

# 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：太原理工大学

队伍名称：晋速二队

参赛队员：柴庆龙

潘俊

马亚丽

带队教师：谢刚、闫高伟

# 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：\_\_\_\_\_

# 目录

|                      |    |
|----------------------|----|
| 第一章 引言 .....         | 1  |
| 第二章 系统整体设计 .....     | 2  |
| 2.1 系统概述 .....       | 2  |
| 2.2 整体布局 .....       | 2  |
| 第三章 机械结构设计 .....     | 4  |
| 3.1 舵机安装及调整 .....    | 4  |
| 3.2 摄像头的安装 .....     | 5  |
| 3.3 编码器安装 .....      | 6  |
| 3.4 前轮调整 .....       | 6  |
| 3.5 后轮差速调整 .....     | 7  |
| 第四章 硬件系统设计及实现 .....  | 8  |
| 4.1 最小系统板的设计 .....   | 8  |
| 4.2 供电底板的设计 .....    | 9  |
| 4.3 电机驱动电路 .....     | 11 |
| 第五章 软件系统设计与实现 .....  | 14 |
| 5.1 程序流程图 .....      | 14 |
| 5.2 单片机资源利用 .....    | 15 |
| 5.3 图像采集 .....       | 15 |
| 5.4 黑线提取 .....       | 17 |
| 5.4.1 整体黑线提取思想 ..... | 17 |
| 5.4.2 单行跳变的提取 .....  | 18 |
| 5.4.3 黑线的完整提取 .....  | 18 |
| 5.5 黑线效果 .....       | 19 |
| 5.6 PID 算法简介 .....   | 21 |
| 5.6.1 位置式 PID .....  | 21 |
| 5.6.2 增量式 PID .....  | 22 |
| 5.7 转向控制算法 .....     | 22 |
| 5.7.1 赛车位置偏移计算 ..... | 23 |
| 5.7.2 转弯斜率计算 .....   | 23 |
| 5.7.3 转向控制算法 .....   | 24 |
| 5.8 电机速度控制 .....     | 25 |
| 第六章 开发工具及调试说明 .....  | 26 |
| 6.1 软件开发平台 .....     | 26 |
| 6.2 智能车上位机设计 .....   | 27 |
| 第七章 模型车技术参数统计 .....  | 29 |
| 第八章 总结 .....         | 30 |
| 参考文献 .....           | I  |
| 附录 .....             | II |

## 第一章 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。

教育部为了加强大学生实践、创新能力和团队精神的培养，经研究决定，委托教育部高等学校自动化教学指导分委会主办每年一度的全国大学生智能汽车竞赛，并成立了由教育部、自动化分教指委、清华大学、飞思卡尔半导体公司等单位领导及专家组成的“飞思卡尔”杯全国大学生智能汽车竞赛。该竞赛是为了提高大学生的动手能力和创新能力而举办的，具有重大的现实意义。与其它大赛不同的是，这个大赛的综合性很强，是以迅猛发展的汽车电子为背景，涵盖了控制、模式识别、传感、电子、电气、计算机和机械等多个学科交叉的科技创意性比赛，这对进一步深化高等工程教育改革，培养本新意识，培养硕士生从事科学、技术研究能力，培养博士生知识、技术创新能力具有重要意义。

智能车比赛是搭载组委会提高的标准汽车模型上的，采用飞思卡尔半导体公司的 8 位、16 位、32 位指定类型的微控制器为核心控制模块，通过指定种类的路径检测传感器，设计供电电路、驱动电路及编写相应的控制程序，装配出一辆能够自主识别道路的汽车模型，并且能够按照道路，以最快时间跑完整个道路者获胜。

该项比赛现在主要分为摄像头，光电，电磁三大类，并且已经成功举行了六届，比赛的规模也在不断的扩大，比赛的成绩也在逐年提高。不断挑战着参赛队伍的团队合作能力，动手能力和耐压能力。

## 第二章 系统整体设计

### 2.1 系统概述

本系统是以 MC9S12XS128 为核心，需要设计最小系统板，电源模块和电机驱动电路。需要通过摄像头读取赛道信息，通过黑白跳变的思路提取赛道引导黑线，并根据提取到的引导线特征对舵机转向和电机速度进行控制。通过使用编码器测速，构成闭环 PID 控制。为了提高调试效率，使用了上位机。通过单片机发送需要观察的数据给上位机，提高调试效率。整体框架如图 2.1 所示。

