

# 第十一届“恩智浦”杯全国大学生 智能汽车竞赛

## 技术报告



学校：安徽三联学院

队伍名称：高瞻远瞩

参赛队员：程 祯

娄 凯

吴晓华

指导教师：梁月放

刘姗姗



# 关于技术报告和研究论文使用授权的说明

本人完全了解第十届“恩智浦”杯全国大学生智能汽车竞赛关保留、使用技术报告和研究论文的规定，即：参赛作品著作权归参赛者本人，比赛组委会和恩智浦半导体公司可以在相关主页上收录并公开参赛作品的设计方案、技术报告以及参赛模型车的视频、图像资料，并将相关内容编纂收录在组委会出版论文集中。

参赛队员签名：程祯 娄凯 吴晓华

带队教师签字：梁月放 刘姗姗

日 期：2016 年 8 月

## 摘要

本文以历届全国大学生智能车竞赛为背景，利用恩智浦半导体公司生产的 MK60DN512ZVLQ10 高性能 32 位单片机和 OV7725 摄像头的配合来实现自动识别道路，让赛车可以在特定的跑道上行驶。通过摄像头成像原理进行图像识别, 控制车模转向。车模采用组委会统一提供的 B 车模，为了提高智能车的行驶速度和稳定性，对比了不同方案的优缺点，并在原有的基础上，尝试各种算法和控制策略，最终得出适合智能车的整体方案。

**关键词：**PID 算法    OV7725

## Abstract

In this paper, the previous National Students' smart car race background, with the use of NXP Semiconductors MK60DN512ZVLQ10 company's high-performance 32-bit microcontroller and OV7725 camera to automatically identify the road, so the car can travel on a particular runway. Image recognition by camera imaging principle, the steering control car models. Cars using the Organizing Committee provided B car models, in order to improve the speed and stability of the smart car, comparing the advantages and disadvantages of different options, and the basis of the original, try a variety of algorithms and control strategies, the final draw for Smart car overall program.

**Keywords:** PID algorithm    OV7725

# 目录

第一章 引言.....	1
1.1 智能车大赛简介.....	1
1.2 智能汽车制作情况概述.....	1
第二章 机械结构设计说明.....	2
2.1 车模机械结构设计.....	2
2.2 各机械的安装.....	4
2.2.1 摄像头的安装 .....	4
2.2.2 编码器的安装 .....	4
2.2.3 PCB 板的安装 .....	4
2.2.4 电池的安装 .....	5
2.2.5 整车设计效果图 .....	5
第三章 硬件系统设计.....	6
3.1 图像采集模块.....	6
3.2 电源管理模块.....	6
3.3 电机驱动模块.....	7
3.4 编码器模块 .....	8
3.5 人机交互模块.....	9
第四章 软件系统设计.....	11
4.1 软件系统设计方案.....	11
4.2 车模行驶信号获取与处理.....	12
4.2.1 赛道图像获取与处理 .....	12
4.2.2 车模行驶偏差获取与处理.....	14
4.2.3 障碍的识别 .....	15
4.2.4 起跑线的识别 .....	16
4.2.5 坡道的识别 .....	16

4.3 车模行驶速度控制.....	16
4.3.1 车模速度控制 .....	16
4.3.2 车模转向控制 .....	18
第五章 系统调试过程.....	19
5.1 IAR Embedded Workbench IDE 在线调试.....	19
5.2 串口调试工具.....	19
第六章 总结.....	21
参考文献.....	I
附录 .....	II





# 第一章 引言

## 1.1 智能车大赛简介

全国大学生“恩智浦”杯智能汽车竞赛起源于韩国，是韩国汉阳大学汽车控制实验室在恩智浦半导体公司资助下举办的以恩智浦单片机为核心的大学生课外科技竞赛。组委会提供一个标准的汽车模型、直流电机和可充电式电池，参赛队伍要制作一个能够自主识别路径的智能车，在专门设计的跑道上自动识别道路行驶，最快跑完全程而没有冲出跑道并且技术报告评分较高为获胜者。其设计内容涵盖了控制、模式识别、传感技术、汽车电子、电气、计算机、机械、能源等多个学科的知识，对学生的知识融合和实践动手能力的培养，具有良好的推动作用。

## 1.2 智能汽车制作情况概述

在本次比赛中，本组使用大赛组委会统一提供的竞赛车模，采用恩智浦32位微控制器 MK60DN512ZVLQ10作为核心控制单元，自主构思控制方案及系统设计，包括传感器信号采集处理、控制算法及执行、动力电机驱动等，最终实现一套能够自主识别路线，并且可以实时输出车体状态的智能车控制系统。

在制作小车的过程中，我们对小车的整体构架进行了深入的研究，分别在机械结构、硬件和软件上都进行过改进，硬件上主要是考虑并实践各种传感器的布局，改进驱动电路，软件上先后进行了几次大改，小车的寻线方式采用适应性较强的优化的位置加权的方法。控制算法上，从PID到Bang-Bang, 再到模糊PID都进行了一些研究。

## 第二章 机械结构设计说明

### 2.1 车模机械结构设计

本届智能车大赛摄像头组所使用的是 B 车模，原装的 B 车模上安装有减震设备，为了提高小车的稳定性，我们拆除了减震，更改了舵机的安装位置，降低了底盘高度。在车模高速行驶过程中，底盘越低，车模重心越低，车模运转也会更加平稳。

#### 1. 舵机的安装



图 2.1 舵机的安装

**我们的舵机采取如下装法：**舵机直立安装，并靠近转向桥。舵机安装直接关系到是否能快速灵敏地转向的问题。如果舵机调整不到位，将很大程度上限制转向角度和转向响应速度。

舵机安装有两种方式，一种是卧式安装，另外一种为立式安装。舵机直立安装是为了保证舵机在底盘上的质量分布尽量平均。对于连片的选择，通过试验和理论的分析，我们发现长度较短的连片优点是能够输出更大力矩，但是不足的是反应速度不够快，而对于长的连片优点是反应速度较快，但是输出力矩不足，所以综合考虑舵机的反应特性和输出力矩、转向模型和极限。

## 2. 前轮的调校

为了保持汽车直线行驶的稳定性、转向的轻便性和减少轮胎与机件的磨损，转向车轮、转向节和前轴三者与车架安装时保持一定的相对位置或要求，这种具有一定相对位置的安装称为转向轮定位，也称前轮定位。前轮定位的内容包括：主销后倾、主销内倾、前轮外倾和前轮前束。

