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：北京科技大学

队伍名称：北京科技大学双车追逐组

参赛队员：冯富森

余昉

赵子秋

周程瑜

带队教师：孟宇

靳添絮

关于技术报告和研究论文使用授权的说明

本人完全了解第十一届“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：_____

带队教师签名：_____

日　　期：2016 年 8 月 10 日

摘 要

本文设计的智能车系统以MK60N512VMD100微控制器为核心控制单元，通过CMOS摄像头检测赛道信息，使用模拟比较器对图像进行硬件二值化，提取黑色引导线，用于赛道识别；通过光电编码器检测模型车的实时速度，使用PID控制算法调节驱动左右电机的转速，实现了对车运动速度和运动方向的闭环控制。为了提高模型车的速度和稳定性，使用C#、MFC上位机、SD卡模块、键盘模块等调试工具，进行了大量硬件与软件测试。实验结果表明，该系统设计方案确实可行。

目录

引　　言

全国大学生“恩智浦”杯智能汽车竞赛是以“立足培养、重在参与、鼓励探索、追求卓越”为宗旨，鼓励创新的一项科技竞赛活动。竞赛要求在规定的汽车模型平台上，使用恩智浦半导体公司的微控制器作为核心控制模块，通过增加道路传感器、电机驱动模块以及编写相应控制程序，制作完成一个能够自主识别道路的模型汽车。参赛队员的目标是模型汽车需要按照规则以最短时间完成单圈赛道。

在本次比赛中，本组使用大赛组委会统一提供的竞赛车 A 车模，采用恩智浦（原飞思卡尔）32 位微控制器 MK60DN256ZVLL10 作为核心控制单元，自主构思控制方案及系统设计，融合摄像头图像采集处理、电机驱动输出，最终实现一套能够自主识别路线、进行双车追逐比赛的完备系统。

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。

在这份报告中，我们主要通过对整体方案、机械、硬件、算法等方面的介绍，详细阐述我队在此次智能汽车竞赛中的思想和创新。具体表现在电路的创新设计、算法以及辅助调试模块等方面的创新。我队成员涉及自动化、机械、计算机等专业，在准备比赛的过程中，队员查阅了大量的专业资料，反复地调试汽车模型的各项参数。所有队员都为此次智能汽车竞赛付出了艰苦的劳动。

第一章 方案设计

系统概述

摄像头车的系统整体结构如图所示。MK60DN256ZVLL10 微处理器通过采集模拟 CMOS 摄像头的硬件二值化信号，得到赛道边沿信息；通过采集陀螺仪数据分析计算过桥信息；通过采集光电编码器对车轮转速的脉冲计数，得到车行进的速度数据。通过超声波传感器实现双车间距离测量；通过蓝牙通信实现双车间数据交流；通过微处理器对图像处理，对角度、速度进行 PID 控制，最后 PWM 波输出驱动电机、舵机。在调试过程中，通过 SD 卡上位机观察摄像头采集的图像、车身角度等实时数据，便于车的状态控制进行调试。

第二章 智能车机械结构调整与优化

根据今年组委会的相关规定以及双车比赛规则完成相应超车动作，因此需要车身尽可能灵活紧凑，我们最后选择使用 A 型车模参加双车追逐组的比赛。在比赛备战之初，我们就对该车模进行了详细的系统分析。但由于 A 型车模精度不是很高，因此在规则允许范围内尽量改造车模，提高车模整体精度，以提高车辆行驶的稳定性。本章将主要介绍智能汽车车模的机械结构及调整方案。

2.1 车体机械建模

此次竞赛的赛车车模选用由东莞市博思电子数码科技有限公司提供的 A 型车模。车模外形如图 2.1 所示。

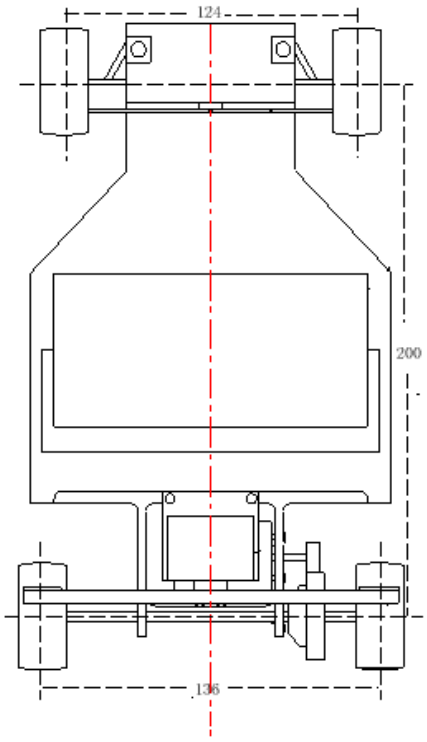

图 2.1 车模外形图

2.2 前轮倾角的调整

在调试过程中，我们发现由于前轮轴和车轮之间的间隙较大，对车高速时转向中心的影响较大，会引起高速转向下模型车的转向不足。然而这里是规则中严禁改动的部分，所以为了尽可能降低转向舵机负载，我们对前轮的安装角度，即前轮定位进行了调整。

前轮定位的作用是保障汽车直线行驶的稳定性，转向轻便和减少轮胎的磨损。前轮是转向轮，它的安装位置由主销内倾、主销后倾、前轮外倾和前轮前束等 4 种可调参数决定，实现转向轮、主销和前轴等三者在车架上的位置关系。

在实际调试中，我们发现适当增大内倾角可以增大转弯时车轮和地面的接触面积，从而增大车了地面的摩擦程度，使车转向更灵活，减小因摩擦不够而引起的转向不足的情况。

2.3 底盘高度的调整

在保证顺利通过坡道的前提下，底盘尽量降低，从整体上降低模型车的重心，可使模型车转弯时更加稳定、高速。

2.4 编码器的安装

为了更为精确的获得电机转速的返回值，本次车模上安装的是编码器，选择质量和体积都较小的编码器以控制车身重量。最终综合考虑了读数精准和重心分布两大因素，用定制加工件和齿轮进行了安装配合，尽量使得传动齿轮轴保持平行，传动部分轻松、流畅，不存在过大噪音和丢数情况。如图 2.2 所示。

图 2.2　编码器安装

2.5 舵机安装结构的安装

原装车模的舵机为卧式安装，考虑到主板的安装方便以及车模转向性能，我们对舵机安装结构进行了较大的调整。比赛车模的转向是通过舵机带动左右横拉杆实现。舵机的转动速度和功率是一定，要想加快转向机构的响应速度，唯一的办法就是优化舵机的安装位置及其力矩延长杆的长度。由于功率是速度与力矩乘积的函数，过分追求速度，必然要损失力矩，力矩太小也会造成转向迟钝，因此设计时就要综合考虑转向机构响应速度与舵机力矩之间的关系，通过优化得到一个最佳的转向效果。利用实际参数经计算，我们得出了一套可以稳定高效工作的参数及结构。

最终，我们设计了一套舵机连片(转向拉杆)，综合考虑了速度与力矩的关系，并根据模型车底盘的具体结构，简化了安装方式，实现了预期目标。

关于舵机的安装方式，我们实验室较为主流的有直立式安装和倒置式安装，我们的舵机安装如图 2.3 所示。

图 2.3 舵机安装

2.6 舵机转角分析

车模的转向运动主要是靠舵机和前轮来实现的，因此，关于舵机和前轮转向关系的分析就显得尤为重要。

依据数学建模，通过 MATLAB 进行分析后，得出了如下结果。

右前轮转角(绿色)及舵机转角（红色）关于转弯半径关系图，如图 2.4 所示。

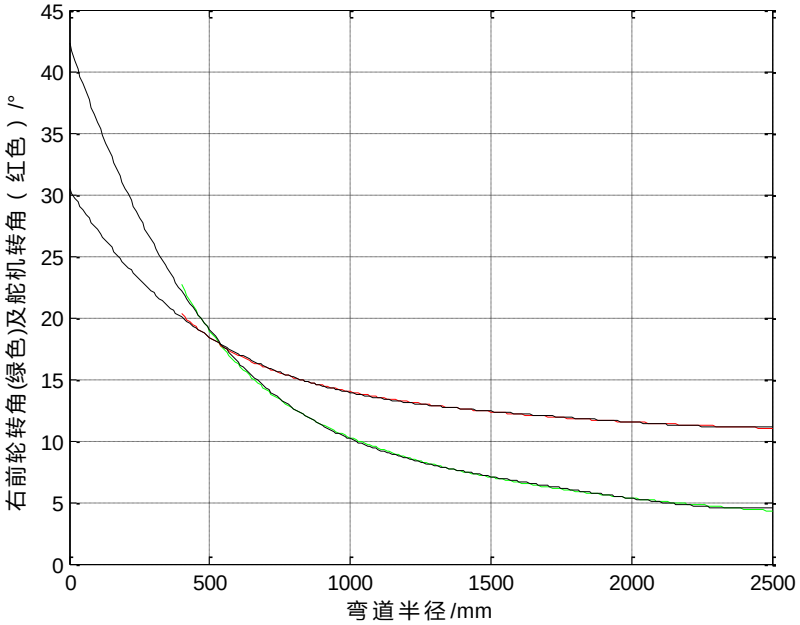

图 2.4

舵机转角关于（右）前轮转角关系图，如图 2.5 所示。

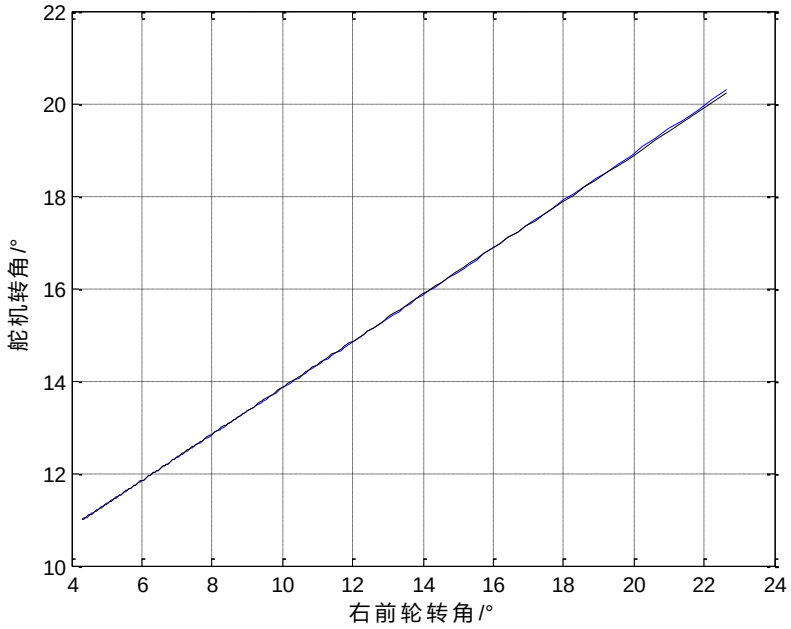

图 2.5

由以上两图得出结论：

舵机转角变化范围即使较小，转弯半径的变化也会很大，因而对多级的控制显得尤为很重要。

前轮打角越小，随着打角变化，转弯半径变化越明显，即小转角对半径的变化会更加明显，因而从前桥到舵机连片的机械固定需牢靠，尽量减少虚位。

舵机转角关于前轮转角呈线性变化，在思路上为通过舵机转角改变从而获得所要前轮转角提供便利。

改变前束，获得新的右前轮转角(绿色)及舵机转角（红色）关于转弯半径关系图，如图 2.6。通过与图 2.4 的对比，获得两种前轮方案对转角的影响，从而选择合适的方案。

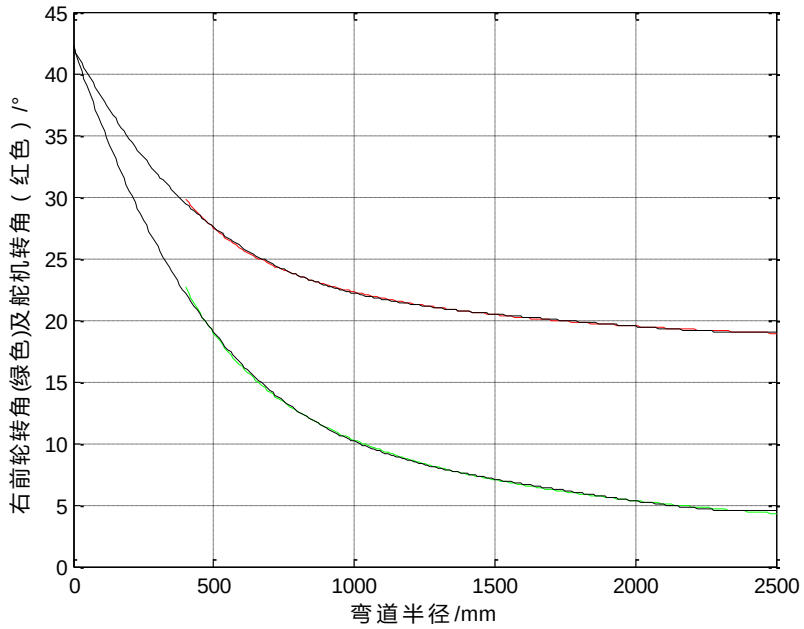

图 2.6

2.7 摄像头的安装

为了降低整车重心，需要严格控制 CMOS 摄像头的安装位置和重量，我们自行设计了轻巧的铝合金夹持组件并采用了碳纤维管作为安装 CMOS 的主桅，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，使摄像头便于拆卸和维修，具有赛场快速保障能力。摄像头的安装如图 2.7 所示。