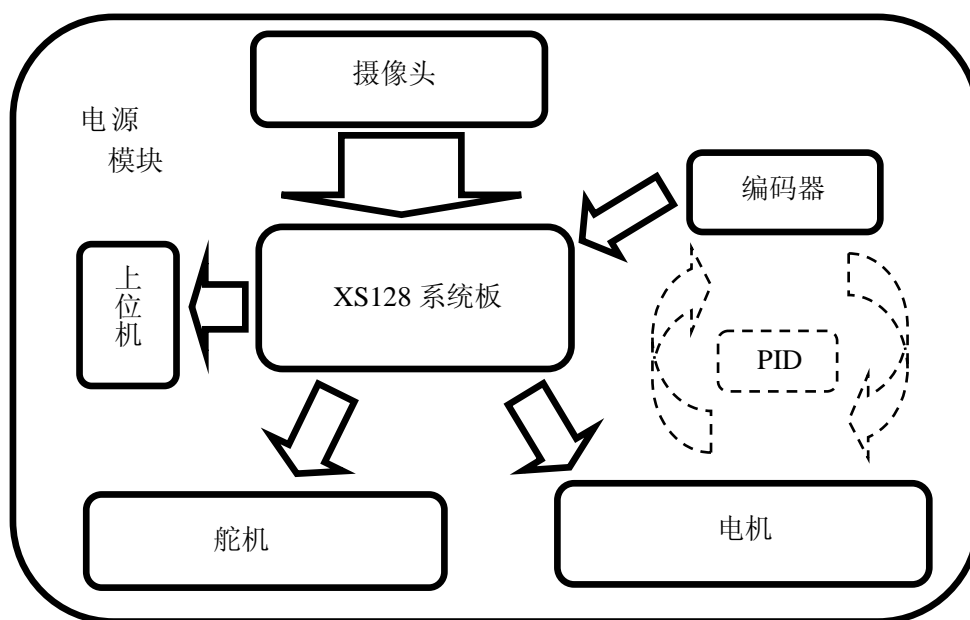


图 2.1 系统总体框架图

### 2.2 整体布局

针对今年的车模，我们采用以下策略提高车子的性能，车子整体图片如图 2.2 所示。

摄像头尽可能后移，减少摄像头前端死区；

尽可能减少车身重量，提高智能车整体性能；

舵机采用竖立安装，加长舵机臂长，提高堆积响应。

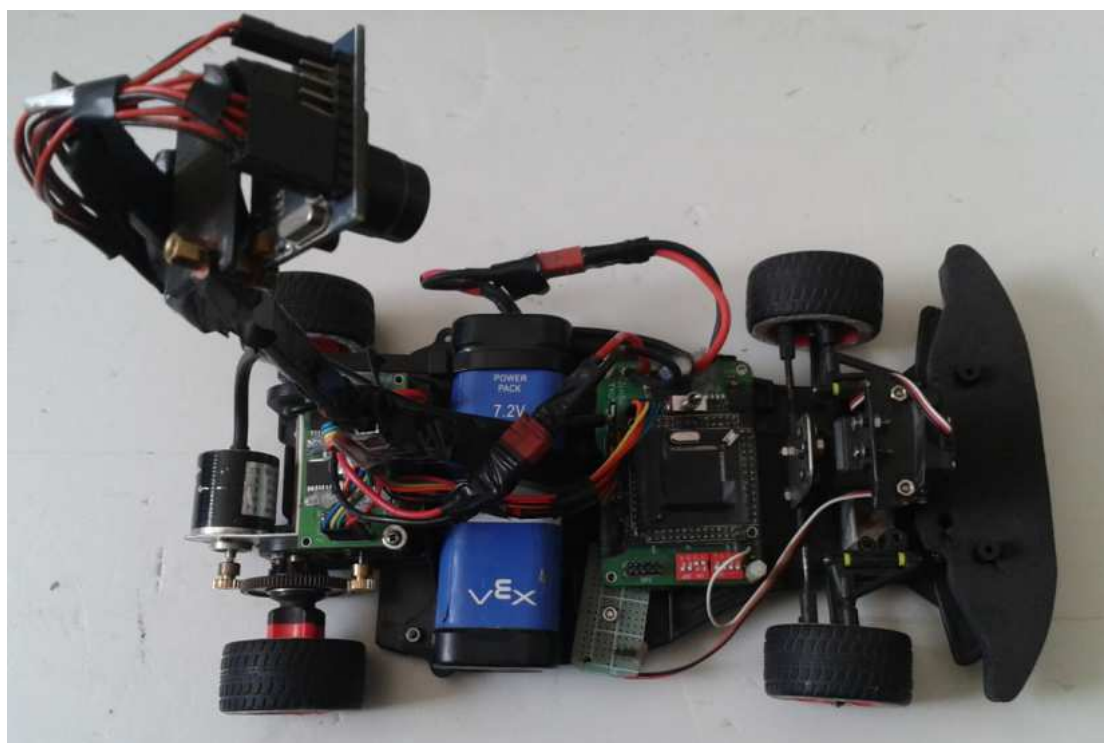


图 2.2 车子整体效果

## 第三章 机械结构设计

根据组委会的相关规定，第七届比赛摄像头组比赛车模采用 A 型车模。机械特性直接影响着车子的转向特性，稳定性。因此对车子进行较好的机械调整十分必要。在对车模进行了详细的系统分析后，对机械进行了适当的调整。

### 3.1 舵机安装及调整

原装车模的舵机为卧式安装，这种安装方式使得两边的摆臂长度不等，在两侧转向时引起偏差。同时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。而采用卧式安装，通过简单的计算发现小车转向响应较慢。所以本设计决定采用立式安装。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。经过多次的实验终于找到了一种较好的安装的方法，不仅使舵机更加稳固，而且增加了舵机转向的响应速度。如图 3.1 所示。



