

# 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：北京科技大学摄像头组

参赛队员：陈伟、金麟、刘舸

带队教师：刘立、杨珏

## 关于技术报告和学术论文使用授权的说明

本人完全了解第十一届“恩智浦”杯全国大学生智能汽车竞赛有关留、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： 陈伟

带队教师签名： 何晓

日期： 2016年8月15日

## 摘 要

本文设计的智能车系统以 K66P144M180SF5RMV2 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，提取黑色边缘线，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

摘 要 .....	IV
目 录 .....	V
引 言 .....	- 1 -
第一章 方案设计 .....	- 1 -
1.1 系统总体方案的设计 .....	- 1 -
第二章 智能车机械结构调整与优化 .....	- 2 -
2.1 智能车车体机械建模 .....	- 2 -
2.2 智能车前轮定位的调整 .....	- 3 -
2.2.1 主销后倾角 .....	- 3 -
2.2.2 主销内倾角 .....	- 4 -
2.2.3 车轮外倾角 .....	- 4 -
2.2.4 前轮前束 .....	- 5 -
2.3 智能车转向机构调整优化 .....	- 5 -
2.4 智能车后轮减速齿轮机构调整 .....	- 7 -
2.5 轮胎的保养与使用 .....	- 7 -
2.6 编码器的安装 .....	- 7 -
2.7 智能车重心位置的调整 .....	- 8 -
2.8 其它机械结构的调整 .....	- 8 -
2.9 摄像头的安装 .....	- 10 -
第三章 电路设计说明 .....	- 11 -
3.1 硬件方案设计 .....	- 11 -
3.2 传感器选择 .....	- 11 -

3.2.1 摄像头选择 .....	- 11 -
3.2.2 陀螺仪选择 .....	- 13 -
3.2.3 编码器选择 .....	- 13 -
3.3 电路模块实现 .....	- 15 -
3.3.1 电源管理模块 .....	- 15 -
3.3.2 电机驱动模块 .....	- 16 -
3.3.3 视频处理模块 .....	- 19 -
3.3.4 接口及外接模块 .....	- 20 -
第四章 智能车控制软件设计说明 .....	- 23 -
4.1 赛道中心线提取及优化处理 .....	- 23 -
4.1.1 原始图像的特点 .....	- 23 -
4.1.2 赛道边沿提取 .....	- 25 -
4.1.3 图像矫正 .....	- 27 -
4.1.4 推算中心 .....	- 29 -
4.1.5 路径选择 .....	- 30 -
4.2 折点求取原理简介 .....	- 32 -
4.3PID 控制算法介绍 .....	- 34 -
4.3.1 位置式 PID .....	- 35 -
4.3.2 增量式 PID .....	- 36 -
4.3.3 PID 参数整定 .....	- 37 -
4.4 转向舵机的 PID 控制算法 .....	- 37 -
4.5 驱动电机的 PID 控制算法 .....	- 39 -
第五章 开发工具、制作、安装、调试过程说明 .....	- 41 -

5.1 开发工具 .....	- 41 -
5.2 上位机图像显示 .....	- 41 -
5.2.1 C#静态上位机 .....	- 41 -
5.2.2 MFC SD 卡上位机 .....	- 43 -
5.3 SD 卡模块 .....	- 46 -
5.3.1 SD 卡介绍 .....	- 46 -
5.3.2 SPI 总线介绍 .....	- 46 -
5.3.3 软件实现 .....	- 47 -
第六章 模型车的主要技术参数说明 .....	- 49 -
结 论 .....	- 50 -
参考文献 .....	I
附录：程序源代码 .....	III

## 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 B 车模，采用恩智浦 32 位微控制器 K66P144M180SF5RMV2 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、陀螺仪角度采集和计算、电机驱动输出，最终实现一套能够自主识别路线控制系统。

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。

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

## 1.1 系统总体方案的设计

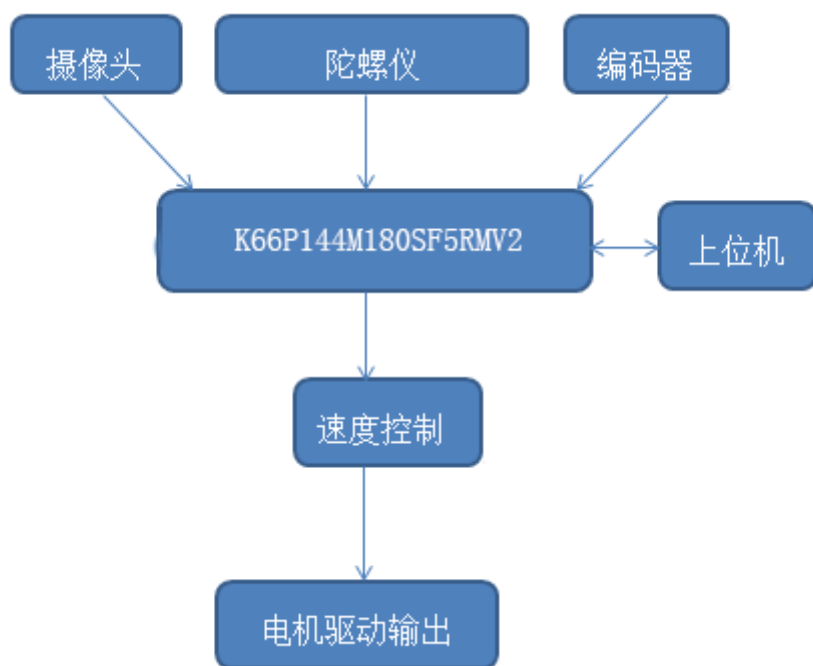


图 1.1 系统结构图

摄像头车的系统整体结构如图所示。K66P144M180SF5RMV2 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；通过采集陀螺仪的模拟信号，得到车身倾角姿态的状态数据；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。通过微处理器对图像处理，对角度、速度进行 PID 控制，最后以 PWM 波输出驱动电机。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

智能车系统任何的控制都是在一定的机械结构基础上实现的，因此在设计整个软件架构和算法之前一定要对整个车模的机械结构有一个感性的认识，然后建立相应的数学模型。从而再针对具体的设计方案来调整赛车的机械结构，并在实际的调试过程中不断的改进和提高。本章将主要介绍智能车车模的机械结构和调整方案。

### 2.1 智能车车体机械建模

此次比赛选用的赛车车模采用 B 型车模。赛车机械结构只使用竞赛提供车模的底盘部分及转向和驱动部分。控制采用前轮转向，后轮驱动方案。

我们对机械结构的要求是：简单而高效。我们在不断的尝试后确定了以下的设计方案：

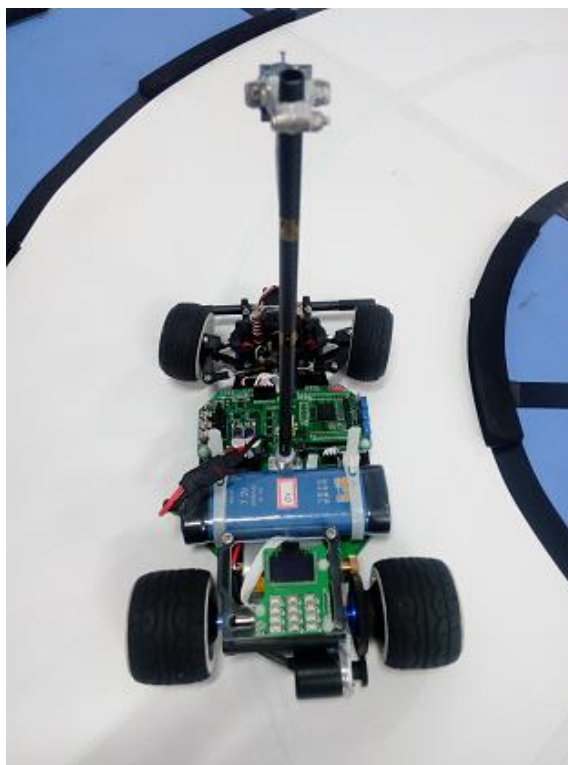


图 2.1 智能车 B 车器件布局图

表 2.1 B 模型车基本参数

	尺寸
轴距	200mm
前轮距	138mm
后轮距	140mm
总长	750mm
总高	340mm
总重	1350g

2.2 智能车前轮定位的调整

现代汽车在正常行驶过程中，为了使汽车直线行驶稳定，转向轻便，转向后能自动回正，减少轮胎和转向系零件的磨损等，在转向轮、转向节和前轴之间须形成一定的相对安装位置，叫车轮定位，其主要的参数有：主销后倾、主销内倾、车轮外倾和前束。智能车竞赛模型车的四项参数都可以调整，但是由于模型车加工和制造精度的问题，在通用的规律中还存在着一些偶然性。