### (1) 主销后倾

主销装在前轴上后，在汽车纵向平面内，其上端略向后倾斜，这种现象称为主销后倾。在纵向垂直平面内，主销轴线与垂线之间夹角  $\gamma$  叫主销后倾角。主销后倾角一般为  $0.5^\circ \sim 3^\circ$ 。主销后倾作用是保持汽车直线行驶的稳定性，并力图使转弯后的前轮自动回正。

### (2) 主销内倾角

主销安装在前轴上后，在汽车的横向平面内，其上端略向内倾斜，这种现象称为主销内倾。在横向垂直平面内，主销轴线与垂线之间的夹角  $\beta$  叫做主销内倾角。主销内倾角一般为  $6^\circ \sim 9^\circ$ 。主销内倾的作用是使前轮自动回正，转向轻便，并减小汽车行驶时路面通过车轮传给转向机构的冲击力。

### (3) 前轮外倾角

转向轮安装在车桥上，其旋转平面上方略向外倾斜，这种现象称为前轮外倾。前轮旋转平面与纵向垂直平面之间的夹角  $\alpha$  叫做前轮外倾角。前轮外倾角一般为  $0.5^\circ \sim 2^\circ$ 。前轮外倾的作用在于提高了前轮工作的安全性和转向操纵轻便性。前轮设置外倾角后，地面对前轮的反作用力沿前轮旋转轴线的分力将前轮压向转向节内侧，可防止汽车行驶中前轮向外脱出，同时地面反力的作用线更接近于转向节轴的根部，可以减小转向力，使转向操纵轻便灵活。

### (4) 前轮前束

汽车两个前轮安装后，左右两前轮的旋转平面不平行，前端略向内收束，这种现象称为前轮前束。左右两前轮间后端距离 A 与前端距离 B 之差（A-B）称为前轮前束值。前轮前束值一般为  $0 \sim 12 \text{ mm}$ 。前轮前束的作用消除或减小汽车行驶过程中因前轮外倾而使两前轮前端向外张开的不良影响。前束过小，方向不能自动回正（前轮前束过小）；过大会导致轮胎外侧过度偏磨。

## 2.2 各机械的安装

### 2.2.1 摄像头的安装

为了保证摄像头拥有足够的视野，我们将摄像头尽可能的安装高，这样就可以保证以一定的角度俯视地面，一定程度上可以解决反光的影响。

### 2.2.2 编码器的安装

编码器安装位置不要贴近电机，以免造成干扰。为了减轻重量，我们使用排线接口作为编码器接口。

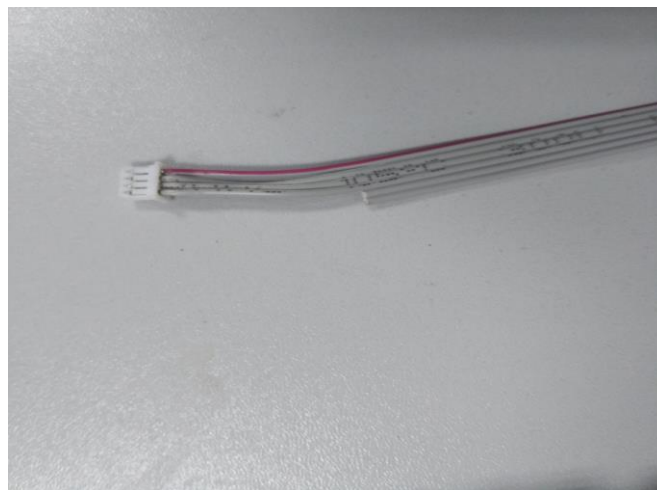


图 2.2 编码器接口

### 2.2.3 PCB 板的安装

为了减小体积，我们将电机驱动和单片机集成到了一片板子上。安装时使用尼龙柱固定在地盘上。以下是效果图：

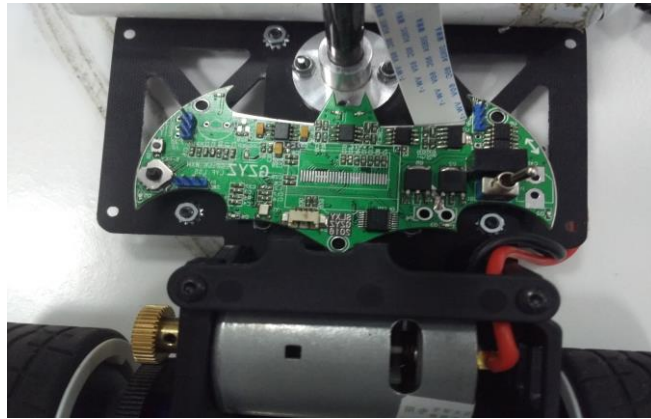


图 2.3 PCB 板的安装

#### 2.2.4 电池的安装

为了增大车模的稳定性,我们将电池安装在了车模的前部。并用扎带固定好。

#### 2.2.5 整车设计效果图



图 2.4 整车效果图





3.3 电机驱动模块

我们使用四个 MOS 管构成桥式电路，驱动电机。由于本届大赛摄像头组所使用的 B 车模电机为 540 电机，功率较大，工作电流也较大，我们选用了 1r7843mos 芯片作为驱动芯片。以下是该 mos 管的部分特性。

FET 类型	MOSFET N 通道，金属氧化物
FET 功能	标准
漏源极电压 (Vdss)	30V
电流 - 连续漏极 (Id) (25°C 时)	161A (Tc)
不同 Id, Vgs 时的 Rds On (最大值)	3.3 毫欧 @ 15A, 10V
不同 Id 时的 Vgs (th) (最大值)	2.3V @ 250μA
不同 Vgs 时的栅极电荷 (Qg)	50nC @ 4.5V
不同 Vds 时的输入电容 (Ciss)	4380pF @ 15V
功率 - 最大值	140W
工作温度	-55°C ~ 175°C (TJ)
安装类型	表面贴装
封装/外壳	TO-252-3, DPak (2 引线+接片), SC-63

