一、 实验名称：

竞速寻线小车完整设计

二、 实验目的：

1. 完成寻线小车的整体设计。
2. 理解各模块的实现及综合应用。

三、 实验器材：

FRDM-KL25z 实验板（编号：5XT6CONT173401）、笔记本电脑（内装 Keil V5 软件）、电机、小车底板、小车轮胎、洞洞板、光电二极管模块、直流 5V 稳压电源模块、电机驱动模块

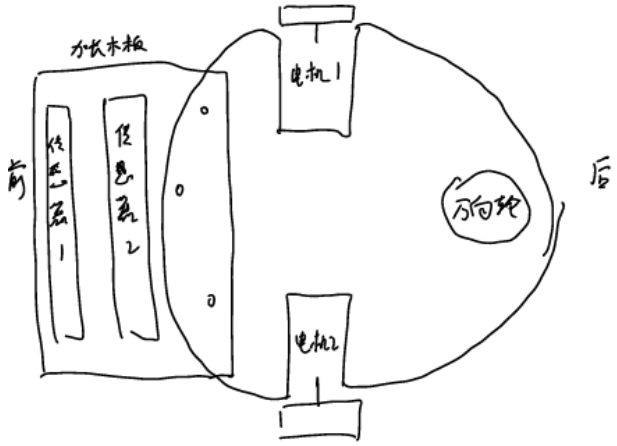
四、 实验过程及相应结果：

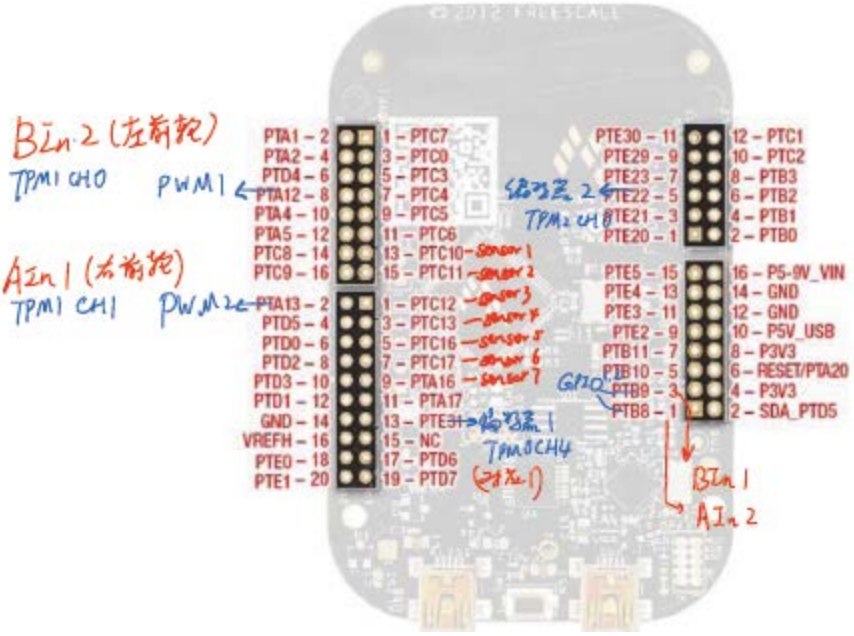
1. 小车设计思路：

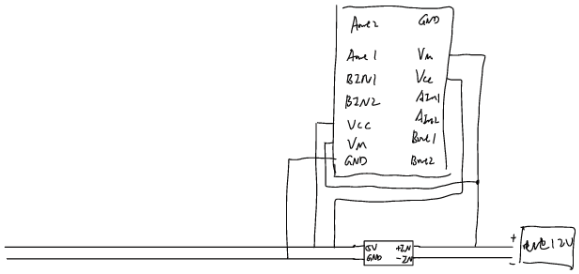
小车物理侧设计，搭建小车模型，连接 5V 电源模块，电机驱动模块，并固定电机等，完成小车物理侧设计。小车传感器置于车前端，采用 7 个光电二极管。物理侧输入单片机处理器的信号有光电二极管信号用于巡线，电机编码器信号用于电机转速的负反馈控制。单片机输出信号为 PWM 占空比可调信号，用于控制电机供电电压。

小车软件侧设计为两部分。PID 反馈控制电机驱动程序，通过输入的电机转速和要求达到转速进行 PWM 占空比调节，为模拟系统控制。小车寻线状态机控制模块，通过接收传感器的信号调整小车各轮应达转速，为数字控制系统。

