



北京科技大学  
University of Science and Technology Beijing

# 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：北京科技大学

队伍名称：北京科技大学摄像头一队

参赛队员：刘珂屹

舒伯特

徐国强

带队教师：

---

## 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：\_\_\_\_\_

带队教师签名：\_\_\_\_\_

日 期：\_\_\_\_\_

## 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 控制算法调节驱动电机的转速和转向舵机的角度，实现了对模型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。

**关键字：** MK60N512VMD100，CMOS，PID，C#，SD 卡

---

## Abstract

In this paper we will design a smart car system based on MK60N512VMD100 as the micro-controller unit. We use a CMOS image sensor to obtain lane image information. Then convert the original image into the binary image by the analog comparator circuit in order to extract black guide line for track identification. An infrared sensor is used to measure the car's moving speed. We use PID control method to adjust the rotate speed of driving electromotor and direction of steering electromotor, to achieve the closed-loop control for the speed and direction. In order to increase the speed and the reliability of the car, a great number of the hardware and software tests are carried on and the advantages and disadvantages of the different schemes are compared by using the C#、MFC simulation platform, SD card module and the keyboard module. The results indicate that our design scheme of the smart car system is feasible.

**Keywords:** MK60N512VMD100, CMOS, PID, C#, SD card

---

## 目 录

摘    要.....	III
Abstract.....	IV
目    录.....	V
引    言.....	1
第一章  系统总体设计.....	2
1.1 系统概述.....	2
1.2 整车布局.....	3
第二章  机械系统设计及实现.....	4
2.1 车体机械建模.....	4
2.2 前轮倾角的调整.....	4
2.3 底盘高度的调整.....	5
2.4 编码器的安装.....	5
2.5 舵机安装结构的调整.....	5
2.6 舵机转角分析.....	6
2.7 摄像头的安装.....	8
第三章  硬件系统设计及实现.....	11
3.1 硬件设计方案.....	11
3.2 传感器的选择.....	11
3.2.1 摄像头.....	11
3.2.2 编码器.....	13
3.3 电路设计方案.....	14
3.3.1 单片机最小系统板.....	15
3.3.2 电源稳压电路及检测电路.....	16
3.3.3 图像处理电路.....	17
3.3.4 电机驱动电路.....	18

---

3.3.5 舵机接口电路 .....	19
3.3.6 键盘拨码电路 .....	19
第四章 软件系统设计及实现 .....	21
4.1 赛道中心线提取及优化处理 .....	21
4.1.1 原始图像的特点 .....	21
4.1.2 赛道边沿提取 .....	22
4.1.3 图像校正 .....	24
4.1.4 推算中心 .....	26
4.1.5 路径选择 .....	27
4.2 PID 控制算法介绍 .....	28
4.2.1 位置式 PID .....	29
4.2.2 增量式 PID .....	29
4.2.3 PID 参数整定 .....	30
4.3 转向舵机的 PID 控制算法 .....	30
4.4 驱动电机的 PID 控制算法 .....	32
第五章 系统开发及调试工具 .....	34
5.1 开发工具 .....	34
5.2 上位机图像显示 .....	34
5.2.1 C#静态上位机 .....	34
5.2.2 MFC SD 卡上位机 .....	35
5.3 SD 卡模块 .....	38
5.3.1 SD 卡介绍 .....	38
5.3.2 SPI 总线介绍 .....	38
5.3.3 软件实现 .....	39
5.4 键盘模块 .....	40
第六章 模型车的主要技术参数 .....	41
结 论 .....	42

---

参 考 文 献 .....	I
附录：程序源代码 .....	II





## 引 言

随着科学技术的不断发展进步，智能控制的应用越来越广泛，几乎渗透到所有领域。智能车技术依托于智能控制，前景广阔且发展迅速。目前，掌握着汽车工业关键技术的发达国家已经开发了许多智能车的实验平台和商品化的车辆辅助驾驶系统。有研究认为智能汽车作为一种全新的汽车概念和汽车产品，在不久的将来会成为汽车生产和汽车市场的主流产品。

面向大学生的智能汽车竞赛最早始于韩国，在国内，全国大学生“飞思卡尔”杯智能汽车竞赛从2006年开始已经举办了七届，得到了各级领导及各高校师生的高度评价。大赛为智能车领域培养了大量后备人才，为大学生提供了一个充分展示想象力和创造力的舞台，吸引着越来越多来自不同专业的大学生参与其中。

全国大学生“飞思卡尔”杯智能汽车竞赛包括光电组、摄像头组和电磁组，其中数摄像头组的智能车速度最快，备受关注。

本技术报告主要包括机械系统、硬件系统、软件系统等，详尽地阐述了我们的设计方案，具体表现在硬件电路的创新设计以及控制算法的独特想法。智能车的制作过程包含着我们的辛勤努力，这份报告凝聚了我们智慧，是我们团队共同努力的成果。

在准备比赛的过程中，我们小组成员涉猎控制、模式识别、传感器技术、汽车电子、电气、计算机、机械等多个学科，几个月来的经历，培养了我们电路设计、软件编程、系统调试等方面的能力，锻炼了我们知识融合、实践动手的能力，对今后的学习工作都有着重大的实际意义。

# 第一章 系统总体设计

## 1.1 系统概述

智能车系统的总体工作模式为：CMOS 图像传感器拍摄赛道图像，输出 PAL 制式信号，经过信号处理模块进行硬件二值化，采用 LM1881 进行视频同步分离，二值化图像信号、奇偶场信号、行同步信号输入到 MK60N512VMD100 微控制器，进行进一步处理获得主要的赛道信息；通过光电编码器来检测车速，并采用 MK60N512VMD100 的输入捕捉功能进行脉冲计算获得速度和路程；转向舵机采用 PD 控制；驱动电机采用 PID 控制，通过 PWM 控制驱动电路调整电机的功率；而车速的目标值由默认值、运行安全方案和基于图像处理的优化策略进行综合控制。

根据智能车系统的基本要求，我们设计了系统结构图，如图 1.1 所示。在满足比赛要求的情况下，力求系统简单高效，因而在设计过程中尽量简化硬件结构，减少因硬件而出现的问题。

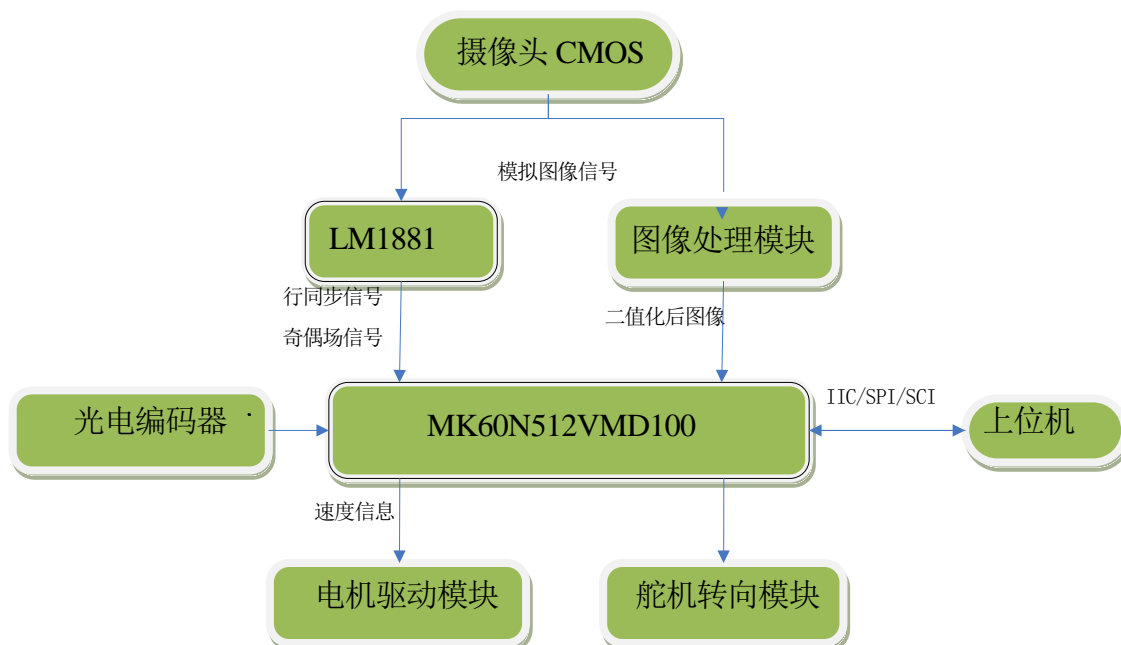


图 1.1 系统结构图

## 1.2 整车布局

- (1) 车模底盘降低，主板低放，以降低重心。
- (2) 舵机竖直放置，方便控制。
- (3) 用轻便坚固的碳纤杆作为摄像头杆的材料。
- (4) 摄像头安于电池的前方，有利于重心分布和盲区与前瞻的匹配。