图 3.3 mos 管的部分特性

为了防止电机驱动电路对单片机造成干扰，我们在驱动电路的输入级接入了逻辑门电路进行隔离。如下是隔离电路的原理图。

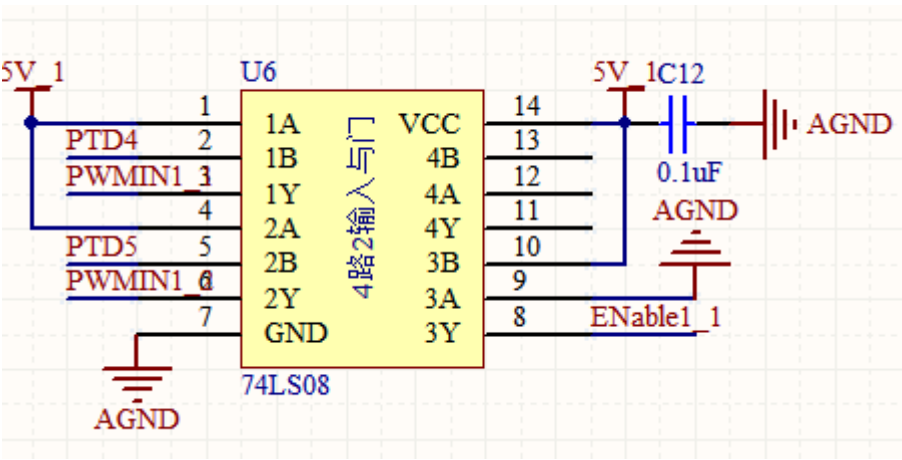


图 3.4 隔离电路原理图

### 3.4 编码器模块

为了使小车更轻，我们选用了泰庆电子生产的 512 线迷你编码器。该编码器特点是体积非常小巧，输出性能稳定，可以 3.3v 和 5v 供电。

编码器可以输出两相脉冲，经过单片机正交解码，我们就可以得到具有正负的脉冲数。

尺寸	13mm*18mm
重量	10g
输出轴	3mm D 型轴
排线长度	220mm

图 3.5 编码器规格

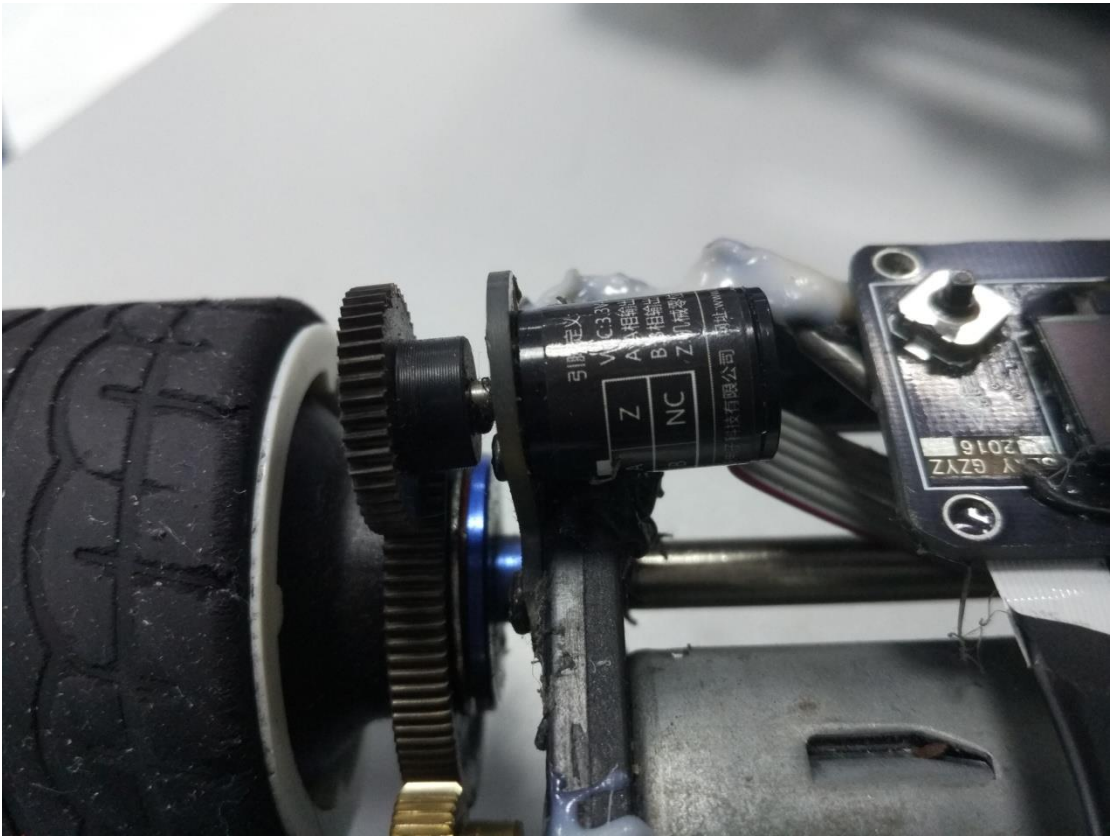


图 3.6 编码器效果图



在编码器电路中，我们在编码器的输出信号端接了 10K 欧姆的上拉电阻。以下是编码器接口电路。

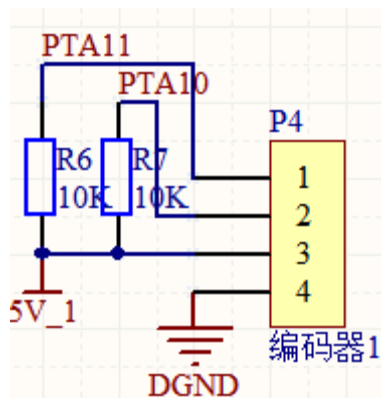


图 3.7 编码器接口电路

### 3.5 人机交互模块

人机交互模块我们使用五向按键和oled 液晶构成人机交互模块。五向按键和液晶模块构成类似于菜单栏的人机交互界面，所更改的数据存入 flash。以下是人机交互模块图。

