归位    Home
        delay_ms(500); //等待舵机到位 Waiting for the servo to arrive
        if((LeftDistance < 13 ) &&( RightDistance < 13 ))//当左右两侧均有障碍物靠得比
较近  when there are obstacles on both sides,
        {
            Contrl_Speed(-300,-300,300,300);//旋转掉头  Rotate U-turn
            delay_ms(600); //等待舵机到位 Waiting for the servo to arrive
//            printf("Turn Around");
        }
        else if(LeftDistance >= RightDistance)//左边比右边空旷 The left side is
more spacious than the right side
        {
            Contrl_Speed(-200,-200,200,200);//左转    Turn left
            delay_ms(400); //等待舵机到位 Waiting for the servo to arrive
//            printf("Left");
        }
        else//右边比左边空旷   The right side is more spacious than the left side
        {
            Contrl_Speed(200,200,-200,-200); //右转   Turn right
            delay_ms(400); //等待舵机到位 Waiting for the servo to arrive
//            printf("Right");
        }
    }

    else if(Len > 15)//当遇到障碍物时  When encountering obstacles
    {
        Contrl_Speed(100,100,100,100);   //无障碍物，直行     No obstacles, go
straight
    }
}
```

Ultrasonic_Init: Initialize the ultrasonic module

Hcsr04GetLength: Get the data of the ultrasonic module

Set_Servo_Angle: Control the servo

Ultrasonic_servo_mode: Detect obstacles through the data obtained by ultrasound. When encountering obstacles, control the servo to rotate and measure the distance to the left and right sides respectively. The car will choose to turn to the direction with the larger distance.

- 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));
}
```

```c
//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));
}

//控制速度  Controlling Speed
void Contrl_Speed(int16_t M1_speed,int16_t M2_speed,int16_t M3_speed,int16_t 
M4_speed)
{
    
sprintf((char*)send_buff,"$spd:%d,%d,%d,%d#",M1_speed,M2_speed,M3_speed,M4_speed
);
    Send_Motor_ArrayU8(send_buff, strlen((char*)send_buff));
}
```

Configure the motor parameters of the 4-way motor driver board

Contrl_Speed: Control the speed of the 4-way motor

- empty.c

```c
#define MOTOR_TYPE 5      //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)
{
    //开发板初始化  Development board initialization
    USART_Init();
    Ultrasonic_Init();
    ...
    #elif 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);
    #endif
    ...
    Set_Servo_Angle(90);
    delay_ms(1000);
    printf("start");
    while(1)
    {
        Ultrasonic_servo_mode();   //舵机超声波避障模式  Servo ultrasonic obstacle 
avoidance mode
    }
}
```

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.

USART_Init: Initialize the serial port for communicating with the 4-way motor driver board

Ultrasonic_Init: Initialize the ultrasonic module

Set_Servo_Angle: Set the servo to 90 degrees

Ultrasonic_servo_mode: Run the ultrasonic servo obstacle avoidance mode

4. Experimental phenomenon

After connecting the car and burning the program to MSPM0, put the car on the ground, confirm that the servo pan/tilt is facing forward at 90 degrees, connect the power supply, and the car will automatically drive forward. When encountering an obstacle in front, the car's servo will turn left, then right, and measure the distance respectively. If the distance on the left is greater than the distance on the right, the car will turn left, otherwise it will turn right; thus avoiding the obstacle in front. (Note: Ultrasonic waves can only measure the distance perpendicular to the ultrasonic wave (directly in front))