图 1.2 整车布局图

---

## 第二章 机械系统设计及实现

根据组委会的相关规定，今年摄像头组比赛车模更换为 A 型车模。针对不同的车模，必然会有不同的调整方式。在比赛备战之初，我们就对该车模进行了详细的系统分析。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，提高车模整体精度是很必要的。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。

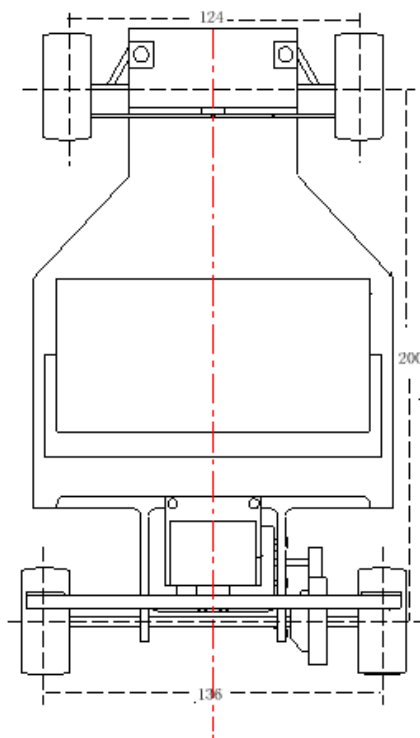


图 2.1

### 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 个项目决定，反映了转向轮、主销和前轴等三者 in 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的确可以增大转弯时车轮和地面的接触面积，从而增大车子地面的摩擦程度，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，不同与上一届的码盘，质量和体积都较大的编码器给安装带来了一定不便。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 舵机安装结构的调整

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。

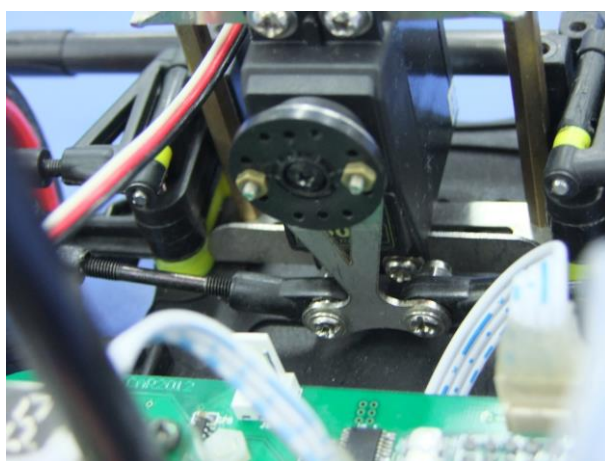


图 2.3 舵机安装

## 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。

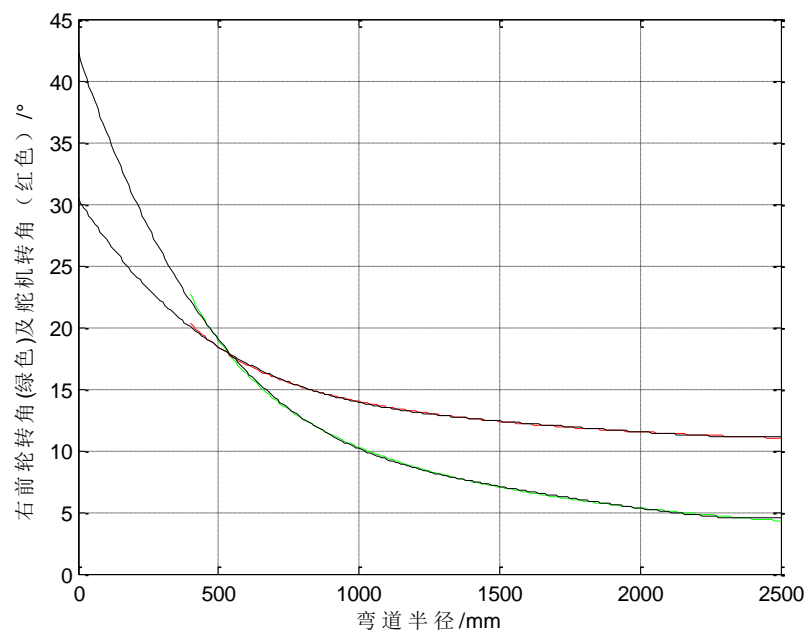


图 2.4

舵机转角关于（右）前轮转角关系图，如图 2.5 所示。

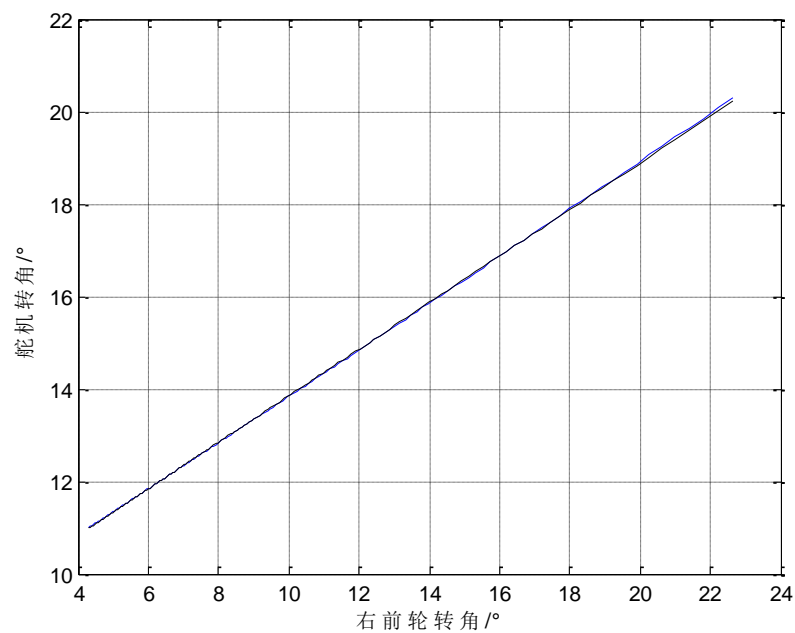


图 2.5

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。

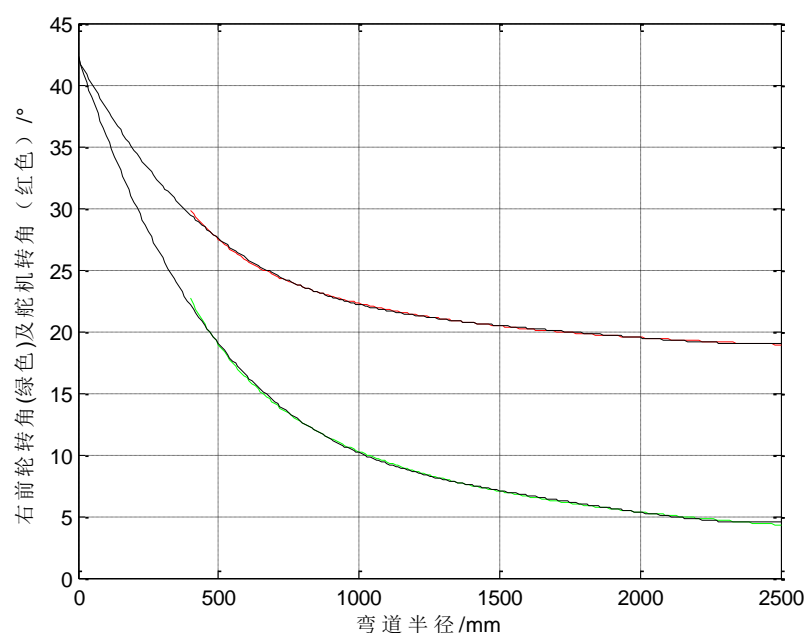


图 2.6

## 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。



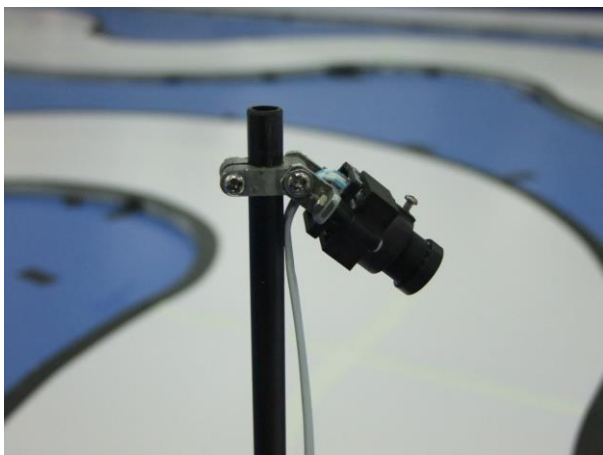


图 2.7 摄像头安装



---

## 第三章 硬件系统设计及实现

### 3.1 硬件设计方案

从最初进行硬件电路设计时就既定了系统的设计目标：可靠、高效、简洁，在整个系统设计过程中严格按照规范进行。

可靠性是系统设计的第一要求，我们对电路设计的所有环节都进行了电磁兼容性设计，做好各部分的接地、屏蔽、滤波等工作，将高速数字电路与模拟电路分开，使本系统工作的可靠性达到了设计要求。