2.2.1 主销后倾角

所谓主销后倾，是将主销（即转向轴线）的上端略向后倾斜。从汽车的侧面看去，主销轴线与通过前轮中心的垂线之间形成一个夹角，即主销后倾角。主销后倾的作用是增加汽车直线行驶时的稳定性和在转向后使前轮自动回正。由于主销后倾，主销（即转向轴线）与地面的交点位于车轮接地点的前面。这时，车轮所受到的阻力的作用点总是在主销轴线之后，相当于主销拖着车轮前进。这样，就能保持行驶方向的稳定性。当车转弯时，由于车轮所受阻力作用线，不通过主销轴线，这样，车轮所受阻力在主销方向有力矩作用产生，迫使车轮自动偏转直到到车轮所受阻力作用线通过主销轴线，此时，车轮已回正，这就是转向车轮的自动回正功能。

主销后倾角越大，方向稳定性越好，自动回正作用也越强，但转向越沉重。汽车主销后倾角一般不超过3°，由前悬架在车架上的安装位置来保证。现代轿车由于采用低压宽幅子午线轮胎，高速行驶时轮胎的变形加大，接地点后移，因此主销后倾角可以减小，甚至为负值（变成主销前倾），以避免

由于回正力矩过大而造成前轮摆振。

模型车通过增减黄色垫片的数量来改变主销后倾角的，由于竞赛所用的转向舵机力矩不大，过大的主销后倾角会使转向变得沉重，转弯反应迟滞，所以设置为 $0^{\circ}$ ，以便增加其转向的灵活性。

### 2.2.2 主销内倾角

所谓主销内倾，是将主销（即转向轴线）的上端向内倾斜。从汽车的前面看去，主销轴线与通过前轮中心的垂线之间形成一个夹角，即主销内倾角。主销内倾的作用是使车轮转向后能及时自动回正和转向轻便。对于模型车，通过调整前桥的螺杆的长度可以改变主销内倾角的大小，由于过大的内倾角也会增大转向阻力，增加轮胎磨损，所以在调整时可以近似调整为 $0^{\circ} \sim 3^{\circ}$ 左右，不宜太大。

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。不同之处是主销内倾的回正与车速无关，主销后倾的回正与车速有关，因此高速时主销后倾的回正作用大，低速时主销内倾的回正作用大。

### 2.2.3 车轮外倾角

前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。如果车轮垂直地面一旦满载就易产生变形，可能引起车轮上部向内倾侧，导致车轮联接件损坏。所以事先将车轮校偏一个正外倾角度，一般这个角度约在 $1^{\circ}$ 左右，以减少承载轴承负荷，增加零件使用寿命，提高汽车的安全性能。

模型车提供了专门的外倾角调整配件，近似调节其外倾角。由于竞赛中模型主要用于竞速，所以要求尽量减轻重量，其底盘和前桥上承受的载荷不大，所以外倾角调整为 $0^{\circ}$ 即可，并且要与前轮前束匹配。

### 2.2.4 前轮前束

所谓前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。前轮在滚动时，其惯性力自然将轮胎向内偏斜，如果前束适当，轮胎滚动时的偏斜方向就会抵消，轮胎内外侧磨损的现象会减少。像内八字那样前端小后端大的称为“前束”，反之则称为“后束”或“负前束”。在实际的汽车中，一般前束为  $0.12\text{mm}$ 。

在模型车中，前轮前束是通过调整伺服电机带动的左右横拉杆实现的。主销在垂直方向的位置确定后，改变左右横拉杆的长度即可以改变前轮前束的大小。在实际的调整过程中，我们发现较小的前束，约  $0.2\text{mm}$  可以减小转向阻力，使模型车转向更为轻便，但实际效果不是十分明显。

虽然模型车的主销后倾角、主销内倾角、车轮外倾角和前束等均可以调整，但是由于车模加工和制造精度的问题，在通用的规律中还存在着不少的偶然性，一切是实际调整的效果为准。

## 2.3 智能车转向机构调整优化

理想的转向模型，是指在轮胎不打滑时，忽略左右两侧轮胎由于受力不均产生的变形，忽略轮胎受重力影响下的变形时车辆的转向建模。在这种理想的模型下，车体的转向半径可以计算得到。

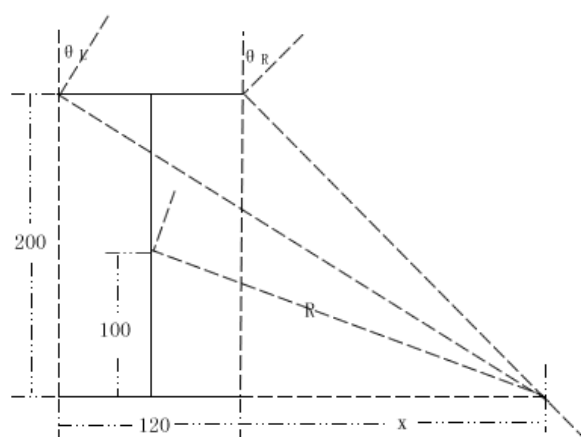


图 2.2 智能车转向示意图

如图 2.2，假设智能车系统为理想的转向模型，且其重心位于其几何中心。车轮满足转向原理，左右轮的轴线与后轮轴线这三条直线必然交于一点。

转向机构在车辆运行过程中有着非常重要的作用。合适的前桥和转向机构可以保证在车辆直线行驶过程中不会跑偏，能保证车辆行驶的方向稳定性；而在车辆转向时，合适的转向机构可以使车辆自行回到直线行驶状态，具有好的回正性。正是由于这些原因，转向系统优化设计成为智能车设计中机械结构部分的重点，直接关系到赛车能否顺利地完比赛。在实际操作中，我们通过理论计算的方案进行优化，然后做出实际结构以验证理论数据，并在实际调试过程中不断改进。

在模型车制做过程中，赛车的转向是通过舵机带动左右横拉杆来实现的。转向舵机的转动速度和功率是一定，要想加快转向机构响应的速度，唯一的办法就是优化舵机的安装位置和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。经过最后的实际的参数设计计算，最后得出一套可以稳定、高效工作的参数及机构。

如图 2.3，我们最终设计的这套转向拉杆，我们综合考虑了速度与扭矩间的关系，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。

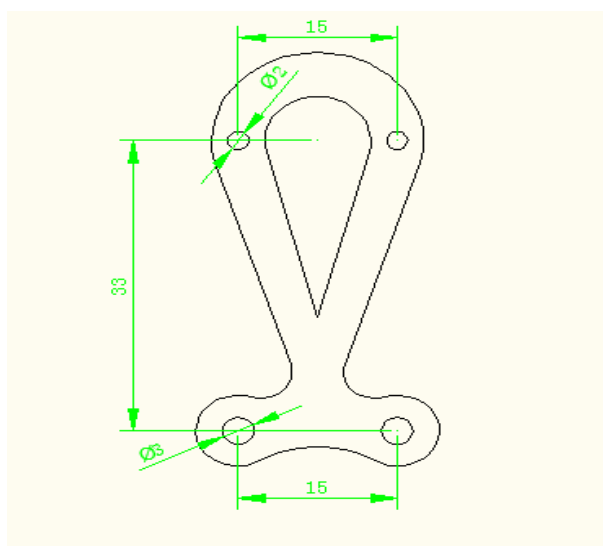


图 2.3 转向拉杆图

## 2.4 智能车后轮减速齿轮机构调整

模型车后轮采用 540 电机驱动，电机轴与后轮轴之间的传动比为 36：105（电机轴齿轮齿数为 36，后轴传动齿数为 105）。齿轮传动机构对车模的驱动能力有很大的影响。齿轮传动部分安装位置的不恰当，会大大增加电机驱动后轮的负载，会严重影响最终成绩。调整的原则是：两传动齿轮轴保持平行，齿轮间的配合间隙要合适，过松容易打坏齿轮，过紧又会增加传动阻力，浪费动力；传动部分要轻松、顺畅，不能有迟滞或周期性振动的现象。判断齿轮传动是否良好的依据是，听一下电机带动后轮空转时的声音。声音刺耳响亮，说明齿轮间的配合间隙过大，传动中有撞齿现象；声音闷而且有迟滞，则说明齿轮间的配合间隙过小，或者两齿轮轴不平行，电机负载变大。调整好的齿轮传动噪音很小，并且不会有碰撞类的杂音，后轮减速齿轮机构就基本上调整好了，动力传递十分流畅。

## 2.5 轮胎的保养与使用

本届比赛B车模采用材料为橡胶的带花纹轮胎（也可以使用光胎），在使用的过程中主要暴露出的问题有：高速行驶时轮胎抓地力不足；轮胎材料强度不足，长期使用容易发粘、磨损。规则中要求轮胎表面要有清晰可辨的花纹，因此在调试过程中，不可以对轮胎表面进行过度处理，我们尽量选择对称、圆的轮胎；尽可能改变前轮倾角，使轮胎与地面的接触面积在各个工况下综合最大；在不调试的时间里尽量避免四轮与地面接触，保证轮胎的形状不发生改变。

