

第七届“飞思卡尔”杯全国大学生 智能汽车竞赛

技 术 报 告



学 校：东北林业大学

队伍名称：赛豹号

参赛队员：卢炯健

王薪淼

徐加彦

带队教师：王琢 白雪冰

关于技术报告和学术论文使用授权的说明

本人完全了解第七届“飞思卡尔”杯全国大学生智能汽车竞赛关保留、使用技术报告和学术论文的规定，即：参赛作品著作权归参赛者本人，比赛组委会和飞思卡尔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：卢炯健

王薪淼

徐加彦

带队教师签名：王琢

日 期：2012 年 8 月 10 日

摘 要

随着电子科技的不断发展,越来越多的自动化设备开始进入到人们的生产生活中,嵌入式的迅猛发展为智能研究提供了更广阔的平台。

本文设计的智能车系统以 MC9S12XS128 微控制器为核心控制单元,通过选用 SONY CCD 模拟摄像头来检测赛道信息,采用 TCL5510 高速 A/D 转换芯片,提取灰度图像后对对其进行软件二值化,提取赛道信息;通过使用光电编码器,检测模型车的实时速度,采用 PID 控制算法调节驱动电机的转速和转向舵机的角度,从而实现了模型车运动速度和运动方向的闭环控制。

关键字: MC9S12XS128, SONY CCD, 软件二值化, PID

Abstract

With the continuous development of electronic technology, more and more automation equipment into the production life of the people, the rapid development of embedded intelligent study provides a broader platform.

In this paper, the design of intelligent vehicle system MC9S12XS128 microcontroller as the core control unit, the selection of SONY CCD analog cameras to detect the track information, to using TCL5510 high-speed AD converter chip, software binarization image, extract the white guide line for identification of the track information; optical encoder to detect the real-time speed of the model car, using the PID control algorithm to adjust the speed of the drive motor and steering the angle of the steering gear, in order to achieve closed-loop control of velocity and direction of the model car.

Keywords: MC9S12XS128 , SONY CCD, software binarization, PID

目录

摘 要	3
Abstract	4
目 录	5
第一章 引言.....	6
1.1 前言.....	6
1.2 论文概述.....	6
第二章 智能车机械分析与设计.....	7
2.1 车模舵机位置的设计与安.....	7
2.2 前轮的调整.....	8
2.3 后轮差速的调整.....	10
2.4 底盘的固定.....	11
2.5 摄像头的选择与安装.....	12
2.6 编码器的安装.....	15
2.7 整车效果图.....	16
第三章 智能车硬件电路部分.....	
3.1 电源系统.....	18
3.2 电机驱动电路设计.....	20
3.3 视频信号采集的处理.....	21
第四章 软件系统的设计与实现.....	
4.1 赛道信息的提取.....	22
4.2 PID 算法介绍.....	24
4.3 PID 参数整定.....	25
4.4 转向舵机控制算法.....	26
4.5 驱动电机控制策略.....	27
4.6 起跑线识别.....	27
第五章 智能车软件开发平台.....	
5.1 CodeWarrior 简介.....	31
5.2 调试工具 USBDM 简介.....	32
第六章 总结.....	33
参考文献.....	34

第一章 引言

1.1 前言

飞思卡尔（原摩托罗拉半导体部）是全球领先的半导体公司，为规模庞大、增长迅速的市场提供嵌入式处理产品和连接产品。飞思卡尔公司在中国有着“悠久”的历史，作为摩托罗拉公司的一部分，飞思卡尔公司是最早在中国建立分公司的首批半导体公司之一，为汽车、消费、工业、网络和无线市场设计并制造嵌入式半导体产品，其中包括传感器、射频半导体、功率的管理及其它模拟和混和信号集成电路。

设立飞思卡尔杯的目的是促进学生以协作、娱乐的方式学习汽车嵌入式控制技术，提高大学生的动手能力和创新能力，与其他大赛不同，飞思卡尔大赛综合性很强，以迅猛发展的汽车电子为背景，涵盖了控制、模式识别、传感、电子、电气、计算机和机械等多个学科交叉的科技创意性比赛。

在半导体技术日渐发展的今天，汽车电子化已成为行业发展的必然趋势。汽车电子化被认为是汽车技术发展进程中的一次革命，它是车体汽车电子控制装置和车载汽车电子控制装置的总称。车体汽车电子控制装置，包括发动机控制系统、底盘控制系统和车身电子控制系统（车身电子ECU）。汽车电子最显著特征是向控制系统化推进，用传感器、微处理器MPU、执行器、数十甚至上百个电子元器件及其零部件组成的电控系统，正获得极其广泛的市场。

1.2 论文概述

本文主要就模型车的制作过程进行介绍，文章中主要涉及的有机械部分、硬件部分以及软件部分，机械部分就车模的分析与改装、传感器的选择与安装以及电路板的安装等进行了分析和概括，硬件部分主要就电源供电电路、电机驱动电路、视频信号采集处理电路等进行了分析设计，软件部分主要介绍了赛道信息采集处理、赛道识别、软件算法与调试过程、各个核心模块的子程序编写等，在制作和调试过程中遇到的重点介绍了硬件电路的设计和路径识别的算法，接着描述了智能车的制作及调试过程，其中包含我们问题及其解决方法，并列出了模型车相关的技术参数。

第二章 智能车机械设计与安装

第七届“飞思卡尔”杯全国大学生智能车竞赛组委会提供的摄像头组车模为 A 车模，为了让智能车在直道和过弯道时反应灵敏，速度更快并且更加流畅，除了对硬件设计和软件设计有相应的要求以外，机械部分也对智能车的整体性能起着非常重要的作用，本章将介绍我们在由组委会给所提供的 A 车模的基础上对车模进行机械分析与改装，以达到使智能车机械性能最佳的要求。

2.1 车模舵机位置的设计与安装

今年摄像头组车模采用的舵机型号为 Futaba S3010，如果按照车模原型去安装舵机，车模舵机初始位置的那种安装方案采用了空间连杆，根据力学原理空间连杆的效率小，这样就造成了力传递过去的就非常少，而且这种舵机的力矩小，造成整体机构反应速度比较慢，并且经过调试，效果不是很理想，因此，我们在车模原型的基础上，对舵机的安装方式和安装位置进行了改装。

为了能够更加充分有效地利用车模底盘上的空间，我们决定采用倒置式安装的安装方式来安装舵机，这样就腾出了底盘中间一大块地方用来安装电路板，同时根据相关的机械参数加长连杆的长度，加大力矩以提高舵机的反应速率，是舵机反应更加灵敏。安装完成之后的舵机位置如图 2.1：

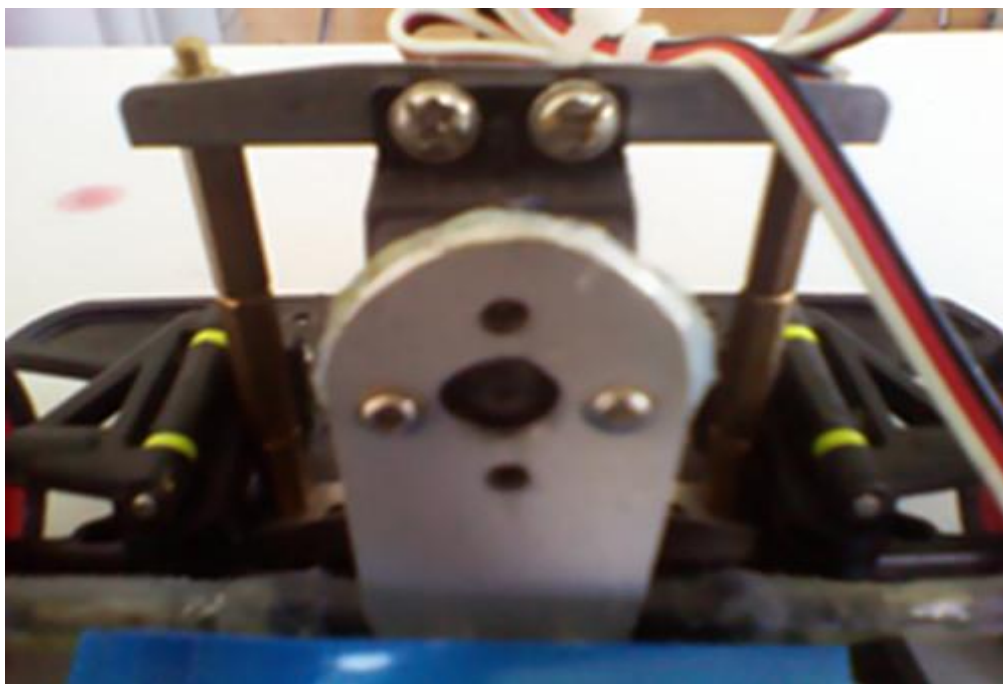


图 2.1 舵机安装效果图

2.2 前轮的调整

前轮调整的主要参数有：主销后倾、主销内倾、前束和车轮外倾。

主销角度的调整，小车的前轮调整，对真个小车的机械性能至关重要，角度调整的好有利于小车转弯，并且减小小车转弯时的抖动。

主销后倾角是指在纵向平面内主销轴线与地面垂直线之间的夹角。它在车辆转弯时会产生与车轮偏转方向相反的回正力矩，使车轮自动恢复到原来的中间位置上。经过实际测试，在调节时要把握好左边和右边打角的平衡，同时，后倾的角度也不是越大越好，基本在 1° ~ 3° 度之间，具体的在调试时由车模决定。

主销内倾角是指在横向平面内主销轴线与地面垂直线之间的夹角，它的作用也是使前轮自动回正。通过查阅资料，主销内倾的调整应该保持在一个合适的范围，“一般来说 0° ~ 8° 度范围内皆可”。在调试的过程中我们通过软件和机械的结合，来对其进行适当的微调，以寻找到这个角度的最佳值。

主销内倾和主销后倾都有使汽车转向自动回正，保持直线行驶的功能。

前轮约束是指两轮之间的后距离数值与前距离数值之差，也指前轮中心线与纵向中心线的夹角。前轮前束的作用是保证汽车的行驶性能，减少轮胎的磨损。在调整的过程中我们采用了硬软件和机械的配合调节的调节方法。

轮外倾角是指通过车轮中心的汽车横向平面与车轮平面的交线与地面垂线之间的夹角，对汽车的转向性能有直接影响，它的作用是提高前轮的转向安全性和转向操纵的轻便性。在汽车的横向平面内，轮胎呈“八”字型时称为“负外倾”，而呈现“V”字形张开时称为正外倾。轮外倾角的调整中，我们采用了负外倾的方式，通过车模在跑道上的测试，以提高前轮的转向安全性和转向操纵的轻便性。

前轮经过最终的调整，得到的效果图如图2.2:

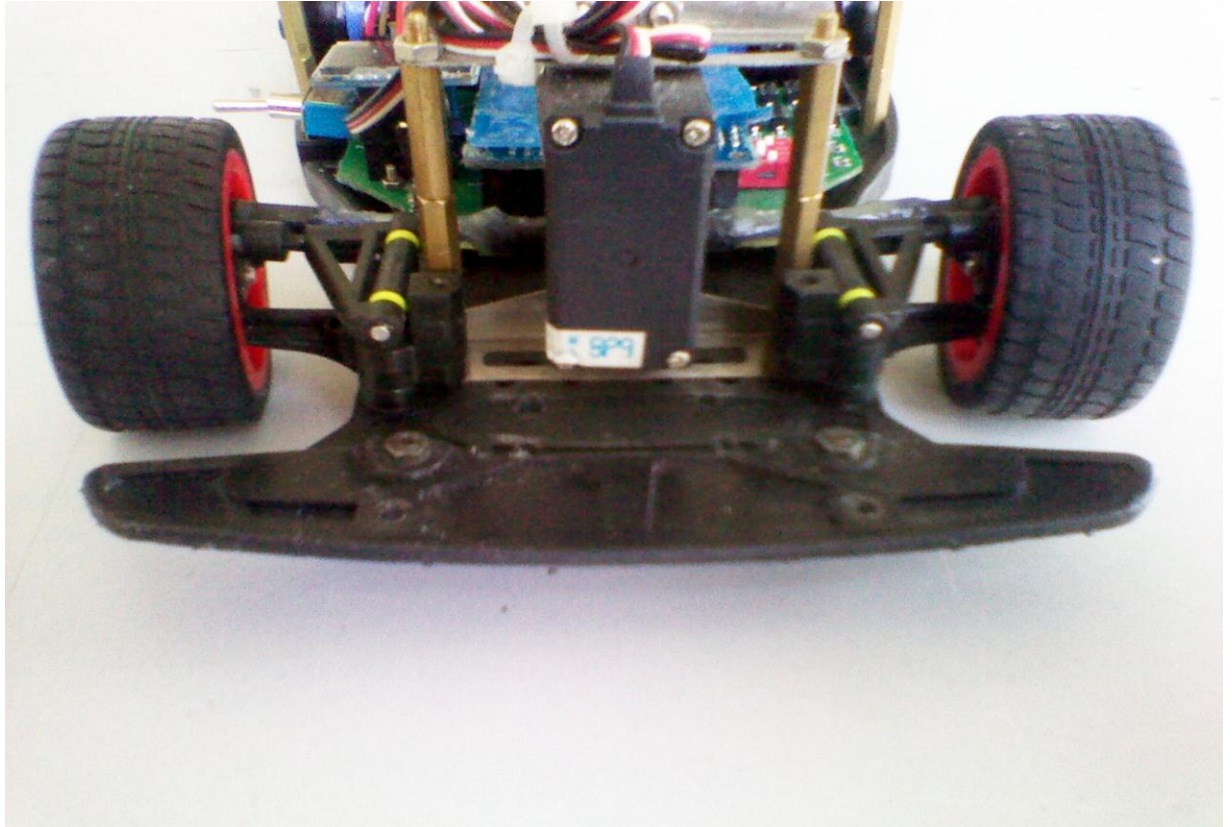


图2.2 前轮调整效果图

2.3 后轮差速的调整

后轮差速的调整对车模的性能也有着很重要的作用，差速过紧，有利于车模在直线上的加速，但是不利于车模顺利通过弯道；差速太松，有利于车模顺利通过弯道，但是在直道上，这会导致车模速度加不起来，从而增加了车模的比赛时间，这也是不利的。

因此，经过软件和机械的不断结合在一起调试，均衡直道和弯道的不同需求，我们找到了差速调节的最佳状态，后轮齿轮差速调整之后的效果图如图2.3：

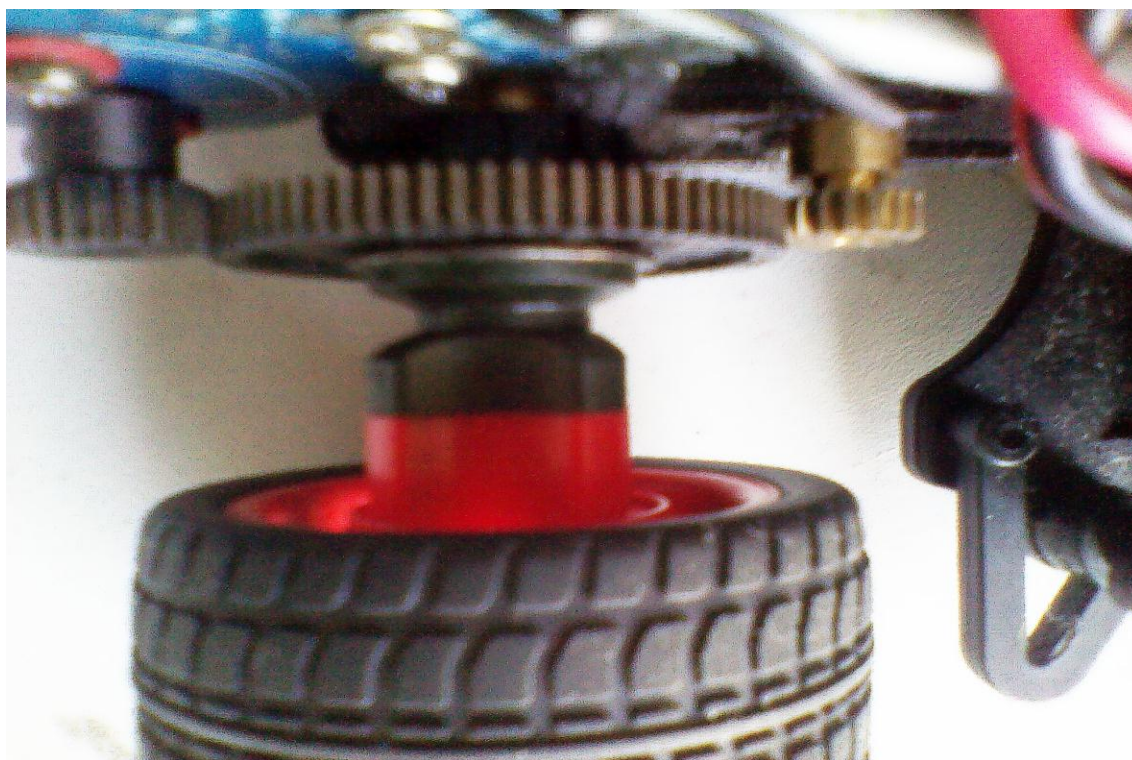


图2.3 后轮差速调整

2.4 底盘的固定

摄像头对图像采集的要求很严格，车模在跑道上运行的过程当中如果摄像头经常抖动，就会导致摄像头所采集到的赛道图像产生一定的偏差，通过摄像头反映回来的赛道信息就会不准确，从而导致误判或者其他更为严重的后果。而摄像头的抖动通常与底盘是否牢固有关，底盘不牢固，会导致车模在赛道上运行的过程当中尤其是在弯道时整个车体抖动的比较厉害，进而摄像头也会跟着抖动，所以底盘不能抖动太厉害，为尽量避免底盘的抖动，使车模能在赛道上更加快速稳定的识别赛道信息，采集更加准确的图像，让车模更加快速稳定地在赛道上奔跑。

为此我们参照组委会的相关规定，用热熔胶对底盘与后轮电机连接部分进行了固定，这样使底盘与后轮电机连接部分连接成为一个整体，从而大大地减小了车模在奔跑的过程当中摄像头的抖动，提高了车模的整体性能，固定之后的底盘如图2.4:



图2.4 底盘的固定

2.5 摄像头的选择与安装

目前市场上普遍常见的的摄像头分为 CCD摄像头和 CMOS摄像头两种，CMOS 摄像头有很多都自带有可编程控制模块和模数转换模块，与普通的存储器有非常相似之处。与CCD 摄像头相比，CMOS 摄像头不需要斩波升压来提供电源，有较强的可编程控制能力，不需要额外地进行A/D 采样，各方面都更适合于作为小车的视觉传感器。从小车需要识别的目标特征来看，CIF 分辨率甚至QCIF 分辨率的黑白摄像头足够提供路径识别和规划所需要的信息，因此我们最终选择CMOS摄像头来作为识别赛道的传感器。

为了降低整车重量和重心，我们需要严格控制CMOS摄像头的安装位置和重量，因此我们在安装时采用了碳纤维管作为安装CMOS摄像头的主桅，自行设计轻巧的铝合金夹持组件并这样可以获得最大的刚度质量比，整套装置具有很高的定位精度和刚度，同时便于摄像头位置的调整，拆卸或者是维修。摄像头的安装如图2.5.1、图2.5.2、图2.5.3所示：



图2.5.1 摄像头的固定

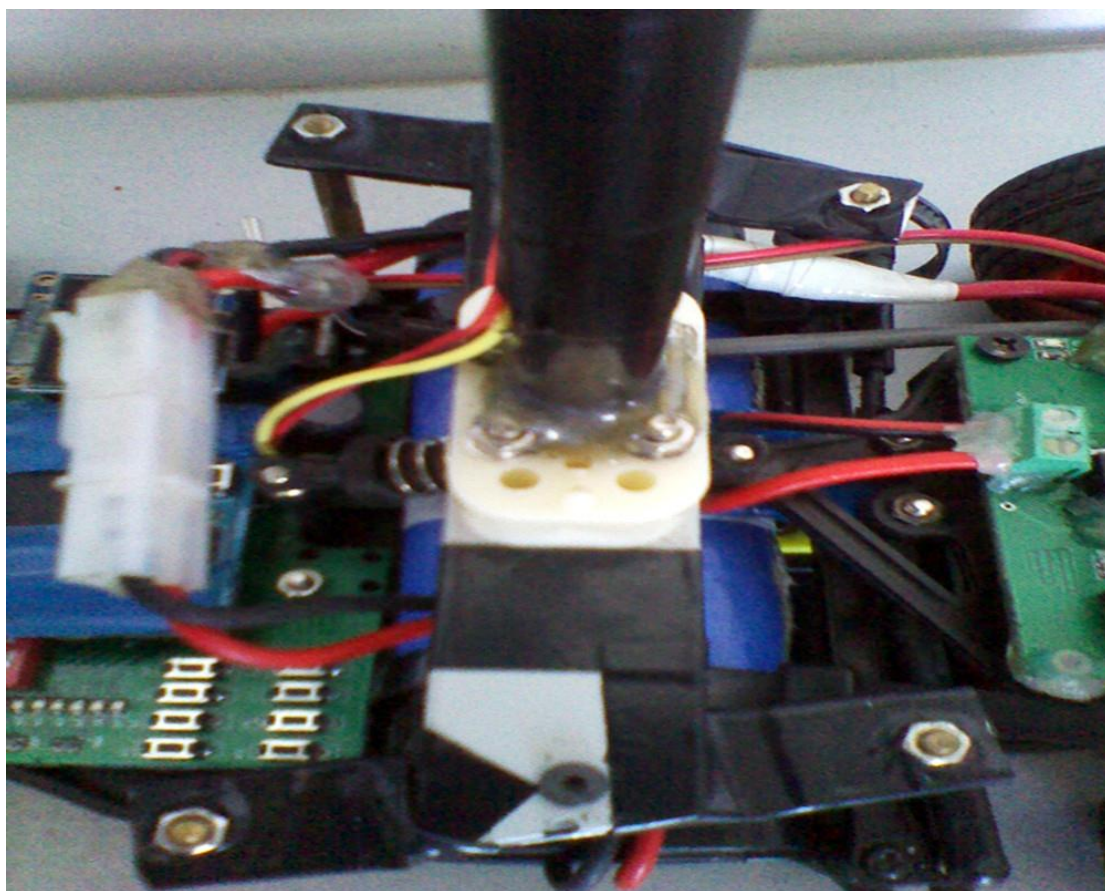


图2.5.2 摄像头底座安装



图2. 5. 3 摄像头安装效果图

2.6 编码器的安装

编码器在智能车当中的作用是很大的，通过编码器对电机转速的测定我们可以知道车模在某个时刻的速度，从而通过软件调整差值，是车模的速度达到我们想要的效果，让车模电机运转处于我们所要求的理想状态，从而形成一个速度闭环控制系统，我们使用了测速编码器去测量脉冲数，就可以得到当前电机的转速。