高效是指本系统的性能要足够强劲。我们主要是从以下两个方面实现的：

(1) 采用运算放大器设计的比较器实现了图像二值化的高速转换，大大提高了图像采集的分辨率；

(2) 使用了由分立元件制作的直流电动机可逆双极型桥式驱动器，该驱动器的额定工作电流可以轻易达到 20A 以上，保证了电动机的工作转矩和转速。

简洁是指在满足了可靠、高效的要求后，为了尽量减轻整车重量，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。我们在对电路进行了详细分析后，对电路进行了简化，合理设计元件排列、电路走线，使本系统硬件电路部分轻量化指标都达到了设计要求。

### 3.2 传感器的选择

#### 3.2.1 摄像头

##### 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。

世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。

智能车的摄像头对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。

### OV5116, BF3003 比较

OV5116 具有成像稳定，技术成熟等优点，但是同时也有供货不稳定（已停产），图像动态特性差等缺点。于是我们选用了新的摄像头 bf3003。经试验证明，与 ov5116 相比，bf3003 具有动态特性好，反应快，曝光时间短等优势，且可以进行 sccb 通信，可以实现动态调节，但是 BF3003 功耗大，对电源芯片要求较高，复合信号纹波较大，稳定性不如 OV5116。

OV5116 是 Omni Vision 公司生产的较为典型的 CMOS 图像传感器模块，芯片阵列大小为  $352 \times 288$ ，有效光敏面为  $312 \times 215$  像素，电源是 5 V(DC)，28 个引脚的 PLCC 型封装。摄像头输出的黑白全电视信号为 PAL 制式模拟信号，每秒 25 帧，电视扫描线为 625 线，奇场在前，偶场在后。

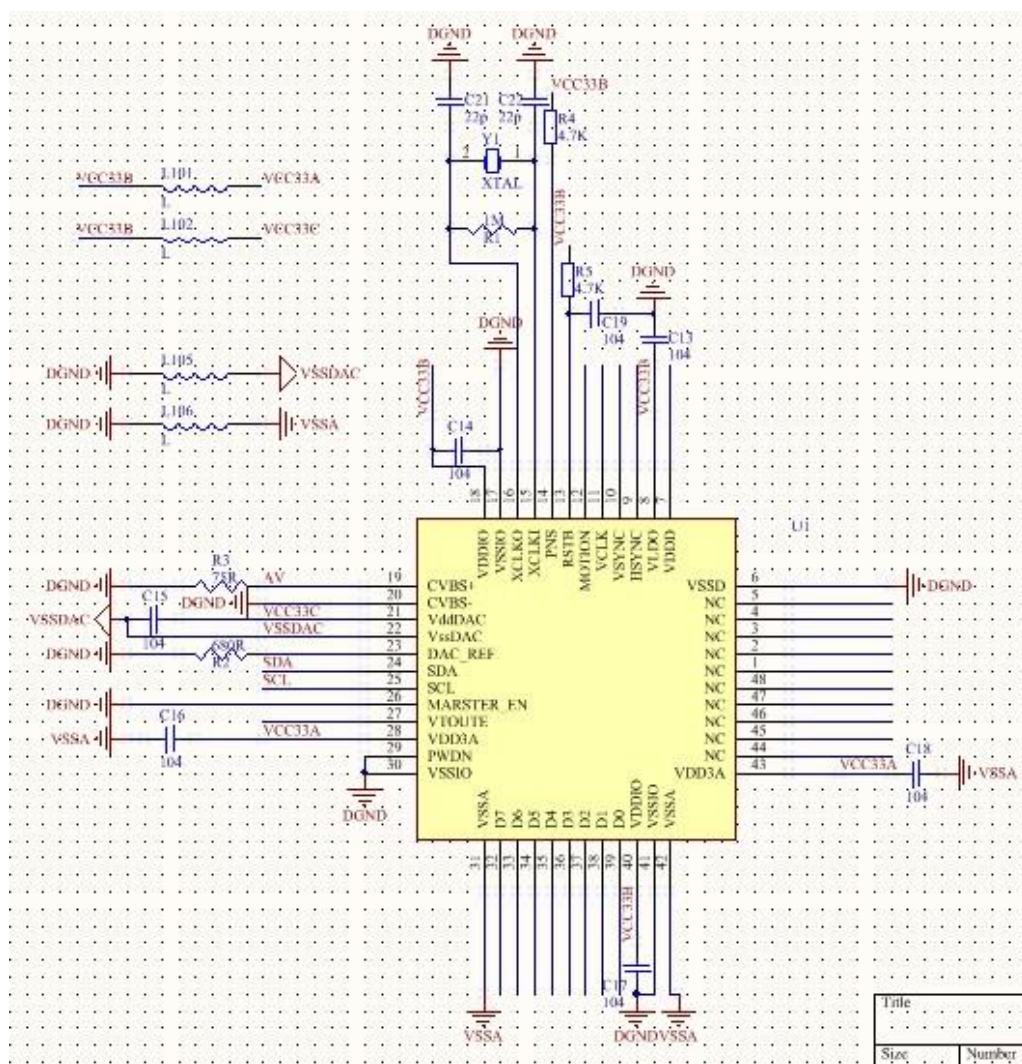


图 3.1 bf3003 电路原理图

### 3.2.2 编码器

由于我们在华北预赛中考虑到智能车在实际速度控制中对反应车速的控制信号波形要求不是太高，因此在满足比赛要求的基础上尽量简化电路，使用自制的光电编码器来测速。我们使用线切割在直径为 30mm 的圆盘周围加工出 100 个细缝，使用红外光电对射管作为采集码盘脉冲可鉴向的传感器。速度传感器实物图如图所示：

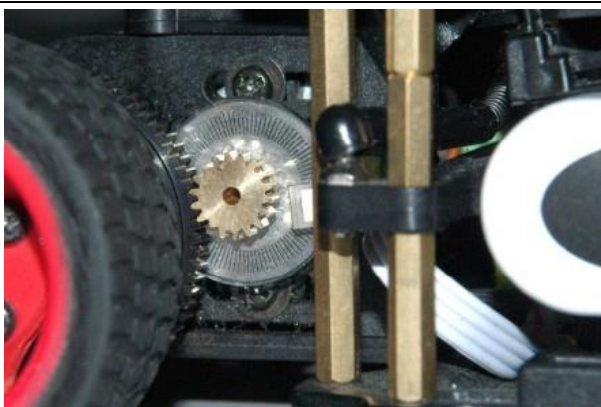


图 3.2 光电编码器

在预赛之后，我们感觉到虽然自制的码盘精度够用，价格也比较低，但是容易受到外部光线和灰尘的影响，会影响测速的精度。因此改用了欧姆龙的小码盘。因此我们选用了方波编码器，有两相信号反馈给单片机，其输入波形如下图所示：

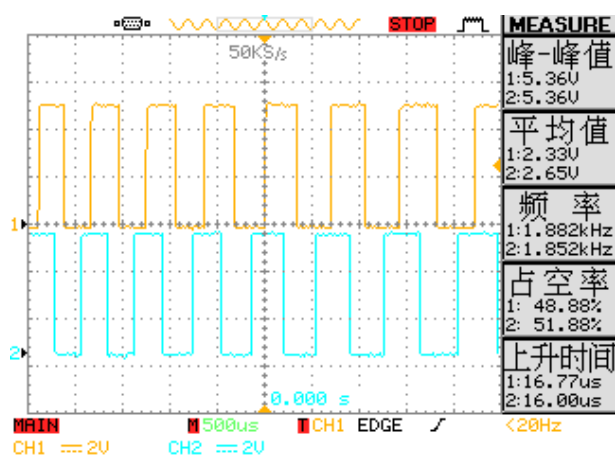


图 3.3 方波编码器波形图

### 3.3 电路设计方案

智能车控制系统电路由三部分组成：MK60N512VMD100 为核心的最小系统板、主板、ZLG7290 键盘。最小系统板可以插在主板上，组成了信号采集、信号处理、电机控制、舵机控制单元。为了减小电机驱动电路带来的电磁干扰，我们把控制单元部分和电机驱动部分分开来，排布在主板的两端。

主板上集成了本系统的主要电路，它包括如下部件：电源稳压电路、最小系统板插座、视频同步分离电路、AD 转换电路、陀螺仪、摄像头接口、舵机接口、电机驱动模块、编码器模块、键盘接口、拨码开关、指示灯等。