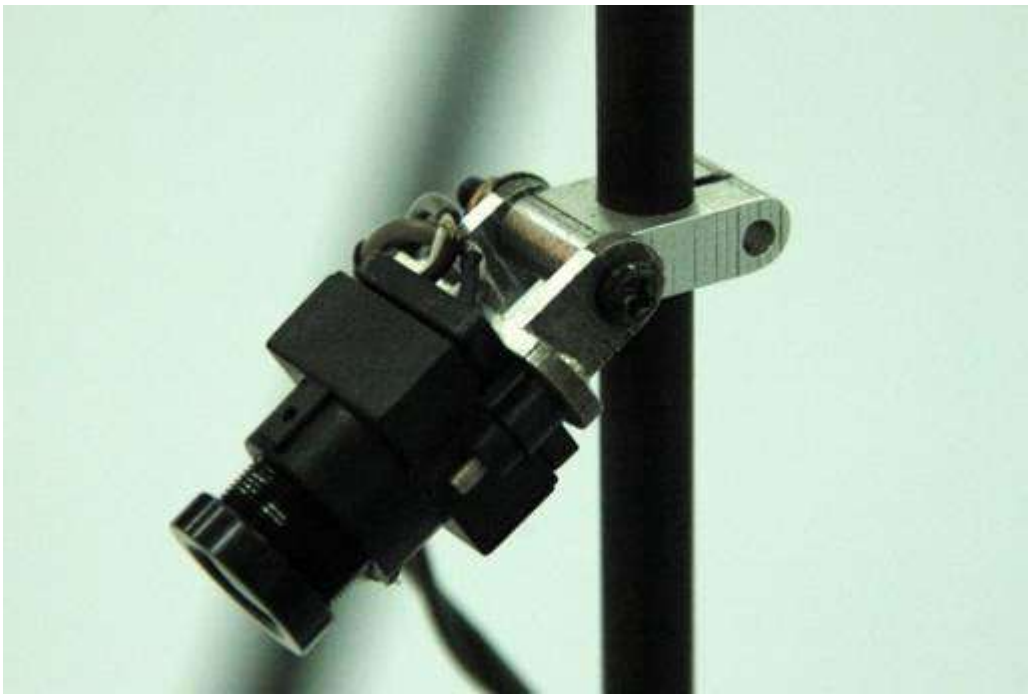

图 2.7 摄像头安装

第三章 电路设计说明

3.1 硬件方案设计

我们主要从系统的稳定性、可靠性、高效性、实用性、简洁等方面来考虑硬件的整体设计。从最初方案设定到最终方案的敲定，我们经历各种讨论与大的改动才有了如下的硬件方案。

可靠性与稳定性是一个系统能够完成预设功能的最大前提。在原理图与 PCB 的设计过程中，我们考虑到各个功能模块的电特性以及之间的耦合作用。对易受干扰的模块做了电磁屏蔽作用，而其他部分则做了相应的接地、滤波、模拟与数字电路的隔离等。

高效与实用性是指本系统的各模块能充分完美的实现相应的功能。从以下两点可以体现出：

a) 视频信号的提取一般有三种方法：片内 AD 转化、基于 TLC5510 的 8 位并行 AD、硬件二值化。第一种方通过片内 AD 转换把连续的模拟视频信号转为数字信号存储起来。第二种方法通过外部 AD 芯片直接把视频信号转为并行的数字信号，而处理芯片只要通过普通 IO 来读取存储；前者不需要外部电路但浪费系统时间，对于主频不高的处理芯片会造成很大负担。而后者虽然精度高但浪费过多的硬件资源。硬件二值化是通过比较器去捕捉视频信号的跳变沿，这不仅减少了采集时间，也节约硬件资源，还省去了许多存储空间。

b) 对于电机驱动，由于 A 车模电机对驱动性能要求高。我们设计了由单独的驱动芯片组成驱动器，该驱动器瞬间驱动电流最大可以达到几十安。

简洁是指在满足了可靠、高效的要求后，为了尽量减轻车模的负载，降低模型车的重心位置，应使电路设计尽量简洁，尽量减少元器件使用数量，缩小电路板面积，使电路部分重量轻，易于安装。在设计完原理图后，注重 PCB 板的布局，优化电路的走线，整齐排列元器件，最终做到电路板的简洁。

3.2 传感器选择

3.2.1 摄像头选择

COMS 与 CCD

CCD 摄像头具有对比度高、动态特性好的优点，但需要工作在 12V 电压下，对于整个系统来说过于耗电,而且 CCD 体积大，质量重，会抬高车体的重心，这对于高速情况下小车的行驶非常不利。

与之相比，COMS 摄像头具有体积小、质量轻、功耗低，图像动态特性好等优点，因为小车队图像的清晰度，分辨率要求并不高，所以选用 COMS 摄像头。

对于摄像头的选择，主要考虑以下几个参数：

1 芯片大小

2 自动增益

3 分辨率

4 最小照度

5 信噪比

6 标准功率

7 扫描方式

其中芯片大小主要会对视场的范围会有影响，扫描方式主要有追行扫描以及隔行扫描，像 ov5116 就是隔行扫描。

市面上的摄像头主要分为数字和模拟两种，数字摄像头主要有 OV7620,OV6620，OV7670,OV7725,模拟摄像头主要有 OV5116，BF3003，MT9V136。大多数摄像头都支持 sccb 通信，可以很好的实现单片机与摄像头之间的交互。

智能车的摄像头队图像的分辨率要求并不高，但是对动态特性要求非常高，特别是小车在高速行驶入弯或者出弯的时候，图像变化较大，这就对摄像头的自动增益有较高的要求。一般来说，在摄像头图像发生突变时，感光芯片本身会有一段适应时间，这段时间要求越小越好。

我们先后试用了 OV5116、BF3003、MT9V136 以及 PC1030N 等摄像头，后三种均为彩色摄像头，在经过硬件二值化后可以取其有效信息，与黑白摄像头 OV5116 得到的信息类似。经过对比，虽然后三种摄像头图像效果更好，动态性能也更好，但是在弯道、回环等元素中由于自动曝光导致得到赛道信息不准确，在智能车应用中 OV5116 已足够满足需求，因此我们最终选择了 OV5116 黑白摄像头。

3.2.2 陀螺仪选择

本届大赛对陀螺仪的型号没有限制，经过挑选我们最终确定陀螺仪使用 L3G4200D。

该产品为 ST 推出采用一个感应结构检测三条正交轴向运动的 3 轴数字陀螺仪。L3G4200D 是三轴共用一个感应结构，这一突破性概念可以消除轴与轴之间的信号干扰，避免输出信号收到干扰信号的影响。

3.2.3 编码器选择

为了使用闭环控制，我们在汽车模型上附加了编码器。经过机械和电路性能的考虑和挑选，最终我们选用了 ABI Mini 增量式旋转编码器。

和其他元件相比，选用编码器可以使电路更加完善，信号更加精确。编码器功耗低，重量轻，抗冲击抗震动，精度高，寿命长，非常实用。编码器内部无上拉电阻，因此编码器接口出需要设计上拉电阻。同时为了保证波形的稳定，主控板上使用了 74HC14 非门隔离。K60 自身具有正交解码功能，因此这里无需使用任何外围计数辅助器件，只需要将接口连接到单片机上相应的接口即可。接口如图 3.1 和图 3.2 所示。

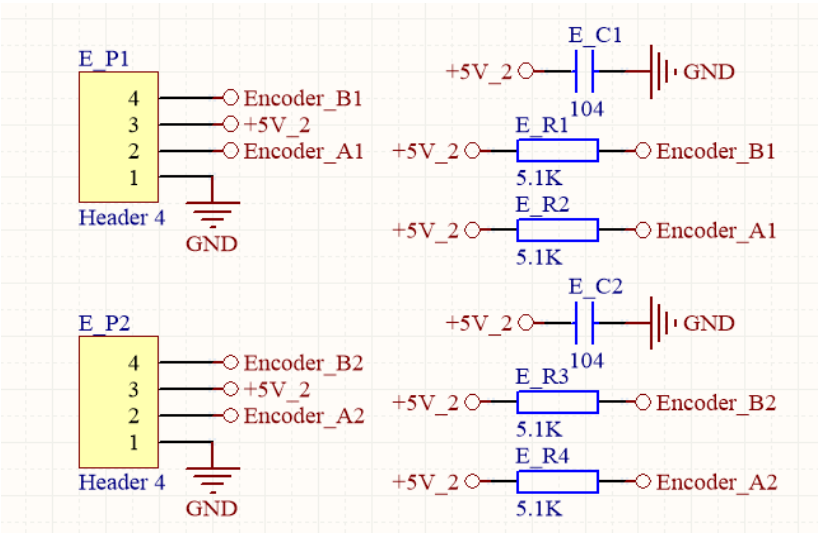

图 3.1 编码器的接口部分

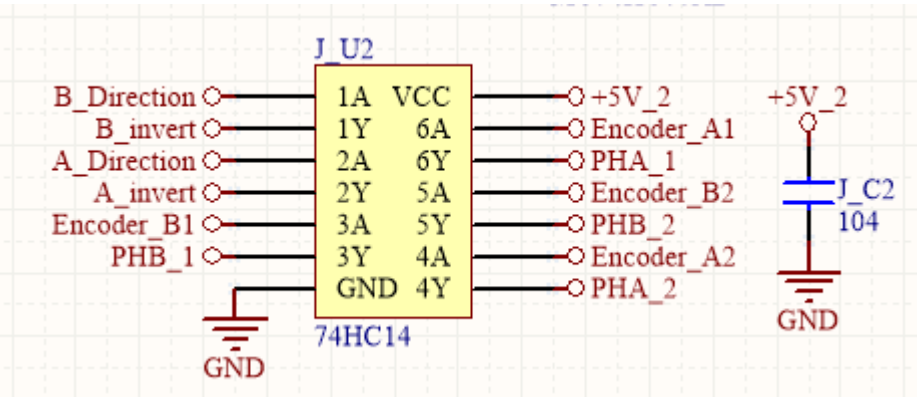

图 3.2 非门电路

3.2.4 超声波传感器

根据市面上大众超声波测距传感器特性，结合本次双车追逐要求，我们设计并制作了符合双车间控距的超声波模块。旨在对运动中的小车进行可靠的距离测量，测量周期为 20ms，最终距离信号以方波脉冲的形式输出由主控单片机检测接收。

3.3 电路模块实现

3.3.1 电源管理模块

首先了解一下不同电源的特点，电源分为开关电源和线性电源，线性电源的电压反馈电路是工作在线性状态，开关电源是指用于电压调整的管子工作在饱和和截至区即开关状态的。线性电源一般是将输出电压取样然后与参考电压送入比较电压放大器，此电压放大器的输出作为电压调整管的输入，用以控制调整管使其结电压随输入的变化而变化，从而调整其输出电压，但开关电源是通过改变调整管的开和关的时间即占空比来改变输出电压的。

从其主要特点上看：线性电源技术很成熟，制作成本较低，可以达到很高的稳定度，波纹也很小，而且没有开关电源具有的干扰与噪音，开关电源效率高、损耗小、可以降压也可以升压，但是交流纹波稍大些。

电源模块对于一个控制系统来说极其重要，关系到整个系统是否能够正常工作，因此在设计控制系统时应选好合适的电源模块。竞赛规则规定，比赛使用智能汽车竞赛统一配发的标准车模用 7.2V 2000mAh Ni-cd 供电，而单片机系统、路径识别的 CCD 传感器、陀螺仪均使用的是 3.3V 的电源。编码器需要 5V 电源，伺服电机工作电压范围为 4V 到 6V（为提高伺服电机响应速度，采用 7.2V 供电），直流电机可以使用 7.2V 2000mAh Ni-cd 蓄电池直接供电，智能汽车电压调节电路示例见图 3.3。

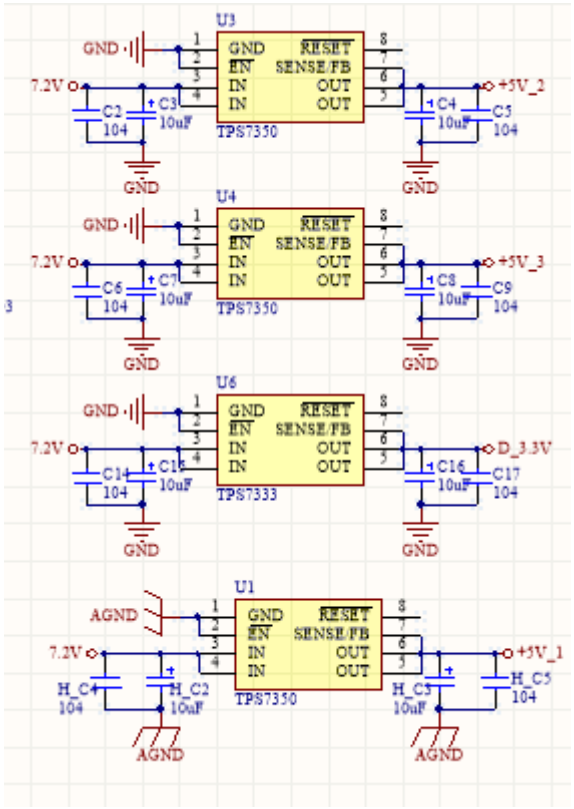

图 3.3 电源管理模块原理图

在电源管理模块中，我们选用了三片 TPS7350 和一片 TPS7333，其中 DCDC5-12 和 IR2184S 共用一个 5V 电源，逻辑元件和编码器共用另一个 5V 电源；其余模块共用 3.3V 电源。

