CCD tracking based on Guosai chassis

CCD tracking based on Guosai chassis

1. Opening instructions
2. Experimental preparation
   - The relationship between the 4 motor interfaces and the car is as follows:
   - Hardware wiring:
     - Wiring using MSPM0 robot expansion board
     - Wiring using MSPM0G3507 core board (Yahboom)
     - Wiring pins
3. Key code analysis
4. Experimental phenomenon

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.

2. Experimental preparation

Guosai chassis V2 four-wheel drive version, 4*L520 motors, 12V lithium battery, linear CCD camera 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)

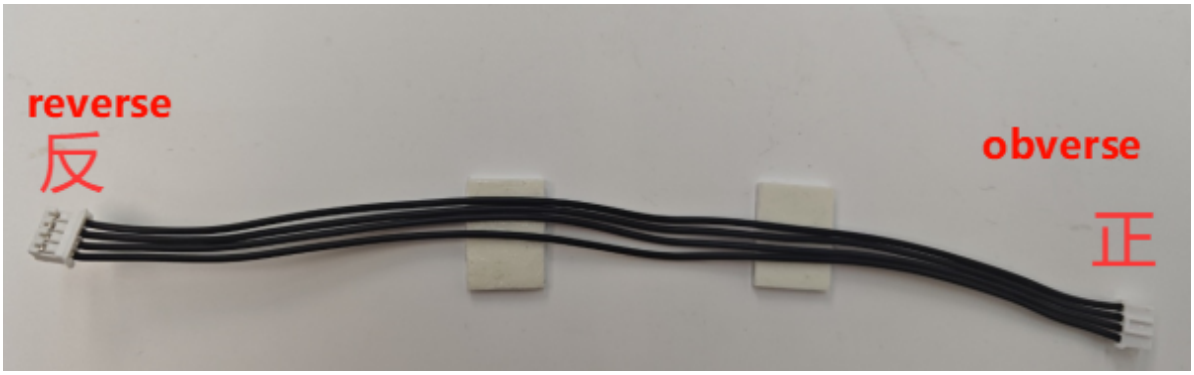
M4 -> lower right motor (right rear wheel of the car)

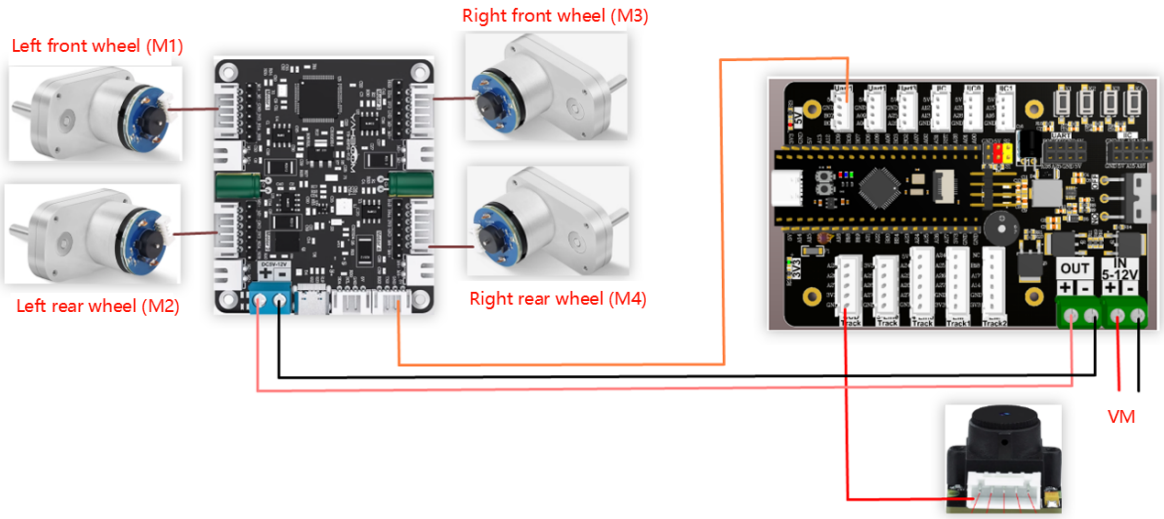
Hardware wiring:

Wiring using MSPM0 robot expansion board

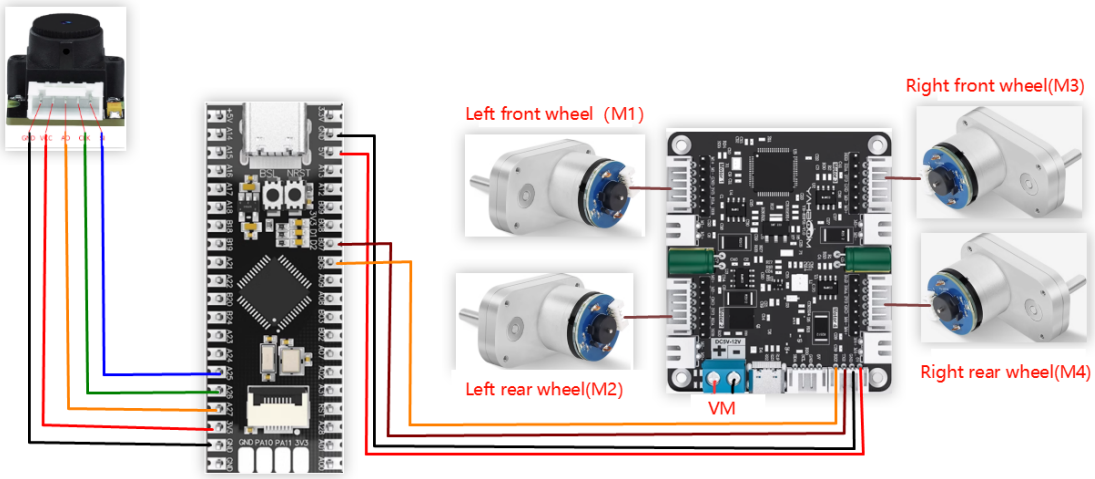
Note: The cable used for the linear CCD camera module is: XH2.54 cable (5pin double-ended, all black, reverse 25cm), the direction of the reverse cable holder is as shown in the figure below