2. 小车模型蓝图和各模块连接图



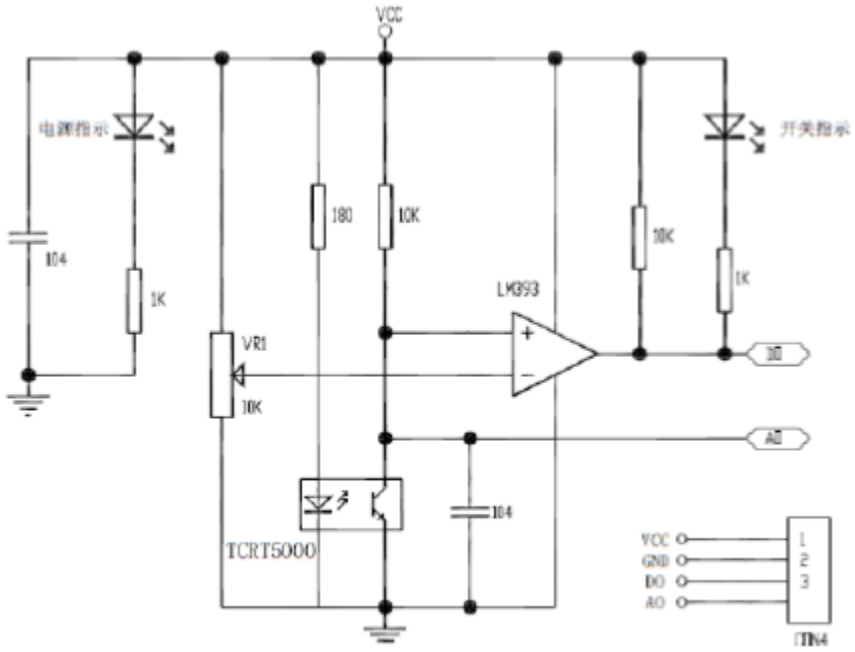




该图均为小车完成前原始设计蓝图。单片机各端口连接均已标注，驱动模块和直流电源模块均已通过布线连接。详情均见上图。

3.传感器电路与工作原理

本车所用传感器为光电二极管模块，且使用数字信号输出。

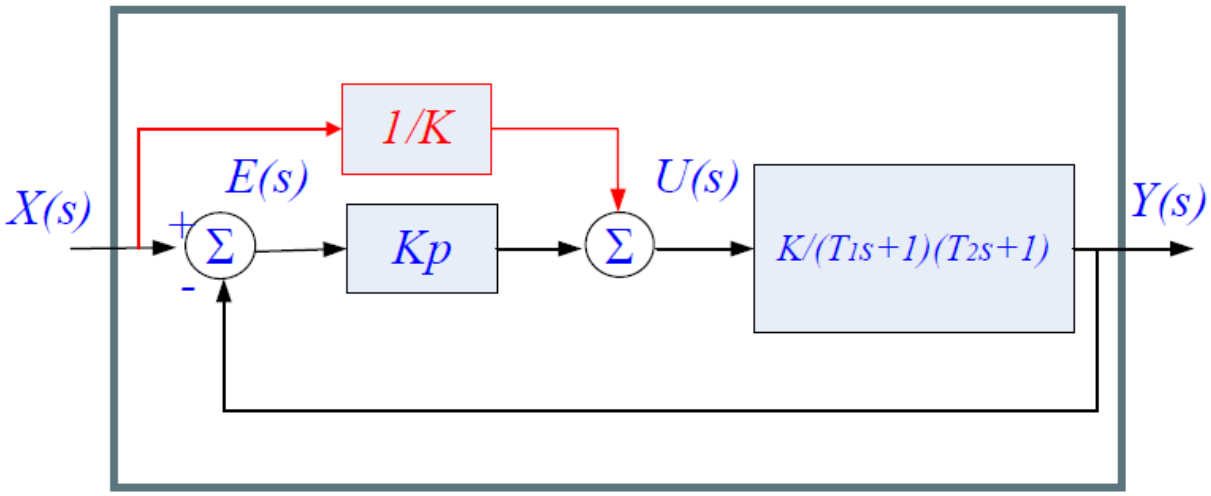


光电二极管模块电路如上，输出信号从 D 端口输出。传感器一端为发光二极管，发出红外光，另一端为接收端的光电二极管。传感器光电二管接收光信号强弱变化引起光电二极管阻值发生变化，导致比较器两边电压发生变化，输出高电平或地电平信号。当光电二极管接收到反射信号，即反射性强的介质（白区）时，会输出高电平信号，反之，介质反射性弱（黑区），输出低电平信号。同时可以通过调整滑动变阻器 VR1 来调整阈值电压。