3.3.2 电机驱动模块

常用的电机驱动有两种方式：

一、集成电机驱动芯片；

二、采用 N 沟道 MOSFET 和专用栅极驱动芯片设计。市面上常见的集成 H 桥式电机驱动芯片中，33886 型芯片性能较为出色，该芯片具有完善的过流、欠压、过温保护等功能，内部 MOSFET 导通电阻为 120 毫欧，具有最大 5A 的连续工作电流。使用集成芯片的电路设计简单，可靠性高，但是性能受限。由于比赛电机内阻仅为几毫欧，而集成芯片内部的每个 MOSFET 导通电阻在 120 毫欧以

上，大大增加了电枢回路总电阻，此时直流电动机转速降落较大，驱动电路效率较低，电机性能不能充分发挥。

由于分立的 N 沟道 MOSFET 具有极低的导通电阻，大大减小了电枢回路总电阻。另外，专门设计的栅极驱动电路可以提高 MOSFET 的开关速度，使 PWM 控制方式的调制频率可以得到提高，从而减少电枢电流脉动。并且专用栅极驱动芯片通常具有防同臂导通、硬件死区、欠电压保护等功能，可以提高电路工作的可靠性。

1.专用栅极驱动芯片的选择：

IR 公司号称功率半导体领袖，所以我们主要在 IR 公司的产品中进行选择。其中 IR2184 型半桥驱动芯片可以驱动高端和低端两个 N 沟道 MOSFET，能提供较大的栅极驱动电流，并具有硬件死区、硬件防同臂导通等功能。使用两片 IR2184 型半桥驱动芯片可以组成完整的直流电机 H 桥式驱动电路。由于其功能完善，价格低廉容易采购，所以我们选择它进行设计，如图 3.4 所示。

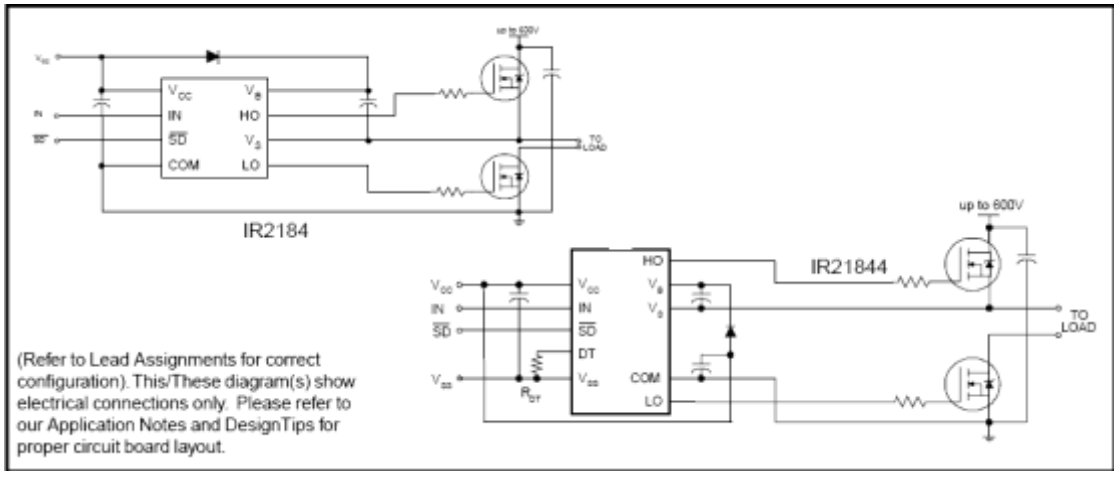

图 3.4 IR2184 应用图

2.MOSFET 的选择：

选择 MOSFET 时主要考虑的因素有：耐压、导通内阻和封装。智能汽车电源是额定电压为 7.2V 的电池组，由于电机工作时可能处于再生发电状态，所以驱动部分的元件耐压值最好取两倍电源电压值以上，即耐压在 16V 以上。而导通内阻则越小越好。封装越大功率越大，即同样导通电阻下通过电流更大，但封

装越大栅极电荷越大，会影响导通速度。常用的 MOSFET 封装有 TO-220、TO-252、SO-8 等，TO-252 封装功率较大、而栅极电荷较小。于是我们最终选择了 IR 公司 TO-252 封装的 LR7843 型 N 沟道 MOSFET，VDSS=55 伏、RDS(on)=8.0 毫欧、ID=110 安。

3、驱动电路的原理分析及元件参数确定

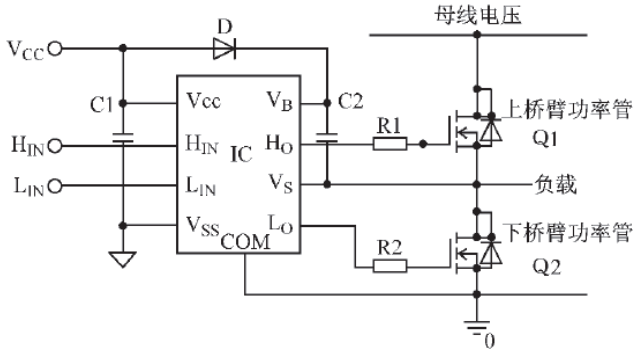

图 3.5 电机驱动分析图

这个驱动设计单从信号逻辑上分析比较容易理解，但要深入的理解和更好的应用，就需要对电路做较深入的分析，对一些外围元件的参数确定做理论分析计算。图 3.8 中 IC 是一个高压驱动芯片,驱动 2 个半桥 MOSFET。Vb,Vs 为高压端供电;Ho 为高压端驱动输出;COM 为低压端驱动供电,Lo 为低压端驱动输出;Vss 为数字电路供电.此半桥电路的上下桥臂是交替导通的,每当下桥臂开通,上桥臂关断时 Vs 脚的电位为下桥臂功率管 Q2 的饱和导通压降,基本上接近地电位,此时 Vcc 通过自举二极管 D 对自举电容 C2 充电使其接近 Vcc 电压.当 Q2 关断时 Vs 端的电压就会升高,由于电容两端的电压不能突变,因此 Vb 端的电平接近于 Vs 和 Vcc 端电压之和,而 Vb 和 Vs 之间的电压还是接近 Vcc 电压.当 Q2 开通时,C2 作为一个浮动的电压源驱动 Q2;而 C2 在 Q2 开通其间损失的电荷在下一个周期又会得到补充,这种自举供电方式就是利用 Vs 端的电平在高低电平之间不停地摆动来实现的。由于自举电路无需浮动电源,因此是最便宜的，如图所示自举电路给一只电容器充电，电容器上的电压基于高端输出晶体管源极电压上下浮动。图 2.6 中的 D 和 C2 是 IR2184 在脉宽调制（PWM）应用时应严格挑选和设计的元器件,根据一定的规则进行计算分析;并在电路实验时进行调整,使电路工作处于最佳状态,其中 D 是一个重要的自举器件,应能阻断直流干线上的高压,其承受的电流是

栅极电荷与开关频率之积,为了减少电荷损失,应选择反向漏电流小的快恢复二极管,芯片内高压部分的供电都来自图中自举电容 C2 上的电荷;为保证高压部分电路有足够的能量供给应适当选取 C2 的大小.

MOSFET 具有相似的门极特性,开通时需要在极短的时间内向门极提供足够的栅电荷,在自举电容的充电路径上,分布电感影响了充电的速率,下桥臂功率管的最窄导通时间应保证自举电容有足够的电荷以满足栅极所需要的电荷量再加上功率器件稳态导通时漏电流所失去的电荷量.因此,从最窄导通时间为最小值考虑,自举电容应足够小;综上所述,在选择自举电容大小时应考虑,既不能太大影响窄脉冲的驱动性能;也不能太小影响宽脉冲的驱动要求,应从功率器件的工作频率、开关速度、门极特性等方面进行选择、估算后调试而定。

3.3.3 视频处理模块

我们的智能模型车自动控制系统中使用黑白全电视信号格式 CMOS 摄像头采集赛道信息。摄像头视频信号中除了包含图像信号之外，还包括了行同步信号、行消隐信号、场同步信号、场消隐信号以及槽脉冲信号、前均衡脉冲、后均衡脉冲等。因此，若要对视频信号进行采集，就必须通过视频同步分离电路准确地把握各种信号间的逻辑关系。我们使用了 LM1881 芯片对黑白全电视信号进行视频同步分离，得到行同步、场同步信号。

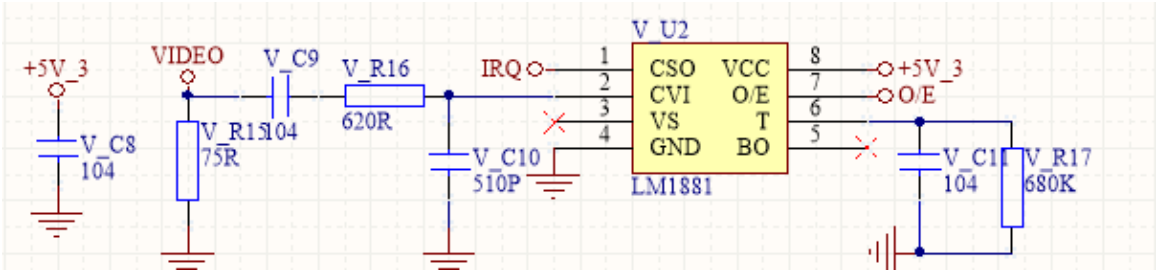

图 3.6　LM1881 外围电路原理图

在对硬件二值化的研究中，我们也从数字比较器以及模拟比较器几个方向进行了试探性研究，从图像的稳定性及清晰性等方面进行筛选，最终决定采用模拟电路搭建而成的比较器对图像进行二值化处理。

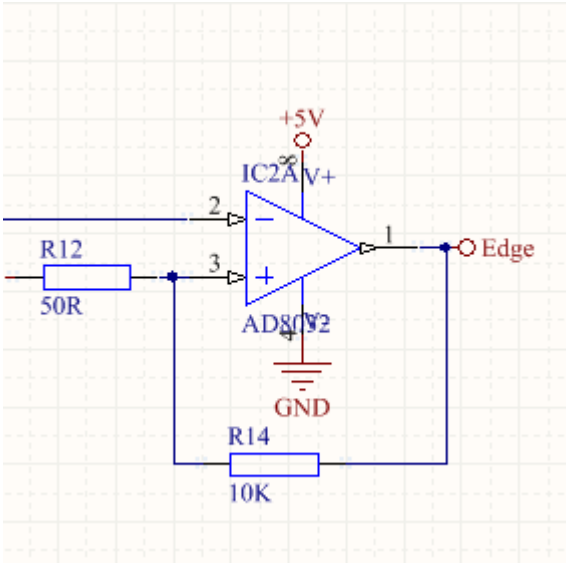

图 3.7　比较电路原理图

3.3.4 接口及外接模块

对于单片机最小系统、陀螺板、视频模块、键盘、超声波模块、蓝牙，我们在主板上设计了外接接口，用于连接。

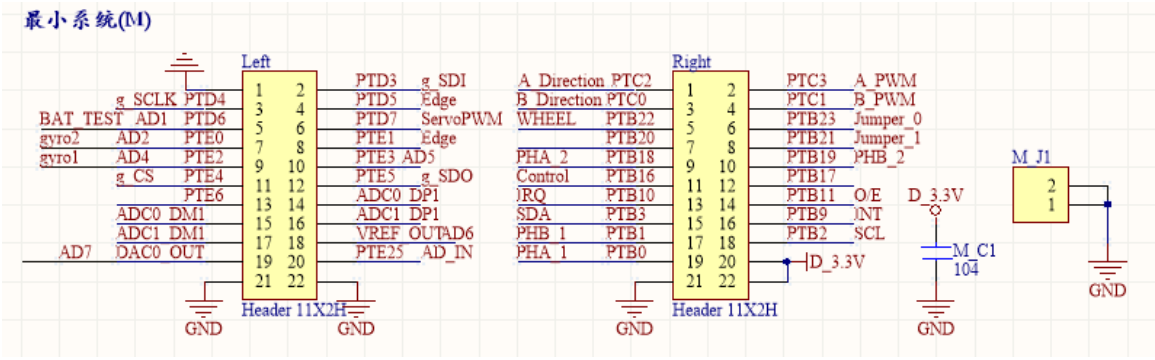

图 3.8 最小系统接口

MK60N512VMD100 是 K60 系列 MCU。Kinetis 系列微控制器是 Cortex-M4 系列的内核芯片。K60 内存空间可扩展，从 32 KB 闪存/ 8 KB RAM 到 1 MB 闪存 / 128 KB RAM，可选的 16 KB 缓存用于优化总线带宽和闪存执行性能。