如图2.6，为了能让编码器更好的固定在车模上面，我们采用了比较厚比较坚固的PCB板来对测速编码器进行了安装，安装之后的效果图如图2.8：



图2.6 编码器安装

2.7 整车效果图

安装完以后的整个智能车的效果图如图 2.7:

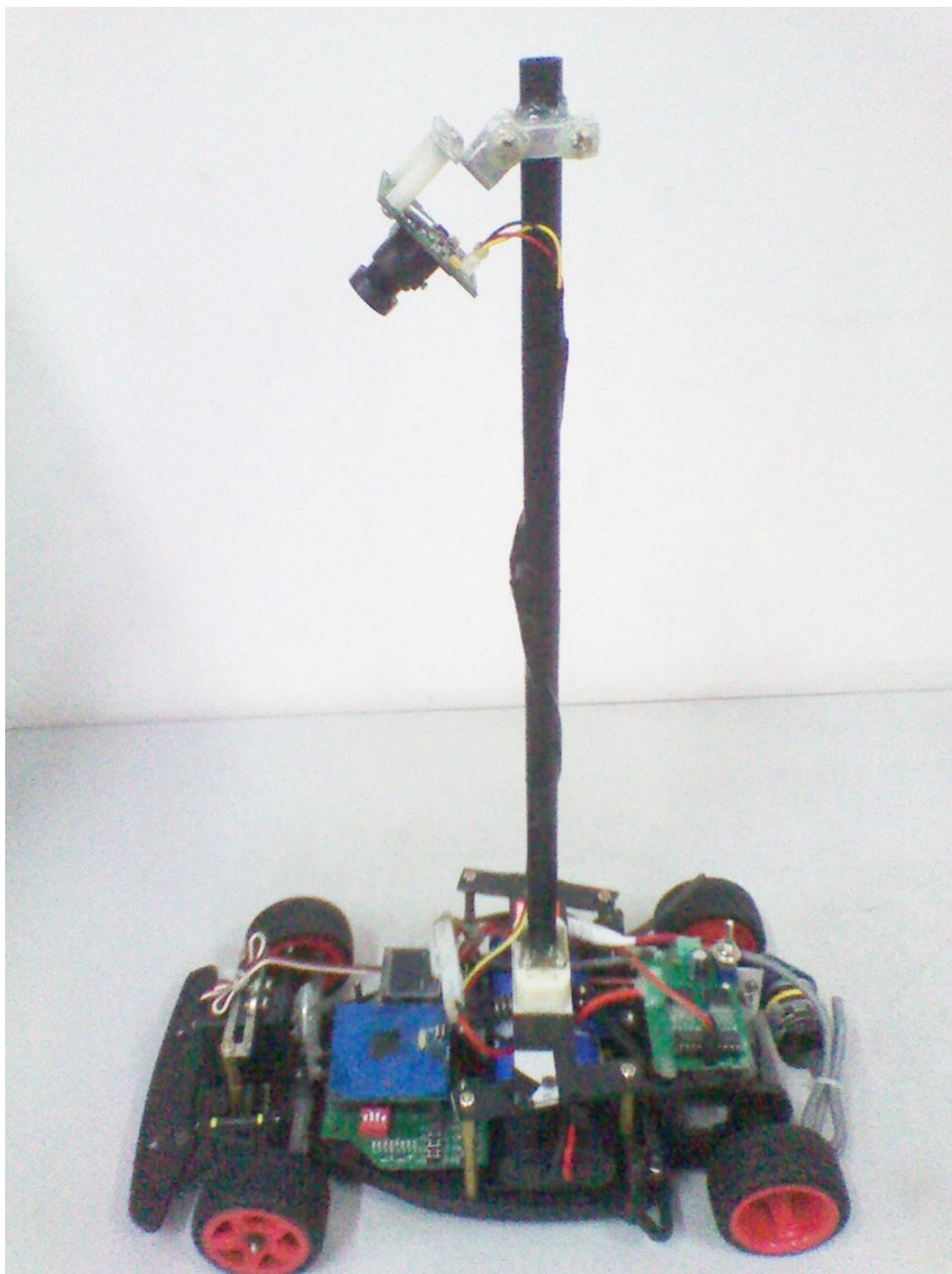
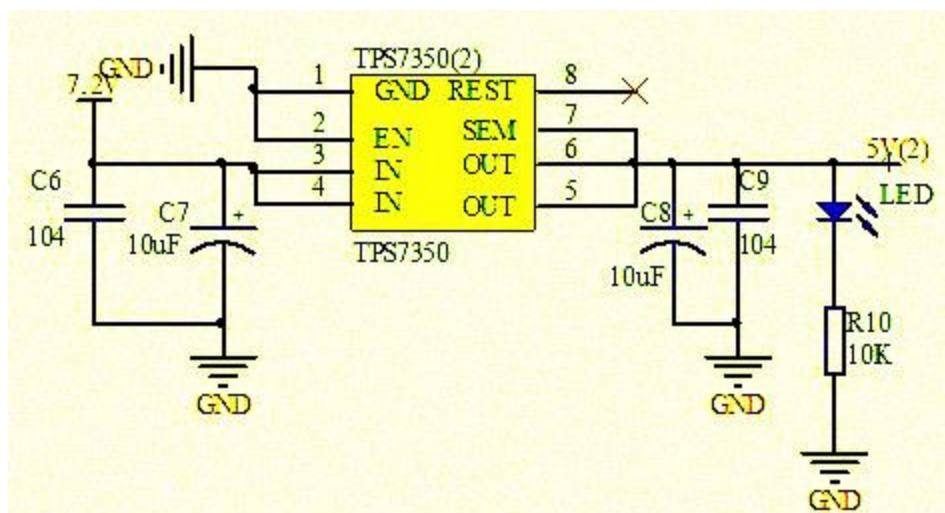


图 2.7 整车效果图

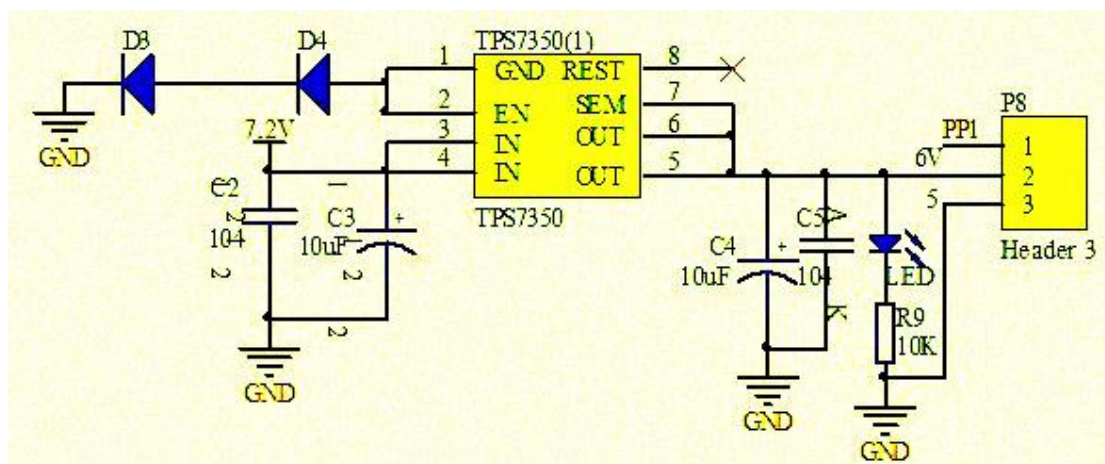
第三章 智能车硬件电路部分

3.1 电源系统

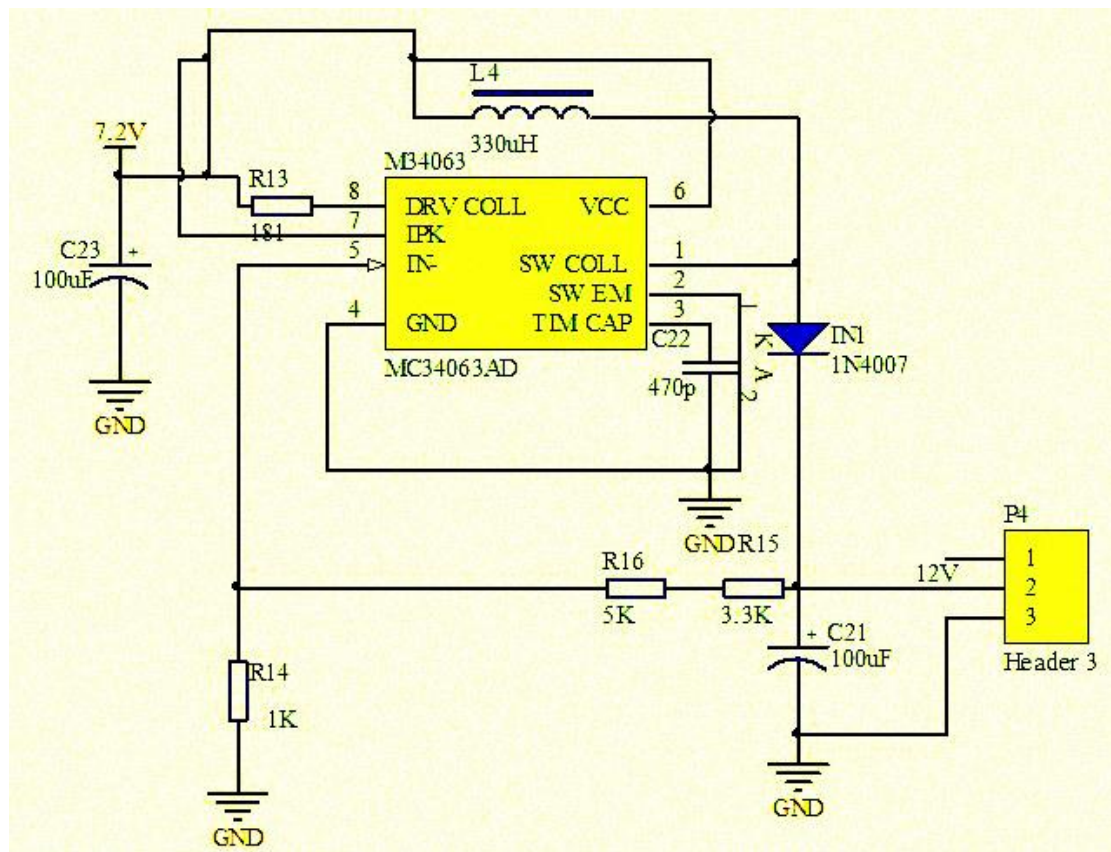
单片机供电系统考虑到系统的稳定性，我们选用 TPS7350 为单片机供电，TPS7350 芯片输出电压稳定，纹波小，不会引起单片机复位。