陀螺仪信号输入：PTE25

键盘输入信号：PTB2、PTB3、PTB9

拨码开关输入信号：PTD3、PTD4

### 3.3.2 电源稳压电路及检测电路

本系统中电源稳压电路分别需要有+5V，+3.3V，+12V 供电。+3.3V 给单片机供电；+5V 为摄像头、摄像头模块、电机模块、键盘拨码、编码器模块供电；+12V 和+5V 共同给电机驱动电路中 IR2184 供电。

由于整个系统中+5V 电路功耗较小，为了降低电源纹波，我们考虑使用线性稳压电路。另外，后轮驱动电机工作时，电池电压压降较大，为提高系统工作稳定性，必须使用低压降电源稳压芯片，我们选用了 TPS7350。TPS7350 是微功耗低压差线性电源芯片，具有完善的保护电路,包括过流,过压,电压反接保护。使用这个芯片只需要极少的外围元件就能构成高效稳压电路。

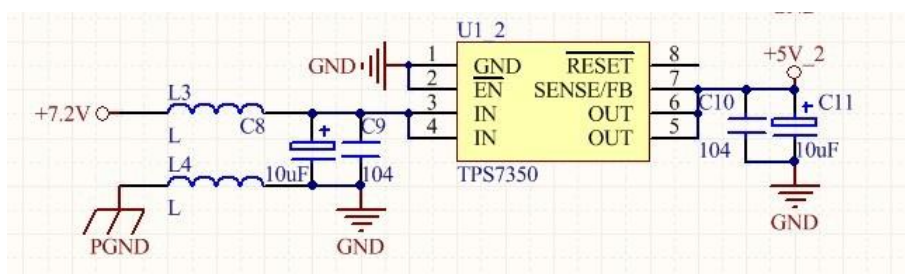


图 3.5 TPS7350 原理图

TPS79333 超是低噪，高电源电压抑制比，高速射频，线性调节的低压稳压芯片。由 TPS7350 输出为其供电，我们选用 fixed version TPS79333, 输出+3.3V 为单片机供电。

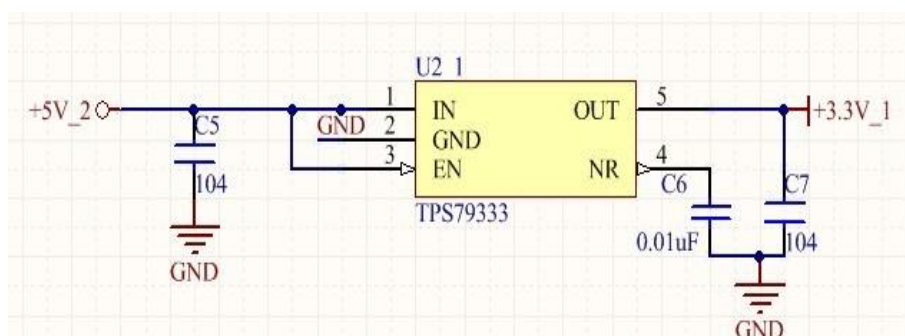


图 3.6 TPS79333 原理图



### 3.3.3 图像处理电路

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。

由于摄像头输出的黑白全电视信号为 PAL 制式模拟信号，所以必须经过相应的图像处理模块进行相应转换之后才能由单片机进行处理。解决方案有以下四种：对 PAL 信号进行硬件二值化是为了降低单片机的计算负荷，通过调节阈值而将灰度图像转换成黑白图像，这样就不需要用 AD 就可以采集图像了。其最明显的优点在于普通 IO 的操作速度要比 AD 快，使提高分辨率成为可能。由于通过模拟电路实现二值化的硬件比较容易实现，所以这个方案最快进入了测试阶段并取得满意果，所以最后采用了此技术路线而放弃了其它方案的研究。

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。

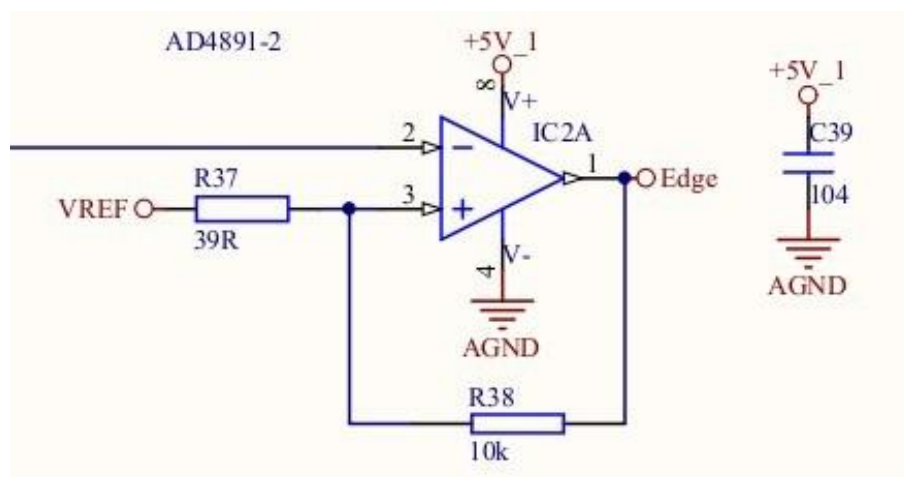


图 3.7 比较电路原理图

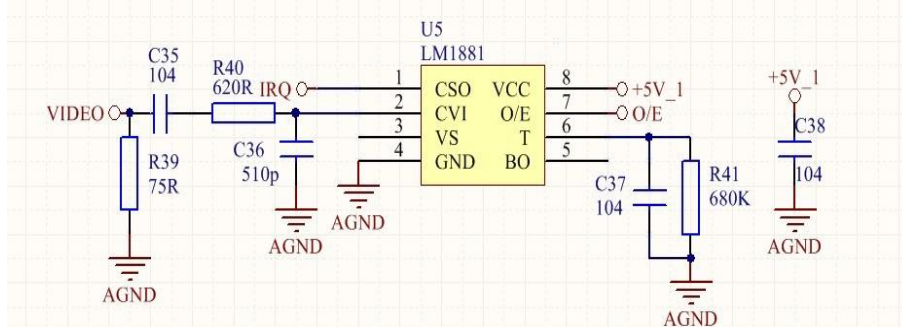


图 3.8 LM1881 外围电路原理图

### 3.3.4 电机驱动电路

电机驱动板为一个由分立元件制作的直流电动机可逆双极型桥式驱动器，其功率元件由四支 N 沟道功率 MOSFET 管组成，额定工作电流可以轻易达到 100A 以上，大大提高了电动机的工作转矩和转速。该驱动器主要由以下部分组成：PWM 号输入接口、逻辑换向电路、死区控制电路、电源电路、上桥臂功率 MOSFET 管栅极驱动电压泵升电路、功率 MOSFET 管栅极驱动电路、桥式功率驱动电路、缓冲保护电路等。