## 2.6 编码器的安装

为了提高精度，本车使用了编码器。为了便于安装，使用加工件将编码器与后桥进行连接，较好的保证了两啮合齿轮轴的平行度及传动的平顺性。

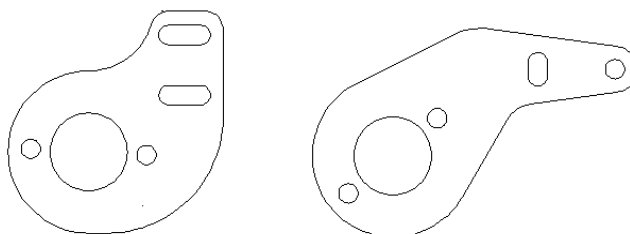


图 2.4 编码器安装与安装连片

## 2.7 智能车重心位置的调整

为了达到较远前瞻，必须把摄像头架到较高的位置，会引起车重心过高，导致甩尾。为了使重心降低，我们尝试了很多降低重心的方法方式，我们把电池重心降低，达到了预期的效果。

## 2.8 其它机械结构的调整

另外，在模型车的机械结构方面还有很多可以改进的地方，比如说车轮、悬架、底盘、车身高度等。

模型车在高速的条件下（ $2.3\text{m/s} \sim 3.5\text{m/s}$ ），由于存在快速变化的加减速过程，使得模型车的轮胎与轮辋之间很容易发生相对位移，可能导致在加速时会损失部分驱动力。在实验中调试表明，赛车在高速下每跑完一圈，轮胎与轮辋

之间通常会产生几个厘米的相对位移，严重影响了赛车的加速过程。为了解决这个问题，我们在实际调试过程中对车轮进行了粘胎处理，可以有效地防止由于轮胎与轮辋错位而引起的驱动力损失的情况。

此外，我们还对车身高度，以及底盘的形状和质量等，都进行了相应的改进和调整，均取得了不错效果。



## 2.9 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 2.3 所示。



图 2.5 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。从以下可以体现出：

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。第一种方法通过片内 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。

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，最终我们选用了欧姆龙的旋转编码器。

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。K66 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。

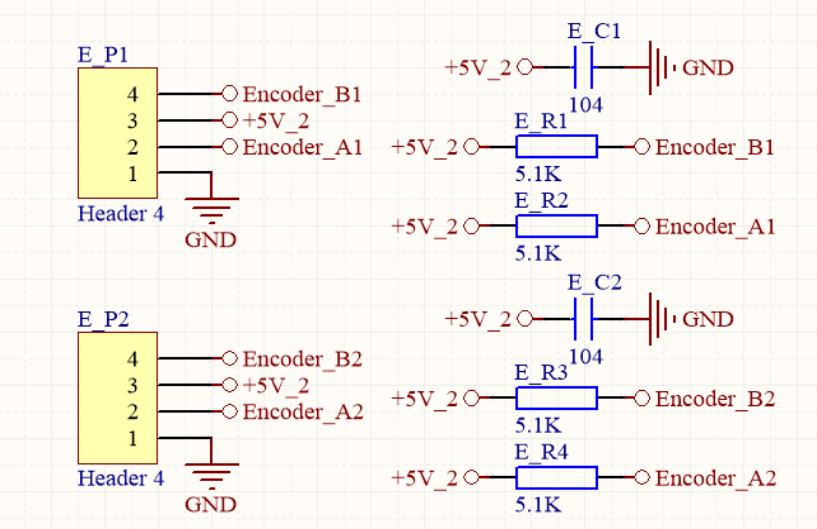


图 3.1 编码器的接口部分

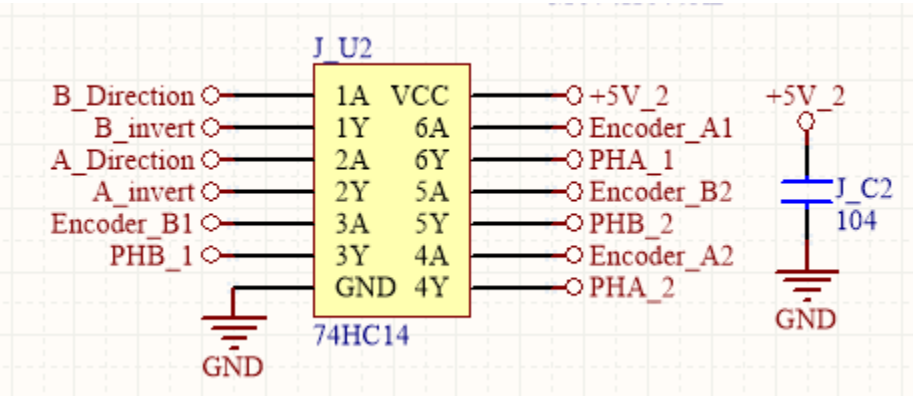


图 3.2 非门电路

### 3.3 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、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。

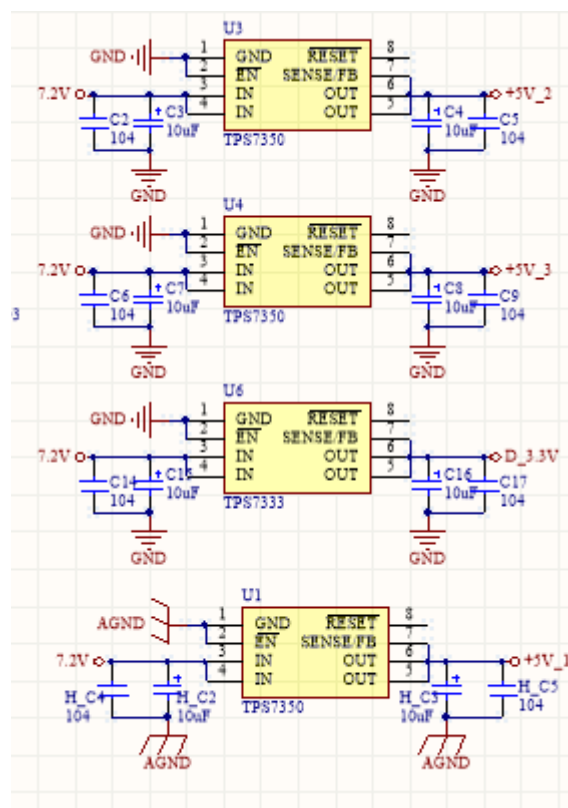


图 3.3 电源管理模块原理图

在电源管理模块中，我们选用了三片 TPS7350 和两片 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

本次摄像头组需要驱动 1 个电机，常用的电机驱动有两种方式：一、采用集成电机驱动芯片；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，恩智浦公司的 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。

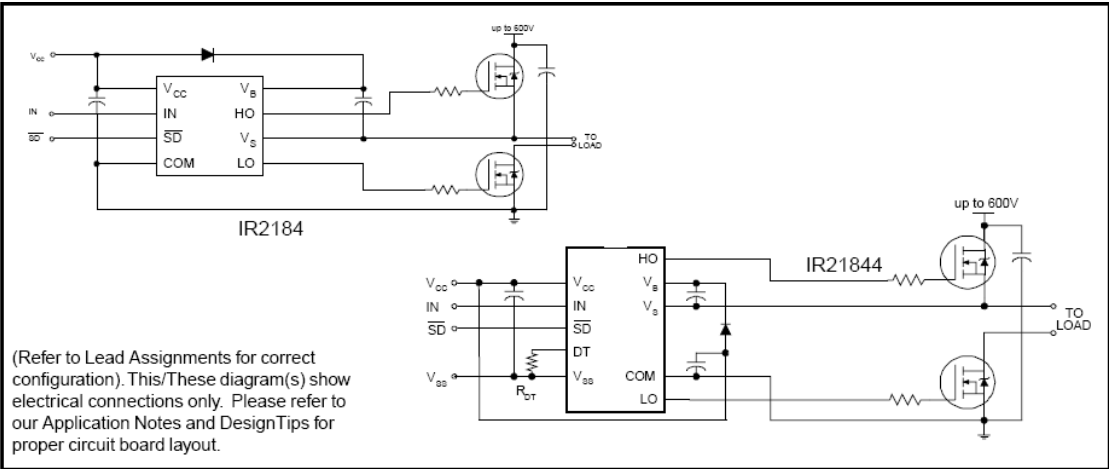


图 3.4 IR2184 应用图

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， $V_{DS} = 55$  伏、 $R_{DS(on)} = 8.0$  毫欧、 $I_D = 110$  安。