Note: The cable 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 as 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

Linear CCD camera module	MSPM0G3507 core board (Yahboom)
GND	GND
VCC	3V3
AO	PA27
CLK	PA26
SI	PA25

3. Key code analysis

- ccd.c

```
void  Find_CCD_Zhongzhi(void)
{
     static uint16_t i,j,Left,Right;
     static uint16_t value1_max,value1_min;

       value1_max=ADV[0];  //动态阈值算法，读取最大和最小值    Dynamic threshold
algorithm, reads maximum and minimum values
     for(i=5;i<123;i++)   //两边各去掉5个点    Remove 5 points from each side
     {
        if(value1_max<=ADV[i])
        value1_max=ADV[i];
     }
       value1_min=ADV[0];  //最小值    Minimum
     for(i=5;i<123;i++)
     {
```

```
        if(value1_min>=ADV[i])
        value1_min=ADV[i];
     }
   CCD_Yuzhi=(value1_max+value1_min)/2;   //计算出本次中线提取的阈值    Calculate the
threshold for this midline extraction
     for(i = 5;i<118; i++)   //寻找左边跳变沿  Find the left edge
    {
        if(ADV[i]>CCD_Yuzhi&&ADV[i+1]>CCD_Yuzhi&&ADV[i+2]>CCD_Yuzhi&&ADV[i+3]
<CCD_Yuzhi&&ADV[i+4]<CCD_Yuzhi&&ADV[i+5]<CCD_Yuzhi)
        {
            Left=i;
            break;
        }
    }
     for(j = 108;j>15; j--)//寻找右边跳变沿    Find the right edge
  {
        if(ADV[j]<CCD_Yuzhi&&ADV[j+1]<CCD_Yuzhi&&ADV[j+2]
<CCD_Yuzhi&&ADV[j+3]>CCD_Yuzhi&&ADV[j+4]>CCD_Yuzhi&&ADV[j+5]>CCD_Yuzhi)
        {
          Right=j;
          break;
        }
  }
    CCD_Zhongzhi=(Right+Left)/2;//计算中线位置    Calculate the centerline position
}

//开始CCD采集并处理输出数据    Start CCD acquisition and process output data
void deal_data_ccd(void)
{
        RD_TSL();
        Find_CCD_Zhongzhi();
}


void use_ccd_line_motion_PID(void)
{
    g_ccd_median=CCD_Zhongzhi - 55;
    pid_output_ele = (int)(APP_ELE_PID_Calc(g_ccd_median));//位置式PID Position
PID
    motion_car_control(CCD_SPEED, 0, pid_output_ele);//直接控制电机       Direct
motor control

}
```

Find_CCD_Zhongzhi: Get the median value after getting the CCD value

deal_data_ccd: Collect and process CCD data

use_ccd_line_motion_PID: After getting the median value, use the position PID to calculate the speed of each motor to drive the motor. According to the median value obtained by your own CCD in the middle of the track, you can modify the value of 55 to your own value.

- app_motor.c

```
#define CCD_PID_KP       (60)
#define CCD_PID_KI       (0)
#define CCD_PID_KD       (7)
```

```
float APP_ELE_PID_Calc(int8_t actual_value)
{
    float pid_out = 0;
    int8_t error;
    static int8_t error_last=0;
    static float Integral;//积分    integral

    error=actual_value-MID_ERR;

    Integral +=error;

    //位置式pid Positional pid
    pid_out=error*CCD_PID_KP
                          +CCD_PID_KI*Integral
                          +(error - error_last)*CCD_PID_KD;

    return pid_out;
}
```

Here, modify the PID effect of CCD line patrol. 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

- 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));
}
```

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

- empty.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
...
#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
...
while(1)
    {
        deal_data_ccd();
        use_ccd_line_motion_PID();
        delay_ms(6);
    }
```

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.

deal_data_ccd: processes CCD data and calculates the position of the center line.

use_ccd_line_motion_PID: uses position-based PID to drive the car to patrol the line

Set the motor type, transfer the preset motor parameters to the four-way motor driver board, and then drive the motor to control the car to patrol the line through the data collected by the CCD combined with the PID algorithm.

4. Experimental phenomenon

After connecting the car and burning the program to MSPM0, place the car on a map with a white background and black lines, aim the camera at the black lines, connect the power supply, and the car will patrol the black lines and drive automatically.