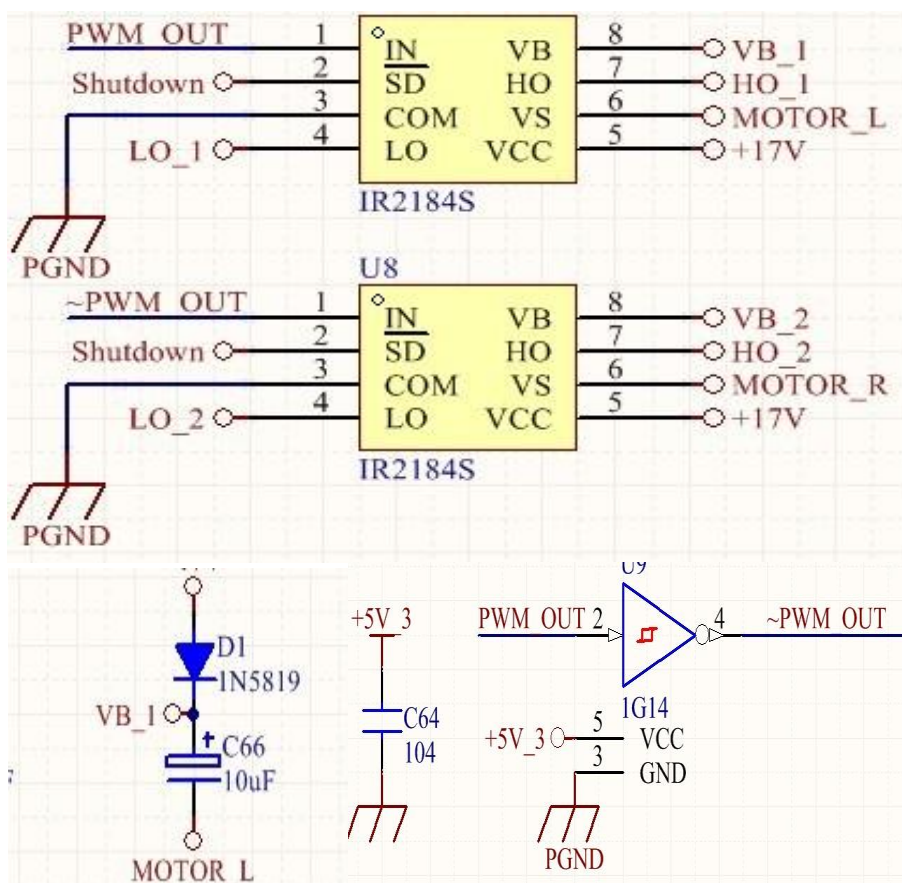


图 3.9 电机驱动模块原理图

### 3.3.5 舵机接口电路

舵机接口电路包括，供电电路和 PWM 号输入接口。

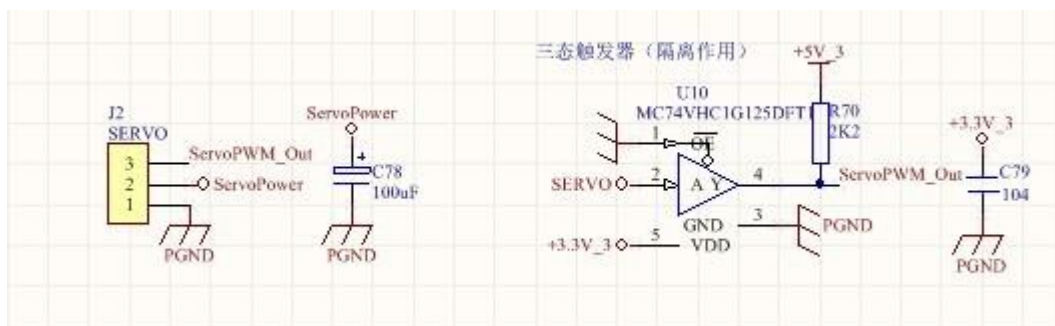


图 3.10 舵机接口原理图

### 3.3.6 键盘拨码电路

主板上还包括拨码开关电路和键盘接口，接外接键盘。

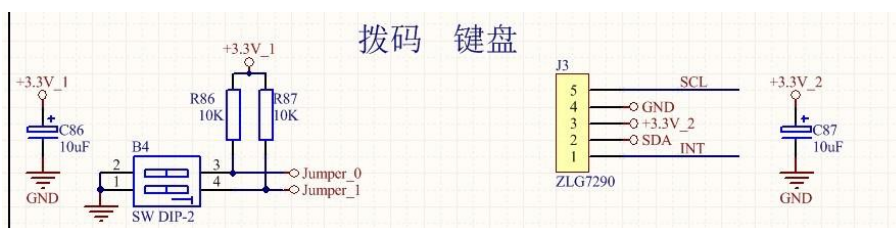


图 3.11 拨码开关、键盘接口原理图



---

## 第四章 软件系统设计及实现

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。

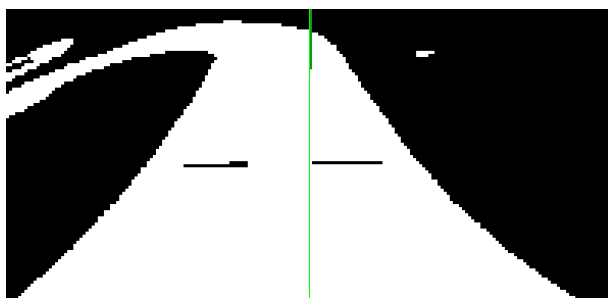


图 4.1 起点原始图像

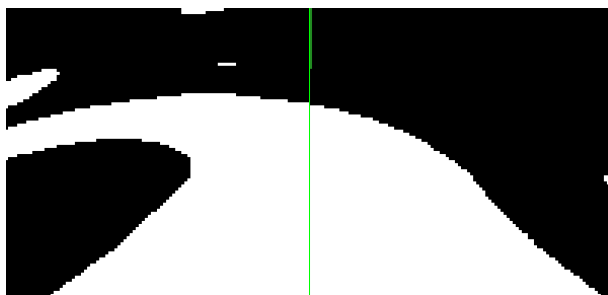


图 4.2 弯道原始图像

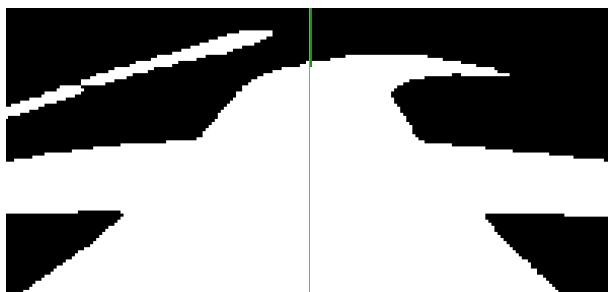


图 4.3 十字交叉原始图像

#### 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) 在弯道的时候赛道有回拐的情况，所以提取本行边沿之前还要向前搜索回拐的边沿点；
- (5) 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；
- (6) 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；
- 边沿提取算法的程序流程如图 4.4 所示。

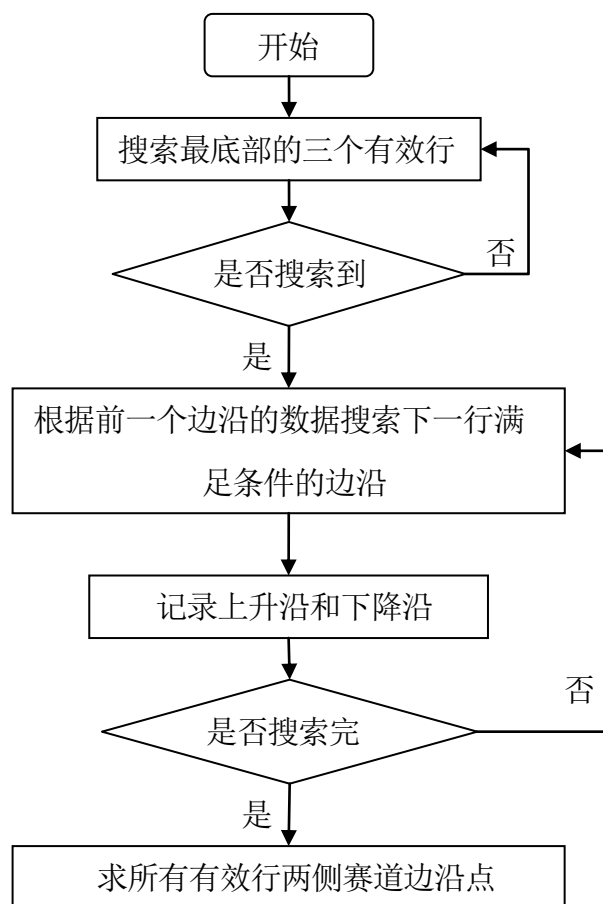


图 4.4 算法流程图

处理后得到的黑线中心如图 4.5~4.7 图所示。

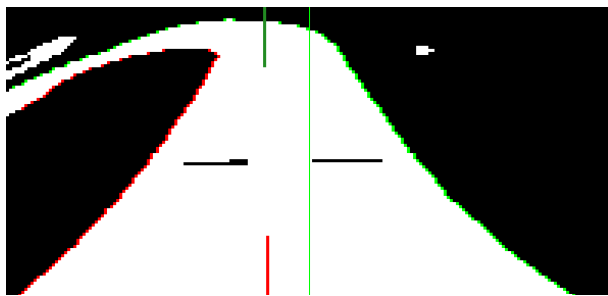


图 4.5 起点处理后图像

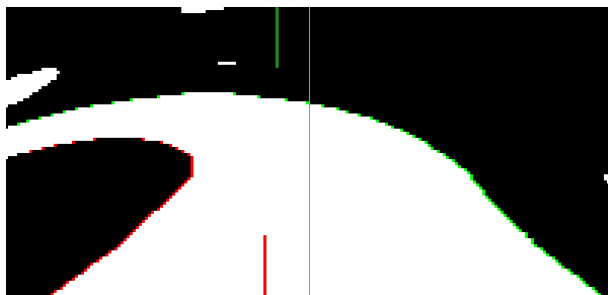


图 4.6 弯道处理后图像

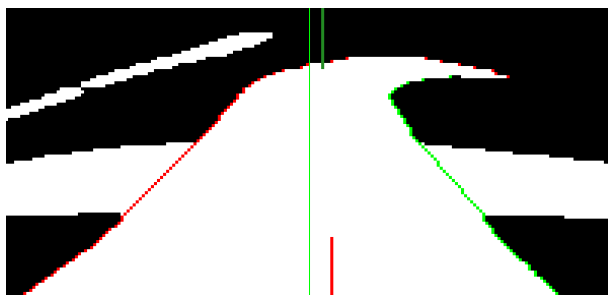


图 4.7 十字交叉处理后图像

#### 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；
- (2) 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

- (3) 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；

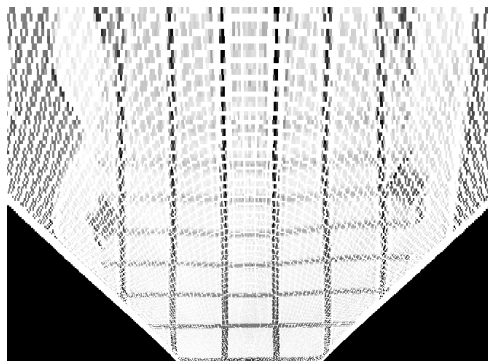


图 4.9 校正后的效果

- (4) 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。
- (5) 用上位机观察的校正效果如下：