该传感器缺少施密特整流电路，引起在阈值附近信号的不稳定是其非常大的缺点。

3. 速度控制系统模型

速度控制利用 PID 控制。该控制系统模型如下图所示。



其中 X 输入为电机编码器测速数值，Y 输出为 PWM 占空比数值。

测速方法为通过周期计数方法实现。编码器工作原理为：编码器上存在两组霍尔元件，分别对应 A 输出和 B 输出。编码器盘上有 13 组磁标器，每当磁标器通过编码盘时，即会产生一个方波脉冲。同时具有 30 倍的加速比。利用 TPM 模块记录两个上升沿间的系统时钟个数，即可描述两上升沿间的周期，从而标定速度。

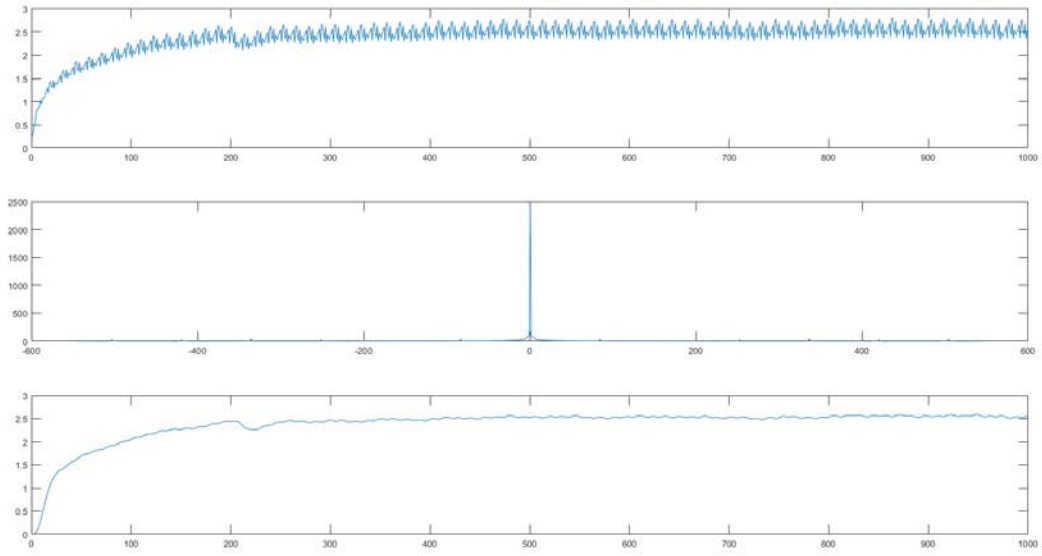
该测速方法面临一个问题，即当轮子停转时，会引起测速无法触发中断，从而无法触发正确速度返回。解决方法为，设置最低 PWM 为 10%左右，即 PWM 最低时电机仍有一定转速，保证测速中断一直可以被触发。

利用该模型进行建模，通过对组装成车后（存在负载和摩擦），两侧轮子的转速和 PWM 占空比控制采样，从而得到系统参数。实现两侧轮子分别采样的代码见附件 test_two_motor.c 文件。使用方法为注释掉右轮中断，测试左轮，再注释掉左轮，测试右轮。因为双侧采样时均以 50%占空比测试，所以两次分别采样并不会影响效果，而只是将左右轮数据分别上传至上位机。