### 3、驱动电路的原理分析及元件参数确定

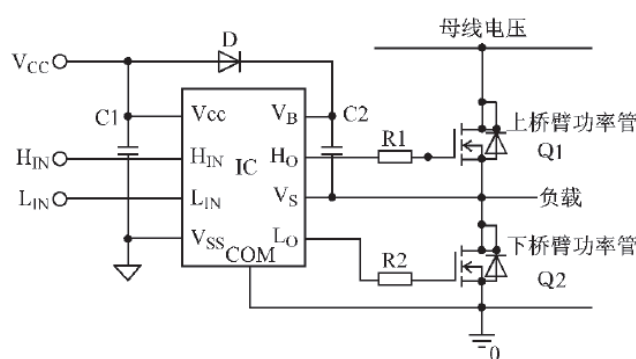


图 3.5 电机驱动分析图

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。 $V_b, V_s$  为高压端供电； $H_o$  为高压端驱动输出； $COM$  为低压端驱动供电， $L_o$  为低压端驱动输出； $V_{ss}$  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  $V_s$  脚的电位为下桥臂功率管  $Q_2$  的饱和导通压降，基本上接近地电位，此时  $V_{cc}$  通过自举二极管  $D$  对自举电容  $C_2$  充电使其接近  $V_{cc}$  电压。当  $Q_2$  关断时  $V_s$  端的电压就会升高，由于电容两端的电压不能突变，因此  $V_b$  端的电平接近于  $V_s$  和  $V_{cc}$  端电压之和，而  $V_b$  和  $V_s$  之间的电压还是接近  $V_{cc}$  电压。当  $Q_2$  开通时， $C_2$  作为一个浮动的电压源驱动  $Q_2$ ；而  $C_2$  在  $Q_2$  开通期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  $V_s$  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。

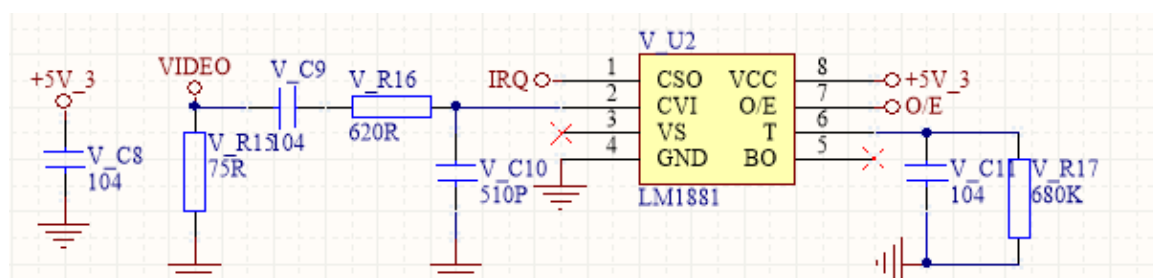


图 3.6 LM1881 外围电路原理图

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。

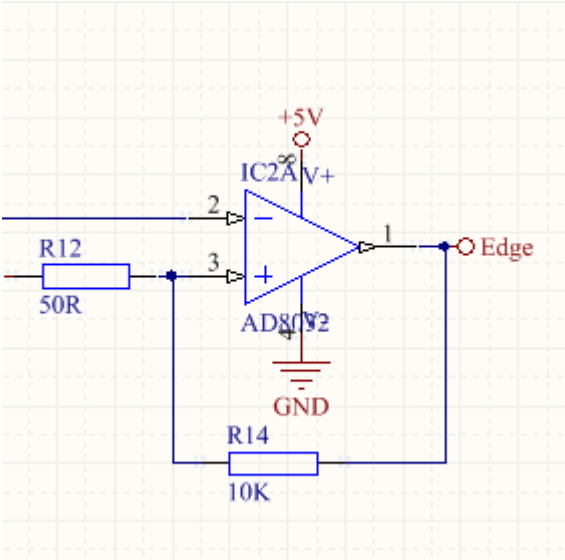


图 3.7 比较电路原理图

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、灯塔接收模块，我们在主板上设计了外接接口，用于连接。

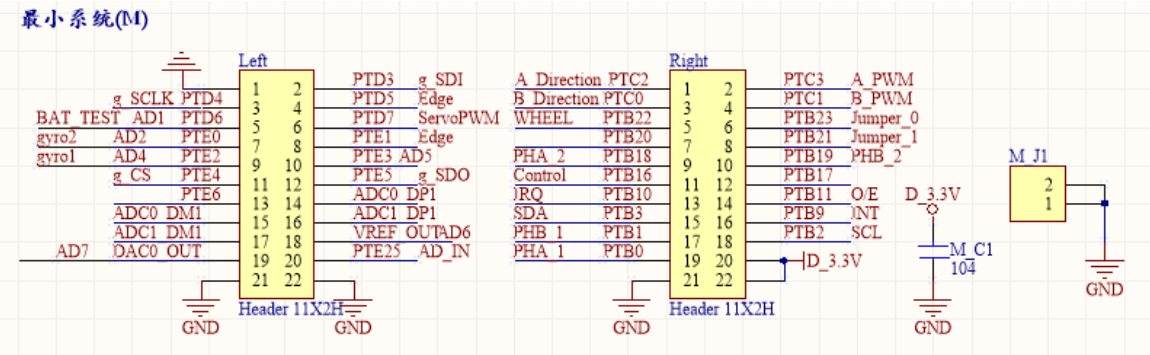


图 3.8 最小系统接口

K66P144M180SF5RMV2 是 K60 系列 MCU。Kinetis 系列微控制器是 Cortex-M4 系列的内核芯片。K66 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。

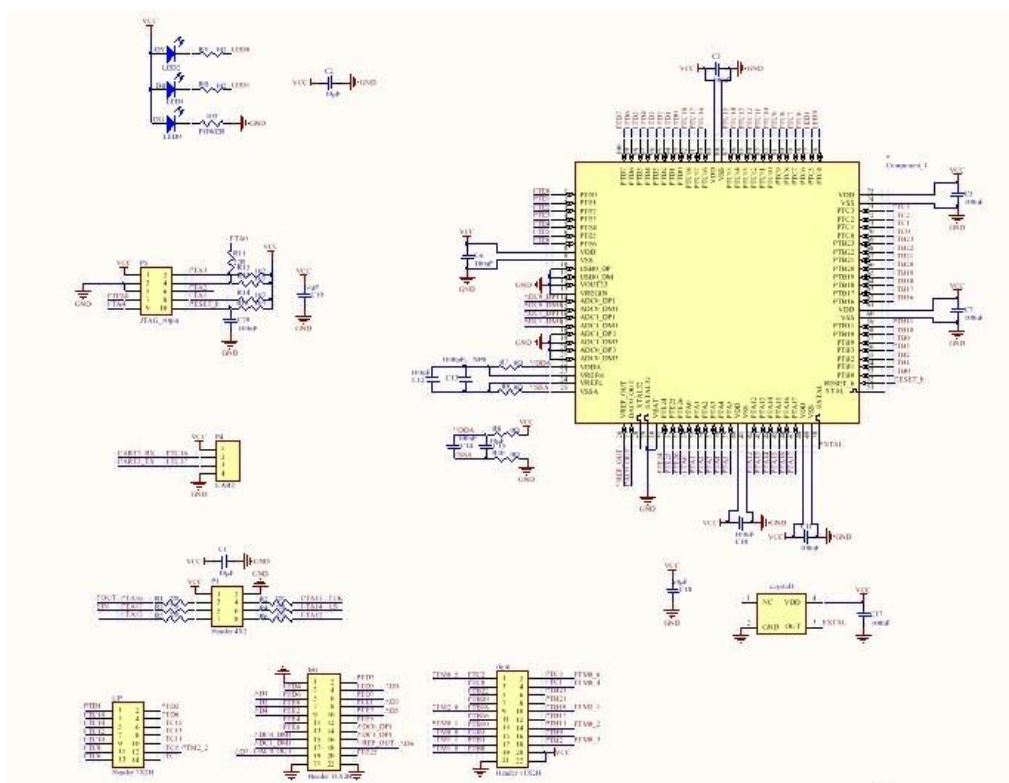


图 3.9 最小系统板

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	INT	键盘中断请求信号，低电平（下降沿）有效
15	RESET	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 <sup>2</sup> C 总线时钟信号
20	SDA	I <sup>2</sup> C 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