舵机需用 6V 电源供电，我们依次试验了 2941—ADJ 芯片，TPS7350 芯片。在使用 2941—adj 芯片产生 6V 电压时，当电源开关时，舵机会向一边方向转动，效果不好，考虑的 TPS7350 纹波小，电压稳定的特点，我们用几个二极管将 tps7350 的电压抬高到 6V，使用效果很好。

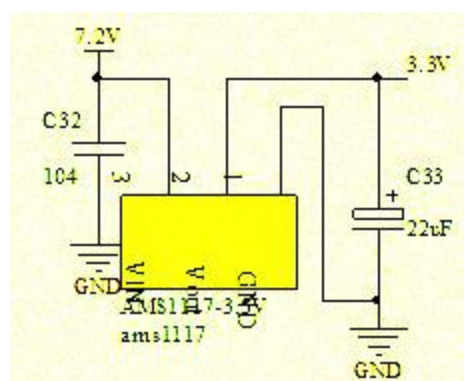


模拟摄像头 CCD 摄像头需要 12V 供电，我们采用 MC34063 集成电路产生 12V 电压；输出电压范围广，输出电压范围 3V 到 40V；MC34063 驱动电流大，最高可以达到 1.5A.

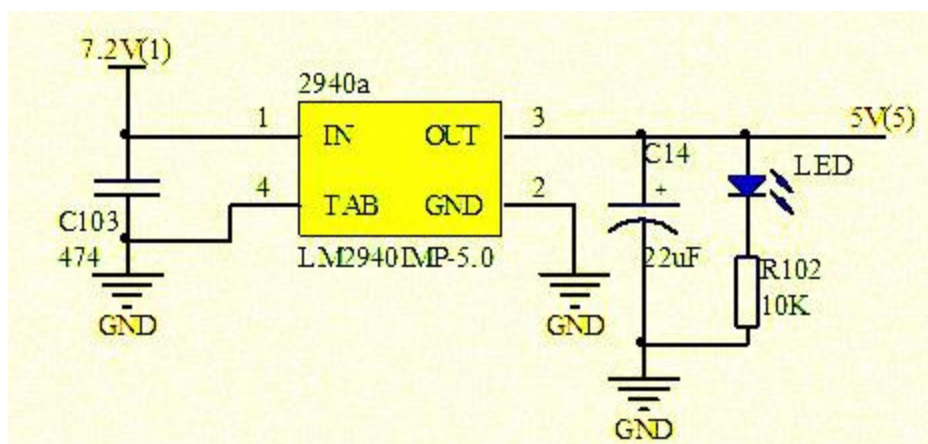


输出电压 $V_{out} = 1.25(R_{15} + R_{16}) / R_{14}$.

为了便于调试，我们采用了小液晶作为调试工具，液晶需要用 3.3V 电压供电，我们选用 AMS1117-3.3V, 效果很好。

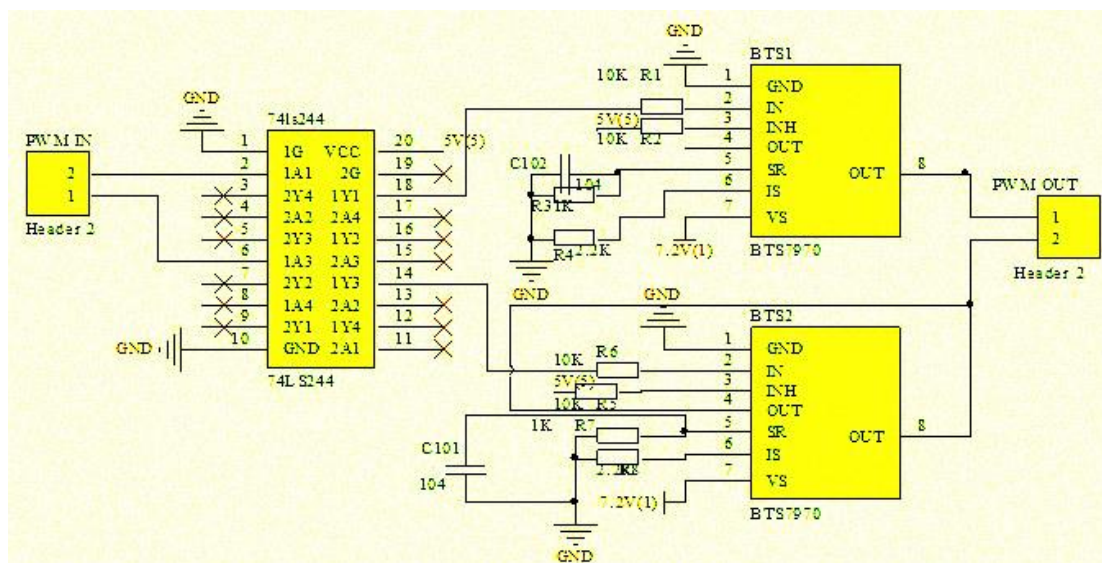


光电编码器等其它模块，考虑到驱动功率问题，我们选用 LM2940，LM2940 虽然稳压效果不如 TPS7350，但驱动功率大。



3.2 电机驱动电路设计

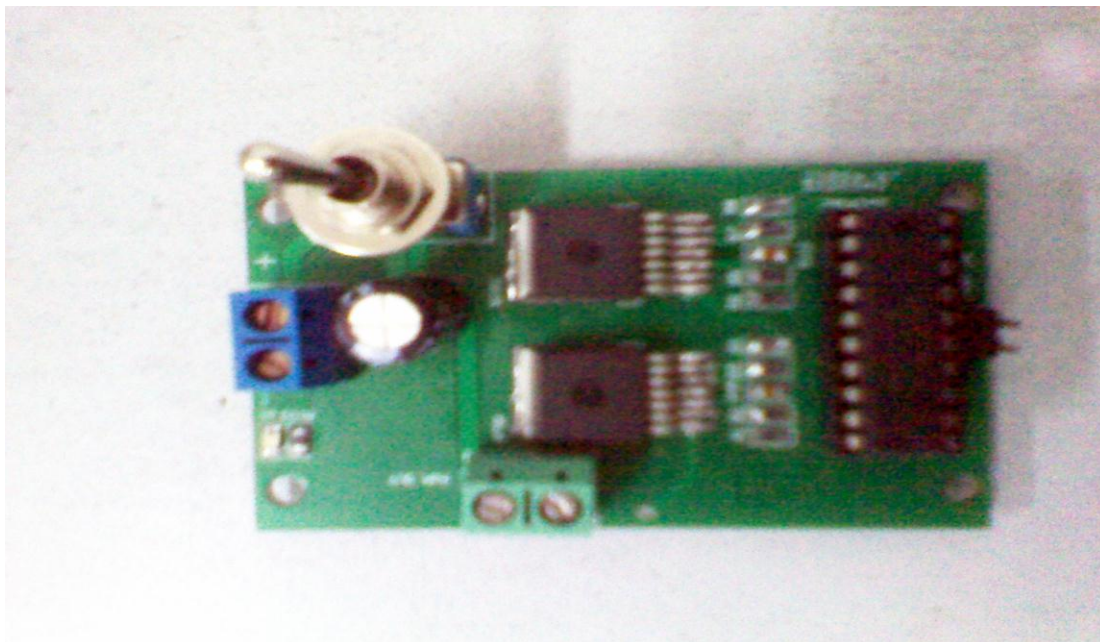
参考前几届的技术报告，我们选用英飞凌公司的 BTS7970 作为驱动芯片。BTS7970 驱动功率大，可以满足大电流的负载。



在设计驱动电路考虑到电机堵转可能产生回流，烧毁单片机，因此我们加上了 74LS244 隔离芯片。

在设计驱动电路时，电机堵转，反转过程中可能产生反电动势，对电池电压影响，我们采用在给电机供电的电源线上加二极管抑制反电动势对电池电压的影响；同时我们在 BTS7970 的使能端用 LM2940 产生 5V 单独供电，这样避免了电机堵转时，对电机控制的影响。

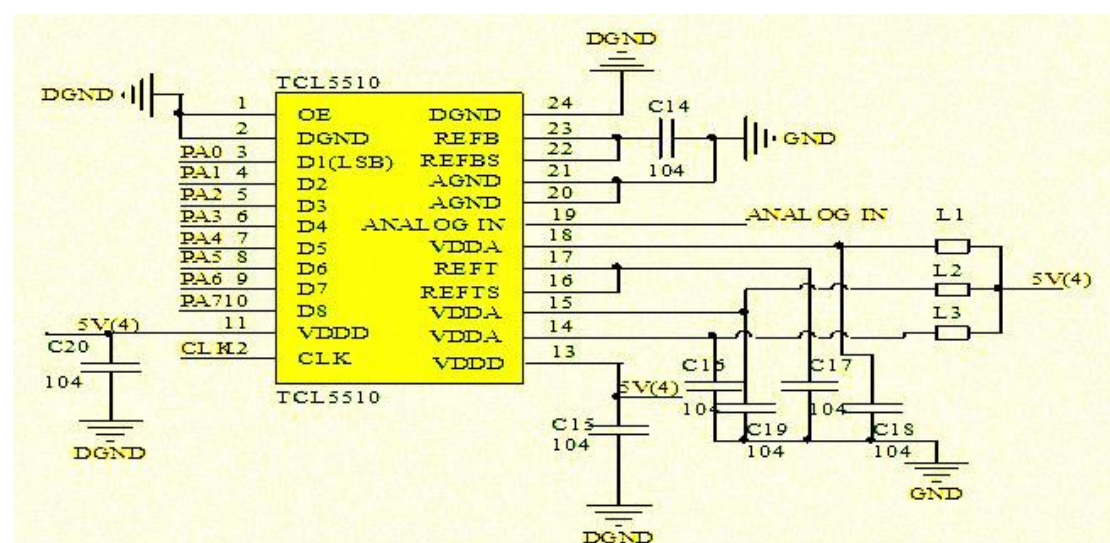
电机驱动模块如下图所示：



3.3 视频信号采集的处理

数字摄像头相比模拟 CCD 摄像头成像质量差，分辨率低，噪声多。但模拟 CCD 摄像头功耗大。我们选用 SONY CCD 模拟摄像头。用 TTL5510 处理模拟信号。