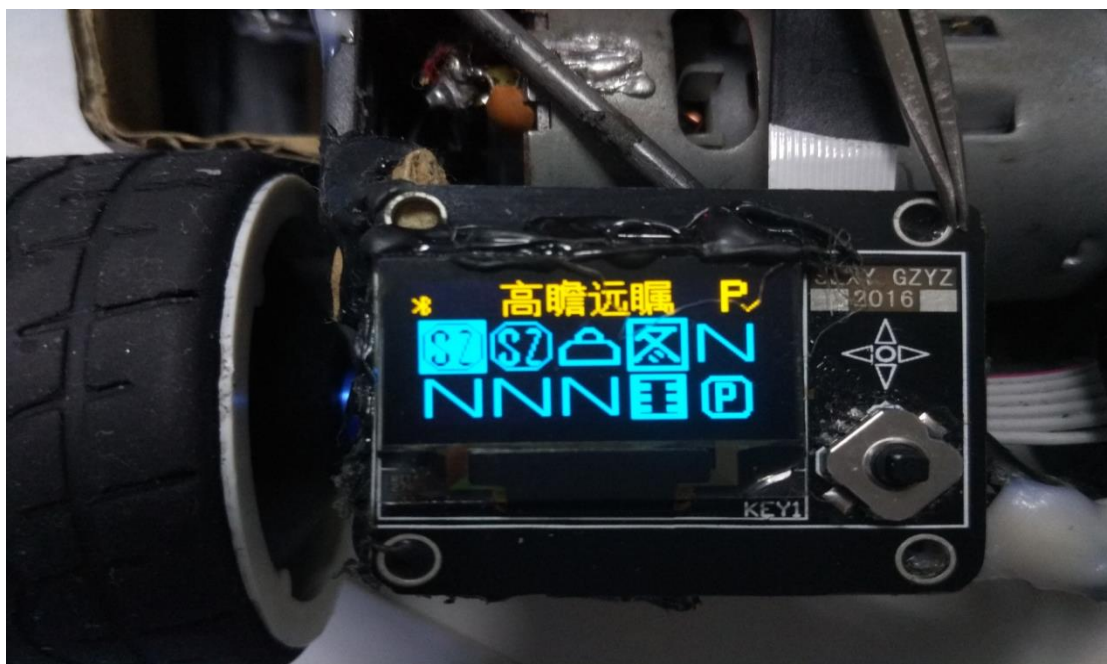


图 3.8 人机交互界面



## 第四章 软件系统设计

### 4.1 软件系统设计方案

首先通过摄像头提取赛道图像，经过图像处理程序后，提取车模的偏差，然后将偏差进行 PID 运算，控制舵机打角，并且图像处理中，算出弯道或者直道的曲线通过曲率判断弯道的缓急程度，从而使车模在弯道时减速，直道时加速。车模的速度我们使用 PID 算法控制。如图 4.1 和 4.2 所示：

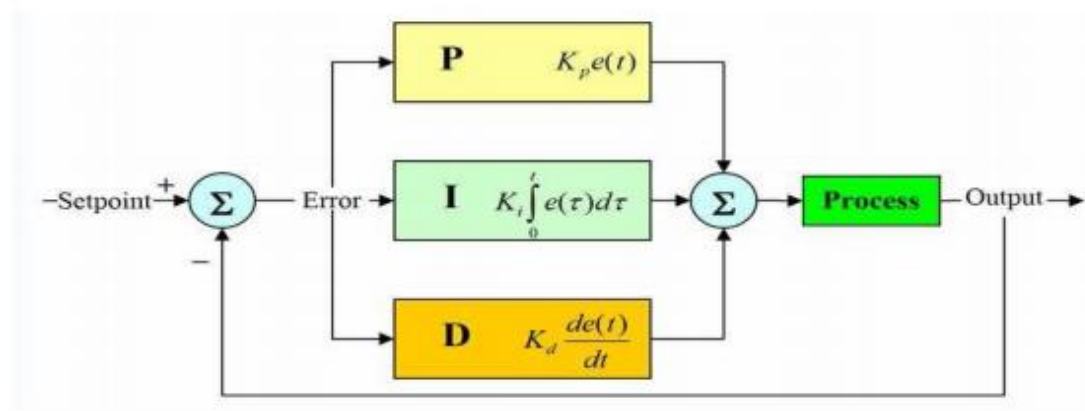


图 4.1 PID 算法运算解析

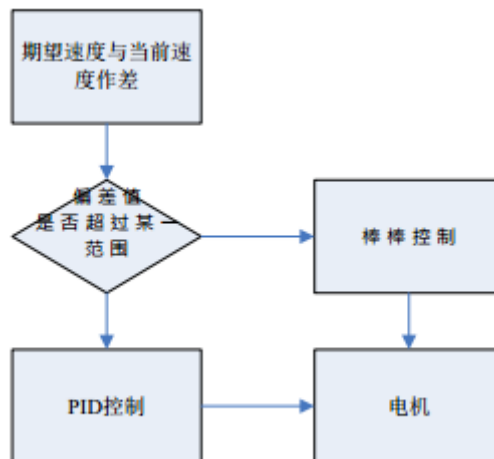


图 4.2 PID 算法流程图

## 4.2 车模行驶信号获取与处理

### 4.2.1 赛道图像获取与处理

我们将采集到的图像解压为 80\*60 分辨率的图像。将整个图像存储在一个二维数组里，对赛道的分析最终转化为对数组的处理。为了提取赛道的偏差，就必须首先得到赛道的中线，综合分析后，我们采用了从中间往两边扫描的策略。当发现图像有从白跳变为黑的现象，就记录下此时的坐标。最后我们将右边界加左边界除以二，就是赛道的中间线。当车模处于弯道时会出现丢的情况，此时我们就要进行补线，补线的思路是以没有丢的一边为基础加上整个赛道的宽度，就可以补齐丢失的一边。以下是补线后的效果图 4.3：