图 3.1 智能车舵机安装

### 3.2 摄像头的安装

摄像头是智能车提取赛道信息的关键。摄像头的安装直接影响着摄像头所采集图像的视野，清晰度，因此摄像头的安装十分重要。在摄像头安装的过程中主要考虑以下几点：

#### （1）摄像头的安装位置

摄像头的安装位置，主要分为前部和后部，经过试验发现摄像头安装在前部容易导致车子前端盲区过大，这会对引导线的提取产生过大的影响作用。所以本设计决定将摄像头的位置后移，通过验证能够得到合适的视野范围，减少了视野死区和图像畸变。

#### （2）摄像头的高度

摄像头的高度主要影响摄像头的前瞻和畸变率，过低会导致图像顶端畸变过大，给舵机控制带来困难。过高虽然有利于提高摄头的前瞻，但会导致车子的重心偏高，高速情况下容易侧翻，增加了智能车过弯道的难度。

#### （3）摄像头与支架的连接方式

摄像头与支架的连接方式同样重要，因为摄像头在刚安装上去的时候，视野很难合适，需要进行一定程度的调节。所以设计的架法需要能够实现图像的上下矫正，和轻微的左右矫正。

在多次测试后采用了图 3.2 的安装方式。不仅底座稳定，而且可以对摄像头的的事业进行灵活的调节。



图 3.2 舵机安装



### 3.3 编码器安装

码器是测速的核心器件。齿轮传动机构对车模的驱动能力有很大的影响。齿轮传动部分安装不恰当，会增大电机驱动后轮的负载；齿轮配合间隙过松则容易打坏齿轮过紧则会增加传动阻力。所以在编码器的安装过程中尽量使得传动部分轻松、流畅，不存在卡壳或迟滞现象。如图 3.4 所示



图 3.4 编码器安装

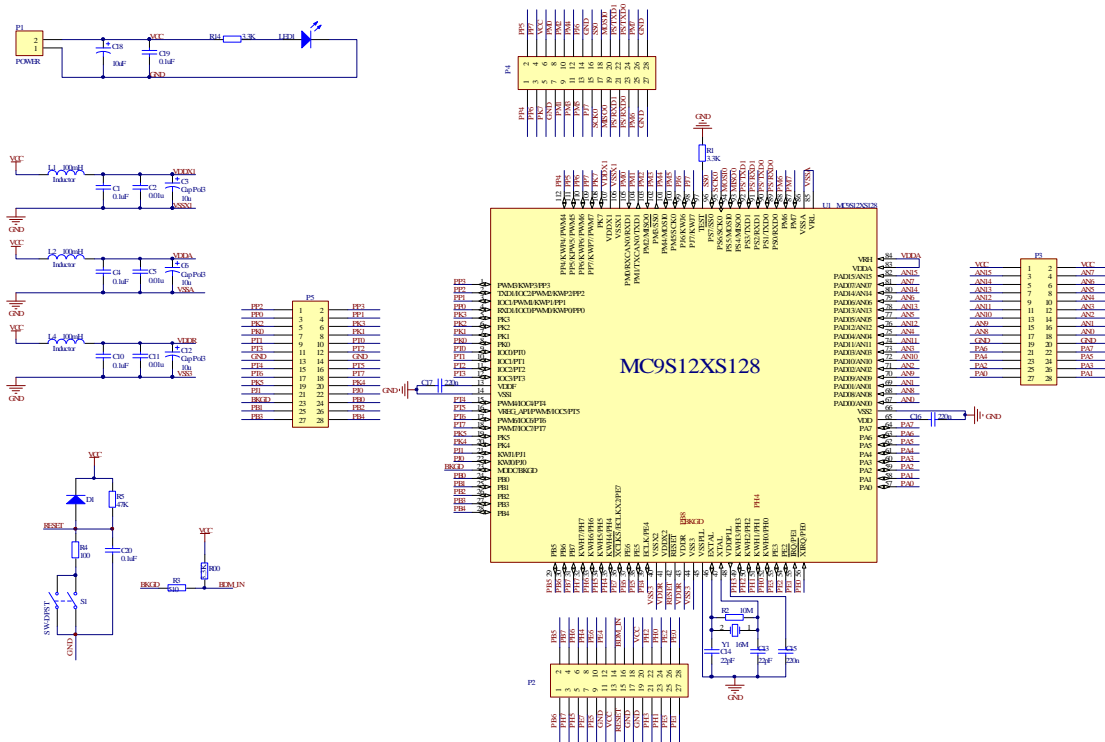
### 3.4 前轮调整

在调试过程中，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，决定对前轮的安装角度，即前轮定位进行调整。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、主销和前轴等三者车架上的位置关系。