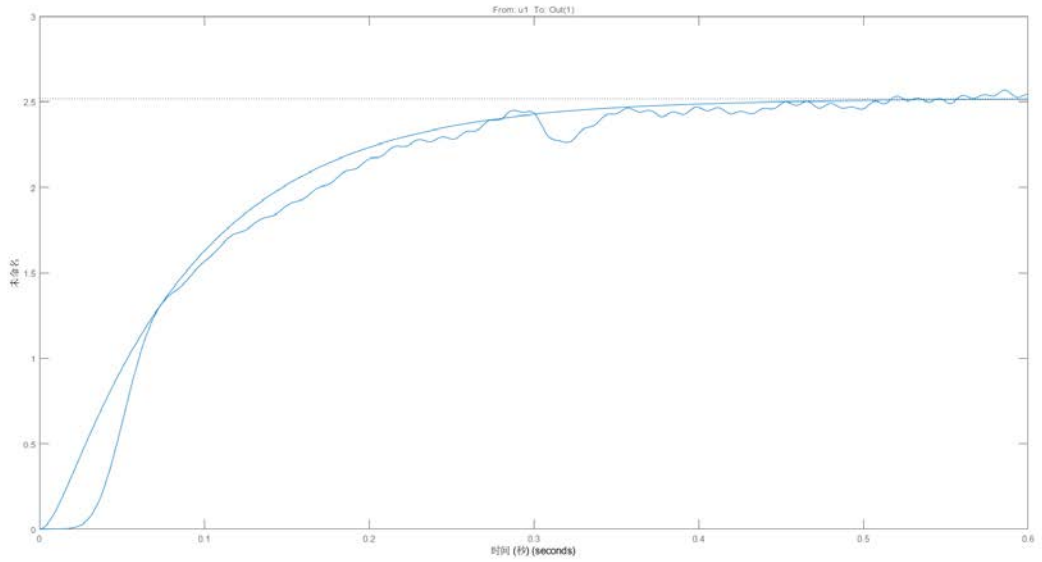
值得注意，该文件在建模时已包含滤波器，滤波器系数由 IIRfilter 定义。该滤波器系数与 MATLAB 模型计算中的滤波器系数完全一致，以保证模型效果统一。

左右轮分别测量的数据见 two_left.txt 和 two_right.txt 文件。两份文件均为车完全组装后，在地面进行的存在负载和摩擦的测试。原始的单电机无摩擦测试对现在小车建模仅有参考意义，而实际参数不同，原来的数据可以见之前的实验数据。

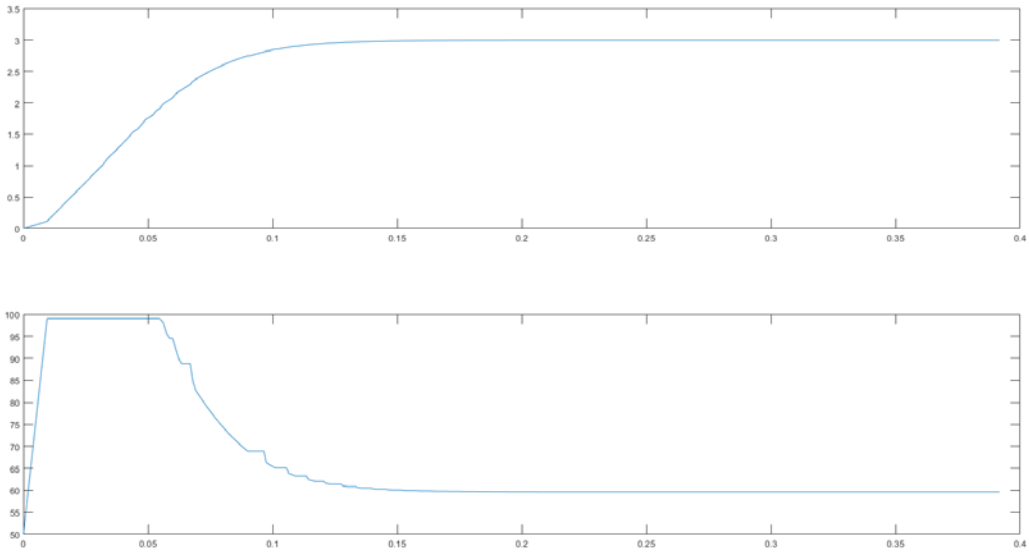
以左轮建模举例，运行 MATLAB 代码，可以得到如下的建模曲线。



原始数据为最上面的图像，频域为中间图像，通过滤波器后得到最下侧图像。



通过拟合后，得到的拟合曲线和滤波后数据进行对比。



通过上述数据计算出的 k 值和 kp 值设置仿真，得到的从 0 转加速到 3 转的仿真图像如上，从 0 转加速至 3 转约 0.1s 即可。

最终实现的 PID 控制电机驱动代码见 MotorDrive.h 文件。K 和 Kp 值最终计算得到分别如下。

左轮：float k_1=19.87; float kp_1=35.4650;