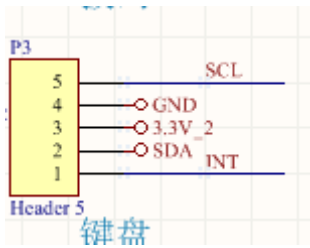


图 3.11 键盘接口

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 4.3 所示。

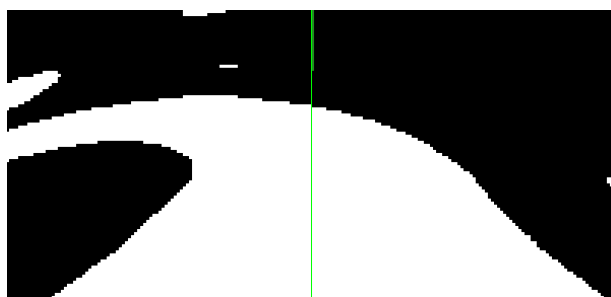


图 4.1 弯道原始图像

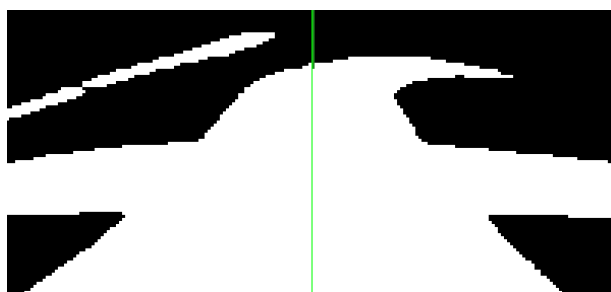


图 4.2 十字交叉原始图像

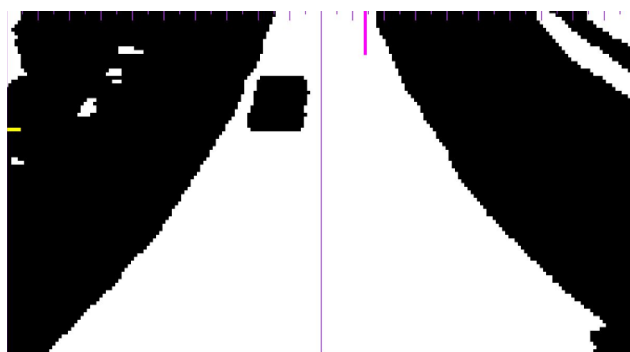


图 4.3 障碍原始图像

#### 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) 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 4.4 所示。