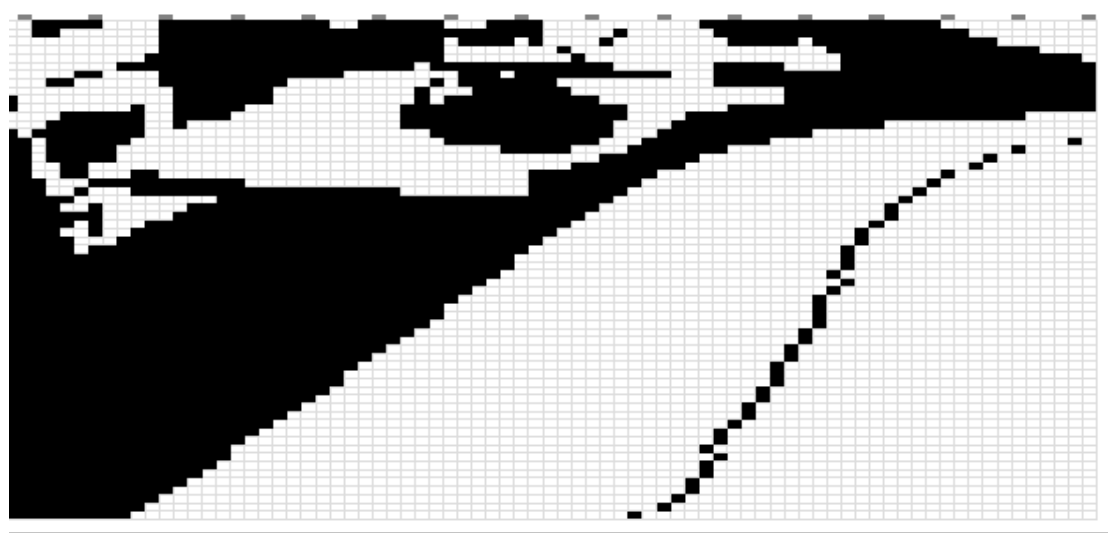


图 4.3 赛道处理效果图

赛道中会出现十字的情况，为此我们必须进行十字补线，十字补线的思路是根据图像底部的斜率向上延伸，就可以保证安全的通过十字，以下是十字补线的效果图 4.4：

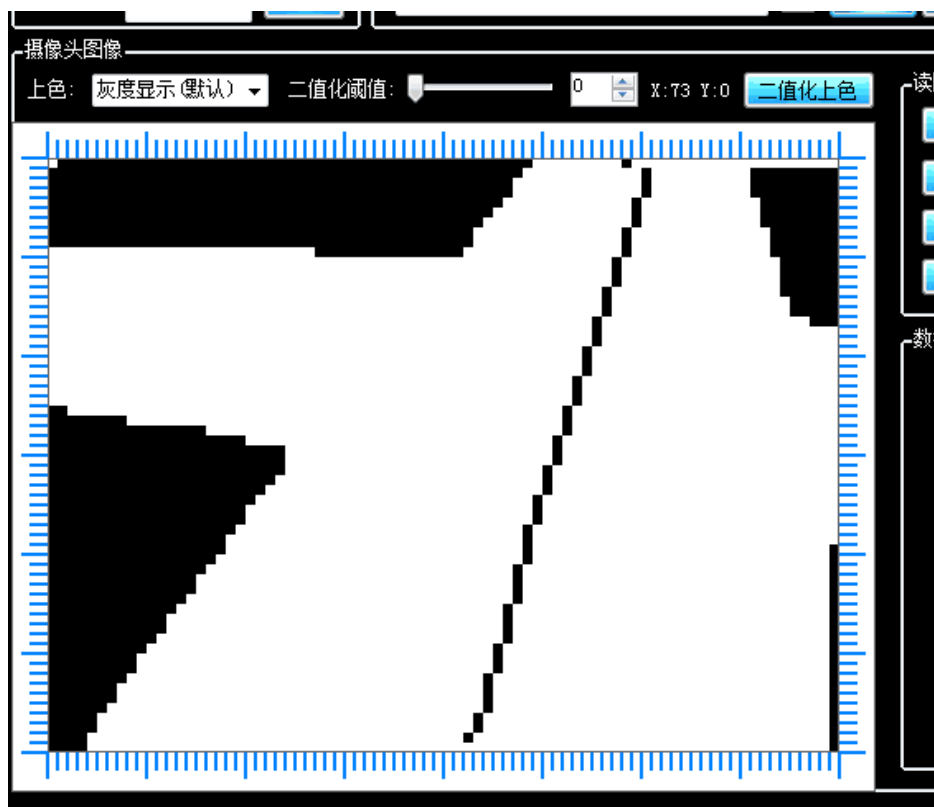


图 4.4 十字补线效果图

十字补线所用到的延伸函数：

```
void zhixian_yanshen(int Y1,int X1,int Y2,int X2,int Y3)
{
    int U,HY;
    for(U=Y1;U>=Y3;U--)
    {
        HY=(U-Y1)*(X2-X1)/(Y2-Y1)+X1;
        currentzhongjian[U]=HY;
    }
}
```

#### 4.2.2 车模行驶偏差获取与处理

车模行驶过程中为了使车模用最短的时间跑完全程，要求车模走最短路径，为此，在偏差的提取上我们去离车身大约 50cm 处的一点的偏差和摄像头视野范围内一段中线的斜率，以一定的权重相加拟合成总的偏差。再将偏差进行 pid 运算即可控制舵机打角。斜率计算方法最简单的是取两点，根据两点横纵坐标直接计算出斜率，但是这种计算方法不可靠，最经典的斜率计算方法即是最小二乘法。以下是最小二乘法计算斜率的 c 语言函数：

```
int regression(int startline,int endline)
{
    if(endline>55)
        endline=55;
    int i;
    int sumX=0,sumY=0,avrX=0,avrY=0 ;
    int num=0,B_up1=0,B_up2=0,B_up,B_down;
    for(i=startline;i<=endline;i++)
    {
        num++;
        sumX+=i;
        sumY+=currentzhongjian[i];
    }
    avrX=sumX/num;
    avrY=sumY/num;
    B_up=0;
    B_down=0;
    for(i=startline;i<=endline;i++)
    {
        B_up1=(int)(currentzhongjian[i]-avrY);
        B_up2=i-avrX;
```

```

        B_up+=(int)(10*(B_up1*B_up2));
        B_up=B_up/100*100;
        B_down+=(int)(10*((i-avrX)*(i-avrX)));
    }
    if(B_down==0)
        B=0;
    else
        B=B_up*10/B_down;
    return B;
}

```

#### 4.2.3 障碍的识别

障碍的图像如下：

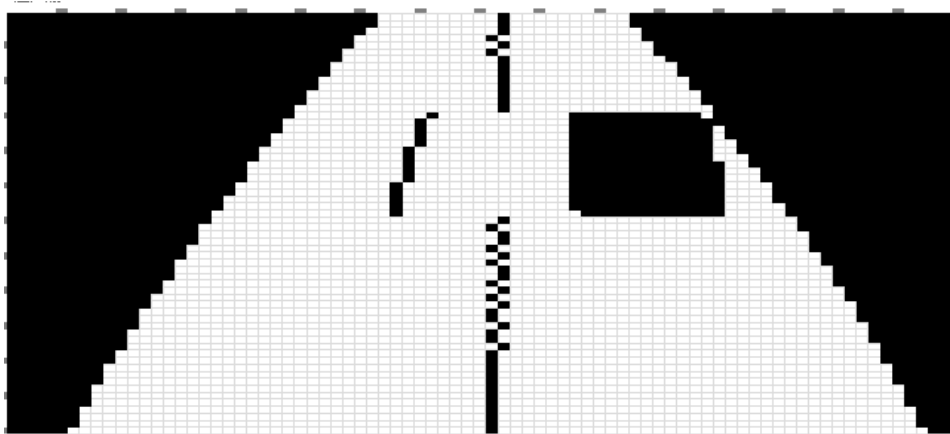


图 4.5 障碍识别图像

当检测到车模处于直道上且整副图像呈宽——窄——宽变化时，就认定为障碍。至于如何检测赛道变窄或者变宽，我们首先测取出车模在直道上检测到的每一行对应的宽度，作为标准宽度保存到数组里。只要与此数组进行对比就可以直道赛道是变宽还是变窄。检测到障碍后，只需要控制舵机打一定的角度，并保持一会，就可以绕过障碍。