右轮：float k_2=17.98; float kp_2=44.41;

4. 角度控制系统

角度控制系统为状态机控制，对于不同传感器的返回值，分别设置不同转速。对于中间三个传感器，目的为巡直，可以在直线或者极小弧度时巡正车身，保持直线快速行驶。外围四个传感器用于转弧度较大的弯道，转速差设置较大。

状态机代码为 sensor.h 文件。提取其中重点状态和设置参数如下：

```
//0001000 or 0011100

#define STRAIGHTSPEED 3.0


//0011000

#define FIRST_LITTLE_CURVESPEED_HIGH 3.0
```

```
#define FIRST_LITTLE_CURVESPEED_LOW 2.5

//0010000
#define SECOND_LITTLE_CURVESPEED_HIGH 3.0
#define SECOND_LITTLE_CURVESPEED_LOW 1.0

//0110000
#define THIRD_MIDDLE_CURVESPEED_HIGH 3.5
#define THIRD_MIDDLE_CURVESPEED_LOW 0.5

//0100000
#define FOURTH_MIDDLE_CURVESPEED_HIGH 4.0
#define FOURTH_MIDDLE_CURVESPEED_LOW 0.5

//1100000
#define FIFTH_LARGE_CURVESPEED_HIGH 4.5
#define FIFTH_LARGE_CURVESPEED_LOW 0.5

//1000000
#define SIXTH_LARGE_CURVESPEED_HIGH 5.0
#define SIXTH_LARGE_CURVESPEED_LOW 0.5

//1111111 or 0111110
```

```
#define CROSSSPEED 3.0


//unknow state
#define UNKNOWNSPEED 1.0
SPEEDSET Judge_Speed()
{
    SPEEDSET judge_speed_result;

    //judge speed

        //hei wei 1
        sensor[0]=(PTC->PDIR & MASK(17))>>(17);
        sensor[1]=(PTC->PDIR & MASK(10))>>(10);
        sensor[2]=(PTC->PDIR & MASK(11))>>(11);
        sensor[3]=(PTC->PDIR & MASK(12))>>(12);
        sensor[4]=(PTC->PDIR & MASK(13))>>(13);
        sensor[5]=(PTC->PDIR & MASK(16))>>(16);
        sensor[6]=(PTA->PDIR & MASK(16))>>(16);

        if(sensor[0]==0 && sensor[1]==0 && sensor[2]==0 && sensor[3]==1 && sensor[4]==0 &&
sensor[5]==0 && sensor[6]==0)//0001000 straight
        {
            judge_speed_result.main_speed_set_1=STRAIGHTSPEED;
            judge_speed_result.main_speed_set_2=STRAIGHTSPEED;
```

```
        last_state=0; //straight
    }
    else if(sensor[0]==0 && sensor[1]==0 && sensor[2]==1 && sensor[3]==1 && sensor[4]==1 && sensor[5]==0 && sensor[6]==0)//0011100 straight
    {
        judge_speed_result.main_speed_set_1=STRAIGHTSPEED;
        judge_speed_result.main_speed_set_2=STRAIGHTSPEED;
        last_state=0;//straight
    }
    else if(sensor[0]==0 && sensor[1]==0 && sensor[2]==1 && sensor[3]==1 && sensor[4]==0 && sensor[5]==0 && sensor[6]==0)//0011000 1_little_left
    {
        judge_speed_result.main_speed_set_1=FIRST_LITTLE_CURVESPEED_LOW;
        judge_speed_result.main_speed_set_2=FIRST_LITTLE_CURVESPEED_HIGH;
        last_state=1;//1_little_left
    }
    else if(sensor[0]==0 && sensor[1]==0 && sensor[2]==1 && sensor[3]==0 && sensor[4]==0 && sensor[5]==0 && sensor[6]==0)//0010000 2_little_left
    {
        judge_speed_result.main_speed_set_1=SECOND_LITTLE_CURVESPEED_LOW;
        judge_speed_result.main_speed_set_2=SECOND_LITTLE_CURVESPEED_HIGH;
        last_state=2;//2_little_left
    }
    else if((sensor[0]==0 && sensor[1]==1 && sensor[2]==1 && sensor[3]==0 && sensor[4]==0 &&
```

```
sensor[5]==0 && sensor[6]==0)|| //0110000 3_middle_left

                (sensor[0]==0 && sensor[1]==1 && sensor[2]==1 && sensor[3]==1 && sensor[4]==0 &&
sensor[5]==0 && sensor[6]==0)|| //0111000 3_middle_left

                (sensor[0]==0 && sensor[1]==1 && sensor[2]==1 && sensor[3]==1 && sensor[4]==1 &&