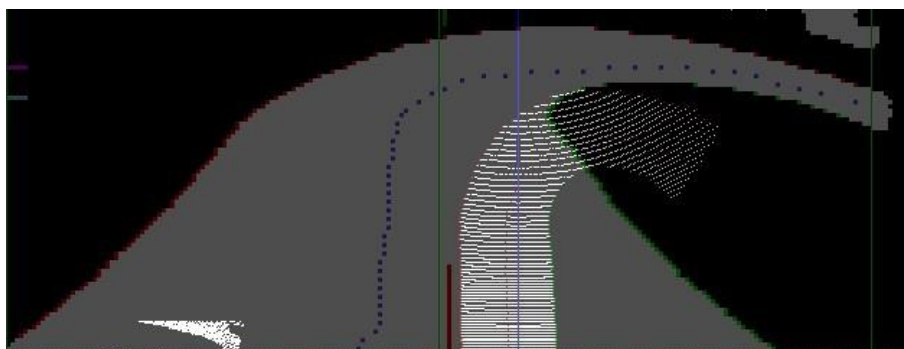


图 4.10 上位机模拟校正

#### 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：

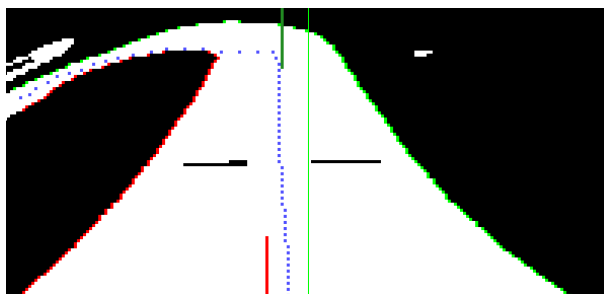


图 4.11 起点处理后图像

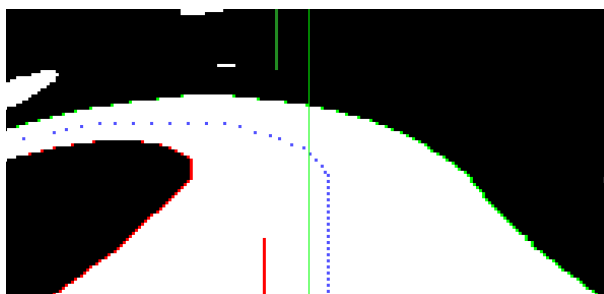


图 4.12 弯道处理后图像

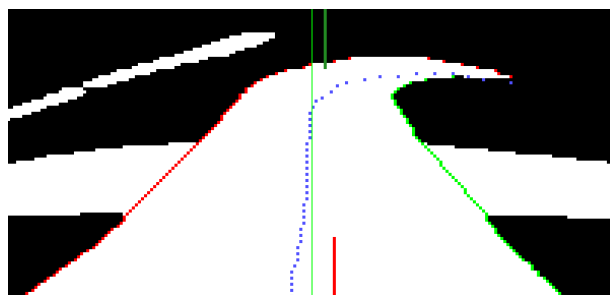


图 4.13 十字交叉处理后图像

### 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，我们从两方面来完成：

#### 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。

## 4.2 PID 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 4.10 所示。

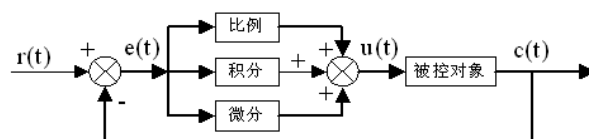


图 4.14 PID 控制器原理框图

在计算机控制系统中，使用的是数字 PID 控制器，控制规律为：

$$e(k) = r(k) - c(k) \quad (\text{公式 4.1})$$

$$u(k) = K_p \{e(k) + \frac{T}{T_i} \sum_{j=0}^k e(j) + \frac{T_d}{T} [e(k) - e(k-1)]\} \quad (\text{公式 4.2})$$

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。

#### 4.2.1 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(4.2)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  $\Delta u(k)$ 。

#### 4.2.2 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(4.2)推导出提供增量的 PID 控制算式。由式(4.2)可以推出式(4.3), 式(4.2)减去式(4.3)可得式(4.4)。

$$u(k-1) = K_p \{ e(k-1) + \frac{T}{T_i} \sum_{j=0}^{k-1} e(j) + \frac{T_d}{T} [e(k-1) - e(k-2)] \} \quad (\text{公式 4.3})$$

$$\begin{aligned}\Delta u(k) &= K_p \{ [e(k) - e(k-1)] + \frac{T}{T_i} e(k) + \frac{T_D}{T} [e(k) - 2e(k-1) + e(k-2)] \} \quad (\text{公式 4.4}) \\ &= K_p \Delta e(k) + K_I e(k) + K_D [\Delta e(k) - \Delta e(k-1)]\end{aligned}$$

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

公式(4.4)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(4.4)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。

### 4.2.3 PID 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。

## 4.3 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  $K_p$  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一种类型，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 120 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  $K_I$  置零的情况下，我们通过合理调节  $K_p$ ，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:

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

其中, local\_error 是中心位置与中心值的偏差。

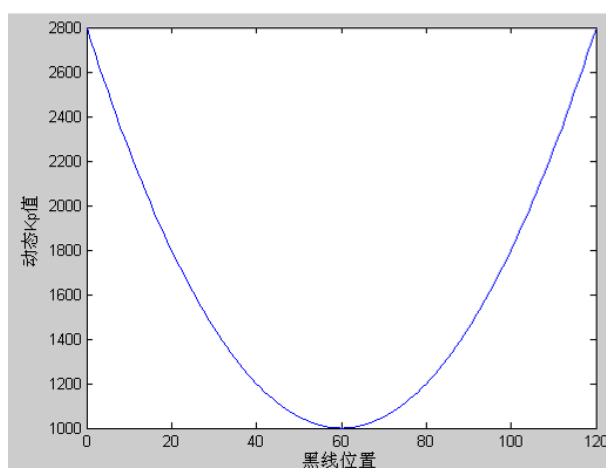


图 4.15 中心位置和动态 Kp 值的二次函数曲线

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。

#### 4.4 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。



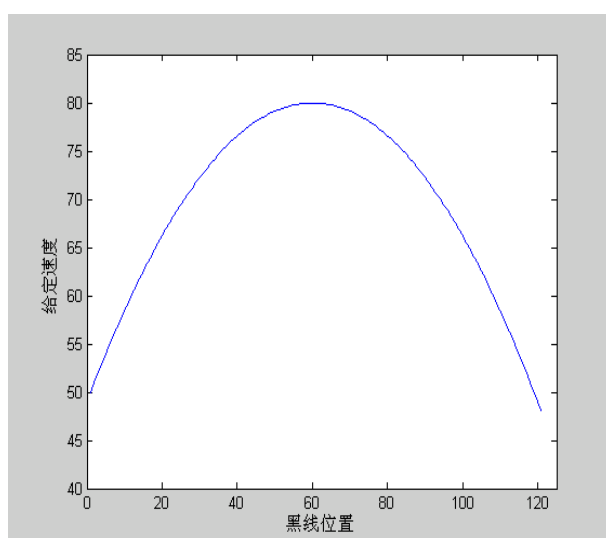


图 4.16 黑线位置和给定速度的二次函数曲线

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 系统开发及调试工具

### 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。

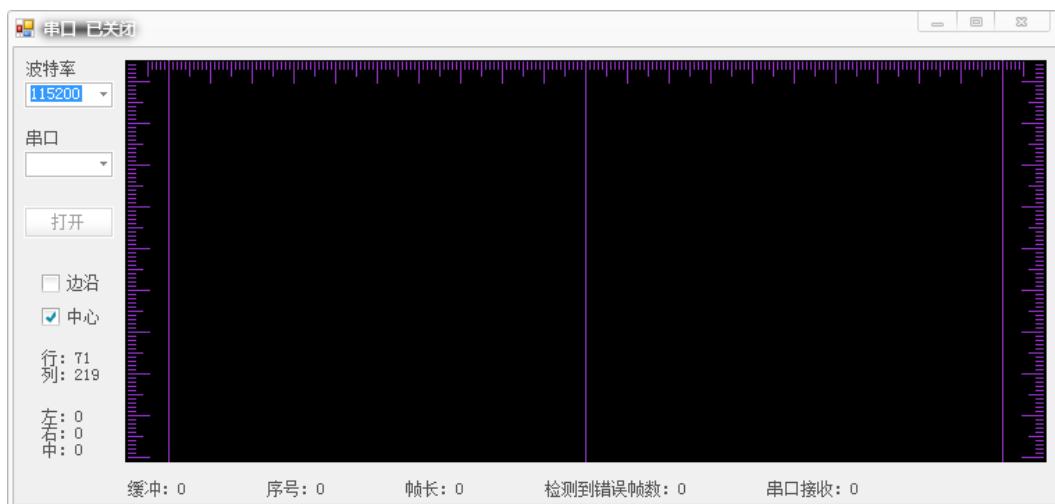


图 5.1 C#程序主界面

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。

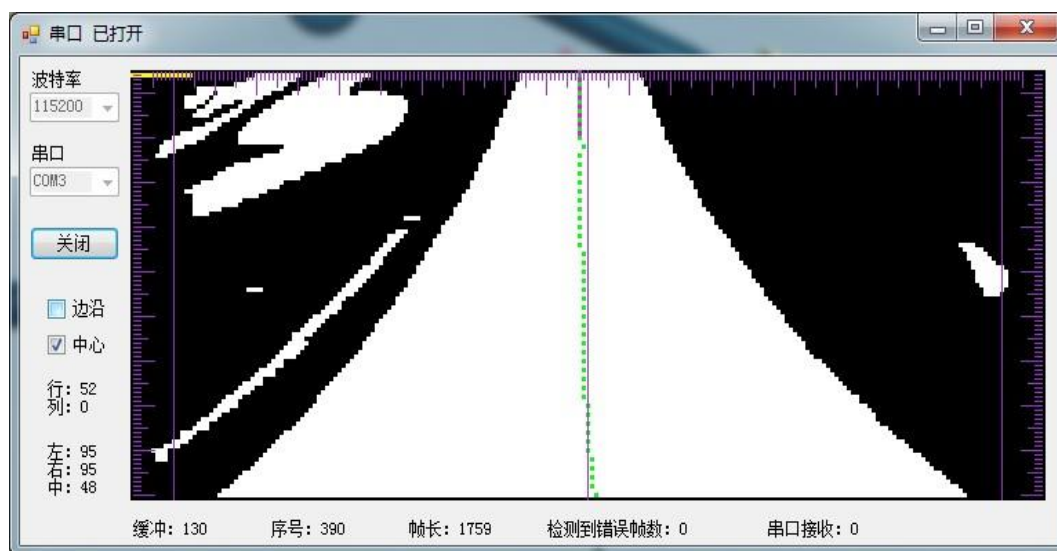


图 5.2 C#图像显示界面

### 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）。

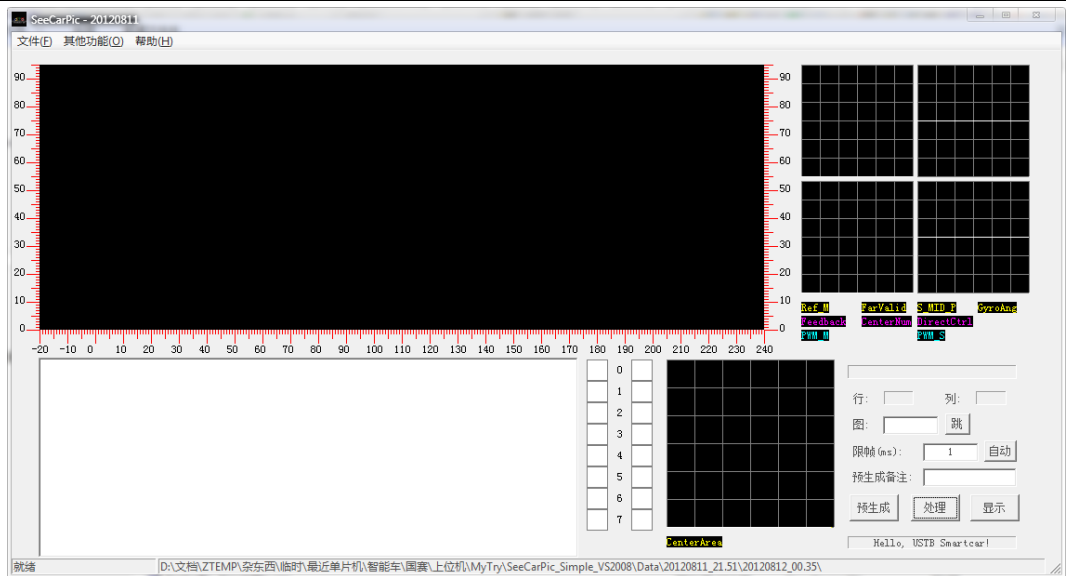


图 5.3 SD 卡上位机主界面

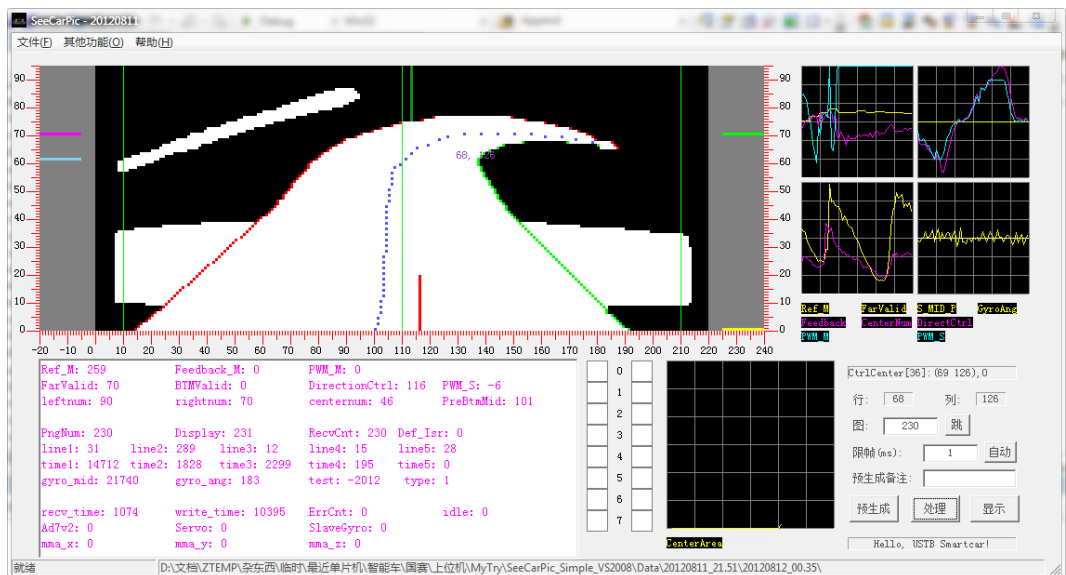


图 5.4 SD 卡上位机运行界面

模型车在运动过程中，记录下的典型赛道图像，如下图 5.3~5.8 所示。

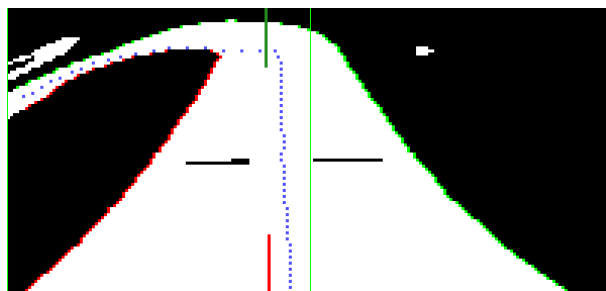


图 5.5 SD 卡记录的起点图像

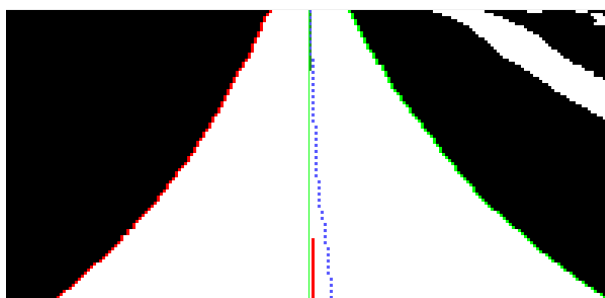


图 5.6 SD 卡记录的直道图像

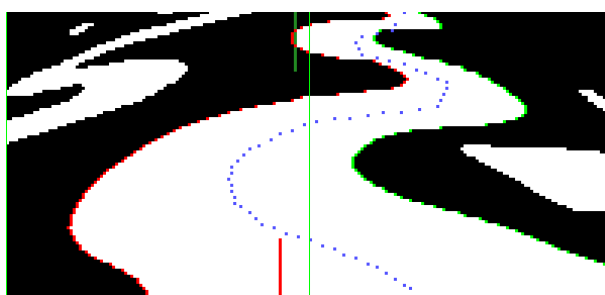


图 5.7 SD 卡记录的 S 型弯道图像

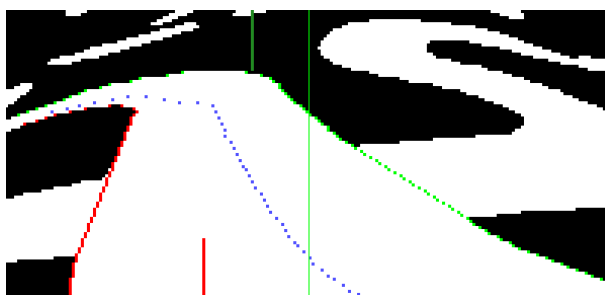


图 5.8 SD 卡记录的十字交叉图像

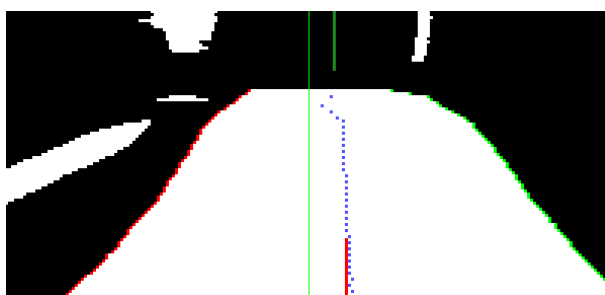


图 5.9 SD 卡记录的桥图像

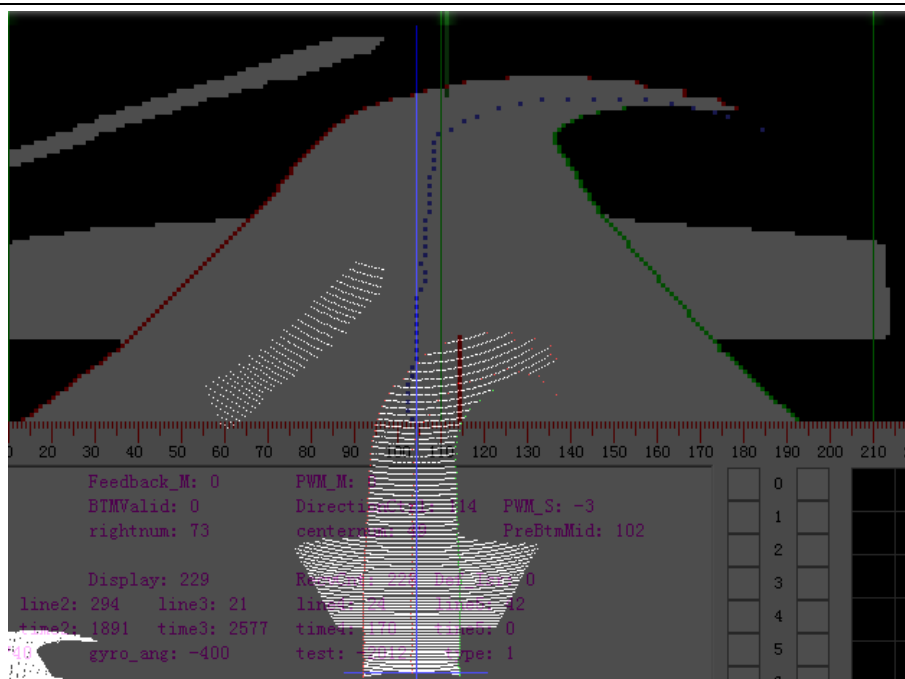


图 5.10 上位机校正效果显示

## 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。

### 5.3.2 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。

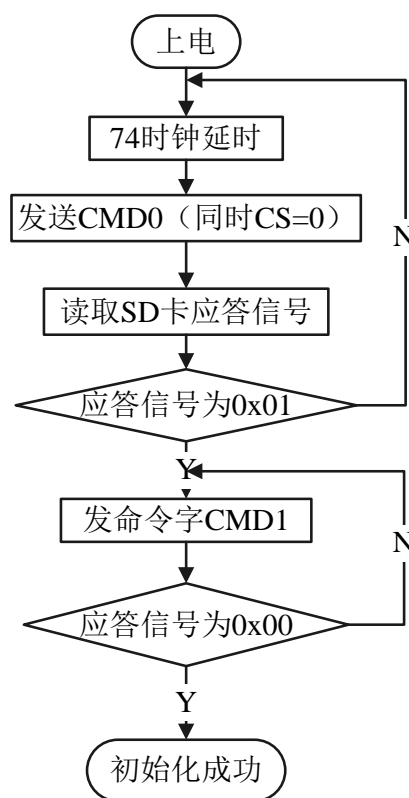


图 5.11 SD 卡初始化步骤