在实际调试中，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。

### 3.5 后轮差速调整

在车实际行驶的过程中发现，后面两个轮的相对松紧程度，也就是差速对拐弯有很大的影响。差速太紧，则拐弯容易甩尾，速度快的时候很容易甩出去，但是差速如果太松，虽然会改善转弯性能，但是严重影响了直线上的加减速，而且齿轮的咬合也不是很好，对齿轮会有一定的损坏。所以，后面两个轮子的相对松紧程度要适中，经过多次的调试，得出了比较满意的效果，即将模型车放到跑到上，用手抓住后轮的一只轮子使其不能转动，在赛道上推车子转弯，如果车子能够稍轻松的推动，则此时的差速器为最适合。当然，实际还需要根据不同的赛道和车的机械性能进行相应的微调。调节差速可以通过工具旋紧或者旋松右后轮来得到合适的差速控制，其实就是适当的调整两片轴承的压力以满足后轮驱动和差速的要求。



PT7: 脉冲累加器; ,

PWM3: 舵机控制;

PWM1、PWM5: 电机正反转控制;

PS0-3: SCI 通信口;

PS4-7: SPI 通信口;

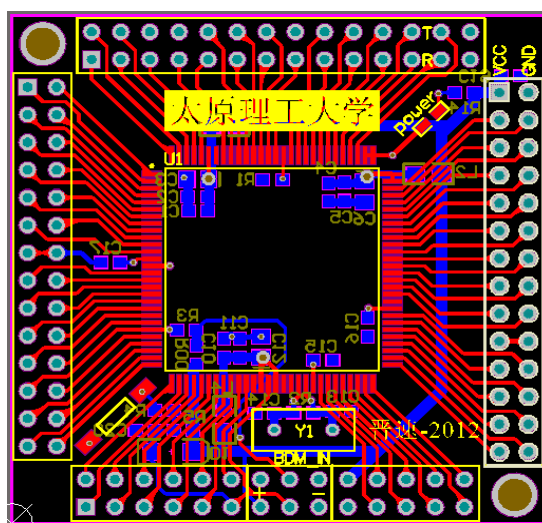


图 4.1.2 最小系统板 PCB 图

## 4.2 供电底板的设计

电源是一个系统正常工作的基础，电源模块为系统其他各个模块提供所需要的能源保证，因此电源模块的设计至关重要。模型车系统中接受供电的部分包括：传感器模块、单片机模块、电机驱动模块、伺服电机模块等。同时，为了减少杜邦线的使用，减少外部的干扰和出错的几率，同时使整车布局更加简洁合理，重心适中，方便车子高速，决定将各类电源模块、拨码开关、指示灯、其它辅助调试设备等都集成在该底板上，通过专门的接口和最小系统板相连接。在实际应用中大大方便了各类器件连线的插拔，给调试带来了很大的方便。其硬件结构原理图如下图 所示：