视频采集电路我们先后试验了 AD811, AD8032 芯片采用硬件二值化电路，边缘检测，但处理完的波形出现微弱的抖动，由于比赛时间紧迫，我们采用了 TCL5510 高速 A/D 转换芯片。TCL5510 是一款高速 A/D 转换芯片，最高频率可达 20M。



第四章 软件系统的设计与实现

4.1 赛道信息的提取

CCD摄像头输出的黑白图像经过TL5510高速AD转换成灰度值为0~255的灰度图像，通过软件建立二维数组把灰度图像存储起来。由于比赛背景颜色与赛道颜色一般会有明显的差异，且一般为蓝色，经试验测试，我们发现假如白色赛道的灰度值为110左右，那么两条黑线的灰度值为55左右，蓝色背景的灰度值则为75左右。可以看出，为了能提取出赛道的两个左右边沿，有两种方法：一种是边沿跳变检测法；另一种是二值化法。

(1) 边沿跳变检测法

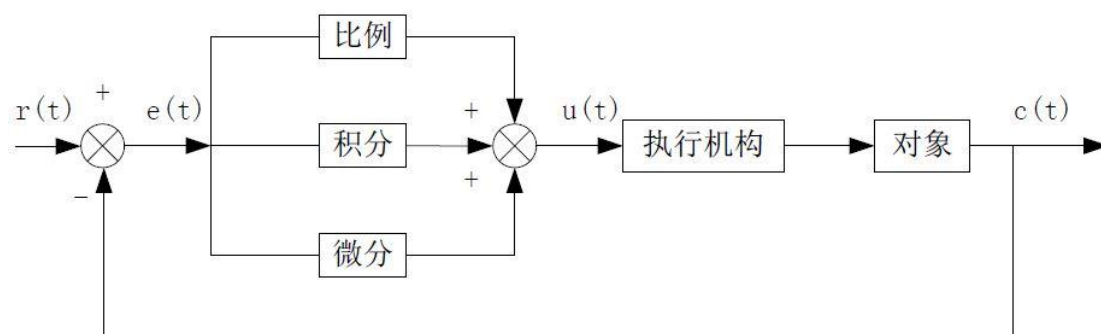
具体原理是判断两个灰度值的差值是否在限定的值域范围内，比如白色赛道的灰度值与黑线的灰度值的差值为55左右（灰度值跳变大），而黑线的灰度值与背景的灰度值的差值为20左右（灰度值跳变小），则值域范围可限定在50~60，这样通过判断跳变的程度即可求出赛道左右边沿的位置。

(2) 二值化法

二值化法可分为固定阈值法和动态阈值法。

固定阈值法：通过设定一个固定的阈值（灰度值）把灰度图像硬生生的分割成只有0和1的二值图像。动态阈值则是通过找出二维数组里每一行的最大值和最小值，求出两者的平均值即为该行的阈值。动态阈值法理论上阈值是可以随着光线明暗的变化而自动调整的。但从实践中我们发现采用动态阈值法，直道和弯道的成像较好，但在十字路口处看到的二值图像不理想，同样是白色赛道，有的地方的灰度值是110左右，有的地方则变为80~90左右，求出的求出行阈值就会比80的大，从而造成误判。最初我们采用的是动态阈值法，后来发现了上述问题，经研究比较，固定阈值二值化是能胜任比赛场地光线的。并且固定阈值法不需复杂的计算，占用单片机资源少，我们最终选择了固定阈值二值化法。

4.2 PID算法介绍



PID算法是一个闭环控制算法，P是比例控制算法，I是积分控制算法，D是微分控制算法，对闭环控制的最朴素的想法就是只有P控制算法，当然我们也可以灵活运用，根据不同需求而采用PI或者PD或者是PID算法，比例（P）、积分（I）、微分（D）各自有着不同的作用：比例（P）反应系统的基本（当前）偏差 $e(t)$ ，系数大，可以加快调节，从而减小误差，但是比例系数过大会导致系统稳定性下降，甚至是导致系统不稳定。积分（I）反映了系统的累计偏差，使系统消除稳态误差，提高无差度，因为有误差，积分调节就进行，直到误差消除为止。微分（D）反应系统偏差信号的变化率，具有预见性，能遇见偏差变化的趋势，产生超前的控制作用，在偏差形成之前，就已经被微分作用消除，因此，微分作用可以改善系统的动态性能。缺点是它对噪声干扰有放大作用，微分系数过大的话对系统的抗干扰不利。

4.3 PID参数整定

由于自动控制系统被控对象的千差万别，PID 的参数也必须随之变化，以满足系统的性能要求。因此PID 参数的确定是PID 控制中重要的部分之一，在智能车中也是一样的，PID参数的整定是一项非常繁杂的工作，参数的整定即便是熟练的工作人员也只是按照经验来一步一步不断的调试，而对于我们初学者来说就更不容易了，我们查阅了网上大量的相关资料，最终选定了以下参数整定的方法：

(1) 确定比例系数 K_p

确定比例系数 K_p 时，首先去掉PID的积分项和微分项，可以令 $T_i=0$ 、 $T_d=0$ ，使之成为纯比例调节。输入设定为系统允许输出最大值的60%~70%，比例系数 K_p 由0开始逐渐增大，直至系统出现振荡；再反过来，从此时的比例系数 K_p 逐渐减小，直至系统振荡消失。记录此时的比例系数 K_p ，设定PID的比例系数 K_p 为当前值的60%~70%。

(2) 确定积分时间常数 T_i

比例系数 K_p 确定之后，设定一个较大的积分时间常数 T_i ，然后逐渐减小 T_i ，直至系统出现振荡，然后再反过来，逐渐增大 T_i ，直至系统振荡消失。记录此时的 T_i ，设定PID的积分时间常数 T_i 为当前值的150%~180%。

(3) 确定微分时间常数 T_d

微分时间常数 T_d 一般不用设定，为0即可，此时PID调节转换为PI调节。如果需要设定，则与确定 K_p 的方法相同，取不振荡时其值的30%。

(4) 系统空载、带载联调

对PID参数进行微调，直到满足性能要求。

4.4 转向舵机控制算法

为了提高舵机的响应能力，舵机部分我们采用PD控制。其中比例部分是理想图像中心线与摄像头实际拍到的图像的中心线的偏差值，微分部分是当前的偏差值与上一次偏差值之差。舵机控制频率为50Hz。为了提高控制精度，我们将两路八位的PWM级联后供给舵机。

舵机PD控制的子程序如下

```
/*舵机PD控制*/  
  
void Futaba_PD(uchar kp, uchar kd)  
{  
    volatile static int LastErr=0,Derr=0;  
  
    Err = IMAGE_WIDTH - LeftTemp - RightTemp;  
    Derr = Err - LastErr;  
    Duoji = DUOJI_CENTER + kp*Err+ kd*Derr;  
  
    if(Duoji < DUOJI_MIN) Duoji = DUOJI_MIN;  
    if(Duoji > DUOJI_MAX) Duoji = DUOJI_MAX;  
  
    LastErr = Err;  
    futaba(Duoji);  
}
```

4.5 驱动电机控制策略

因为积分项存在滞后性，舵机部分我们也采用的是PD控制。控制频率为15KHz，为了提高控制精度，我们也采取了将两路PWM级联的方法控制电机。由于每台车不一样，具体的P、D参数需经过反复实践才能得出。

4.6 起跑线识别

起跑线的识别对参赛队伍来说至关重要。起跑线的识别是根据赛道特征来判断的。首先，起跑线一定是在直道上；其次两条起跑线的宽度及其两者间的距离在二值图像中相对稳定的。起跑线能否识别好，关键在于图像能否处理好。之前我们用动态阈值法，得到的十字路口图像很不理想，经常误判为起跑线，所以我们才改为固定阈值法。起跑线的图像下图所示：

[illegible]

起跑线识别算法如下：

首先，判断是否为直道，然后从图像中心左右等距离的地方从下往上扫，遇到0则记住该行的位置，判断两黑线是否在同一行或者相邻的两行，若成立，则求出两黑线的间距，再判断间距是否满足起跑线的要求即可。实践证明该方法切实可行。

起跑线程序代码如下：

```
//-----起跑线识别-----//

if (PTT_PTT3==0&&flag== 1) //拨码开关控制是否启用起跑线的识别
{
    if (EffectiveLine < FlagLine) //判断是否为直道
    {
        for (i=IMAGE_ROW-1;i>9;i--)
        {

pucTemp1=puca_BufferProcess+(i-1)*IMAGE_COLUMN + Center - 20;
            if (*pucTemp1 == 0) //从距中心左边20处从下往上扫
                                起跑线。如果扫到，则记住改行位置

                {
                    FlagLeft = 1;
                    LocalLeft = i-1;
                    break;
                }
        }
    }
}

for (i=IMAGE_ROW-1;i>9;i--)
{
```

```

pucTemp2=puca_BufferProcess+(i-1)*IMAGE_COLUMN + Center + 20;
    if(*pucTemp2 == 0)                //从距中心右边20处从下往上扫
                                        起跑线。如果扫到，则记住改行位置
    {
        FlagRight = 1;
        LocalRight = i-1;
        break;
    }
}

i = ( LocalLeft <= LocalRight )? LocalLeft:LocalRight;

pucTemp1 = puca_BufferProcess + (i)*IMAGE_COLUMN + Center;

for(i=4;i<20;i++) //30
{
    if(* (pucTemp1+i)==0    &&    * (pucTemp1+i+1)==0    &&
        * (pucTemp1+i+2)==0    &&    * (pucTemp1+i+3)==0    &&
        * (pucTemp1+i+4)==0    &&    * (pucTemp1+i+5)==0    &&
        * (pucTemp1+i+6)==0    &&    * (pucTemp1+i+7)==0    &&
        * (pucTemp1+i+8)==0    &&    * (pucTemp1+i+9)==0    &&
        * (pucTemp1+i+10)==0    )
    {

        BlackRight = Center+i-1;    //求出左边黑线右端点的位置
        break;
    }
}

for(i=4;i<20;i++) //30