sensor[5]==0 && sensor[6]==0))  //0111100 3_middle_left

        {

            judge_speed_result.main_speed_set_1=THIRD_MIDDLE_CURVESPEED_LOW;

            judge_speed_result.main_speed_set_2=THIRD_MIDDLE_CURVESPEED_HIGH;

            last_state=3;//3_middle_left

        }

        else if(sensor[0]==0 && sensor[1]==1 && sensor[2]==0 && sensor[3]==0 && sensor[4]==0 &&
sensor[5]==0 && sensor[6]==0)//0100000 4_middle_left

        {

            judge_speed_result.main_speed_set_1=FOURTH_MIDDLE_CURVESPEED_LOW;

            judge_speed_result.main_speed_set_2=FOURTH_MIDDLE_CURVESPEED_HIGH;

            last_state=4;//4_middle_left

        }

        else if(sensor[0]==1 && sensor[1]==1 && sensor[2]==0 && sensor[3]==0 && sensor[4]==0 &&
sensor[5]==0 && sensor[6]==0)//1100000 5_mlarge_left

        {

            judge_speed_result.main_speed_set_1=FIFTH_LARGE_CURVESPEED_LOW;

            judge_speed_result.main_speed_set_2=FIFTH_LARGE_CURVESPEED_HIGH;

            last_state=5;//5_mlarge_left

        }
```

```
else if((sensor[0]==1 && sensor[1]==0 && sensor[2]==0 && sensor[3]==0 && sensor[4]==0 &&
sensor[5]==0 && sensor[6]==0)|| //1000000 6_mlarge_left

        (sensor[0]==1 && sensor[1]==1 && sensor[2]==1 && sensor[3]==0 && sensor[4]==0 &&
sensor[5]==0 && sensor[6]==0)|| //1110000

        (sensor[0]==1 && sensor[1]==1 && sensor[2]==1 && sensor[3]==1 && sensor[4]==0 &&
sensor[5]==0 && sensor[6]==0)|| //1111000

        (sensor[0]==1 && sensor[1]==1 && sensor[2]==1 && sensor[3]==1 && sensor[4]==1 &&
sensor[5]==0 && sensor[6]==0)) //1111100

    {

        judge_speed_result.main_speed_set_1=SIXTH_LARGE_CURVESPEED_LOW;

        judge_speed_result.main_speed_set_2=SIXTH_LARGE_CURVESPEED_HIGH;

        last_state=6;//6_mlarge_left

    }

    else if(sensor[0]==0 && sensor[1]==0 && sensor[2]==0 && sensor[3]==1 && sensor[4]==1 &&
sensor[5]==0 && sensor[6]==0)//0001100 1_little_right

    {

        judge_speed_result.main_speed_set_1=FIRST_LITTLE_CURVESPEED_HIGH;

        judge_speed_result.main_speed_set_2=FIRST_LITTLE_CURVESPEED_LOW;

        last_state=7;//1_little_right

    }

    else if(sensor[0]==0 && sensor[1]==0 && sensor[2]==0 && sensor[3]==0 && sensor[4]==1 &&
sensor[5]==0 && sensor[6]==0)//0000100 2_little_right

    {

        judge_speed_result.main_speed_set_1=SECOND_LITTLE_CURVESPEED_HIGH;

        judge_speed_result.main_speed_set_2=SECOND_LITTLE_CURVESPEED_LOW;
```

```
        last_state=8;//2_little_right
    }
    else if((sensor[0]==0 && sensor[1]==0 && sensor[2]==0 && sensor[3]==0 && sensor[4]==1 && sensor[5]==1 && sensor[6]==0)|| //0000110 3_middle_right
            (sensor[0]==0 && sensor[1]==0 && sensor[2]==1 && sensor[3]==1 && sensor[4]==1 && sensor[5]==1 && sensor[6]==0))    //0011110
    {
        judge_speed_result.main_speed_set_1=THIRD_MIDDLE_CURVESPEED_HIGH;
        judge_speed_result.main_speed_set_2=THIRD_MIDDLE_CURVESPEED_LOW;
        last_state=9;//3_middle_right
    }