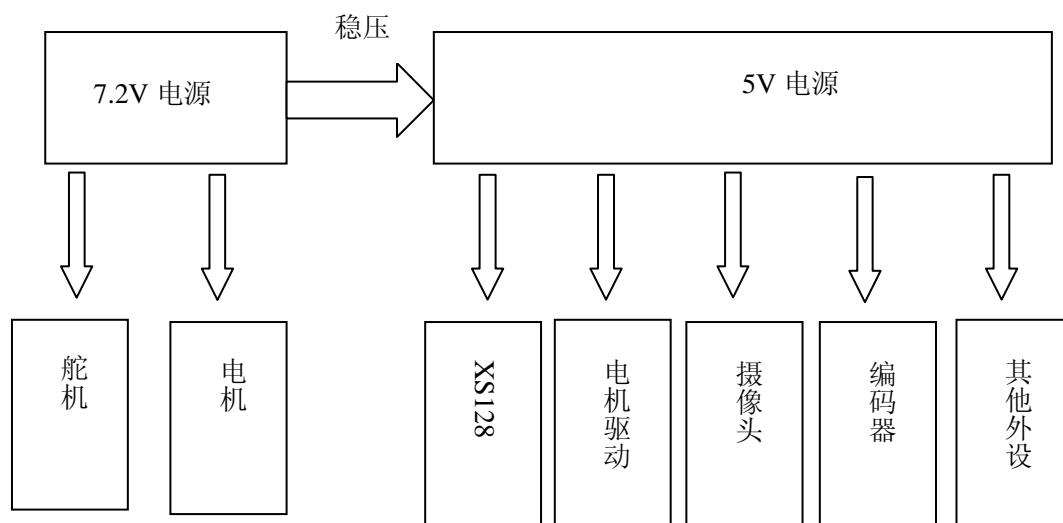


图 4.2.1 底板电路框架

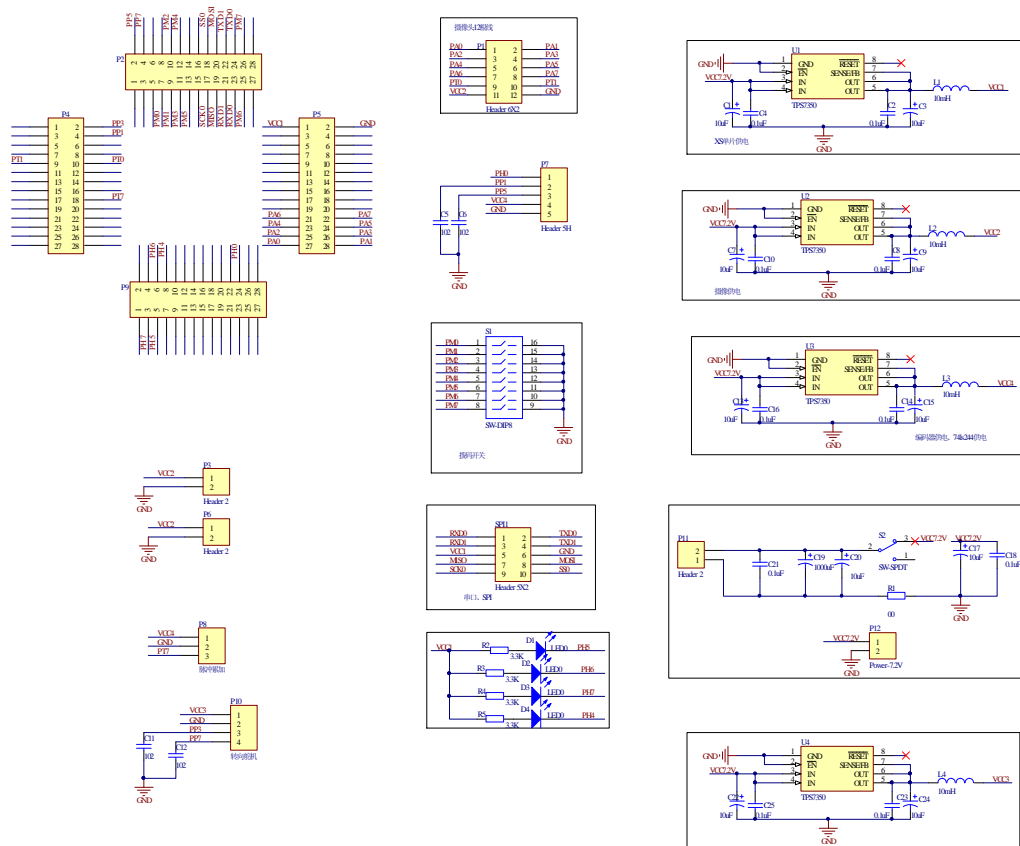


图 4.2.2 底板电路原理图

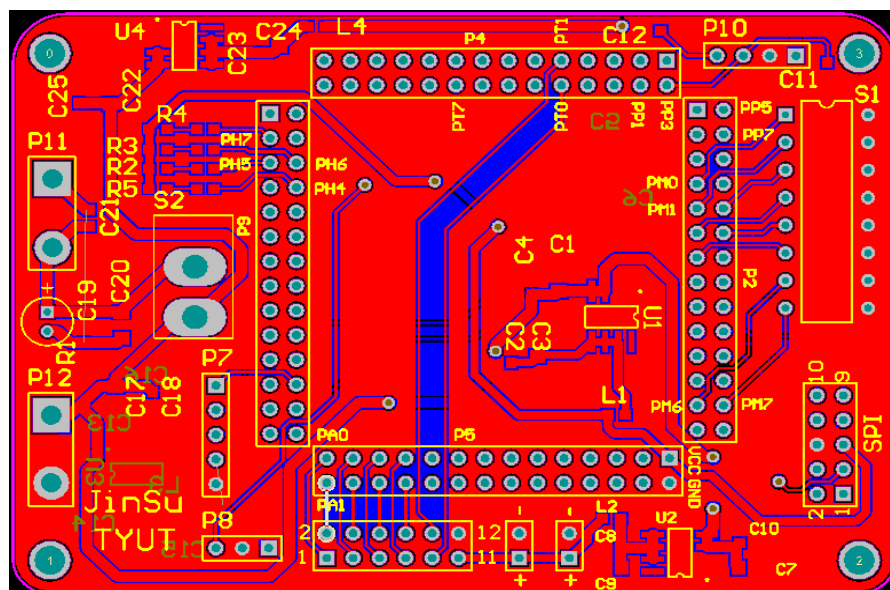


图 4.2.3 底板电路 PCB 图

### 4.3 电机驱动电路

为了使电机能够较快的实现正反转，设计中采用了 H 桥电路。通过 PWM 来控制 H 桥上的电流方向和通电时间来控制直流电机。驱动芯片采用 BTS7970。BTS7970 芯片是个理想的含半桥的单块集成电路。BTS7970 具有较低的工作压降和较小的静态工作电流，可以使电池获得相对更长的使用时间。由于热损失小，因此不需要专门考虑散热问题。其特性为：

- (1) 工作电压：-0.3-45V；
- (2) 导通电阻：16 毫欧姆；
- (3) 输入信号：驱动电路和逻辑电平输入；
- (4) PWM 频率：< 25KHZ；
- (5) 具有短路保护、欠压保护、过电压保护等；
- (6) 连续控制电流 68.0A。