#### 4.2.4 起跑线的识别

起跑线的图像如下：

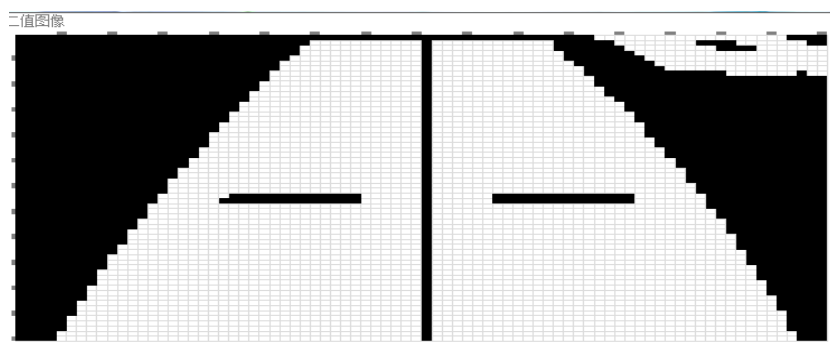


图 4.6 起跑线识别图像

起跑线识别我们利用了条边沿的思想，当车模处于直道上时，从底部几行中线的平均值向左向右分别推移一定的距离的点位置，当此点的图像有白黑白跳变时，就判定为起跑线。经过试验，这种方法没有出现过误判，很少出现漏判。

#### 4.2.5 坡道的识别

坡道图像如下：

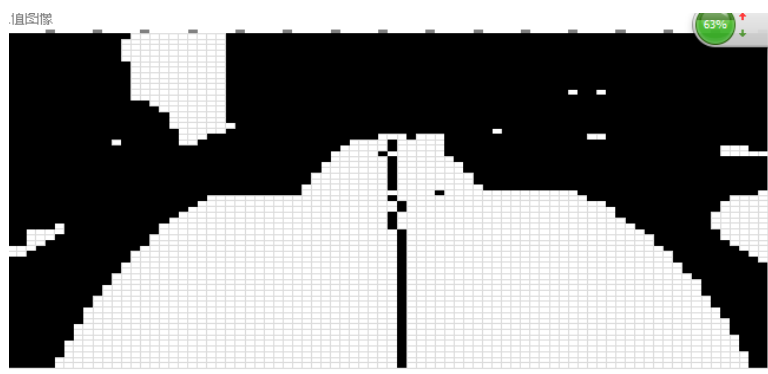


图 4.7 坡道识别图像

当检测到车中线和直道上的特征一样，且赛道逐渐变窄，并且左右是对称的就可以判定为坡道。

### 4.3 车模行驶速度控制

#### 4.3.1 车模速度控制

为了让车模直道加速弯道减速，我们通过曲率判断赛道的缓急程度。由于使



用数学上的曲率计算方法较为复杂，单片机执行起来较为吃力，我们简化了处理算法，即中线中某点相对于上一点每向右偏移一次，曲率加一，向左偏移一次减一，最后将整副图像的中线都判断一次就可以得到当前图像的曲率。根据曲率就可以判断赛道的缓急程度。不同的缓急程度给定不同的速度，为了使速度控制流畅，我们采用了 pid 控制车速，以下是 pid 控制车速的代码：

```

Void speed control()
{
    piancha_speed=maichongshu-speed_now;
    speed_wucha=speed_now-maichongshu;// bu
    dwucha=speed_wucha-lastspeed_wucha2;

    zhankongbi=last_zhankongbi+speed_kp*speed_wucha-speed_ki*lastspeed_wuc
    ha+speed_kd*dwucha;//pid 调速
    if(zhankongbi>7000)
        zhankongbi=7000;
    if(zhankongbi<0)
        zhankongbi=0;
    if(piancha_speed>menkan)
    {
        fanzhuan(dut);
    }
    else
        zhengzhuan(zhankongbi);
    last_zhankongbi=zhankongbi;
    lastspeed_wucha3=lastspeed_wucha2;
    lastspeed_wucha2=lastspeed_wucha1;
    lastspeed_wucha1=lastspeed_wucha;
    lastspeed_wucha=speed_wucha;
}

```

#### 4.3.2 车模转向控制

车模的转向控制我们采用了 pd 控制算法。为了补偿 B 车模转向的抖动，我们使用前第四次的偏差量和当前的偏差的差作为微分项，使得抖动现象有了很大好转。为了防止舵机打到死角堵转，需要对舵机角度进行限幅。以下是方向控制的代码：

```
void servo()//舵机控制函数
{
    jiao=(int)(kp*duojijiaodu-kd*(lastpiancha_4-duojijiaodu));
    if(jiao>=1000)
        jiao=1000;
    if(jiao<=-1000)
        jiao=-1000;
    ftm_pwm_duty(FTM1,FTM_CH0,4210+jiao); //20
}
```

## 第五章 系统调试过程

### 5.1 IAR Embedded Workbench IDE 在线调试

程序开放在 IAR Embedded Workbench IDE 下进行，Embedded Workbench for ARM 是 IAR Systems 公司为 ARM 微处理器开发的一个集成开发环境(下面简称 IAR EWARM)。比较其他的 ARM 开发环境，IAR EWARM 具有入门容易、使用方便和代码紧凑等特点。EWARM 中包含一个全软件的模拟程序(simulator)。用户不需要任何硬件支持就可以模拟各种 ARM 内核、外部设备甚至中断的软件运行环境。从中可以了解和评估 IAR EWARM 的功能和使用方法。

### 5.2 串口调试工具

串口调试工具在车模调试过程中起着非常重要的作用，传统的蓝牙模块，传输距离短，速率慢，因此我们采用的是基于 433M 频率的无线通信串口模块，此模块传输距离可达 1000 米，最高波特率可支持 194000b/s。



图 5.1 蓝牙模块

为了方便调试我们基于 C++制作了上位机软件。



图 5.2 上位机界面图

## 第六章 总结

以恩智浦单片机 K60 作为核心控制单元，以摄像头采集路况信息，以光电编码器对小车速度进行实时采集，完成智能车工程制作及调试，自主智能识别轨道、自动控制动力系统和转向系统完成智能控制。整个系统包括车体机械结构设计和系统的软/硬件系统设计。车体机械结构设计主要包括编码器和图像传感器的安装、舵机和前轮调节等；硬件系统设计完成了电源管理模块、电机驱动模块、舵机控制模块、速度测量模块、辅助调试模块、图像采集处理模块和单片机模块等的电路设计和调试；软件系统设计完成了各功能模块的算法及程序设计，包括图像采集算法设计、提取黑线算法设计以及舵机和电机的 PID 算法设计。所设计的系统经过测试，赛车能够快速安全行驶。模型车的主要技术参数如下表：

表 6.1 模型车主要技术参数

电路功耗 (w)	18
模型总质量 (g)	1158
模型车长/宽/高 (cm)	27/18/39
电容总容量 (uF)	800
传感器种类	3
传感器个数	3
除转向外伺服电机个数	0
赛道位置检测精度 (mm)	8
赛道位置检测频率 (Hz)	150