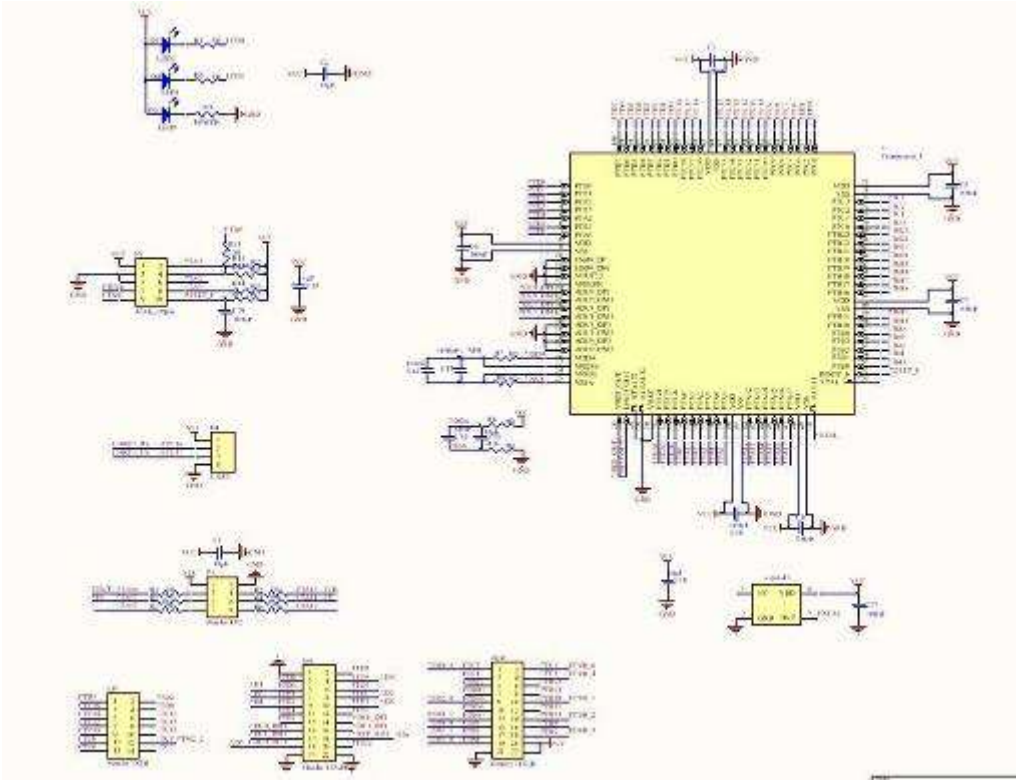

图 3.9 最小系统板

在调试过程之中，我们需要实时的了解与掌握一些车的运行状态，比如说传感器的状态，舵机的转角等，调试时用数码管将这些参数显示出来，让我们实时的监测车的状态，从而做出判断，这样很大程度的方便了对车的调试。有时候需要对参数作修改处理，如果每修改一个数据就下载一次程序的话，就会浪费时间，这时应用键盘，它就起到一个人机交互的作用。

我们选用 ZLG7290 控制芯片，它可以驱动八位数码管，最多可驱动 64 个独立按键。

引脚序号	引脚名称	功能描述
1	SC/KR2	数码管 c 段 / 键盘行信号 2
2	SD/KR3	数码管 d 段 / 键盘行信号 3
3	DIG3/KC3	数码管位选信号 3 / 键盘列信号 3
4	DIG2/KC2	数码管位选信号 2 / 键盘列信号 2
5	DIG1/KC1	数码管位选信号 1 / 键盘列信号 1
6	DIG0/KC0	数码管位选信号 0 / 键盘列信号 0
7	SE/KR4	数码管 e 段 / 键盘行信号 4
8	SF/KR5	数码管 f 段 / 键盘行信号 5
9	SG/KR6	数码管 g 段 / 键盘行信号 6
10	DP/KR7	数码管 dp 段 / 键盘行信号 7
11	GND	接地
12	DIG6/KC6	数码管位选信号 6 / 键盘列信号 6
13	DIG7/KC7	数码管位选信号 7 / 键盘列信号 7
14	$\overline{INT}$	键盘中断请求信号，低电平（下降沿）有效
15	$\overline{RST}$	复位信号，低电平有效
16	Vcc	电源，+3.3~5.5V
17	OSC1	晶振输入信号
18	OSC2	晶振输出信号
19	SCL	I^2C 总线时钟信号
20	SDA	I^2C 总线数据信号
21	DIG5/KC5	数码管位选信号 5 / 键盘列信号 5
22	DIG4/KC4	数码管位选信号 4 / 键盘列信号 4
23	SA/KR0	数码管 a 段 / 键盘行信号 0
24	SB/KR1	数码管 b 段 / 键盘行信号 1

图 3.10 ZLG7290 管脚说明

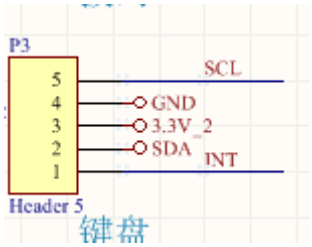

图 3.11 键盘接口

我们设计了八位数码管，3*4 共十二键的键盘，在具体的设计中可依据实际的应用选择限流电阻的大小，这样可以灵活的控制数码管的亮度。

第四章 智能车控制软件设计说明

高效的软件程序是智能车高速平稳自动寻线的基础。我们设计的智能车系统采用 CMOS 摄像头进行赛道识别，图像采集及校正处理就成了整个软件的核心内容。在智能车的转向和速度控制方面，我们使用了鲁棒性很好的经典 PID 控制算法，配合使用理论计算和实际参数补偿的办法，使智能车能够稳定快速寻线。

4.1 赛道中心线提取及优化处理

4.1.1 原始图像的特点

在单片机采集图像信号后需要对其进行处理以提取主要的赛道信息，同时，由于交叉道、起点线的存在，光线、杂点、赛道远处图像不清楚的干扰，图像效果会大打折扣。因此，在软件上必须排除干扰因素，对赛道进行有效识别，并提供尽可能多的赛道信息供决策使用。

在图像信号处理中我们提取的赛道信息主要包括：赛道两侧边沿点位置、通过校正计算的赛道中心位置，中心点规划面积，赛道变化幅度，赛道类型判别。

由于摄像头自身的特性，图像会产生梯形式变形，这使得摄像头看到的信息不真实。因此我们利用赛道进行测量，创建函数还原出了真实赛道信息。原始图像是一个将模拟图像经模拟电路转换得到的二维数据矩阵，矩阵的每一个元素对应一个像素点，图像的第一行对应最远处，大约 220cm，图像的最底部一行对应最近处，大约 5cm。远处的图像小，近处的图像大，黑线为梯形状。

单片机通过比较器电路将每一行的黑白跳变点（跳变点按从右到左的顺序）记录下来，保存到两个二维数组里（分别表示上升沿、下降沿）。通过遍历上升沿和下降沿可以完成赛道边沿的提取。

摄像头采集到几种典型赛道图像如图 4.1～图 4.7 所示。

图 4.1 三角原始图像

图 4.2 弯道原始图像

图 4.3 十字交叉原始图像

图 4.4 前车障碍原始图像

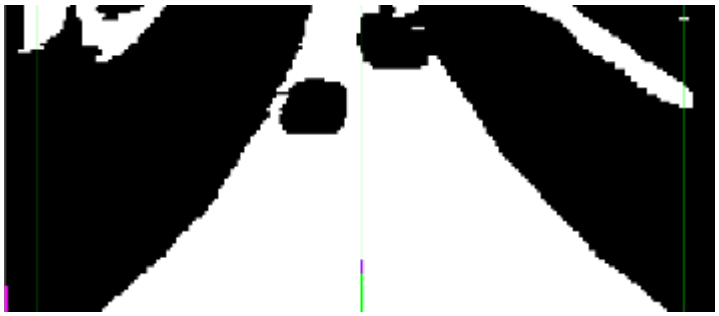

图 4.5 后车障碍原始图像

图 4.6 前车加宽区原始图像

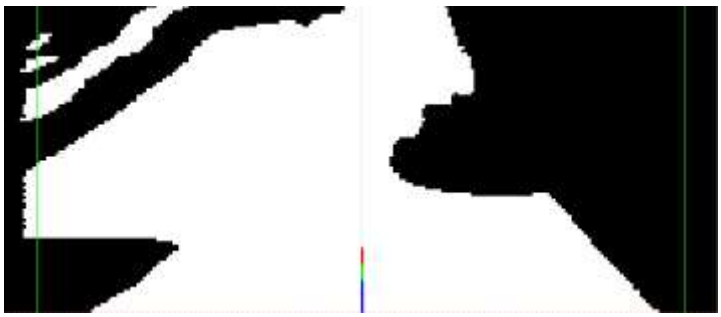

图 4.7 后车加宽区原始图像

4.1.2 赛道边沿提取

边沿提取算法的基本思想如下：

(1) 直接逐行扫描原始图像，根据设定的阈值提取黑白跳变点；

(2) 赛道宽度有一个范围，在确定的赛道宽度范围内提取有效赛道边沿，这样可以滤除不在宽度范围内的干扰；

(3) 利用赛道的连续性，根据上一行白块的位置和边沿的位置来确定本行的边沿点；

(4) 求边沿点时，因为近处的图像稳定，远处图像不稳定，所以采用由近及远的办法；

(5) 进出十字的时候，通过校正计算出边沿角度可较好的滤除十字并补线；

(6) 人字元素是整个赛道边沿角度是尖锐角的部分，根据这个特征通过计算边沿的角度以及内侧两条边沿包络形成的面积可以有效的识别出人字；为了排除赛道方向的突变对控制造成干扰，将人字建模成为具有一定曲率的弯道，并进行补线；

（7）由于权重分配的问题，如果不对障碍进行一定的处理的话，控制量对于远端的障碍相应比较小，可能在很接近障碍的时候才有响应，这样很容易造成撞到障碍，为了消除这种影响，我们利用曲线包络的形式 将障碍作用的区域人为扩大，这样有效避免了碰撞障碍的危险。