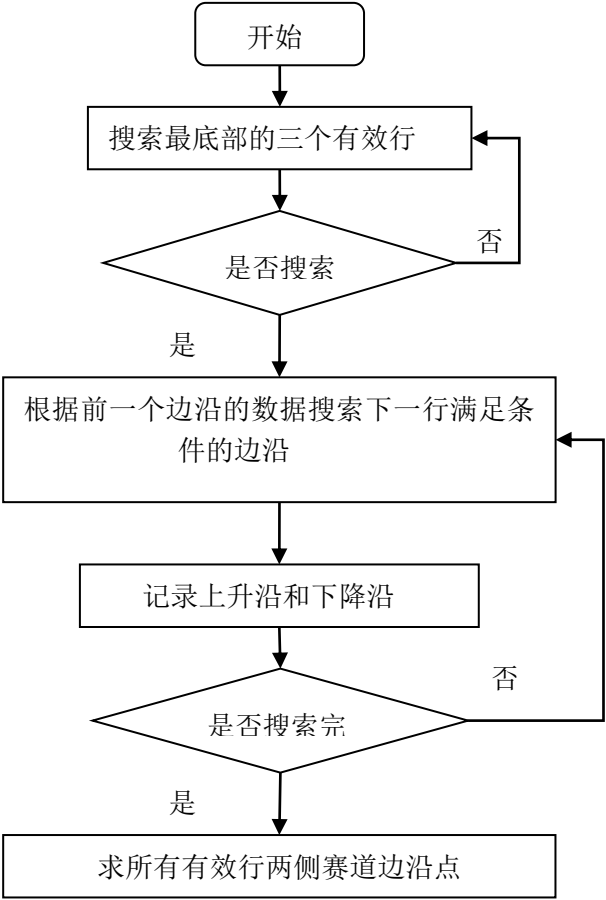


图 4.4 算法流程

处理后得到的黑线中心如图 4.5~4.7 图所示。

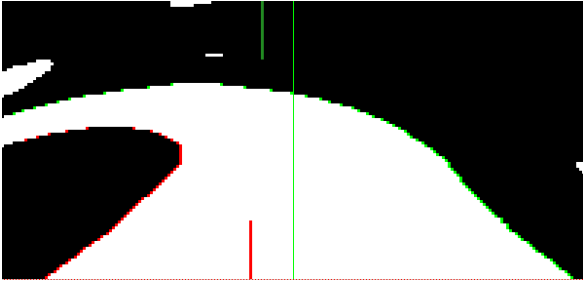


图 4.5 弯道处理后图像

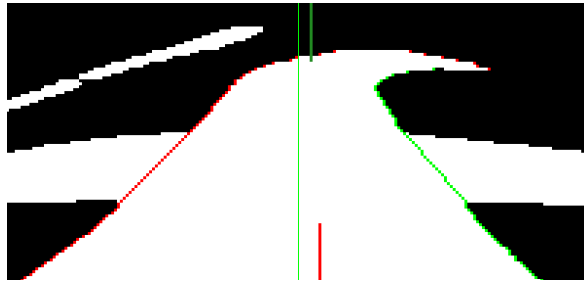


图 4.6 中心引导线处理后图像

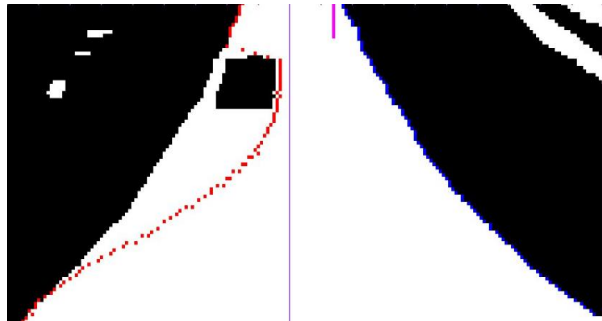


图 4.7 障碍处理图像

#### 4.1.3 图像矫正

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 4.8)；



图 4.8 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；

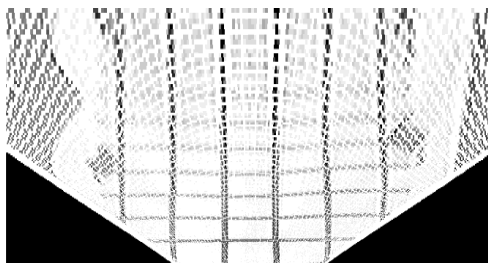


图 4.9 校正后的效果

- (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：

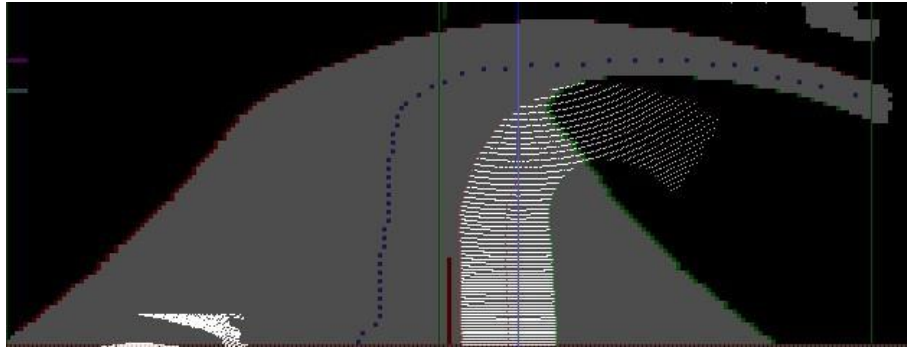


图 4.10 上位机模拟校正

#### 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：

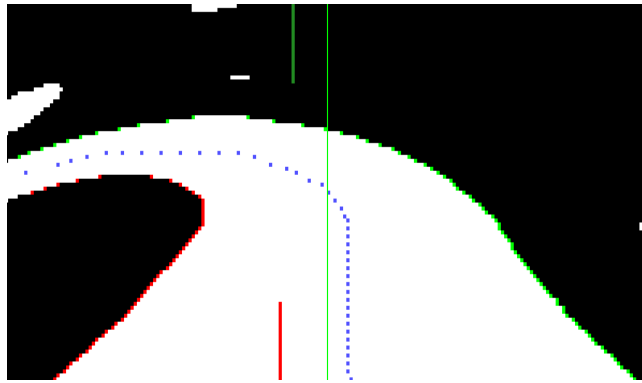


图 4.11 弯道处理后图像

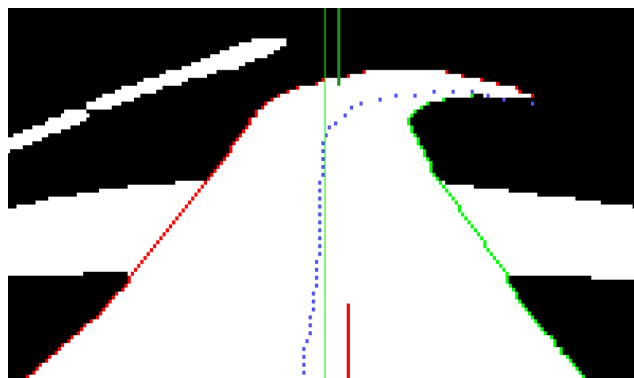


图 4.12 十字交叉处理后图像

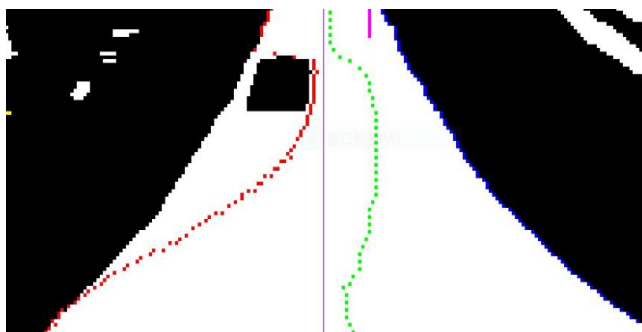


图 4.13 障碍处理之后的图像

#### 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。

视场的长度与单片机可以处理的图像行数成正比。我们采用由运算放大器制作的模拟比较器进行图像二值化，处理速度较 A/D 转换有了很大提高，大大增加了单片机处理的图像行数，最终处理行数为 95 行(隔 3 行提取一行)，达到的视场长度为 200 多 cm。为了增加视场宽度，增加每行采集的图像点数之外，我们采用了广角镜头，从而有效地增加了视场宽度。

## 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，使车模不再出现压路肩的现象。

## 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：

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

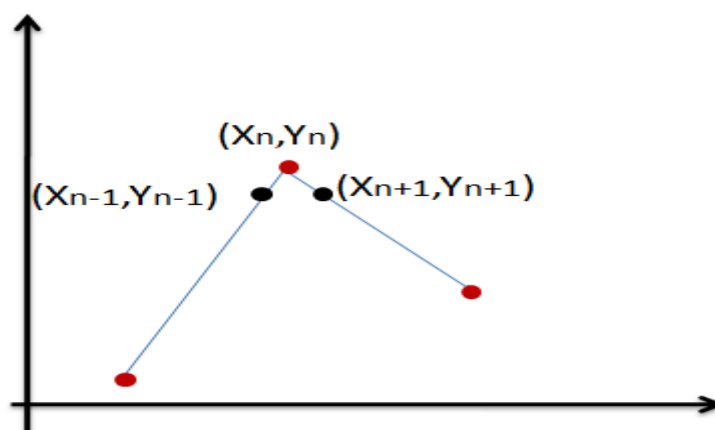


图 4.14 数学极值点

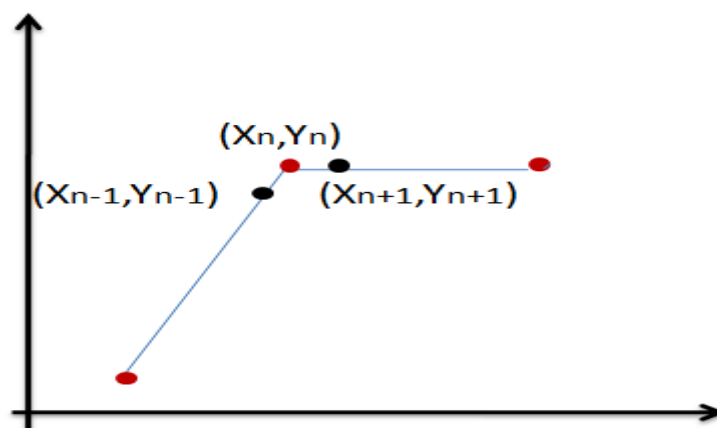


图 4.15 特殊情况下的折点

但这个公式在图 4.15 中存在一些问题, 对于  $(x_n, y_n)$  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:

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

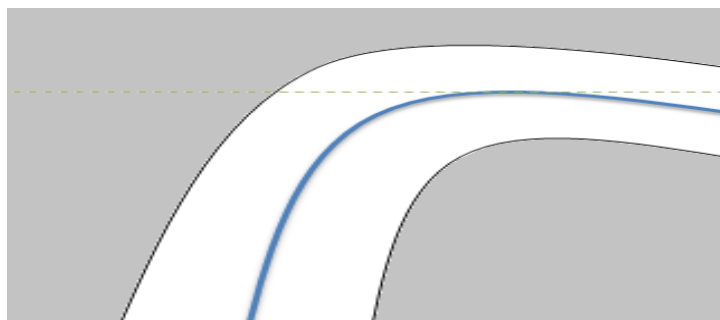


图 4.16 赛道图像中的折点

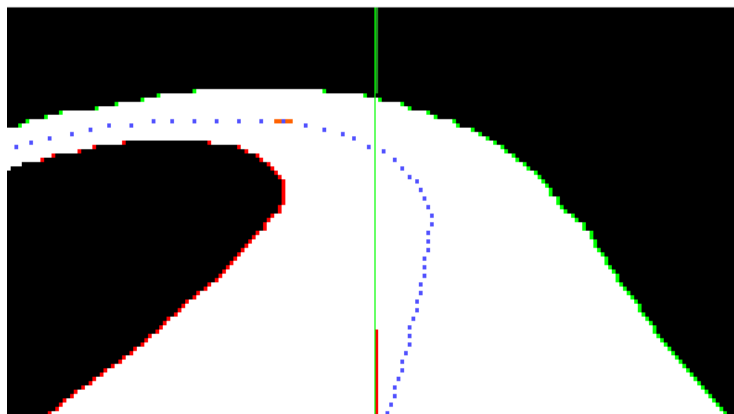


图 4.17 使用折点处理后的图像

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:

$$(X_{n-1} - X_n) \cdot (X_n - X_{n+1}) \leq 0 \ \&\& \ ! (0 = (X_{n-1} - X_n) \ \&\& \ 0 = (X_n - X_{n+1})) \quad (\text{公式 4.3})$$



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 4.18 所示。

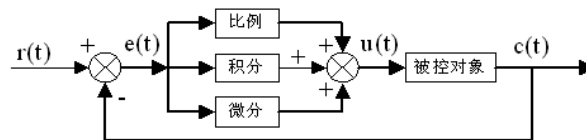