## 参考文献

- [1] 谭浩强. C 程序设计. 北京: 清华大学出版, 2003
- [2] 张军. AVR 单片机应用系统开发典型实例. 北京: 中国电力出版社, 2005.
- [3] 张文春. 汽车理论[M]. 北京. 机械工业出版社. 2005.
- [4] 郭宝龙 孙伟. 数字图像处理工程导论. 西安. 西安电子科技大学. 2012.
- [5] Kenneth R. Castleman. 数字图像处理. 北京. 电子工业出版社. 2002.
- [6] 马光. 单片机原理及应用. 北京. 机械工业出版社. 2006.
- [7] 刘金坤. 先进 PID 控制 Matlab 仿真. 北京. 机械工业出版社. 2004.
- [8] 黄友锐 曲立国. PID 控制器参数整定. 科学出版社. 2010.
- [9] 马光. 机械原理. 北京. 机械工业出版社. 2006.

## 附录

程序源代码:

```
#include "common.h"
#include "include.h"
#include "math.h"
#include "saomiao.h"

int yunsuflag;

uint8 imgbuff[CAMERA_SIZE];

    int currentzhongjian[60];

int quanjuCount,xielv_dian;

int OutData[4],maichongshu;

extern                                                    int

s_wan_flag,S_Z,s_count,dawan_speed,duojijiaodu,dian1,dian2,dian3,kp,zhankongbi,
mubiao_speed,dian,dianyuan,zhidao_flag,qvlv_quanju,control_point2,qulv_jinduan,q
ulv_yuandaun,my_piancha,xielv,xielv_yuan,right_heixian[60],left_heixian[60],contro
l_point,control_point1,qvlv_quanju_right,qvlv_quanju_left;

//int

kp=22,kd=10,right_heixian[60],lastpiancha_1,duojijiaodu,left_heixian[60],my_lastzh
ongjian=40,currentzhongjianzhi=40,,leftheixian_flag,xielv;

//函数声明

void PORTA_IRQHandler();

void DMA0_IRQHandler();

void PIT0_IRQHandler();

void Priority_Set();

extern int regression(int startline,int endline);

char time,star_lineflag,star_line,go,L,R;

extern char xielv_flag,qulv_point,zhangai_right,zhangai_left,podao_flag;

extern void speedcontrol();

extern void servo();
```



```

extern void myself_key();

//void PIT1_IRQHandler();

void hecheng()//虚拟中线函数
{
    unsigned int i;
    for(i=0;i<=59;i++)
    {
        imgyiwei[i][currentzhongjian[i]]=0 ;
        //imgyiwei[i][S_Z]=0 ;
        //imgyiwei[i][left_heixian[i]]=0 ;
    }
}

void sending(uint8 *imgaddr,uint32 imgsize)
{
    img_extract((u8 *)imgyiwei, imgbuff,CAMERA_SIZE);    //解压图像
    congzhongjianwangliangbian();
    uint8 cmd[4] = {0, 255, 1, 0 };    //yy_摄像头串口调试 使用的命令
    hecheng();
    uart_putbuff(VCAN_PORT, cmd, sizeof(cmd));    //先发送命令
    uart_putbuff(VCAN_PORT, imgaddr, imgsize); //再发送图像
}

/*****
*****
*/

* 功能说明：SCI 示波器 CRC 校验
内部调用函数
* 参数说明： 无
* 函数返回：无符号结果值

```

\* 修改时间：2013-2-10

\*/

//\*\*\*\*\*

\*\*\*\*\*

static unsigned short CRC\_CHECK(unsigned char \*Buf, unsigned char CRC\_CNT)

{

    unsigned short CRC\_Temp;

    unsigned char i,j;

    CRC\_Temp = 0xffff;

    for (i=0;i<CRC\_CNT; i++){

        CRC\_Temp ^= Buf[i];

        for (j=0;j<8;j++) {

            if (CRC\_Temp & 0x01)

                CRC\_Temp = (CRC\_Temp >>1 ) ^ 0xa001;

            else

                CRC\_Temp = CRC\_Temp >> 1;

        }

    }

    return(CRC\_Temp);

}

//\*\*\*\*\*

//

/\*

\* 功能说明：SCI 示波器发送函数

\* 参数说明：

OutData[] 需要发送的数值赋予该数组

```
* 函数返回：无符号结果值
* 修改时间：2013-2-10
*/
//*****

void OutPut_Data(void)
{
    int temp[4] = {0};
    unsigned int temp1[4] = {0};
    unsigned char databuf[10] = {0};
    unsigned char i;
    unsigned short CRC16 = 0;
    for(i=0;i<4;i++)
    {

        temp[i] = (int)OutData[i];
        temp1[i] = (unsigned int)temp[i];

    }

    for(i=0;i<4;i++)
    {
        databuf[i*2] = (unsigned char)(temp1[i]%256);
        databuf[i*2+1] = (unsigned char)(temp1[i]/256);
    }

    CRC16 = CRC_CHECK(databuf,8);
    databuf[8] = CRC16%256;
    databuf[9] = CRC16/256;

    for(i=0;i<10;i++)
```

```
{  
    uart_putchar (UART0,(char)databuf[i]);  
}  
}
```

```
void SendHex(unsigned char hex) {  
    unsigned char temp;  
    temp = hex >> 4;  
    if(temp < 10) {  
        uart_putchar(UART0,temp + '0');  
    } else {  
        uart_putchar(UART0,temp - 10 + 'A');  
    }  
    temp = hex & 0x0F;  
    if(temp < 10) {  
        uart_putchar(UART0,temp + '0');  
    } else {  
        uart_putchar(UART0,temp - 10 + 'A');  
    }  
}
```

```
void SendImageData(unsigned char ImageData[][80])  
{  
    int lll1, lll2;  
    unsigned char crc = 0;  
  
    /* Send Data */  
    uart_putchar(UART0, '*');  
    uart_putchar(UART0, 'L');  
    uart_putchar(UART0, 'D');
```

```
SendHex(0);
SendHex(0);
SendHex(0);
SendHex(0);

//  imgyiwei[60][80];

for(l1l2=0;l1l2<80;l1l2++)
{

    for(l1l1=0;l1l1<60; l1l1++)
        SendHex(ImageData[l1l1][l1l2]);
}

SendHex(crc);
uart_putchar(UART0,'#');
}

//***** 起跑线检测
*****

void star_line_judg()//起跑线检测
{
    int
    kk,left_baihei_flag,right_baihei_flag,baihei_flag,left_heibai_flag,right_heibai_flag;;
    for(kk=58;kk>48;kk--)
    {

        if(!right_baihei_flag&&(imgyiwei[kk][currentzhongjian[58]+14]-imgyiwei[kk+1][cu
```

```
rentzhongjian[58]+14]<0))//右跳变
```

```
    right_baihei_flag=1;
```

```
if(!left_baihei_flag&&(imgyiwei[kk][currentzhongjian[58]-14]-imgyiwei[kk+1][currentzhongjian[58]-14]<0))//左跳变
```

```
    left_baihei_flag=1;
```

```
if(right_baihei_flag&&left_baihei_flag)
```

```
    baihei_flag=1;
```

```
if(baihei_flag&&!right_heibai_flag&&(imgyiwei[kk][currentzhongjian[58]+14]-imgyiwei[kk+1][currentzhongjian[58]+14]>0))//右跳变
```

```
    right_heibai_flag=1;
```

```
if(baihei_flag&&!left_heibai_flag&&(imgyiwei[kk][currentzhongjian[58]-14]-imgyiwei[kk+1][currentzhongjian[58]-14]>0))//左跳变
```

```
    left_heibai_flag=1;
```

```
if(right_heibai_flag&&left_heibai_flag)
```

```
{
```

```
    star_lineflag=1;
```

```
    break;
```

```
}
```

```
}
```

```
}
```

```
//*********************************************************************main    函    数
```

```
//*********************************************************************
```

```
void main(void)
```

```
{
```