为了增大驱动能力，采用了两个 H 桥并联的方式，同时决定将控制信号在进入 BTS7970 芯片前，先进入 74LS244 芯片来进一步增大信号的驱动能力。各引脚的功能：

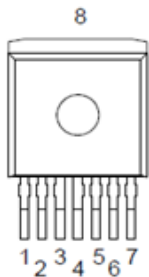


图 4.3.1 BTS7970 引脚

表 0.1 BTS7970 引脚功能

| 引脚   | 引脚名称 | 功能        |
|------|------|-----------|
| 1    | GND  | 数字地       |
| 2    | IN   | 输入        |
| 3    | INH  | 抑制        |
| 4, 8 | OUT  | 功率输出      |
| 5    | SR   | 转换速率      |
| 6    | IS   | 电流检测和故障诊断 |
| 7    | VS   | 电源供电      |

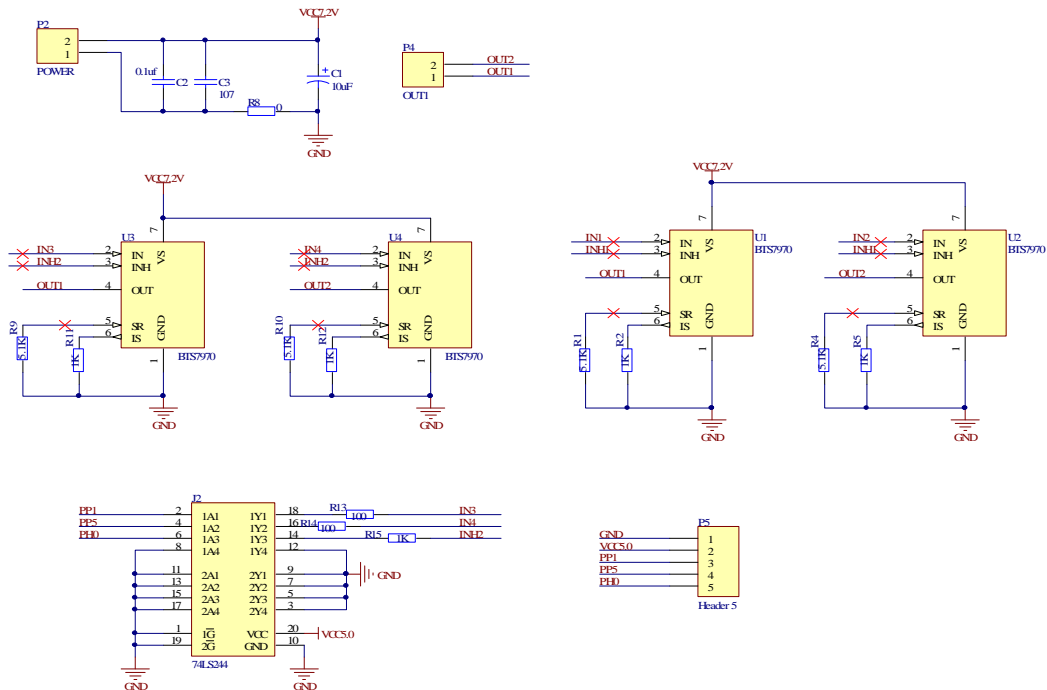


图 4.3.2 驱动电路的原理图

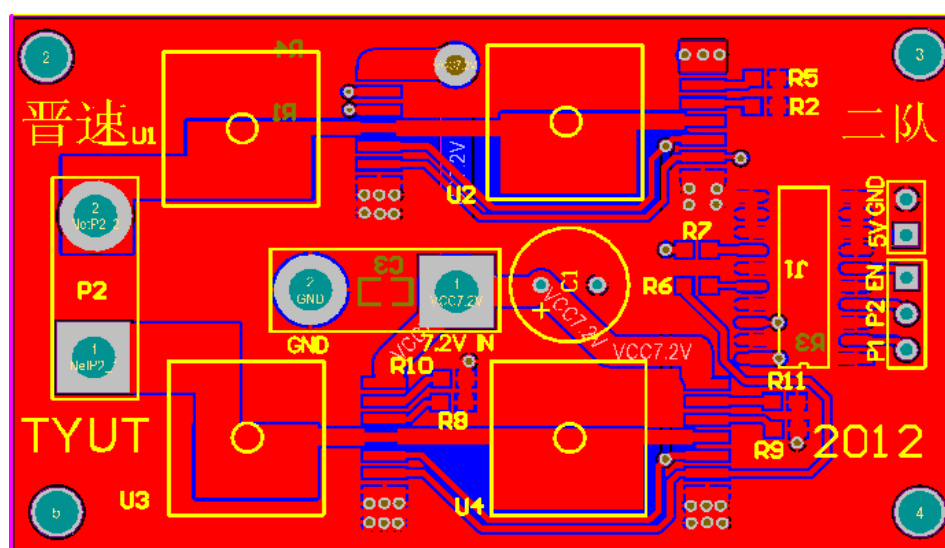


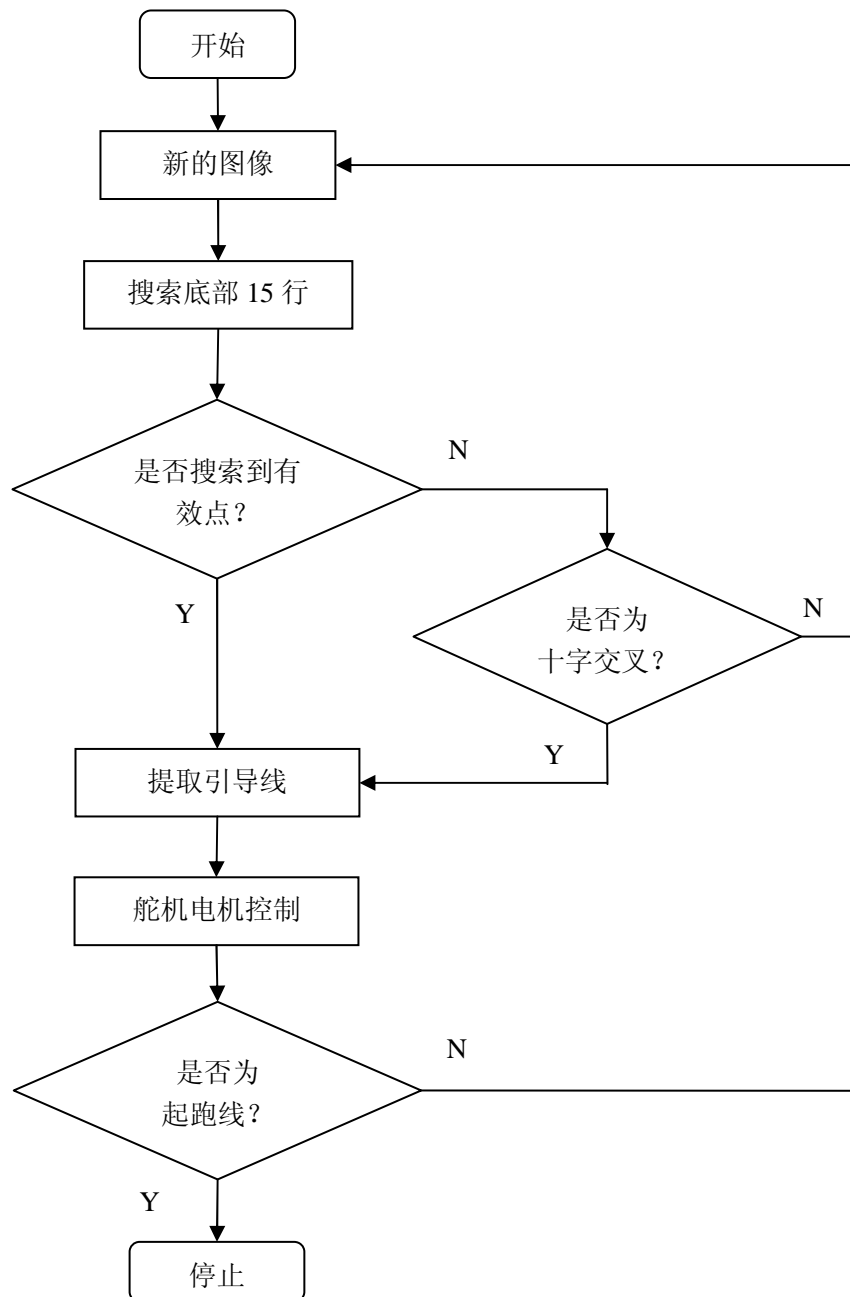
图 4.3.3 驱动电路的 PCB 图



## 第五章 软件系统设计与实现

### 5.1 程序流程图

整个的软件的核心是对舵机和电机进行控制。摄像头采集图像，提取引导线，然后据此控制舵机和电机。系统流程如图 5.1 所示。



5.1 程序流程图

## 5.2 单片机资源利用

在智能车的软件设计中，用到了单片机的 PWM 模块，用于电机的正反转和舵机的转向角控制；用到了增强型 ECT 模块，用于对摄像头的场行中断的捕捉；用到了普通 IO 口，用于对摄像头返回的灰度值的读取以及电机使能等其他功能；用到了特殊功能引脚，通过 SCI、SPI 等协议对智能车进行调试。其整体资源占用如图 5.1 所示。