边沿提取算法的程序流程如图 4.8 所示。

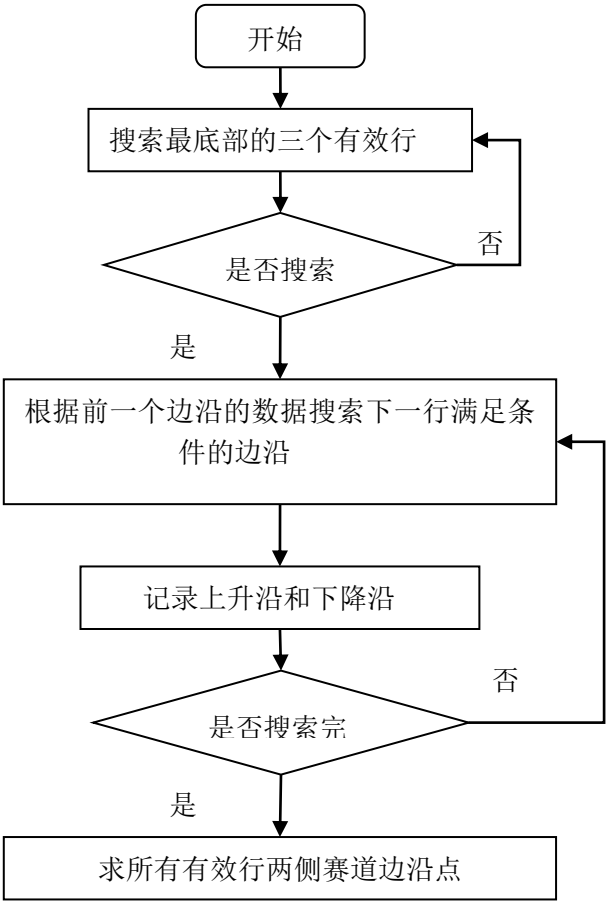

图 4.8

处理后得到的黑线中心如图 4.9～4.15 图所示。

图 4.9 三角处理后图像

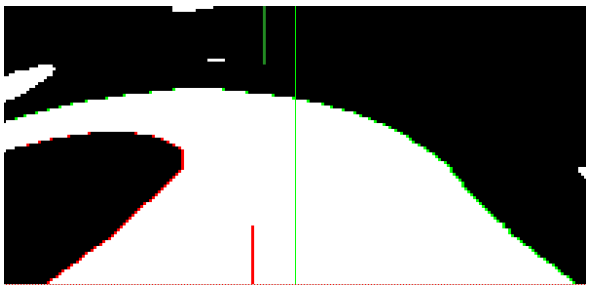

图 4.10 弯道处理后图像

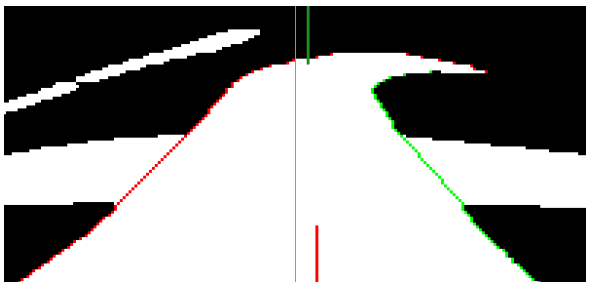

图 4.11 十字交叉处理后图像

图 4.12 障碍前车处理后图像

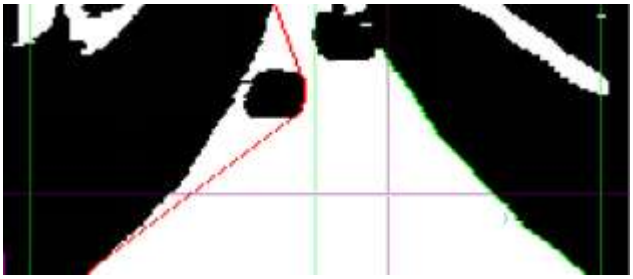

图 4.13 障碍后车处理图像

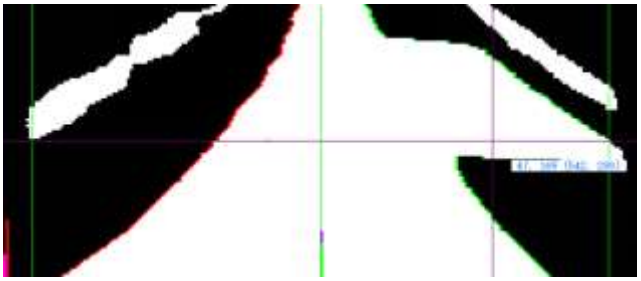

图 4.14 加宽区前车处理图像

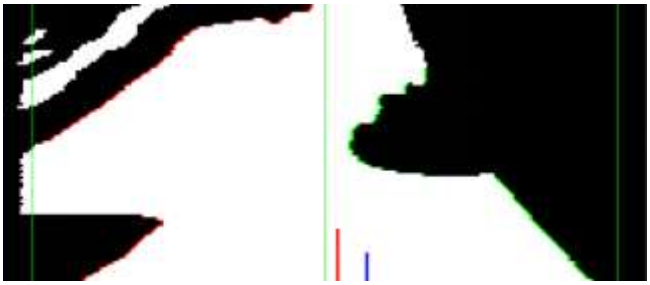

图 4.15 加宽区后车处理图像

4.1.3 图像校正

图像校正的实现如下：

(1) 调整好摄像头位置、前瞻，固定好；将摄像头对准黑白相间的赛道板，然后用电视观看摄像头图像，用照相机对准电视拍照(见图 4.16)；

图 4.16 电视中的图像

(2) 从照片中截取出赛道部分，然后用 matlab 编写程序，载入图片并进行相应的桶形变换、透视变换，调整好参数，生成校正表和反校正表；

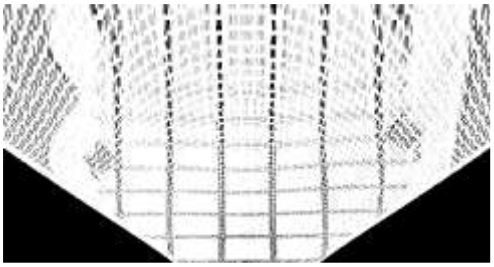

图 4.17 校正后的效果

(3) 在单片机程序中加入常量表，然后就可进行相应的校正和反校正变换了。

(4) 用上位机观察的校正效果如下：

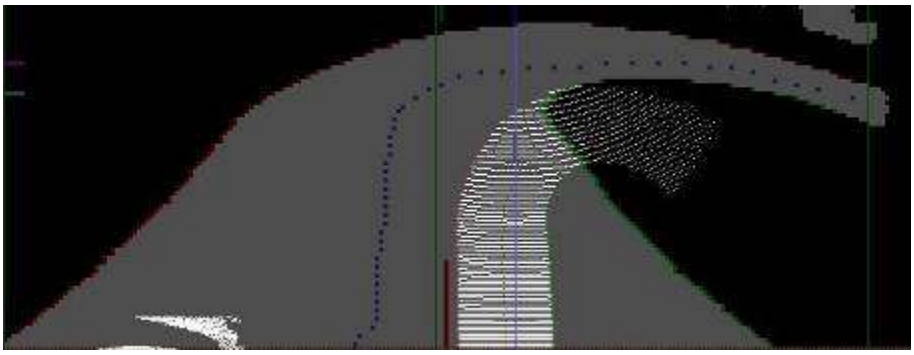

图 4.18 上位机模拟校正

4.1.4 推算中心

通过之前提取的赛道边沿数据推算中心：当左右边沿点总数较少时返回；若只有单边有边沿点数据，则通过校正对单边数据按法线平移赛道宽度一半的距离；当能找到与一边能匹配上的另一边沿点时则直接求其中心作为中心点。推算完中心点后，对中心点进行均匀化，方便之后的控制。计算出的中心点效果如下：

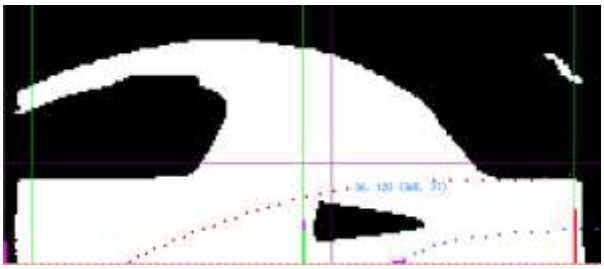

图 4.19 三角处理后图像

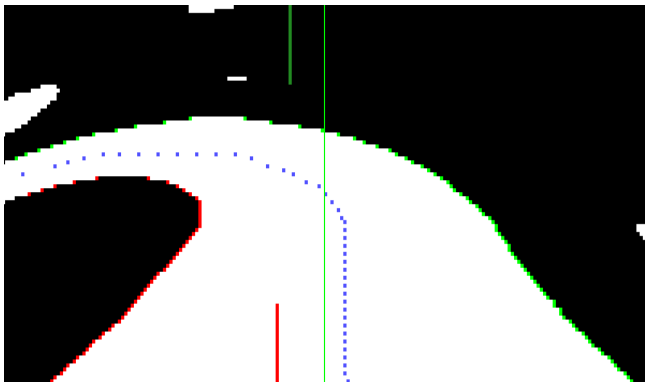

图 4.20 弯道处理后图像

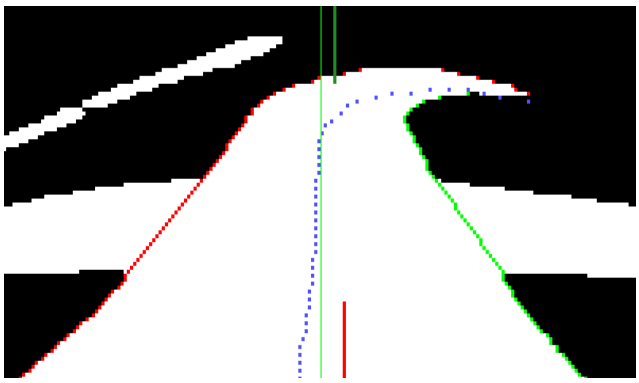

图 4.21 十字交叉处理后图像

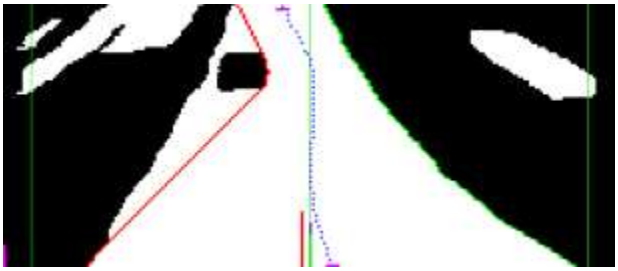

图 4.22 障碍前车处理后图像

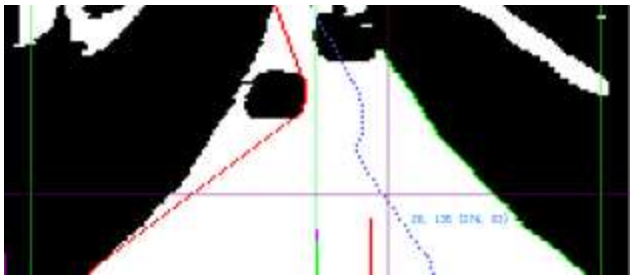

图 4.23 障碍后车处理图像

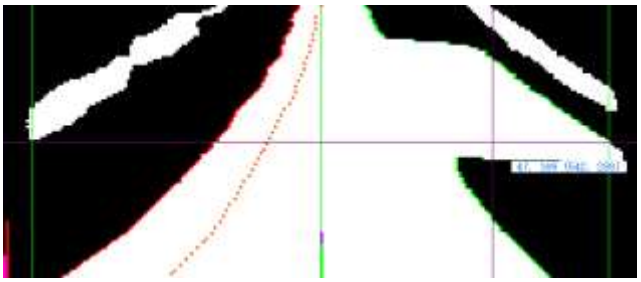

图 4.24 加宽区前车处理图像

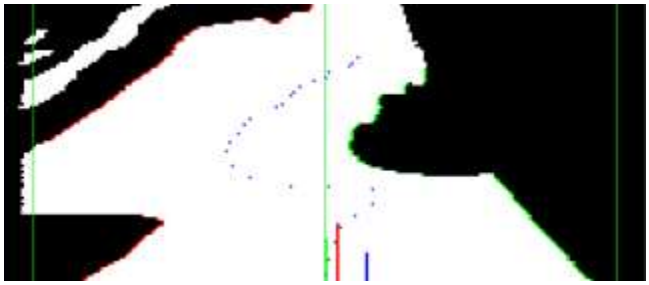

图 4.25 加宽区后车处理图像

4.1.5 路径选择

根据往届比赛以及本届华北赛的经验，赛车能否以最短的时间完成比赛，与赛车的速度和路径都有着密切的关系，因此，如何使赛车以一个最合理、最高效的路径完成比赛是提高平均速度的关键。

对于赛车路径的优化，我们从以下三个方面来完成：

1）增加视场的长度和宽度

根据我们的分析，当赛车采集到的图像能够覆盖一个比较完整的 S 弯道时，通过加权算法计算出来的中心就会处于视场中央附近，此时赛车会以一个比较好的路径快速通过 S 弯道；相反，如果视场无法覆盖一个完整的 S 弯道，赛车就会误处理为普通的单向弯道，这样赛车的速度就会大大减慢。因此，尽量增大视场的长度和宽度就很有必要了。

2）优化加权算法

对整场有效行的中心求加权平均值的算法，在低速情况下可以有效地优化赛车路径，但在赛车速度提高到一定程度之后由于过弯时的侧滑，路径不是很好。而由于图像分布不均，三分之二的行分布于车体前方 40cm 的范围内，求出的加权平均值受车体近处的图像影响较大，因此整场图像求加权的算法对于高速情况下的路径优化效果不是很明显。

为了解决这个问题，我们对于参与加权计算的图像行数及权重进行了处理，减小了车体前部 50cm 范围内的图像参与加权的行数和权重，同时增大视场前部图像的权重。在经过长期调试之后，得到了一套比较合适的参数，能够有效优化高速情况下的赛车路径。

3）对不合理的中心点进行处理

对于在校正后的图像数据中求得的中心线，反校正到原始图像后存在一行中含有多个中心点的情况。在通常情况下，这种情况出现在较远的视野中，但由于我们增大了视场前部图像的权重，这些中心点对权重的影响极大，导致车模容易出现掉轮甚至冲出赛道的现象。

为了解决这个问题，我们利用数学方法求出了中心线的折点，对折点之后的中心点单独处理，使车模不再出现掉轮的现象。

4.2 折点求取原理简介

折点是指数学中的极值点，即一阶导数为 0 的点，但一阶导数为 0 的点不一定都是极值点，究竟是否是极值点，还要判断二阶导数。在图像的离散数据中，极值点的离散化计算公式如式如下：

$$(Y_{n-1}-Y_n)\square(Y_n-Y_{n+1})<0 \qquad \text{(公式 4.1)}$$

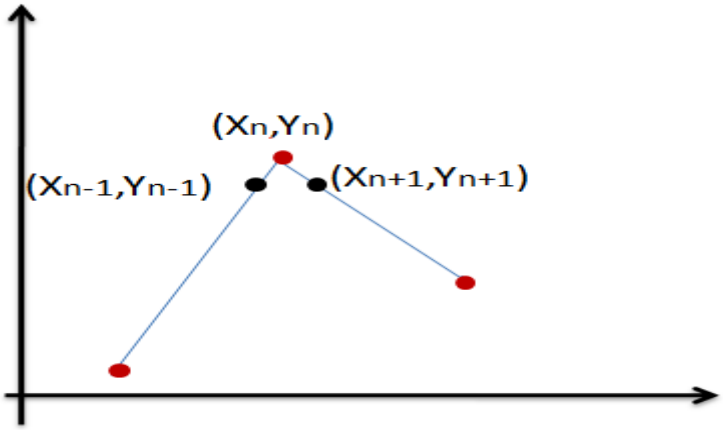

图 4.17 数学极值点

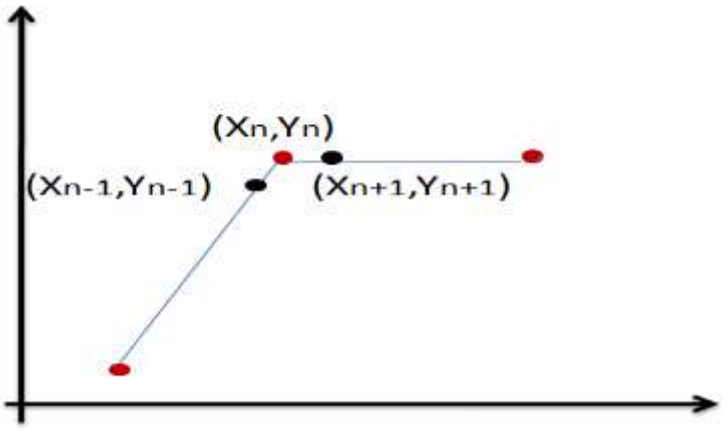

图 4.18 特殊情况下的折点

但这个公式在图 4.18 中存在一些问题，对于（Xn，Yn）点，由于$(Y_n - Y_{n+1})$为0，所以该点不会被判断为极值点。但我们认为这样的点也能反映图像的特征，因此对上式作出修改得到新的折点计算方法：

$$（Y_{n-1} - Y_n）*(Y_n - Y_{n+1}) \le 0 \ \&\& \ !(0 == (Y_{n-1} - Y_n) \ \&\& \ 0 == (Y_n - Y_{n+1})) \quad \text{(公式 4.2)}$$