```

```

{
    if(*(pucTemp1-i)==0    &&    *(pucTemp1-i-1)==0    &&
        *(pucTemp1-i-2)==0    &&    *(pucTemp1-i-3)==0    &&
        *(pucTemp1-i-4)==0    &&    *(pucTemp1-i-5)==0    &&
        *(pucTemp1-i-6)==0    &&    *(pucTemp1-i-7)==0    &&
        *(pucTemp1-i-8)==0    &&    *(pucTemp1-i-9)==0    &&
        *(pucTemp1-i-10)==0    )
    {
        BlackLeft = Center-i+1;    //求出右边黑线左端点的位置
        break;
    }
}

BlackWidth=BlackRight-BlackLeft; //求出求出两黑线的间距

if(FlagLeft==1&&FlagRight==1&&BlackWidth>14&& BlackWidth <30)
//17    20
{
    if( absolute(LocalLeft - LocalRight) < 3 )
    {
        wait(250);
        PWME = 0X02;    //若条件成立，关闭电机
        PORTB = 0X00;

        //flag++;
    }
}
else
{
    FlagLeft = FlagRight = 0;
}
}
}

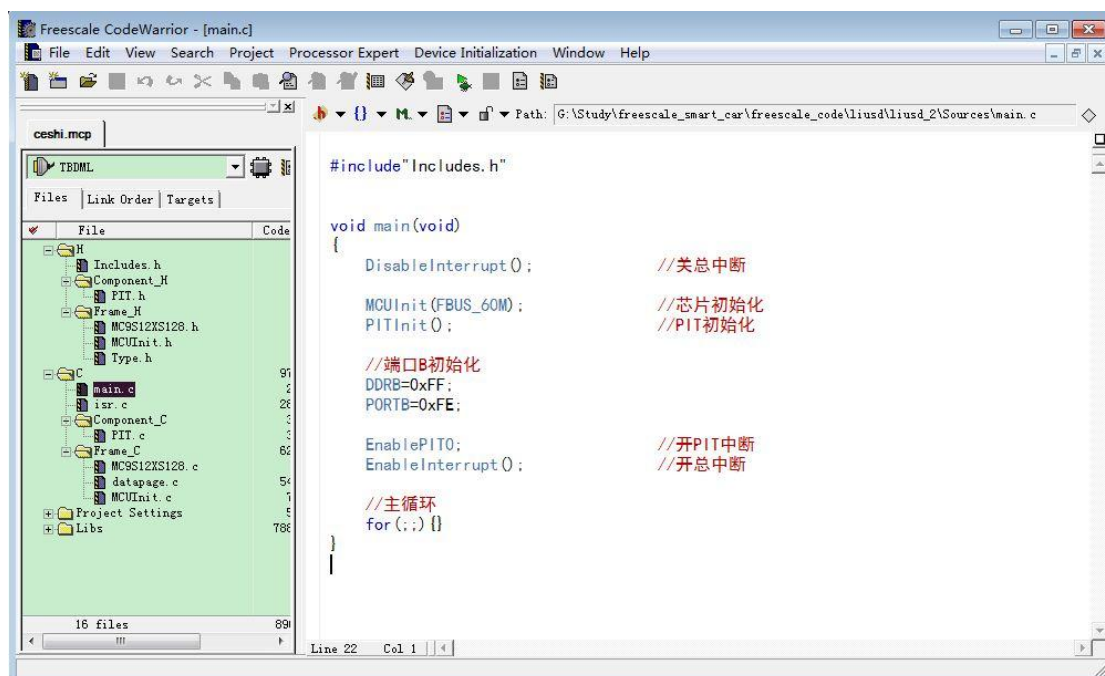
```

第五章 智能车软件开发平台

5.1 CodeWarrior简介

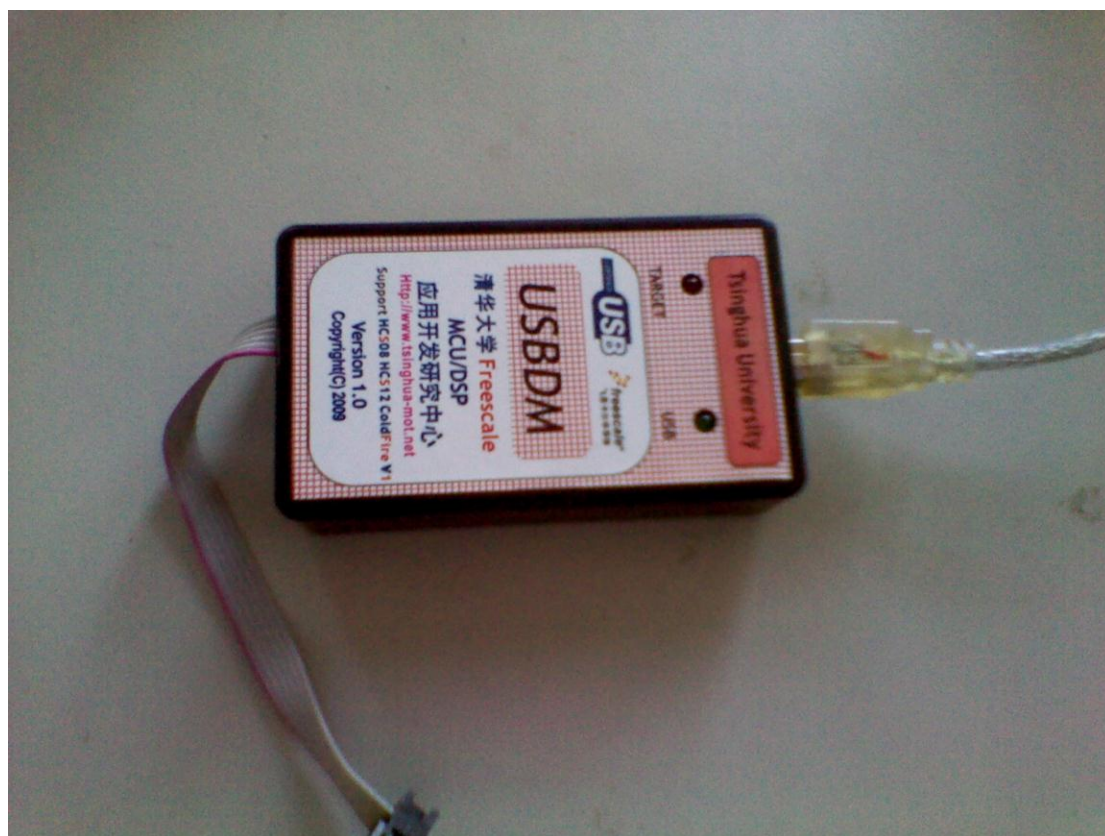
CodeWarrior 是 Metrowerks 公司研发的专门面向 Freescale 所有 MCU 与 DSP 嵌入式应用开发的软件工具，是 Freescale 向用户推荐的产品。CodeWarrior 分为3个版本：特别版（Special Edition）、标准版和专业版。特别版是免费的、用于教学目的，对生成的代码量有一定限制，C 代码不超过32KB，对于工程包含文件数目限制在30个以内，标准版和专业版则没有这种限制。

软件开发平台我们用的版本是CW_HC12_v5.1_SPECIAL。该软件界面简洁、友好但功能强大。CodeWarrior 开发环境如下图所示：



5.2 调试工具 USBDM 简介

新款BDM (Background Debug Mode) 调试器是专门为Freescale 公司的基于HCS08, HCS12 和 ColdFire V1 内核的处理器设计的, 用USB 口接PC 机, 命名为USBDM。USBDM 是清华Freescale MCU&DSP 应用开发中心在研究了Freescale 公司因特网上的技术论坛中的USBDM 开源软件做了些修改, 并重新设计了硬件, 生产的调试器。对三类 Freescale 单片机的试用证明, 新款BDM 调试器能适应CodeWarrior 软件调试环境, 且在和PC 机通讯速率上比老款BDM 快很多。



USBDM for S12

第六章 总结

从去年开始,到现在半年多的时间里,我们一直都在为飞思卡尔智能车忙碌着,从一开始的知识储备,到后来的动手实践,计划一步一步实现,到参加比赛的整个过程单中,我们收获的不仅仅是一辆小车,更收获了更多的知识。从一开始的什么都不懂,到现在的硬件、软件、机械等相关方面知识的掌握与了解,大大的提高了我们的知识面和动手能力,同时也增强了我们的团队协作能力,收获了友谊,在这里同时也感谢学校和我们的老师,正因为背后有他们的大力的支持,我们才能从众多参赛队伍当中脱颖而出,为学校和老师争光。

硬件、软件以及机械的有机结合,团队良好的协作能力,让我们最终收获了一个良好的结果,飞思卡尔智能车为广大的大学生学习提供了另一个良好的平台,让我们从整个过程中学会了画板、焊电路、软件调试、机械设计分析与安装等一大批有用的知识,对我们广大的大学生提高自身的综合素质起到了非常大的帮助。同时,在这里也祝愿飞思卡尔全国大学生智能车竞赛也越办越好。

参考文献

- 【1】 卓晴,黄开胜,邵贝贝,学做智能车[M].北京:北京航空航天大学出版社,2007
- 【2】 李庆常·模拟电子技术基础·北京:机械工业出版社,2007
- 【3】 王美玲·数字电子技术基础·北京:机械工业出版社,2010
- 【4】 谭浩强·C程序设计·北京:清华大学出版社,2001
- 【5】 王宜怀,刘晓升等。《嵌入式系统》 北京航空航天大学出版社 2008
- 【6】 马忠梅,籍顺心,张凯,马岩. 单片机C语言的应用程序设计(第四版. 北京航空航天大学出版社 .2007
- 【7】 王宜怀,曹金华. 嵌入式系统设计实践——基于飞思卡尔S12X微控制器, 北京航空航天大学出版社. 2011
- 【8】 张阳,吴晔,滕勤等. MC9S12XS单片机原理及嵌入式系统开发. 电子工业出版社. 2011
- 【9】 张伟等. Protel DXP高级应用[M]. 北京. 人民邮电出版社. 2002
- 【10】 陈正冲 《C语言深度解剖》 北京航空航天大学出版社 2010

附录

部分源程序代码：

```
#include "includes.h"

volatile uchar heixian=0; // 采集行的计数
volatile uchar g_car_show_yn = 0; // 是否可以显示数据

#define IMAGE_ROW 32 // < v9008 10
#define IMAGE_COLUMN 115 // 87//110-23 // 86//87
// #define IMAGE_ROW_DIS 2//

uchar uca_Buffer1[IMAGE_ROW][IMAGE_COLUMN]; // 图像数据缓存
uchar uca_Buffer2[IMAGE_ROW][IMAGE_COLUMN];

uchar ThresHold[IMAGE_ROW];
uchar *threshold = ThresHold;

unsigned int get_n[]={ 107, 108, 109, 110, 113, 114, 137, 139, 143, 145,
                     147, 151, 154, 157, 161, 164, 168, 174, 178, 184,
                     189, 193, 197, 200, 204, 208, 212, 217, 242, 246,
                     250, 254
                     };
```

```
uchar *puca_BufferSample=&uca_Buffer1[0][0]; //指向当前采集数据采样缓存首地址的指针
```

```
uchar *puca_BufferProcess=&uca_Buffer2[0][0]; //指向当前处理数据采样缓存首地址的指针
```

```
uchar *puca_BufferTemp=&uca_Buffer1[0][0]; //用于图像采集和处理交换缓存
```