    else if(sensor[0]==0 && sensor[1]==0 && sensor[2]==0 && sensor[3]==0 && sensor[4]==0 && sensor[5]==1 && sensor[6]==0)//0000010 4_middle_right
    {
        judge_speed_result.main_speed_set_1=FOURTH_MIDDLE_CURVESPEED_HIGH;
        judge_speed_result.main_speed_set_2=FOURTH_MIDDLE_CURVESPEED_LOW;
        last_state=10;//4_middle_right
    }
    else if(sensor[0]==0 && sensor[1]==0 && sensor[2]==0 && sensor[3]==0 && sensor[4]==0 && sensor[5]==1 && sensor[6]==1)//0000011 5_mlarge_right
    {
        judge_speed_result.main_speed_set_1=FIFTH_LARGE_CURVESPEED_HIGH;
        judge_speed_result.main_speed_set_2=FIFTH_LARGE_CURVESPEED_LOW;
        last_state=11;//5_mlarge_right
```

```
}
else if((sensor[0]==0 && sensor[1]==0 && sensor[2]==0 && sensor[3]==0 && sensor[4]==0 && sensor[5]==0 && sensor[6]==1)|| //0000001 6_mlarge_right
        (sensor[0]==0 && sensor[1]==0 && sensor[2]==0 && sensor[3]==0 && sensor[4]==1 && sensor[5]==1 && sensor[6]==1)|| //0000111
        (sensor[0]==0 && sensor[1]==0 && sensor[2]==0 && sensor[3]==1 && sensor[4]==1 && sensor[5]==1 && sensor[6]==1)|| //0001111
        (sensor[0]==0 && sensor[1]==0 && sensor[2]==1 && sensor[3]==1 && sensor[4]==1 && sensor[5]==1 && sensor[6]==1))    //0011111
{
    judge_speed_result.main_speed_set_1=SIXTH_LARGE_CURVESPEED_HIGH;
    judge_speed_result.main_speed_set_2=SIXTH_LARGE_CURVESPEED_LOW;
    last_state=12;//6_mlarge_right
}
else if(sensor[0]==1 && sensor[1]==1 && sensor[2]==1 && sensor[3]==1 && sensor[4]==1 && sensor[5]==1 && sensor[6]==1)//111111 cross
{
    judge_speed_result.main_speed_set_1=CROSSSPEED;
    judge_speed_result.main_speed_set_2=CROSSSPEED;
    last_state=13;//cross
}
else if(sensor[0]==0 && sensor[1]==1 && sensor[2]==1 && sensor[3]==1 && sensor[4]==1 && sensor[5]==1 && sensor[6]==0)//011110 cross
{
    judge_speed_result.main_speed_set_1=CROSSSPEED;
```

```
        judge_speed_result.main_speed_set_2=CROSSSPEED;
        last_state=13;//cross
    }
    else if(sensor[0]==0 && sensor[1]==0 && sensor[2]==0 && sensor[3]==0 && sensor[4]==0 && sensor[5]==0 && sensor[6]==0)//no_clear_state so keep last state
    {
        if (last_state==0)
        {
            judge_speed_result.main_speed_set_1=STRAIGHTSPEED;
            judge_speed_result.main_speed_set_2=STRAIGHTSPEED;
            last_state=0;//straight
        }
        else if(last_state==1){
            judge_speed_result.main_speed_set_1=FIRST_LITTLE_CURVESPEED_LOW;
            judge_speed_result.main_speed_set_2=FIRST_LITTLE_CURVESPEED_HIGH;
            last_state=1;//1_little_left
        }
        else if(last_state==2){
            judge_speed_result.main_speed_set_1=SECOND_LITTLE_CURVESPEED_LOW;
            judge_speed_result.main_speed_set_2=SECOND_LITTLE_CURVESPEED_HIGH;
            last_state=2;//2_little_left
        }
        else if(last_state==3){
            judge_speed_result.main_speed_set_1=THIRD_MIDDLE_CURVESPEED_LOW;
```

```
        judge_speed_result.main_speed_set_2=THIRD_MIDDLE_CURVESPEED_HIGH;
        last_state=3;//3_middle_left
    }