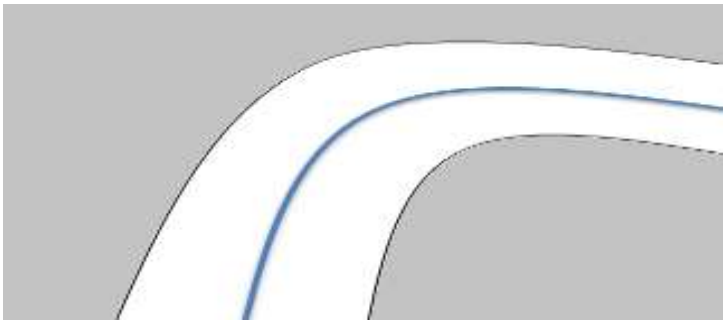

图 4.19 赛道图像中的折点

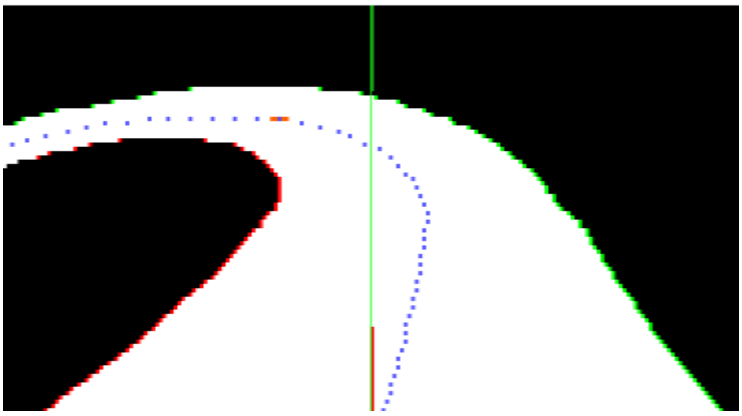

图 4.20 使用折点处理后的图像

极值点是针对某一个坐标轴而言的，对上式稍作修改，即可得到针对另一个坐标轴的极值点的计算公式：

$(X_{n-1}-X_n)*(X_n-X_{n+1})\le 0\ \&\&\ !(0==(X_{n-1}-X_n)\ \&\&\ 0==(X_n-X_{n+1}))$ (公式 4.3)

将两个方向上的极值点结合使用，不仅可以对不合理的中心点加以滤除，还可以实现对赛道类型的粗略判断。

4.3PID 控制算法介绍

在工程实际中，应用最为广泛的调节器控制规律为比例、积分、微分控制，简称 PID 控制，又称 PID 调节。PID 控制器问世至今已有近 70 年历史，它以其结构简单、稳定性好、工作可靠、调整方便而成为工业控制的主要技术之一。当被控对象的结构和参数不能完全掌握，或得不到精确的数学模型时，控制理论的其它技术难以采用时，系统控制器的结构和参数必须依靠经验和现场调试来确定，这时应用 PID 控制技术最为方便。即当我们不完全了解一个系统和被控对象，或不能通过有效的测量手段来获得系统参数时，最适合用 PID 控制技术。PID 控制，实际中也有 PI 和 PD 控制。

PID 控制器是一种线性控制器，它根据给定值与实际输出值构成控制偏差。将偏差的比例(P)、积分(I)和微分(D)通过线性组合构成控制量，对被控对象进行控制，故称 PID 控制器，原理框图如图 4.19 所示。

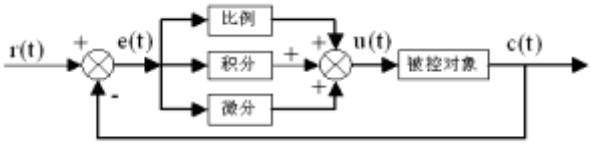

图 4.21 PID 控制器原理框图

在计算机控制系统中，使用的是数字 PID 控制器，控制规律为：

$$e(k)=r(k)-c(k) \qquad \text{(公式 4.4)}$$

$$u(k)=K_P\{e(k)+\frac{T}{T_I}\sum_{j=0}^{k}e(j)+\frac{T_D}{T}[e(k)-e(k-1)]\} \qquad \text{(公式 4.5)}$$

式中：

k——采样序号，k = 0，1，2…；　　r(k)——第 k 次给定值；

c(k)——第 k 次实际输出值；　　u(k)—— 第 k 次输出控制量；

e(k)—— 第 k 次偏差；　　e(k-1)—— 第 k-1 次偏差；

KP——比例系数；　　TI——积分时间常数；

TD——微分时间常数；　　T——采样周期。

简单说来，PID 控制器各校正环节的作用如下：

比例环节：及时成比例地反映控制系统的偏差信号，偏差一旦产生，控制器立即产生控制作用，以减少偏差。

积分环节：主要用于消除静差，提高系统的无差度。积分作用的强弱取决于积分时间常数，越大，积分作用越弱，反之则越强。

微分环节：能反映偏差信号的变化趋势(变化速率)，并能在该偏差信号变得太大之前，在系统中引入一个有效的早期修正信号，从而加快系统的动作速度，减小调节时间。

数字 PID 控制算法通常分为位置式 PID 控制算法和增量式 PID 控制算法。

4.3.1 位置式 PID

位置式 PID 中，由于计算机输出的 u (k) 直接去控制执行机构(如阀门)，u(k)的值和执行机构的位置(如阀门开度)是一一对应的，所以通常称公式(4.5)为位置式 PID 控制算法。

位置式 PID 控制算法的缺点是：由于全量输出，所以每次输出均与过去的状态有关，计算时要对过去 e(k)进行累加，计算机工作量大；而且因为计算机输出的 u(k)对应的是执行机构的实际位置，如计算机出现故障，u(k)的大幅度变化，会引起执行机构位置的大幅度变化，这种情况往往是生产实践中不允许的，在某些场合，还可能造成严重的生产事故。因而产生了增量式 PID 控制的控制算法，所谓增量式 PID 是指数字控制器的输出只是控制量的增量Δu(k)。

4.3.2 增量式 PID

当执行机构需要的是控制量的增量(例如：驱动步进电机)时，可由式(4.5)推导出提供增量的 PID 控制算式。由式(4.5)可以推出式(4.6)，式(4.5)减去式(4.6)可得式(4.4)。

$$u(k-1)=K_P\{e(k-1)+\frac{T}{T_I}\sum_{j=0}^{k-1}e(j)+\frac{T_D}{T}[e(k-1)-e(k-2)]\} \quad \text{(公式 4.6)}$$

$$\Delta u(k)=K_P\{[e(k)-e(k-1)]+\frac{T}{T_I}e(k)+\frac{T_D}{T}[e(k)-2e(k-1)+e(k-2)]\}$$
$$=K_P\Delta e(k)+K_I e(k)+K_D[\Delta e(k)-\Delta e(k-1)]\} \quad \text{(公式 4.7)}$$

式中 $\Delta e(k)=e(k)-e(k-1)$； $K_I=K_P\frac{T}{T_I}$； $K_D=K_P\frac{T_D}{T}$

公式(4.7)称为增量式 PID 控制算法，可以看出由于一般计算机控制系统采用恒定的采样周期 T，一旦确定了 KP、TI 、TD，只要使用前后三次测量值的偏差，即可由式(4.7)求出控制增量。

增量式 PID 具有以下优点：

(1) 由于计算机输出增量，所以误动作时影响小，必要时可用逻辑判断的方法关掉。

(2) 手动/自动切换时冲击小，便于实现无扰动切换。此外，当计算机发生故障时，由于输出通道或执行装置具有信号的锁存作用，故能保持原值。

(3) 算式中不需要累加。控制增量Δu(k)的确定仅与最近 k 次的采样值有关，所以较容易通过加权处理而获得比较好的控制效果。

但增量式 PID 也有其不足之处：积分截断效应大，有静态误差；溢出的影响大。使用时，常选择带死区、积分分离等改进 PID 控制算法。

4.3.3PID 参数整定

运用 PID 控制的关键是调整 KP、KI、KD 三个参数，即参数整定。PID 参数的整定方法有两大类：一是理论计算整定法。它主要是依据系统的数学模型，经过理论计算确定控制器参数；二是工程整定方法，它主要依赖工程经验，直接在控制系统的试验中进行，且方法简单、易于掌握，在工程实际中被广泛采用。由于智能车系统是机电高耦合的分布式系统，并且要考虑赛道的具体环境，要建立精确的智能车运动控制数学模型有一定难度，而且我们对车身机械结构经常进行修正，模型参数变化较为频繁，理论计算整定法可操作性不强，最终我们采用了工程整定方法。此外，我们先后实验了几种动态改变 PID 参数的控制方法。

4.4 转向舵机的 PID 控制算法

对于舵机的闭环控制，我们采用了位置式 PID 控制算法，根据往届的技术资料和实际测试，将每场图像的黑线中心加权平均值与舵机 PID 参考角度值构成一次线性关系。

在较低速(2m/s 以下)试验时，在偏离黑线很少的某个范围，将 Kp 直接置零，在偏离黑线较少的某个范围，将 Kp 值减小为原来的一半，在偏离较大的其他情况，则保持 Kp 原来的大小。取得的实际效果在弯道较多、直道较短的赛道上，车子转弯流畅，直道也能基本保持直线加速，车身左右抖动较小。

在提高车速至高速(2.5m/s 以上)时，我们发现车身在直道上特别是长直道上时，车身左右震荡比较严重，究其原因，硬件上，我们认为首先是轮轴本身的松动并且转向机构左右转向性能可能存在不对称性，设计有待改进，软件上，则是自身编写的 PID 舵机控制还不够精细，动态适应能力不够。在从弯道到直道的过程中，由于小车寻赛道本质上是一个随动系统，积分项在弯道累积的偏差错误地加在直道的跟踪上，造成在进入直道时转向不够准确，跑直道时虽然能跟踪黑线，但是转向调整往往超调，导致车身在直道上左右震荡，这种震荡严重影响了车的整体速度。此外，我们对 S 弯的控制也过于简单，没有特别的处理，导致车在跑 S 弯的时候，几乎完全沿弯走，没有明显的直冲 S 弯的效果，原因是在前瞻有限的情况下，在采集的图像中 S 弯入弯和普通弯道是一样的，导致小车

开始转向，由于中间一直检测到弯道，小车会沿 S 弯道左右震荡，同时相应会减速。

经过反复调试 PID 参数，我们发现只调整 PID 参数很难使车在跑 S 弯和长直道时都选择最佳路径，同时不影响在普通弯处的转向。这就要求系统能够智能地识别出当前赛道是哪种类型，我们没有选择赛道记忆等方法，而采取在不降低远处分辨率的情况下，尽量让摄像头看得更远的方法。最后，在 MCU 超频的条件下，每行图像采集了 210 个点，成功地增大了 CMOS 摄像头采集图像的分辨率。在透视问题影响远处分辨率的制约下，使视场长度(视场最远处和最近处的距离)达到 2m 多，最远前瞻达到 2.20m，足以覆盖赛道中的各种赛道类型，使得我们在程序中并没有加入了对 S 弯、长直道以及大弯进行可靠识别的算法，仅仅根据中心位置动态改变 PID 参数，就得到了较好的控制效果。

经过反复测试，我们选择的 PID 调节策略是：

(1) 将积分项系数置零，我们发现相比稳定性和精确性，舵机在这种随动系统中对动态响应性能的要求更高。更重要的是，在 KI 置零的情况下，我们通过合理调节 Kp，发现车能够在直线高速行驶时仍能保持车身非常稳定，没有震荡，基本没有必要使用 KI 参数；

(2) 微分项系数 KD 使用定值，原因是舵机在一般赛道中都需要较好的动态响应能力；

(3) 对 Kp，我们使用了二次函数曲线，Kp 随中心位置与中心值的偏差呈二次函数关系增大，在程序中具体代码如下：

loca_Kp = (loca_error * loca_error)/2 + 1000

其中，local_error 是中心位置与中心值的偏差。

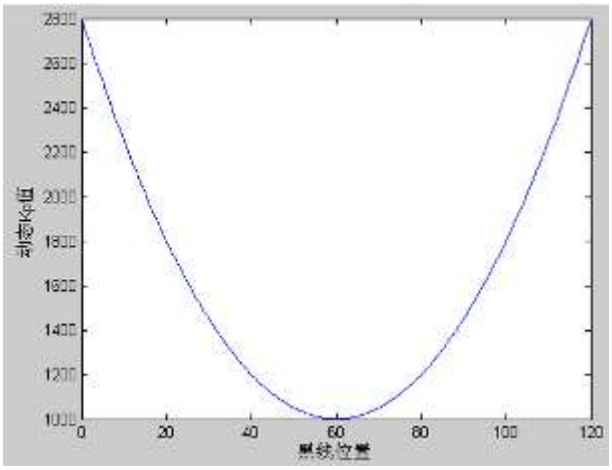

图 4.20 中心位置和动态 Kp 值的二次函数曲线

经不断调试，最终我们选择了一组 PID 参数，得到了较为理想的转向控制效果。

4.5 驱动电机的 PID 控制算法

对于速度控制，我们采用了增量式 PID 控制算法，基本思想是直道加速，弯道减速。经过反复调试，将每场图像得到的黑线位置与速度 PID 参考速度值构成二次曲线关系。在实际测试中，我们发现小车直道和弯道相互过渡时加减速比较灵敏，与舵机转向控制配合得较好。

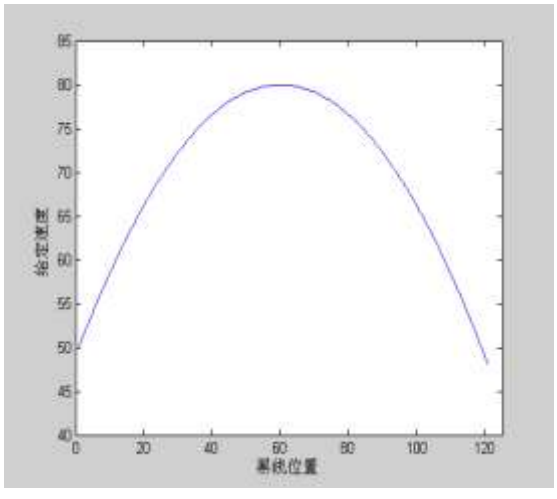

图 4.21 黑线位置和给定速度的二次函数曲线

在程序中具体代码如下：

sPID.vi_Ref = g_HighestSpeed - (59 - g_Control) * (59 - g_Control) * (g_HighestSpeed- g_LowestSpeed）/ 3481