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

```
#define IMAGE_CENTER 57
#define IMAGE_WIDTH 114
#define MinEffectiveLine 1
#define FlagLine 8
#define SpeedLine 5
#define DUOJI_CENTER 2700 //2754
#define DUOJI_MAX 3040
#define DUOJI_MIN 2360
#define KP 6
#define KD 20
```

```
uint chesudu;
```

```
volatile uint pulse;
```

```
uchar i, j, Kp, Kd, speed, sudu, sudu1;
```

```
uchar Line, EffectiveLine, LeftTemp, RightTemp, UseLine;
```

```
//uchar , flag1 = 0, flag2 = 0, LocalLeft=100, LocalRight=200,
LocationLeft = 100, LocationRight =200;

uchar Center, FlagLeft=0, FlagRight=0, LocalLeft=100, LocalRight=150,
BlackLeft=100, BlackRight=100, BlackWidth=100;

uchar flag=0;
uchar Piancha;
//char num;
uint Duoji;

volatile static int Err=0;

/*****函数声明*****/

void Photo_Make01(void);
//void Threshold(void);
void OledDisplay(void);
void Keyscan(void);

/*****舵机PD控制*****/

void Futaba_PD(uchar kp, uchar kd)
{
    volatile static int LastErr=0, Derr=0;
```

```
Err = IMAGE_WIDTH - LeftTemp - RightTemp;
Derr = Err - LastErr;
Duoji = DUOJI_CENTER + kp*Err+ kd*Derr;

if(Duoji < DUOJI_MIN) Duoji = DUOJI_MIN;
if(Duoji > DUOJI_MAX) Duoji = DUOJI_MAX;

LastError = Err;
futaba(Duoji);
}
```

```
/***** system初始化 *****/
```

```
void Init(void)
{
    DisableInterrupts;

    pll_init();
    pwm_init();
    PIT_Init();
    TIM_init();
    InCapture_init();
    LCD_Init();
    DDRA = 0X00;
    DDRB = 0XFF;
    PORTB = 0X55;
    DDRT = 0X00;
    DDRH = 0X00;
```

```
sudu = 55;      //      55      //42      42      40      33
sudu1 = 80;     //      80      //45      47      49      40      33
```

```
LCD_Print(20, 2, "东北林业大学");
LCD_Print(35, 4, "赛 豹 号");
Dly_ms(2000);
LCD_CLS();
LCD_P6x8Str(0, 1, "EffectLine = ");
LCD_P6x8Str(0, 2, "sudu      = ");
LCD_P6x8Str(0, 3, "sudu1     = ");
LCD_P6x8Str(0, 4, "Center      = ");
LCD_P6x8Str(0, 5, "BlackLeft   = ");
LCD_P6x8Str(0, 6, "BlackRight  = ");
LCD_P6x8Str(0, 7, "BlackWidth  = ");
```

```
// chesudu = 0;
```

```
EnableInterrupts;
```

```
while(!PTT_PTT5)
{
    Keyscan();
    OledDisplay();
}
```

```
        wait(3000);
    }

    /*
    ****
    *
    *   主函数
    *
    ****
    */
void main(void)
{

    uchar *pucTemp1, *pucTemp2, *pucTemp3, *pucTemp4;    //

    Init();

    g_car_show_yn = 0;    //等待下场数据

    PITCFLMT_PITE = 1;    // PIT is enabled
    chesudu = 50;
    for(;;)
    {
        _FEED_COP();
    }
}
```



```

/*      if(PTT_PTT3 == 1)
{
}          */
while(g_car_show_yn == 1)
{

    g_car_show_yn = 0;
    //      Threshold();
    Photo_Make01();          //图像2值化!

    for(Line=IMAGE_ROW-1; Line>0; Line--)
    {
        pucTemp1 =
puca_BufferProcess+(Line)*IMAGE_COLUMN+IMAGE_CENTER;
        pucTemp2 =
puca_BufferProcess+(Line-1)*IMAGE_COLUMN+IMAGE_CENTER;
        pucTemp3 =
puca_BufferProcess+(Line-2)*IMAGE_COLUMN+IMAGE_CENTER;
        pucTemp4 =
puca_BufferProcess+(Line-3)*IMAGE_COLUMN+IMAGE_CENTER;

        if(*(pucTemp1) == 0 && *(pucTemp2) == 0 &&
*(pucTemp3) == 0 && *(pucTemp4) == 0)
        {
            EffectiveLine=Line+1;
            if(EffectiveLine < MinEffectiveLine)
            {
                EffectiveLine = MinEffectiveLine;
            }
        }
    }
}

```

```
        else if(EffectiveLine > IMAGE_ROW-1)
        {
            EffectiveLine = IMAGE_ROW-1;
        }
        break;
    }

}

//-----
-----//

Center = (uchar)( (LeftTemp + RightTemp)/2 );

//      Piancha = absolute(Center - IMAGE_CENTER);

if(EffectiveLine < SpeedLine) //4
{
    UseLine = 16; //PORTB = 0XFE;
    Kp = 3; Kd = 7; //3 8 or 3 7
                                kd越大转弯越灵敏
//      if(Piancha<9)
//      {
//          speed = 42;
```

```
        //      }

        //      else
            speed = sudu1;
                        //55  60  56

    }
    else if(EffectiveLine < FlagLine)           //7
    {
        UseLine = 16; //PORTB = 0XF0;
        Kp = 4; Kd =16;                        //4   8   418
        speed = sudu;                          //49

    }
    else if(EffectiveLine < 17)
    {
        UseLine = 16; //PORTB = 0X0F;
        Kp = 6; Kd = 20;
        speed = sudu;
    }
    else
    {
        UseLine = EffectiveLine; //PORTB = 0XFF;
        if(UseLine > IMAGE_ROW-1)
        {
            UseLine = IMAGE_ROW-1;
        }
    }
}
```

```

        pucTemp1 =
        puca_BufferProcess+(UseLine)*IMAGE_COLUMN;

        for(i=0;i<IMAGE_COLUMN;i++)
        {
            if(*(pucTemp1+i)==1 &&
*(pucTemp1+i+1)==1 && *(pucTemp1+i+2)==1 && *(pucTemp1+i+3)==1 &&
*(pucTemp1+i+4)==1 && *(pucTemp1+i+5)==1 && *(pucTemp1+i+6)==1 &&
*(pucTemp1+i+7)==1 && *(pucTemp1+i+8)==1 && *(pucTemp1+i+9)==1 &&
*(pucTemp1+i+10)==1 && *(pucTemp1+i+11)==1 && *(pucTemp1+i+12)==1 &&
*(pucTemp1+i+13)==1 && *(pucTemp1+i+14)==1 && *(pucTemp1+i+15)==1 &&
*(pucTemp1+i+16)==1 && *(pucTemp1+i+17)==1 && *(pucTemp1+i+18)==1 &&
*(pucTemp1+i+19)==1 && *(pucTemp1+i+20)==1 && *(pucTemp1+i+21)==1 &&
*(pucTemp1+i+22)==1 && *(pucTemp1+i+23)==1 && *(pucTemp1+i+24)==1 &&
*(pucTemp1+i+25)==1 && *(pucTemp1+i+26)==1 && *(pucTemp1+i+27)==1 &&
*(pucTemp1+i+28)==1 && *(pucTemp1+i+29)==1 && *(pucTemp1+i+30)==1)
            {
                LeftTemp = i;
                break;
            }
        }

        for(i=IMAGE_COLUMN-1;i>0;i--)
        {
            if(*(pucTemp1+i)==1 &&
*(pucTemp1+i-1)==1 && *(pucTemp1+i-2)==1 && *(pucTemp1+i-3)==1 &&
*(pucTemp1+i-4)==1 && *(pucTemp1+i-5)==1 && *(pucTemp1+i-6)==1 &&

```

```

*(pucTemp1+i-7)==1 && *(pucTemp1+i-8)==1 && *(pucTemp1+i-9)==1 &&
*(pucTemp1+i-10)==1 && *(pucTemp1+i-11)==1 && *(pucTemp1+i-12)==1 &&
*(pucTemp1+i-13)==1 && *(pucTemp1+i-14)==1 && *(pucTemp1+i-15)==1 &&
*(pucTemp1+i-16)==1 && *(pucTemp1+i-17)==1 && *(pucTemp1+i-18)==1 &&
*(pucTemp1+i-19)==1 && *(pucTemp1+i-20)==1 && *(pucTemp1+i-21)==1 &&
*(pucTemp1+i-22)==1 && *(pucTemp1+i-23)==1 && *(pucTemp1+i-24)==1 &&
*(pucTemp1+i-25)==1 && *(pucTemp1+i-26)==1 && *(pucTemp1+i-27)==1 &&
*(pucTemp1+i-28)==1 && *(pucTemp1+i-29)==1 && *(pucTemp1+i-30)==1)
    {
        RightTemp = i;
        break;
    }
}

```

//-----起跑线识别-----//

```

if(PTT_PTT3 == 0 && flag == 1)
{

    if(EffectiveLine < FlagLine)
    {

        for(i=IMAGE_ROW-1;i>9;i--)
        {
            pucTemp1 = puca_BufferProcess + (i-1)*IMAGE_COLUMN + Center - 20;

```

```
        if(*pucTemp1 == 0)
        {
            FlagLeft = 1;
            LocalLeft = i-1;
            break;
        }
    }

    for(i=IMAGE_ROW-1;i>9;i--)
    {
        pucTemp2 = puca_BufferProcess + (i-1)*IMAGE_COLUMN + Center + 20;
        if(*pucTemp2 == 0)
        {
            FlagRight = 1;
            LocalRight = i-1;
            break;
        }
    }

    i = ( LocalLeft <= LocalRight )? LocalLeft:LocalRight;
    pucTemp1 = puca_BufferProcess + (i)*IMAGE_COLUMN + Center;

    for(i=4;i<20;i++)    //30
    {

        if(*(pucTemp1+i)==0 && *(pucTemp1+i+1)==0 && *(pucTemp1+i+2)==0 &&
```

```

*(pucTemp1+i+3)==0 && *(pucTemp1+i+4)==0 && *(pucTemp1+i+5)==0 &&
*(pucTemp1+i+6)==0 && *(pucTemp1+i+7)==0 && *(pucTemp1+i+8)==0 &&
*(pucTemp1+i+9)==0 && *(pucTemp1+i+10)==0)

                                {

                                BlackRight =

Center+i-1;

                                break;

                                }

                                }

                                for(i=4;i<20;i++)

//30

                                {

if(*(pucTemp1-i)==0 && *(pucTemp1-i-1)==0 && *(pucTemp1-i-2)==0 &&
*(pucTemp1-i-3)==0 && *(pucTemp1-i-4)==0 && *(pucTemp1-i-5)==0 &&
*(pucTemp1-i-6)==0 && *(pucTemp1-i-7)==0 && *(pucTemp1-i-8)==0 &&
*(pucTemp1-i-9)==0 && *(pucTemp1-i-10)==0)

                                {

                                BlackLeft =

Center-i+1;

                                break;

                                }

                                }

                                BlackWidth =

BlackRight - BlackLeft;

                                if(FlagLeft == 1 &&
FlagRight == 1 && BlackWidth > 14 && BlackWidth <30) //17    20