    else if(last_state==4){
        judge_speed_result.main_speed_set_1=FOURTH_MIDDLE_CURVESPEED_LOW;
        judge_speed_result.main_speed_set_2=FOURTH_MIDDLE_CURVESPEED_HIGH;
        last_state=4;//4_middle_left
    }
    else if(last_state==5){
        judge_speed_result.main_speed_set_1=FIFTH_LARGE_CURVESPEED_LOW;
        judge_speed_result.main_speed_set_2=FIFTH_LARGE_CURVESPEED_HIGH;
        last_state=5;//5_mlarge_left
    }
    else if(last_state==6){
        judge_speed_result.main_speed_set_1=SIXTH_LARGE_CURVESPEED_LOW;
        judge_speed_result.main_speed_set_2=SIXTH_LARGE_CURVESPEED_HIGH;
        last_state=6;//6_mlarge_left
    }
    else if(last_state==7){
        judge_speed_result.main_speed_set_1=FIRST_LITTLE_CURVESPEED_HIGH;
        judge_speed_result.main_speed_set_2=FIRST_LITTLE_CURVESPEED_LOW;
        last_state=7;//1_little_right
    }
```

```
else if(last_state==8){
        judge_speed_result.main_speed_set_1=SECOND_LITTLE_CURVESPEED_HIGH;
        judge_speed_result.main_speed_set_2=SECOND_LITTLE_CURVESPEED_LOW;
        last_state=8;//2_little_right
}
else if(last_state==9){
        judge_speed_result.main_speed_set_1=THIRD_MIDDLE_CURVESPEED_HIGH;
        judge_speed_result.main_speed_set_2=THIRD_MIDDLE_CURVESPEED_LOW;
        last_state=9;//3_middle_right
}
else if(last_state==10){
        judge_speed_result.main_speed_set_1=FOURTH_MIDDLE_CURVESPEED_HIGH;
        judge_speed_result.main_speed_set_2=FOURTH_MIDDLE_CURVESPEED_LOW;
        last_state=10;//4_middle_right
}
else if(last_state==11){
        judge_speed_result.main_speed_set_1=FIFTH_LARGE_CURVESPEED_HIGH;
        judge_speed_result.main_speed_set_2=FIFTH_LARGE_CURVESPEED_LOW;
        last_state=11;//5_mlarge_right
}
else if(last_state==12){
        judge_speed_result.main_speed_set_1=SIXTH_LARGE_CURVESPEED_HIGH;
        judge_speed_result.main_speed_set_2=SIXTH_LARGE_CURVESPEED_LOW;
```

```
            last_state=12;//6_mlarge_right
        }
        else if(last_state==13){
            judge_speed_result.main_speed_set_1=CROSSSPEED;
            judge_speed_result.main_speed_set_2=CROSSSPEED;
            last_state=13;//cross
        }
        else if(last_state==14){
            judge_speed_result.main_speed_set_1=UNKNOWNSPEED;
            judge_speed_result.main_speed_set_2=UNKNOWNSPEED;
            last_state=14;//unknown
        }
        else{
            judge_speed_result.main_speed_set_1=UNKNOWNSPEED;
            judge_speed_result.main_speed_set_2=UNKNOWNSPEED;
            last_state=14;//unknown
        }
    }
    else{ //unknown
        judge_speed_result.main_speed_set_1=UNKNOWNSPEED;
        judge_speed_result.main_speed_set_2=UNKNOWNSPEED;
        last_state=14;//unknown
    }
```

```
    return judge_speed_result;
}
```

5. 算法主流程

算法主流程见 mainloop.c 文件。

本质上是状态机判断和 PID 控制的循环。计算可以发现，状态机判断函数和电机反馈控制函数均在 600 个时钟周期以内，单片机使用系统时钟为 48MHz，那么对于状态机判断和执行反馈控制（假设触发了中断，此时反馈控制时间最长）过程约占用时间 0.025ms，量级小于 1ms，理论上不会导致车的状态无法及时改变。所以没有采用状态机中断结构，而是直接进行循环结构。实际测量时，用手突然挡住一些传感器，进行测试，轮子转速改变迅速，几乎无延迟。

6. 实物照片