其中，g_HighestSpeed 为最高速，g_LowestSpeed 为最低速，g_Control 为黑线位置，取值范围为 0～120，图 4.10 中，g_HighestSpeed 为 80，g_LowestSpeed 为 50。

但是，该方法存在一定的局限。一方面是车在从弯道入直道时加速和从直道入弯道时减速达不到最好的控制效果，弯道入直道减速不够快速，直道入弯道加速的时机不够及时。因此我们做了进一步的改进，根据入弯时黑线位置的特点动态改变二次曲线中最高点(直道的最高速度)和最低点(弯道的最低速度)的大小，结果表明，控制效果更好。另一方面是没有考虑到实际比赛中长直道急速冲刺的情况，赛前在程序中人为设定直线速度不够灵活不够合理，所以我们在程序中根据赛道状态动态提高了直线速度 g_HighestSpeed，使车能够在长直道上充分发挥潜能。

第五章　开发工具、制作、安装、调试过程说明

5.1 开发工具

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 上位机图像显示

5.2.1C#静态上位机

为了观察摄像头采集图像的直观效果，我们还采用了 VS2008 的 C#作为辅助开发调试工具。

我们设计的智能车系统采用 CMOS 摄像头采集赛道信息，分析处理之后用来编写黑线识别及控制算法。虽然直接将摄像头通过视频接口连接到电视可观察到摄像头所采的图像，但对于图像分析不够方便，且无法实时精确地反馈出一些特定信息。我们在 VS2008 的 C#环境下开发了一套基于 PC 机平台的图像显示与处理程序，可完成赛道显示及相关参数的实时反馈，运行界面如图 5.1 所示。

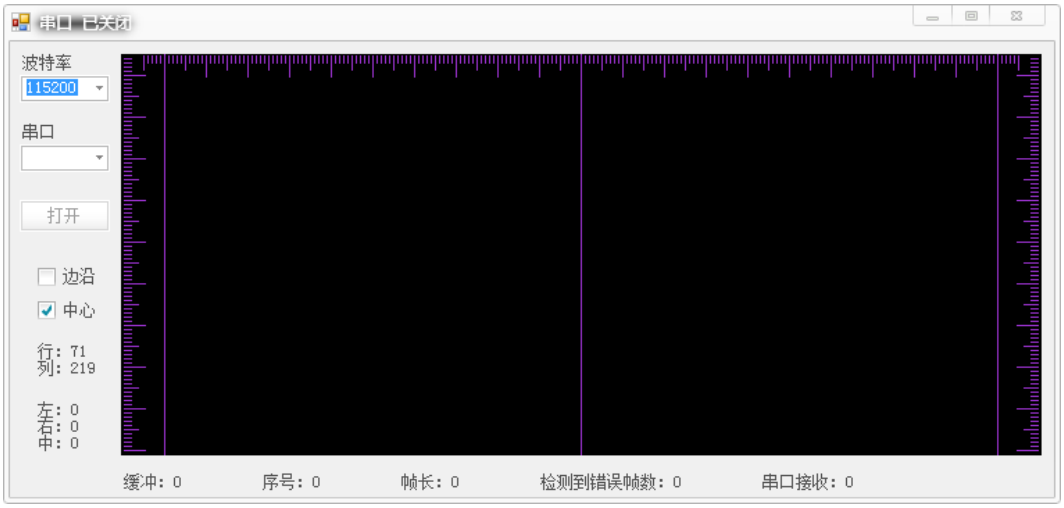

图 5.1 C#程序主界面

显示区域可原始图像及处理后的中心点，这为控制算法的编写提供了非常好的依据，也大大减少了调试者的工作量。

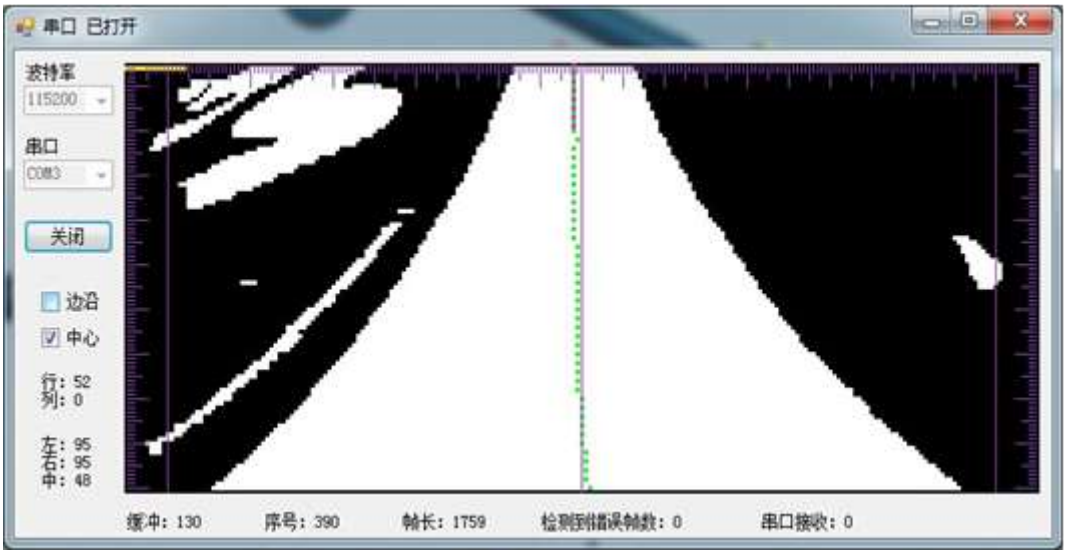

图 5.2 C#图像显示界面

5.2.2MFC SD 卡上位机

我们在 VS2008 的 C++环境下用 MFC 编写了 SD 卡上位机，每次模型车跑完全程之后，通过该上位机程序可得到运动过程中的每一场图像及相关数据及曲线，并可使用原始数据进行相关算法的模拟以及校正效果的显示（上位机主界面见图 5.3）。