图 4.18 PID 控制器原理框图

在计算机控制系统中，使用的是数字 PID 控制器，控制规律为：

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

$$u(k) = K_p \left\{ e(k) + \frac{T}{T_i} \sum_{j=0}^k e(j) + \frac{T_d}{T} [e(k) - e(k-1)] \right\} \quad (\text{公式 4.5})$$

式中

$k$ ——采样序号， $k = 0, 1, 2, \dots$ ；       $r(k)$ ——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  $\Delta u(k)$ 。

#### 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)。

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

$$\begin{aligned} \Delta u(k) &= K_p \{ [e(k) - e(k-1)] + \frac{T}{T_i} e(k) + \frac{T_D}{T} [e(k) - 2e(k-1) + e(k-2)] \} \\ &= K_p \Delta e(k) + K_I e(k) + K_D [\Delta e(k) - \Delta e(k-1)] \end{aligned} \quad (\text{公式 4.7})$$

$$\text{式中 } \Delta e(k) = e(k) - e(k-1); \quad K_I = K_p \frac{T}{T_i}; \quad K_D = K_p \frac{T_D}{T}$$

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  $\Delta u(k)$  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。

### 4.3.3 PID 参数整定

运用 PID 控制的关键是调整  $K_P$ 、 $K_I$ 、 $K_D$  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 and 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  $K_p$  直接置零，在偏离黑线较少的某个范围，将  $K_p$  值减小为原来的一半，在偏离较大的其他情况，则保持  $K_p$  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，车身左右振荡比较严重，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振荡，这种振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 弯道左右振荡，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，没有振荡，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：

$$\text{loca\_Kp} = (\text{loca\_error} * \text{loca\_error}) / 2 + 1000$$

其中，loca\_error 是中心位置与中心值的偏差。

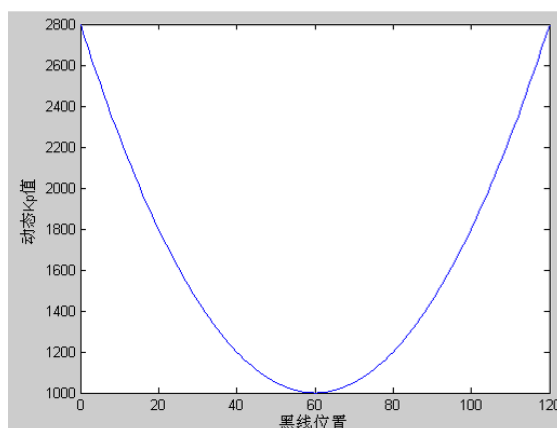


图 4.19 中心位置和动态 Kp 值的二次函数曲线

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。

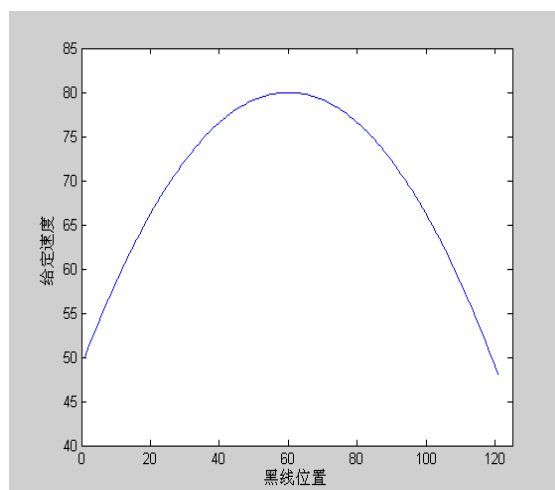


图 4.20 黑线位置和给定速度的二次函数曲线

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 4.20 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