---

```

    DisableInterrupts;
    //初始化摄像头
    ftm_quad_init(FTM2);
    ftm_pwm_init(FTM0,FTM_CH5,10000,0); //电机初始化
    ftm_pwm_init(FTM0,FTM_CH4,10000,0);
    ftm_pwm_init(FTM1,FTM_CH0,300,4230);//舵机初始化
    Priority_Set();
    camera_init(imgbuff); //这里设定
imgbuff 为采集缓冲区!!!!!!
    pit_init_ms(PIT0,5);
    set_vector_handler(PIT0_VECTORn,PIT0_IRQHandler); // 设置中断服务
函数到中断向量表里
    enable_irq(PIT0_IRQn); // 使能 PIT 中断
    set_vector_handler(PORTA_VECTORn,PORTA_IRQHandler); // 设置
PORTA 的中断服务函数为 PORTA_IRQHandler
    set_vector_handler(DMA0_VECTORn,DMA0_IRQHandler); // 设置
DMA0 的中断服务函数为 DMA0_IRQHandler
    DMX_chengxuchushihua();
    //led_init(LED1);
    EnableInterrupts;
    //ftm_pwm_duty(FTM1,FTM_CH0,770);
    while(1)
    {
    // OutData[0]=mubiao_speed;
    //OutData[1]=maichongshu;
    // OutData[2]=zhankongbi;
    //OutPut_Data();
    camera_get_img();
    if(quanjuCount>300&&quanjuCount<308)
        go=1;

```

```
        if(quanjuCount>2000&&quanjuCount<2005)
            star_line=1;
        if(star_line)
            star_line_judg();//起跑线检测
    //    }
    // sending((u8 *)imgyiwei, CAMERA_W * CAMERA_H);
    }
}

/*!
 *   @brief      PORTA 中断服务函数
 *   @since      v5.0
 */
void PORTA_IRQHandler()
{
    uint8  n = 0;    //引脚号
    uint32 flag = PORTA_ISFR;
    PORTA_ISFR  = ~0;    //清中断标志
    位

    n = 29;    //场中断
    if(flag & (1 << n))    //PTA29 触发中断
    {
        camera_vsync();
    }

    #if 0    //鹰眼直接全速采集，不需要行中断
        n = 28;
        if(flag & (1 << n))    //PTA28 触发中断
        {
            camera_href();
        }
    #endif
}
```



```
    }
#endif
}

/*!
 * @brief      DMA0 中断服务函数
 * @since      v5.0
 */

//void PIT1_IRQHandler()
//{
//shizi_count++;
//}

void DMA0_IRQHandler()
{
camera_dma();
img_extract((u8 *)imgyiwei, imgbuff,CAMERA_SIZE);    //解压图像
}

void PIT0_IRQHandler(void)
{
    PIT_Flag_Clear(PIT0);
    if(flag_LCD)
        LCD_view();
else
{
    quanjuCount++;
    if(quanjuCount==5000)
        quanjuCount=0;
    maichongshu=ftm_quad_get(FTM2);
    ftm_quad_clean(FTM2);
}
```

```
    if(star_lineflag&&star_lineflag<20)
        star_lineflag++;
    if(star_lineflag>20)
        star_lineflag=20;

/*****                                直    道    判    定
*****/

if(abs(my_piancha)<=12&&qvlv_quanju<=12&&qulv_point<10||(abs(S_Z-39)<=5&
&qulv_point<10))
{
    zhidao_flag++;
}
else
{
    zhidao_flag=0;
}

/*****                                障    碍    处    理
*****/

if(zhangai_left)
{
    L=1;
    zhangai_left=0;
}
else
if(zhangai_right)
{
    R=1;
    zhangai_right=0;
}
```

```
if(L&&L<45)
{
L++;
//duojijiaodu=5;
}else
if(R&&R<45)
{
R++;
//duojijiaodu=-5;
}else
L=R=0;

/***** 图 像 处 理 及 偏 差 提 取
*****/

congzhongjianwangliangbian();//图像处理
if(xielv_flag)
{
xielv=-regression(12+control_point,xielv_dian+control_point);//斜率计算
}
if(xielv>18)
xielv=18;
if(xielv<-18)
xielv=-18;
pianchachuli();//偏差处理

/***** 舵 机 控 制
*****/

servo();

/***** 坡 道 处 理
*****/
```

```
if(podao_flag&&podao_flag<60)
podao_flag++;
else
podao_flag=0;
/*****          速    度    控    制
*****/
if(!star_lineflag&&go)
{
    if(!yunsuflag)
luduan_panduan();
else
if(yunsuflag)
mubiao_speed=dawan_speed;
//if(s_wan_flag)
//mubiao_speed=60;
}
else
if(star_lineflag==20)
mubiao_speed=0;
speedcontrol(mubiao_speed);
}
}
void Priority_Set(void)
{
    // NVIC_SetPriorityGrouping(4);          //设置优先级分组,4bit 抢占优先
级,没有亚优先级
    NVIC_SetPriority(PORTA_IRQn,0);          //配置优先级
    NVIC_SetPriority(DMA0_IRQn,1);          //配置优先级
    NVIC_SetPriority(PIT0_IRQn,2);          //配置优先级
```

```
}
```