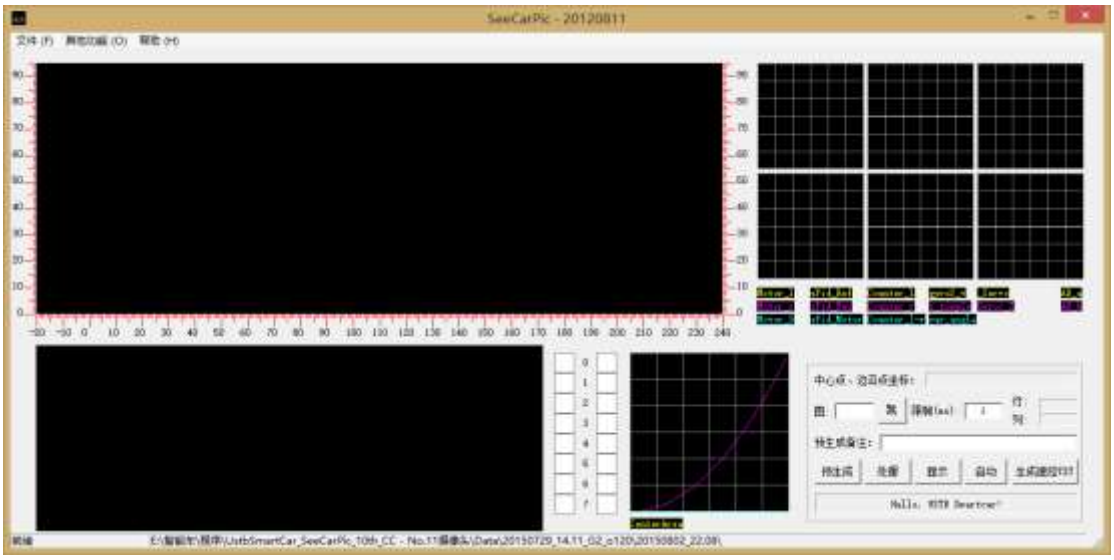

图 5.3 SD 卡上位机主界面

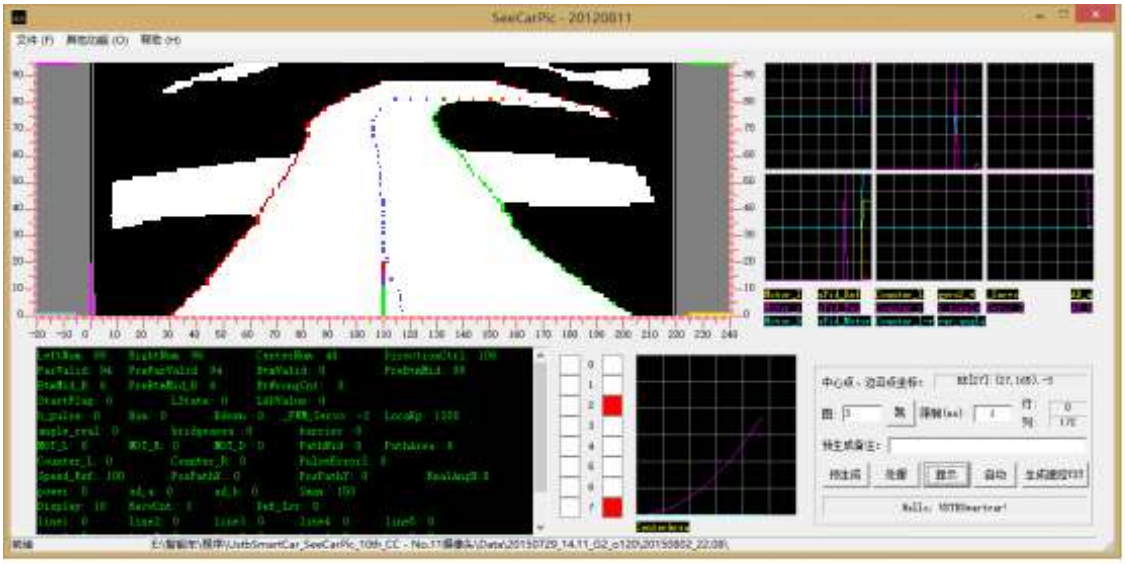

图 5.4 SD 卡上位机运行界面

模型车在运动过程中，记录下的典型赛道图像，如下图 5.5～5.8 所示。

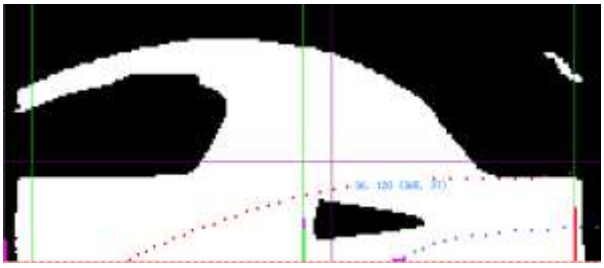

图 5.5 SD 卡记录的三角图像

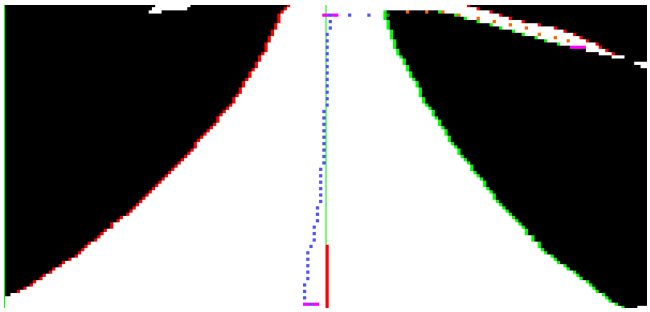

图 5.6 SD 卡记录的直道图像

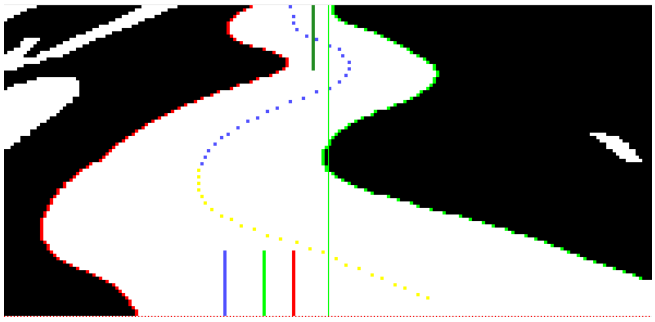

图 5.7 SD 卡记录的 S 型弯道图像

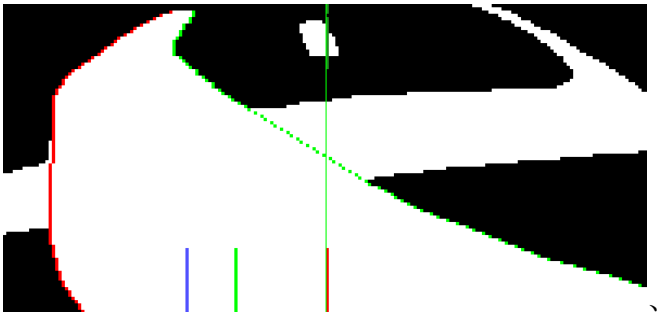

图 5.8 SD 卡记录的十字交叉图像

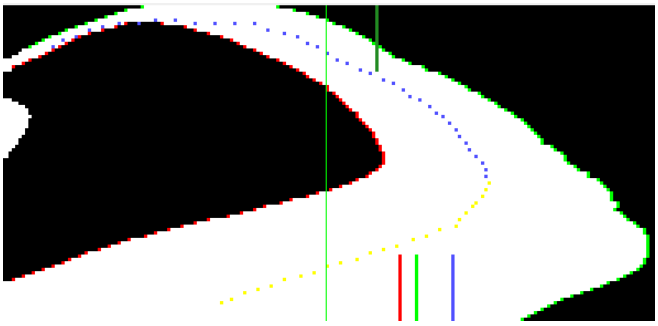

图 5.9 SD 卡记录的普通弯道图像

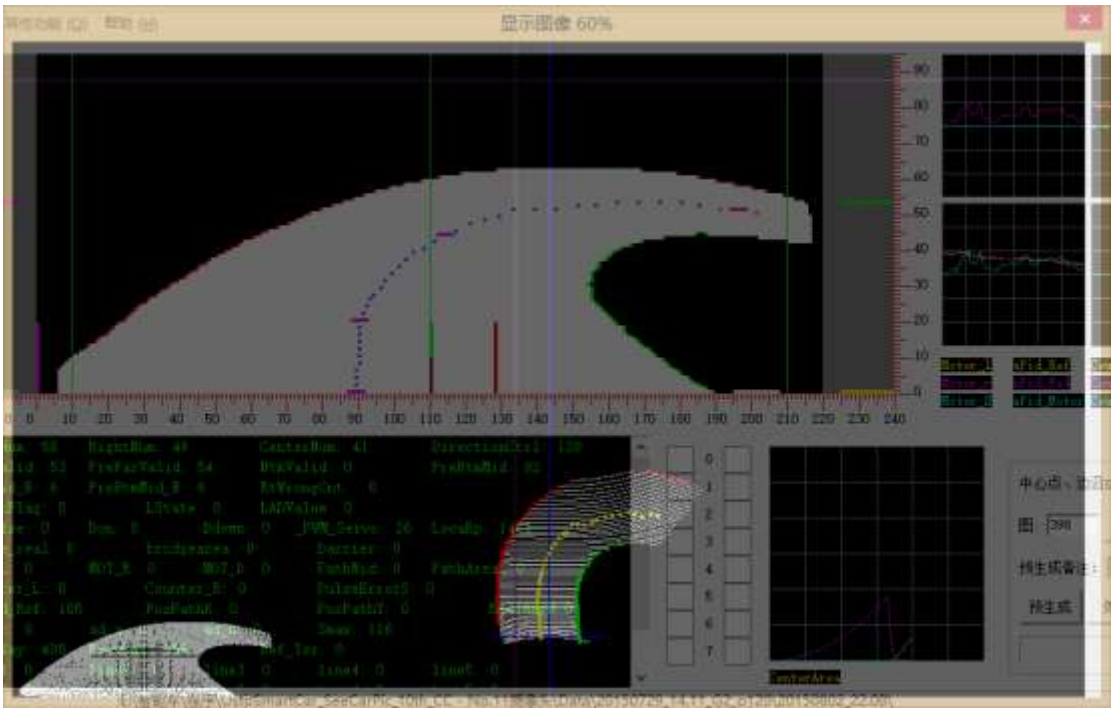

图 5.10 上位机校正效果显示

5.3SD 卡模块

5.3.1SD 卡介绍

SD 卡(Secure Digital Memory Card)是一种基于半导体快闪存的新一代记忆设备。由日本松下、东芝及美国 SanDisk 公司于 1999 年 8 月共同开发研制，其大小犹如一张邮票，重量只有 2 克，却拥有高记忆容量、快速数据传输率、极大的移动灵活性以及很好的安全性。SD 卡的数据存储管理可以类似于硬盘的磁盘管理系统，以 FAT 格式来存储数据。SD 卡的接口支持 SD 模式和 SPI 模式，主机系统可以选择其中任一模式。SPI 模式允许简单通用的 SPI 通道接口，这种模式相对于 SD 模式的不足之处是降低了速度。由于飞思卡尔系列单片机拥有 SPI 接口，所以我们使用了 SD 卡的 SPI 模式。

5.3.2SPI 总线介绍

SPI (Serial Peripheral Interface，串行外围设备接口总线) 总线技术是 MOTOROLA 公司推出的一种同步串行总线接口, 它是目前单片机应用系统中最

常用的几种串行扩展接口之一。SPI 总线主要通过三根线进行数据传输: 同步时钟线 SCK，主机输入/从机输出数据线 MISO、主机输出/从机输入数据线 MOSI，另外还有一条低电平有效的从机片选线 CS。SPI 系统的片选信号以及同步时钟脉冲由主机提供。SPI 总线模式的数据是以字节为单位进行传输的，每字节为 8 位，每个命令或者数据块都以字节对齐的(8 个时钟的整数倍)。主机与 SD 卡的各种通信都由主机控制，主机在对 SD 卡进行任何操作前都必须先要拉低 SD 卡的片选信号 CS (card select) ,然后由主机向 SD 卡发送命令, SD 卡对主机发送的任何命令都要进行响应，不同的命令会有不同的响应格式(1 个字节或 2 个字节响应)。SD 卡除了对命令响应外，在执行写操作时，还要对主机发送的每个数据块进行响应(向主机发送一个特殊的数据响应标志)。

5.3.3 软件实现

首先需要将 SPI 模块设置为主机模式，并设置相关的寄存器使 SPI 模块有高速和低速之分。SD 卡的软件设计主要包括两部分内容：SD 卡的上电初始化过程和对 SD 卡的读写操作。对 SD 卡初始化程序流程如图 5.11 所示。

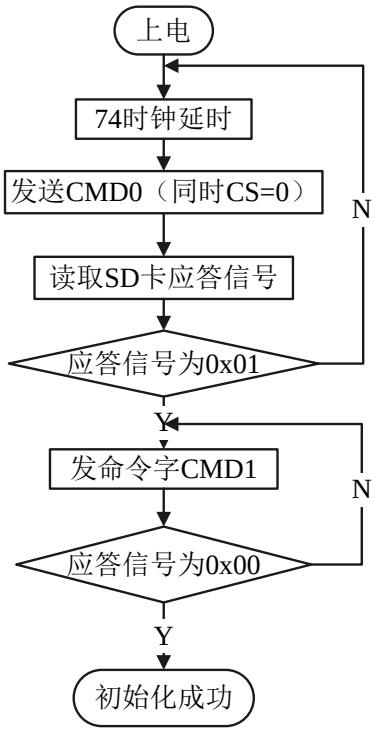

图 5.11　SD 卡初始化步骤

SD 卡上电后，主机必须先向 SD 卡发送 74 个时钟周期，以完成 SD 卡上电过程。SD 卡上电后会自动进入 SD 总线模式，并在 SD 总线模式下向 SD 卡发送复位命令(CMD0)，若此时片选信号 CS 处于低电平态，则 SD 卡进入 SPI 总线模式，否则 SD 卡工作在 SD 总线模式。SD 卡进入 SPI 工作模式会发出应答信号，若主机读到的应答信号为 01，即表明 SD 卡已进入 SPI 模式，此时主机即可不断地向 SD 卡发送命令字(CMD1) 并读取 SD 卡的应答信号，直到应答信号为 00，以表明 SD 卡已完成初始化过程，准备好接受下一命令。

此后，系统便可读取 SD 卡的各寄存器，并进行读写等操作，每次读写数据都是按照扇区操作的，每次操作 512 字节。

第六章　模型车的主要技术参数说明

赛车基本参数	长	30cm
	宽	16cm
	高	34cm
车重		1030g
功耗	空载	10W
	带载	大于 12W
电容总容量		1800uF
传感器	CMOS 摄像头	1 个
	陀螺仪	1 个
	编码器	1 个
	蓝牙	1 个
	超声波	1 对
除了车模原有的驱动电机、舵机之外伺服电机个数		0
赛道信息检测	视野范围（近瞻/远瞻）	20cm/250cm
	精度(近/远)	2/12.5mm
	频率	50Hz

结论

自报名参加“恩智浦”杯智能汽车竞赛以来，我们小组成员从查找资料、设计机构、组装车模、编写程序一步一步的进行，最后终于完成了最初目标，定下了现在这个设计方案。

在此份技术报告中，我们主要介绍了准备比赛时的基本思路，包括机械、电路以及最重要的控制算法的创新思想。在机械结构方面，我们分析了整车质量分布，调整重心位置，优化机械结构。在电路方面，我们以模块形式分类，在最小系统、主板、电机驱动等模块分别设计，在查找资料的基础上各准备了几套方案；然后我们分别实验，最后以报告中所提到的形式决定了我们最终的电路图。在程序方面，我们使用 C 语言编程，利用比赛推荐的开发工具调试程序，经过小组成员不断讨论、改进，终于设计出一套比较通用稳定的程序。在这套算法中，我们结合路况调整车速，做到直道加速、弯道减速，保证在最短时间内跑完全程。

在这几个月的备战过程中，场地和经费方面都得到了学校和学院的大力支持，在此特别感谢一直支持和关注智能车比赛的学校和学院领导以及各位指导老师、指导学长，同时也感谢比赛组委会能组织这样一项有意义的比赛。

现在，面对即将到来的大赛，在历时近五个月的充分准备以及华北赛的考验之后，我们有信心在全国比赛中取得优异成绩。也许我们的知识还不够丰富，考虑问题也不够全面，但是这份技术报告作为我们小组辛勤汗水的结晶，凝聚着我们小组每个人的心血和智慧，随着它的诞生，这份经验将永伴我们一生，成为我们最珍贵的回忆。

参考文献

[1]邵贝贝．单片机嵌入式应用的在线开发方法［M］．北京．清华大学出版社．2004.

[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]王晓明．电动机的单片机控制［M］．北京：北京航空航天大学出版社．2002.

[4]安鹏，马伟．S12 单片机模块应用及程序调试[J]．电子产品世界．2006．第 211 期． 162-163.

[5]张文春．汽车理论[M]．北京．机械工业出版社．2005.

[6]童诗白，华成英．模拟电子技术基础［M］．北京: 高等教育出版社，2001.

[7]阎石．数字电子技术基础［M］．北京: 高等教育出版社，2000.

[8]谭浩强著．C 程序设计．北京：清华大学出版社，2003.

[9]尹勇．Protel DXP 电路设计入门与进阶［M］．北京: 科学出版社，2004.

[10]Park K.H ，Bien Z，Hwang D.H. A study on the robustness of a PID - type iterative learning controller against initial state error [J]. Int. J. Syst. Sci. 1999, 30(1) ，102～135.

[11]殷剑宏，吴开亚．图论及其算法［M］．中国科学技术大学出版社，2003.

[12]夏克俭．数据结构及算法［M］．北京：国防工业出版社， 2001.

[13]尹怡欣，陶永华．新型 PID 控制及其应用．北京：机械工业出版社，1998 年.

[14]李太福．基于在线参数自整定的模糊 PID 伺服控制系统[J]．交流伺服系统，2005，4：203～215.

[15]仲志丹，张洛平，张青霞．PID 调节器参数自寻优控制在运动伺服中的应用[J]．洛阳工学院学报，2000，21（1）：57～60.

[16]卓晴，黄开胜，邵贝贝．学做智能车：挑战“飞思卡尔”杯．北京：北京航空航天大学出版社，2007 年.

附录：程序源代码

```
signed long int loca_PIDCalc(struct PID *pp)
{
    signed long loca_error = 0;
    signed long loca_derror = 0;
    pp->loca_Ref = Set_Mid_Point;
    loca_error = pp->loca_Ref - pp->loca_FeedBack;
    sPID.loca_Kp = 1250 + (((MAX_VIDEO_USEDLINE - v_FarthestValidLine)
* (MAX_VIDEO_USEDLINE - v_FarthestValidLine)) / 6);
    if ((loca_error < LOCA_DEADLINE) && (loca_error > -LOCA_DEADLINE))

    { ;  // 不执行 PID 调节      }// 设置调节死区
    else    // 执行位置 PID 调节
    {
        loca_derror = loca_error - pp->loca_PreError;     // 计算微分项偏差
       pp->loca_PreIntegral += loca_error;                  // 存储当前积分偏差
         pp->loca_PreError = loca_error;                        // 存储当前偏差
         pp->loca_PreU = pp->loca_Kp * loca_error + pp->loca_Ki *
pp->loca_PreIntegral + pp->loca_Kd * loca_derror;    // 位置 PID 算法

         if (pp->loca_PreU >= LOCA_MAX)   // 防止调节溢出
       {
```

```
            pp->loca_PreU = LOCA_MAX;
        }
        else if (pp->loca_PreU <= LOCA_MIN)
        {
            pp->loca_PreU = LOCA_MIN;
        }
    }
    return(pp->loca_PreU / 1000);
}
INT32S speed_PIDCalc(struct PID_Speed *pp )
{
    INT32S  error,d_error,dd_error;
    error = pp->vi_Ref - pp->vi_FeedBack;          //偏差计算(积分)

    d_error = error - pp->vi_PreError;             //偏差计算(比例)
    dd_error = d_error - pp->vi_PreDerror;         //偏差计算(微分)

    pp->vi_PreError = error;                       //存储当前偏差
    pp->vi_PreDerror = d_error;
    if ((error < VV_DEADLINE) && (error > -VV_DEADLINE))// 设置调节死区
    {
        ; // 不执行 PID 调节
    }
```

```
        else//  速度 PID 计算
        {
            pp->vl_PreU += ((pp->v_Kp * d_error) + (pp->v_Ki * error) + (pp->v_Kd
* dd_error));
        }

        if (pp->vl_PreU >= VV_MAX)                    //防止调节溢出
        {
            pp->vl_PreU = VV_MAX;
        }
        else if (pp->vl_PreU <= vv_min)
        {
            pp->vl_PreU = vv_min;
        }
        return (pp->vl_PreU / 10);              //PID 返回值(电机 PWM 波占空比)
    }
```