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。

## 5.4 键盘模块

为了方便智能车调试，利用 ZLG7290 芯片开发了键盘模块，采用为 I2C 总线接口。通过键盘可以方便输入参数、选择策略。ZLG7290 键盘电路图如图 5.12 所示。

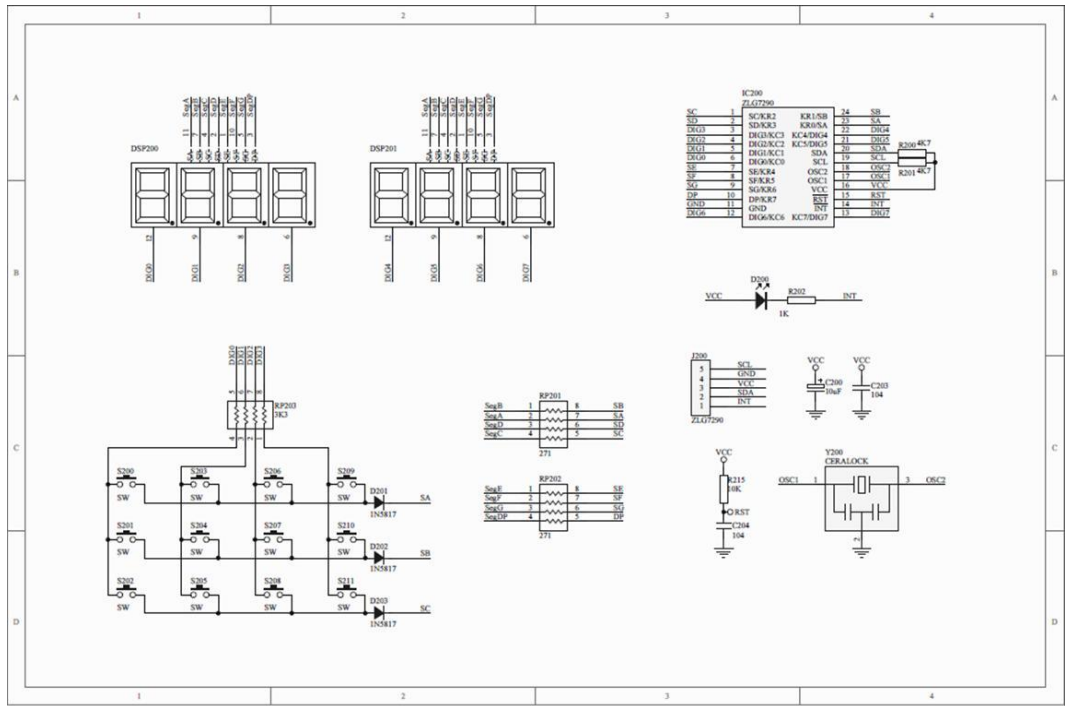


图 5.12 ZLG7290 键盘电路图



## 第六章 模型车的主要技术参数

赛车基本参数	长	27cm
	宽	20cm
	高	34cm
车重		1007g
功耗	空载	8W
	带载	大于 10W
电容总容量		1600uF
传感器	方波编码器	1 个
	CMOS 摄像头	1 个
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）	5/220cm
	精度(近/远)	2/12.5mm
	频率	50Hz

---

## 结 论

自报名参加“飞思卡尔”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，我们分析了舵机转向系统的改进办法，前轮束角和主销倾角的调整以及在其他细节方面的优化。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，经过不断实验，最后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) 臧杰, 阎岩. 汽车构造[M]. 北京: 机械工业出版社, 2005.
- (5) 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.
- (6) 张文春. 汽车理论[M]. 北京: 机械工业出版社, 2005.
- (7) 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.
- (8) 阎石. 数字电子技术基础 [M]. 北京: 高等教育出版社, 2000.
- (9) 谭浩强著. C 程序设计. 北京: 清华大学出版社, 2003.
- (10) 尹勇. Protel DXP 电路设计入门与进阶 [M]. 北京: 科学出版社, 2004.
- (11) 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.
- (12) 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.
- (13) 夏克俭. 数据结构及算法 [M]. 北京: 国防工业出版社, 2001.
- (14) 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.
- (15) 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.
- (16) 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.

## 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);
}
```