```

```

{

if( absolute(LocalLeft - LocalRight) < 3 )

{
//

PWMDTY23 =

wait(250);
PWME = 0X02;
PORTB = 0X00;
//      flag++;

}
}
else
{
FlagLeft =

FlagRight = 0;

}

}

}
```

```

//-----速度方向控制
-----//
```



```
switch(UseLine)
{

/*      case 12:{

                                Futaba_PD(11,KD);
                                chesudu = 39;break;

                                }

      case 13:{

                                Futaba_PD(11,KD);
                                chesudu = 39;break;

                                }

      case 14:{

                                Futaba_PD(11,KD);
                                chesudu = 39;break;

                                }

      case 15:{

                                Futaba_PD(11,KD);
                                chesudu = 39;break;

                                }

*/

      case 16:{
```

```

        Futaba_PD(Kp, Kd) ;
        chesudu = speed;break;
    }

    case 17:{

        Futaba_PD(KP, KD) ;
        chesudu = sudu;break;

    }

    case 18:{

        Futaba_PD(KP, KD) ;
        chesudu = sudu;break;

    }

    case 19:{

        Futaba_PD(KP, KD) ;
        chesudu = sudu;break;

    }

    case 20:{

        Futaba_PD(KP, KD) ;
        chesudu = sudu;break;

    }

    case 21:{

        Futaba_PD(KP, KD) ;
        chesudu = sudu;break;

    }

```

```
case 22: {  
  
    Futaba_PD(KP, KD) ;  
    chesudu = sudu; break;  
  
}  
  
case 23: {  
  
    Futaba_PD(KP, KD) ;  
    chesudu = sudu; break;  
  
}  
  
case 24: {  
  
    Futaba_PD(7, KD) ;  
    chesudu = sudu; break;  
  
}  
  
case 25: {  
  
    Futaba_PD(7, KD) ;  
    chesudu = sudu; break;  
  
}  
  
case 26: {  
  
    Futaba_PD(7, KD) ;  
    chesudu = sudu; break;  
  
}  
  
case 27: {  
  
    Futaba_PD(7, KD) ;  
    chesudu = sudu; break;  
  
}
```

```
case 28:{  
    if((IMAGE_CENTER-LeftTemp) <  
    (RightTemp-IMAGE_CENTER))    //右转  
    {  
        futaba(DUOJI_MIN);  
    }  
    else  
    //左转  
    {  
        futaba(DUOJI_MAX);  
    }  
    chesudu = sudu;break;  
}  
  
case 29:{  
    if((IMAGE_CENTER-LeftTemp) <  
    (RightTemp-IMAGE_CENTER))    //右转  
    {  
        futaba(DUOJI_MIN);  
    }  
    else  
    //左转  
    {  
        futaba(DUOJI_MAX);  
    }  
    chesudu = sudu;break;  
}
```

```
case 30:{
    if((IMAGE_CENTER-LeftTemp) <
(RightTemp-IMAGE_CENTER))    //右转
    {
        futaba(DUOJI_MIN);
    }
    else
//左转
    {
        futaba(DUOJI_MAX);
    }
    chesudu = sudu;break;
}

case 31:{
    if((IMAGE_CENTER-LeftTemp) <
(RightTemp-IMAGE_CENTER))    //右转
    {
        futaba(DUOJI_MIN);
    }
    else
//左转
    {
        futaba(DUOJI_MAX);
    }
    chesudu = sudu;break;
}

default: break;
```

```

    }

    OledDisplay();

}

}

}

//-----拨码开关和键盘-----//
void Keyscan(void)
{
    if(PTH_PTH0 == 0)
    {
        delay(1000);
        if(PTH_PTH0 == 0)
        {
            sudu--;
        }
        while(!PTH_PTH0);
        delay(1000);
        while(!PTH_PTH0);
    }
    if(PTH_PTH3 == 0)
    {
        //15
        61

21  97

        delay(1000);
    }
}

```

```

        if(PTH_PTH3 == 0)
        {
            sudu++;
        }
        while(!PTH_PTH3);
        delay(1000);
        while(!PTH_PTH3);
    }

    if(PTH_PTH4 == 0)
    {
        //15        61

21  97

        delay(1000);
        if(PTH_PTH4 == 0)
        {
            sudu1--;
        }
        while(!PTH_PTH4);
        delay(1000);
        while(!PTH_PTH4);
    }

    if(PTH_PTH7 == 0)
    {
        //15        61

21  97

        delay(1000);
        if(PTH_PTH7 == 0)
        {
            sudu1++;

```

```
        }
        while(!PTH_PTH7);
        delay(1000);
        while(!PTH_PTH7);
    }

}

//-----OLED显示-----//
void OledDisplay(void)
{
    //    uchar iii,jjj;
    if(PTT_PTT2 == 0)
    {

        LCD_P6x8one(90, 1, EffectiveLine);
        LCD_P6x8one(90, 2, sudu);

        LCD_P6x8one(90, 3, sudu1);
        LCD_P6x8one(90, 4, Center);
        //    iii = Duoji/100;
        //    jjj = Duoji%100;
        //    LCD_P6x8one(90, 5, iii);
        //    LCD_P6x8one(90, 6, jjj);
        LCD_P6x8one(90, 5, BlackLeft);
        LCD_P6x8one(90, 6, BlackRight);
    }
}
```



```

        LCD_P6x8one(90, 7, BlackWidth);
    }

}

//-----动态阈值-----//
/*void Threshold(void)
{
    char ii;
    unsigned char jj;
    unsigned char *temp, *maxTemp, *minTemp;

    for(ii=0;ii<IMAGE_ROW;ii++)
    {
        maxTemp = minTemp = puca_BufferProcess+ii*IMAGE_COLUMN;
//?

        for(jj=1; jj<IMAGE_COLUMN; jj++)
        {
            temp = puca_BufferProcess+ii*IMAGE_COLUMN+jj;
            if(*temp > *maxTemp && *temp < 200)  maxTemp = temp;
            if(*temp < *minTemp)  minTemp = temp;
        }

        *(threshold + ii) = (uchar)((*maxTemp + *minTemp)/2);
    }
}

```

```
        for(ii=IMAGE_ROW-1;ii>=0;ii--)
        {
            if( absolute(*(threshold + ii) - 85) > 4 )
                *(threshold + ii) = 85;
        }
    }

    */

/*
*****
****
*           图像2值化
* 黑0 白1
*
*****
****
*/

#define xcs_maxphoto 80    //80

void  Photo_Make01(void)
{
    uchar i1,j1;
    uchar *pucTemp;

    for(i1=0;i1<IMAGE_ROW;i1++)
    {
        for(j1=0; j1<IMAGE_COLUMN; j1++)
        {
```

```

        pucTemp = puca_BufferProcess + i1*IMAGE_COLUMN + j1;

        /*pucTemp= (*pucTemp>=xcs_maxphoto)?1:0;
        *pucTemp = ( *pucTemp >= xcs_maxphoto )?1:0;

        }

    }

    *(puca_BufferProcess+0*IMAGE_COLUMN+IMAGE_CENTER) = 0;
    *(puca_BufferProcess+1*IMAGE_COLUMN+IMAGE_CENTER) = 0;
    *(puca_BufferProcess+2*IMAGE_COLUMN+IMAGE_CENTER) = 0;
    *(puca_BufferProcess+3*IMAGE_COLUMN+IMAGE_CENTER) = 0;
}

/*
*****

**

*           速度控制
* PD控制
*

*****

***

*/

void Moto_PD(void)
{
    volatile static int err=0,last_err=0,derr=0;
    volatile static int M_PWM=0;

    err = chesudu - pulse;

```

```

    derr = err - last_err;

    M_PWM += err*90 + derr*30;          //16 15      //2 5      //80
90

    if(M_PWM < 0)      M_PWM = 0;
    if(M_PWM > 4000) M_PWM = 4000;

    last_err = err;
    move_foward(M_PWM);
}

/*
*****
*   行场中断处理
*功能：
*       b 重新开启了行中断。这个要清行中断的标志
*       c 采集行数的复位
*****
*/

uchar jiange_m=0;    //是不是间隔(jiange)了8行

uchar hang_i=0; //行中断里面用的记录每行点的个数的变量

uchar *H_puTemp= NULL;

//uint count;

#pragma CODE_SEG __NEAR_SEG NON_BANKED

```

```
/******场中断******/
```

```
void interrupt 9 Changzd(void)
```

```
{
```

```
    TFLG1_C1F=1; // 清场中断标志
```

```
    heixian =0; //从置 行里面的信息
```

```
    jiangem=0; //行数清零
```

```
    //交换图像采集和处理缓存
```

```
    puca_BufferSample=puca_BufferProcess;
```

```
    puca_BufferProcess=puca_BufferTemp;
```

```
    puca_BufferTemp=puca_BufferSample;
```

```
    TFLG1_C0F=1; // 清行中断标志
```

```
    TIE_C0I =1; // 行中断
```

```
}
```

```
/******行中断******/
```

```
void interrupt 8 Hangzd(void)
```

```
{
```

```
    TFLG1_C0F=1; // 清中断标志
```

```
    if(heixian < IMAGE_ROW)
```

```
    {
```

```
        if(jiangem == get_n[heixian])
```

```
        {
```

```
        delay(150);

        H_puTemp =puca_BufferSample+heixian*IMAGE_COLUMN;

        for(hang_i=0;hang_i<IMAGE_COLUMN;hang_i++)
        {

            *(H_puTemp+hang_i)=PORTA;

        }

        heixian ++; //记录黑线的数目

        if(heixian == IMAGE_ROW)
        {

            g_car_show_yn = 1;//可以显示，可以处理这场数据了

        }

    }

    jiangem++;
}

} //end inter

uint count=0;
void interrupt 66 timer0(void) //测度
{

    PITTF=0x01;          //注意这。若是入口地址是67，PITTF=0x02
```

```
pulse=PACNT;

Moto_PD();

PACNT=0X0000;


count++;

if(count >= 1600)
{
    flag = 1;
    count = 1600;
}

}


#pragma CODE_SEG DEFAULT
```