#### 5.2.1 C#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。



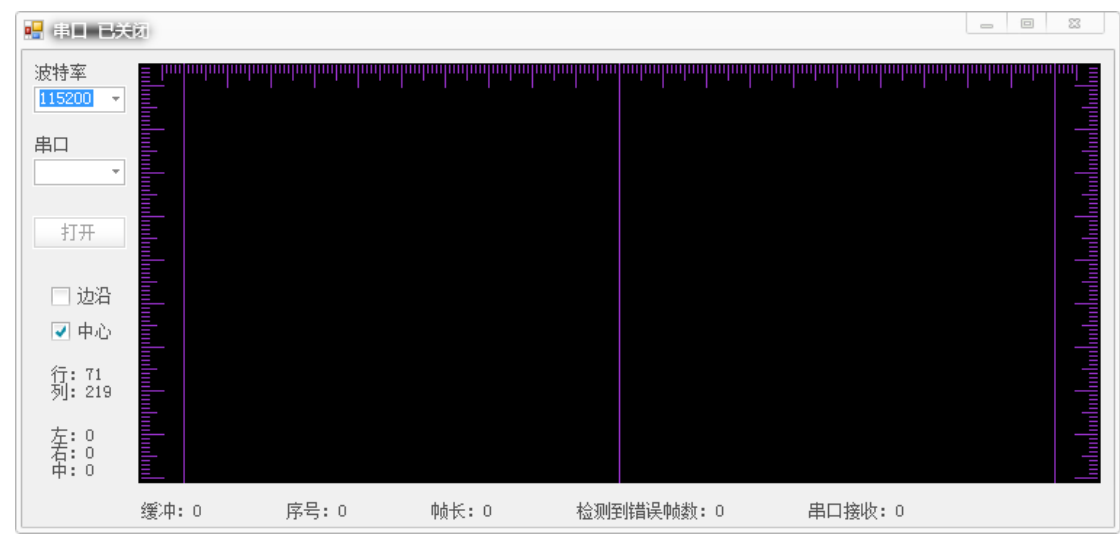


图 5.1 C#程序主界面

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。

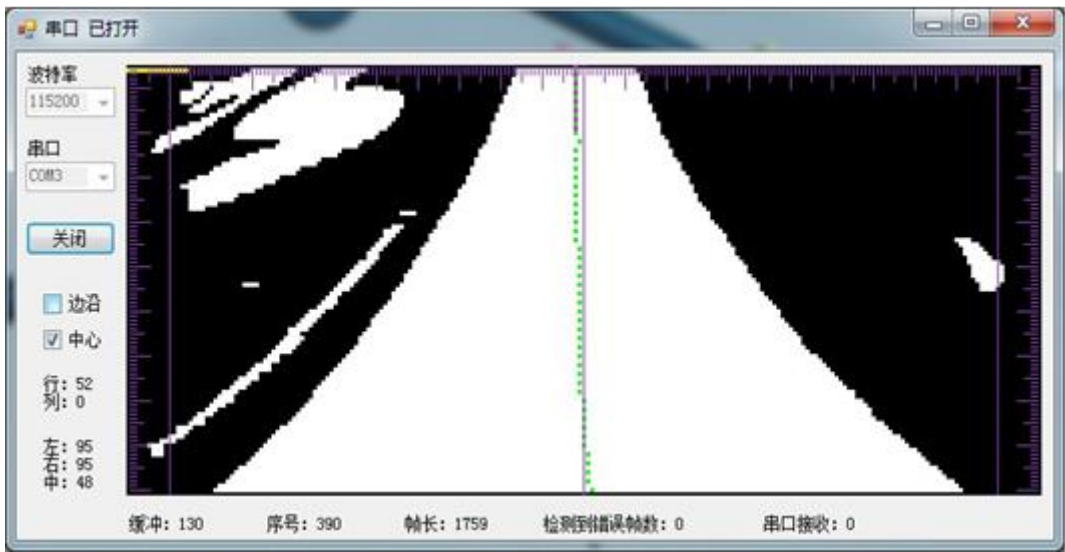


图 5.2 C#图像显示界面

### 5.2.2 MFC 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）。

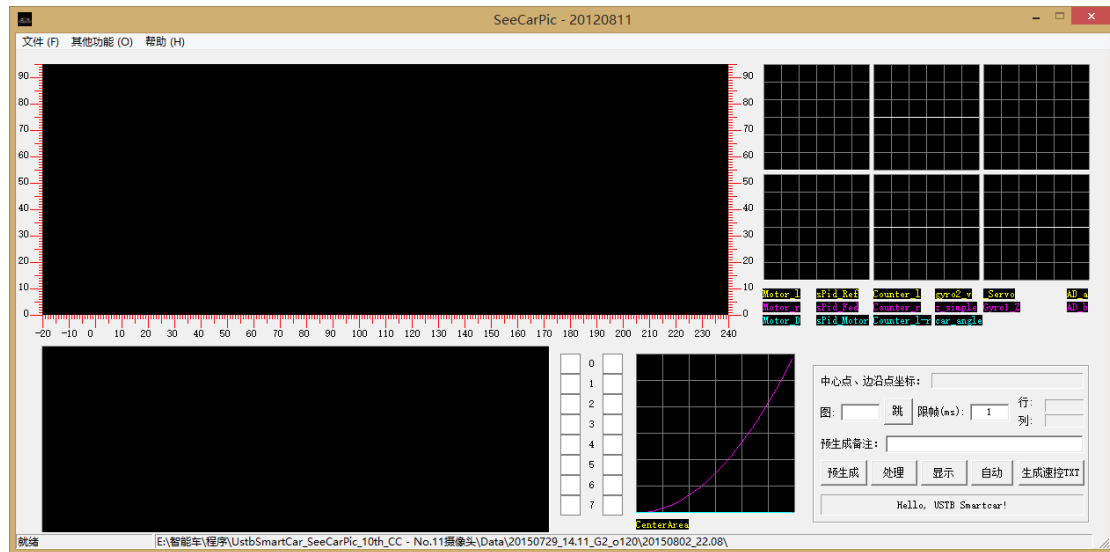


图 5.3 SD 卡上位机主界面

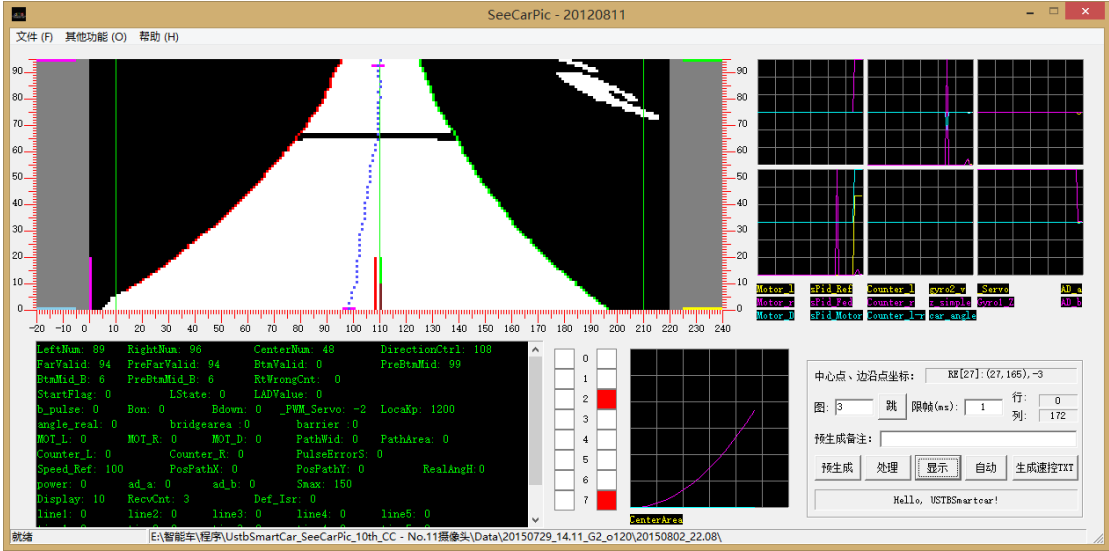


图 5.4 SD 卡上位机运行界面

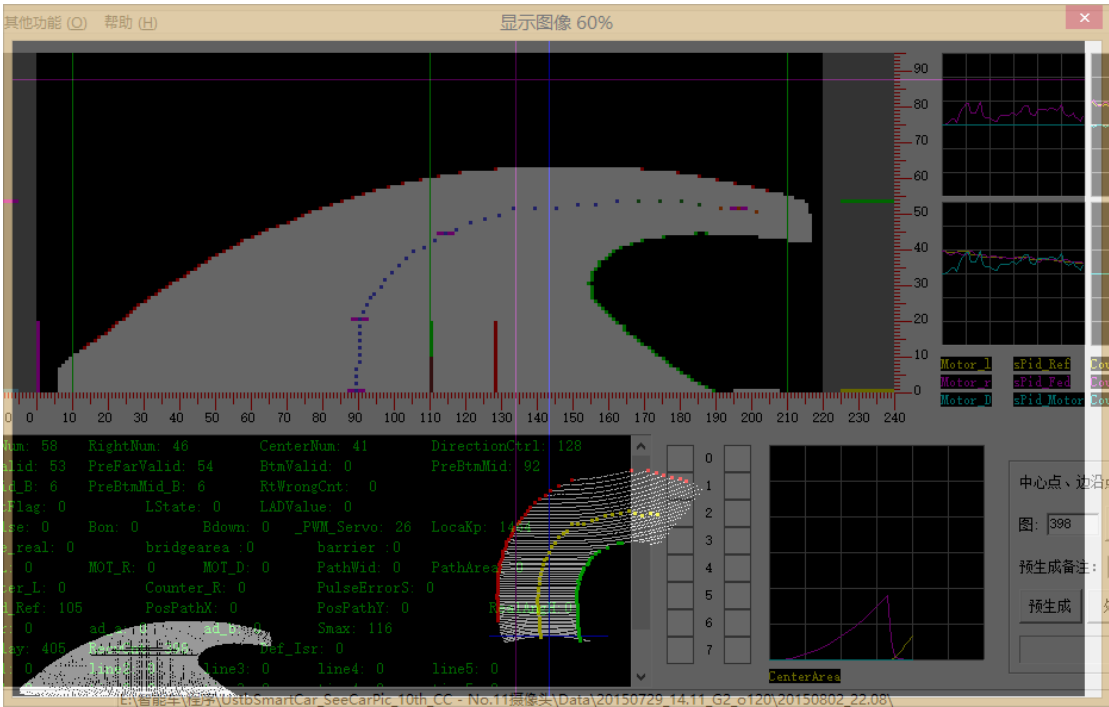


图 5.5 上位机校正效果显示

模型车在运动过程中，记录下的典型赛道图像，如下图 5.6~5.9 所示。

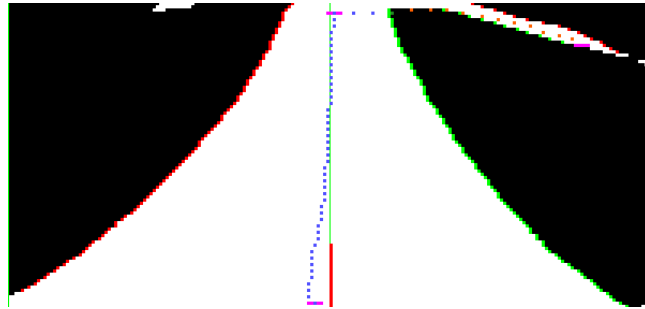


图 5.6 SD 卡记录的直道图像

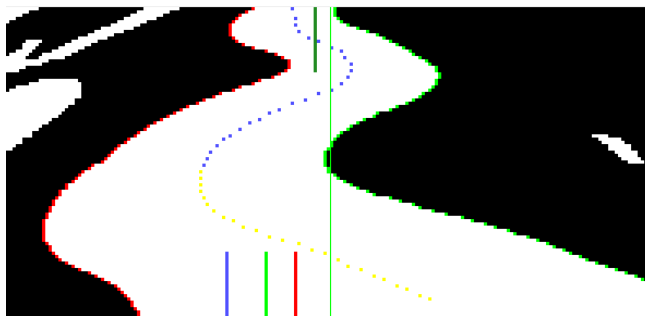


图 5.7 SD 卡记录的 S 型弯道图像

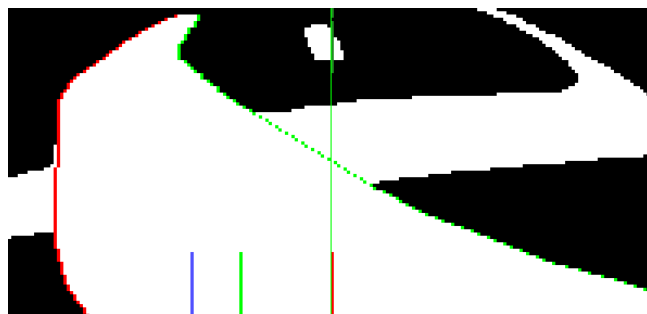


图 5.8 SD 卡记录的十字交叉图像

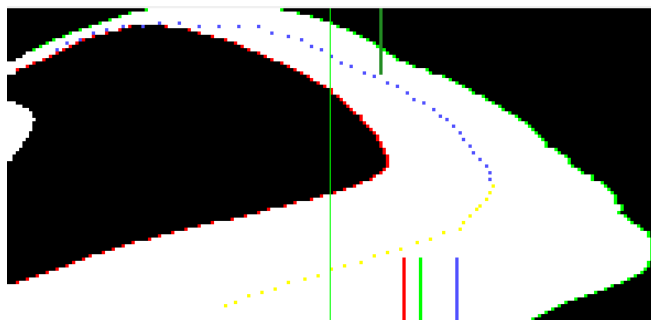


图 5.9 SD 卡记录的普通弯道图像

## 5.3 SD 卡模块

### 5.3.1 SD 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。由于恩智浦系列单片机拥有 SPI 接口，所以我们使用了 SD 卡的 SPI 模式。

### 5.3.2 SPI 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。

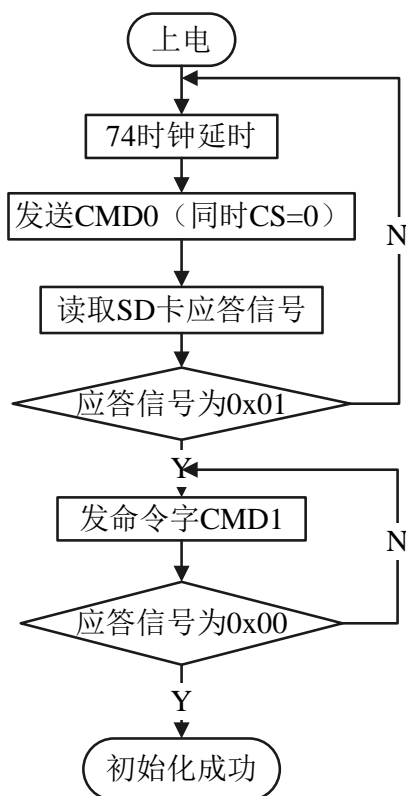


图 5.10 SD 卡初始化步骤

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	75cm
	宽	20cm
	高	34cm
车重		1100g
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	2 个
除了车模原有的驱动电机、舵机之外伺服电机个数	视野范围（近瞻/远瞻）	20cm/250cm
赛道信息检测	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->loc_Ref = Set_Mid_Point;

    loca_error = pp->loc_Ref - pp->loc_FeedBack;

    sPID.loc_Kp = 1250 + (((MAX_VIDEO_USEDLINE - v_FarthestValidLine)
* (MAX_VIDEO_USEDLINE - v_FarthestValidLine)) / 6);

    if ((loc_error < LOCA_DEADLINE) && (loc_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->loc_PreError; // 计算微分项偏差

        pp->loc_PreIntegral += loca_error; // 存储当前积分偏差

        pp->loc_PreError = loca_error; // 存储当前偏差

        pp->loc_PreU = pp->loc_Kp * loca_error + pp->loc_Ki *
pp->loc_PreIntegral + pp->loc_Kd * loca_derror; // 位置 PID 算法

        if (pp->loc_PreU >= LOCA_MAX) // 防止调节溢出

        {

            pp->loc_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)
}

```