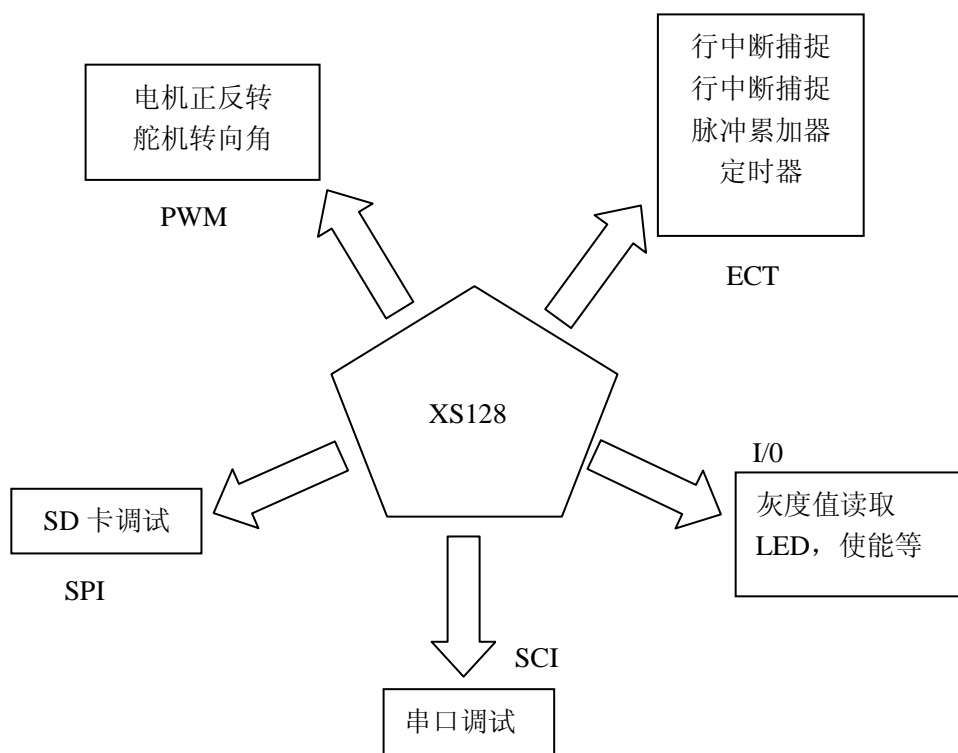


图 5.1 整体资源分配图

## 5.3 图像采集

图像采集的难点在于工作时序的把握。摄像头的图像采集是通过对场行中断捕捉来完成的。场中断标志着一场图像的到来，行中断标志着一行图像到来。行把所读去的行灰度信息组合在一起，就是一幅图像。因此什么时候打开或关闭场行中断是正确采集图像的关键。其实现算法控制流程图如图 5.2 所示。

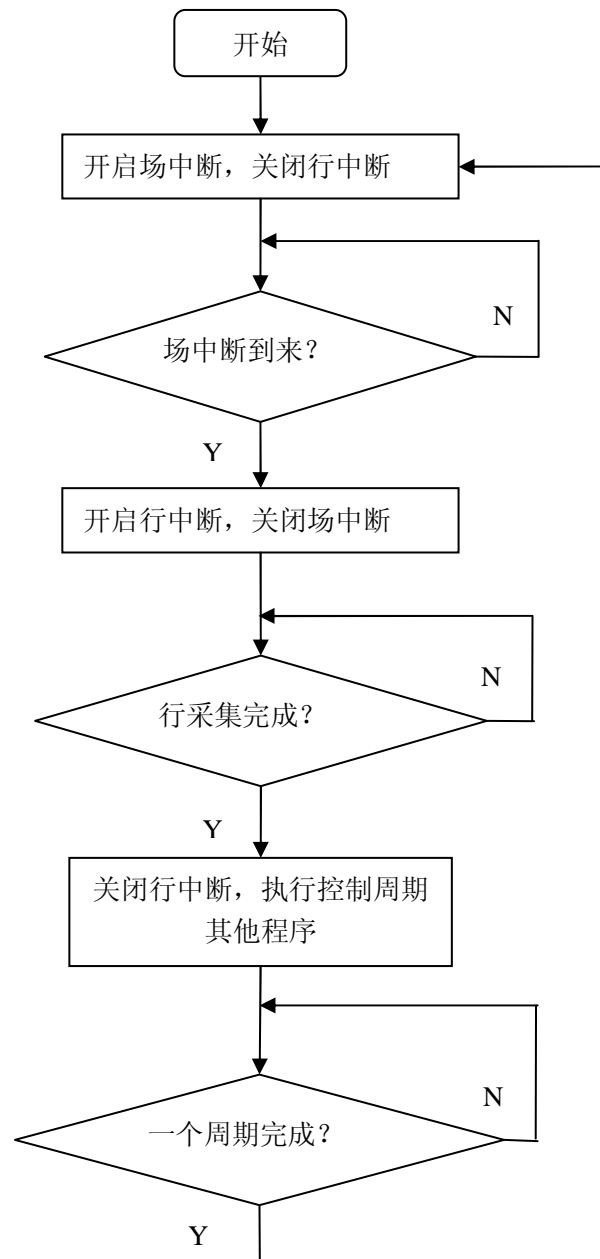


图 5.2 采集图像流程图

## 5.4 黑线提取

### 5.4.1 整体黑线提取思想

赛道黑线提取的关键在于能够找准一个基点，并以此作为起点寻找黑线。为此本设计将赛道分为两部分，即前 15 行整行搜索跳变,并依此确定基点。如果左右两边都不存在基点，则根据实际情况只能为十字交叉，根据赛道特征，可以在 15 行到 30 行再次寻找基准。在确定了基点后，即可限定搜索范围上下快速搜索符合条件的跳变点。其整体算法流程图如图 5.4.1 所示

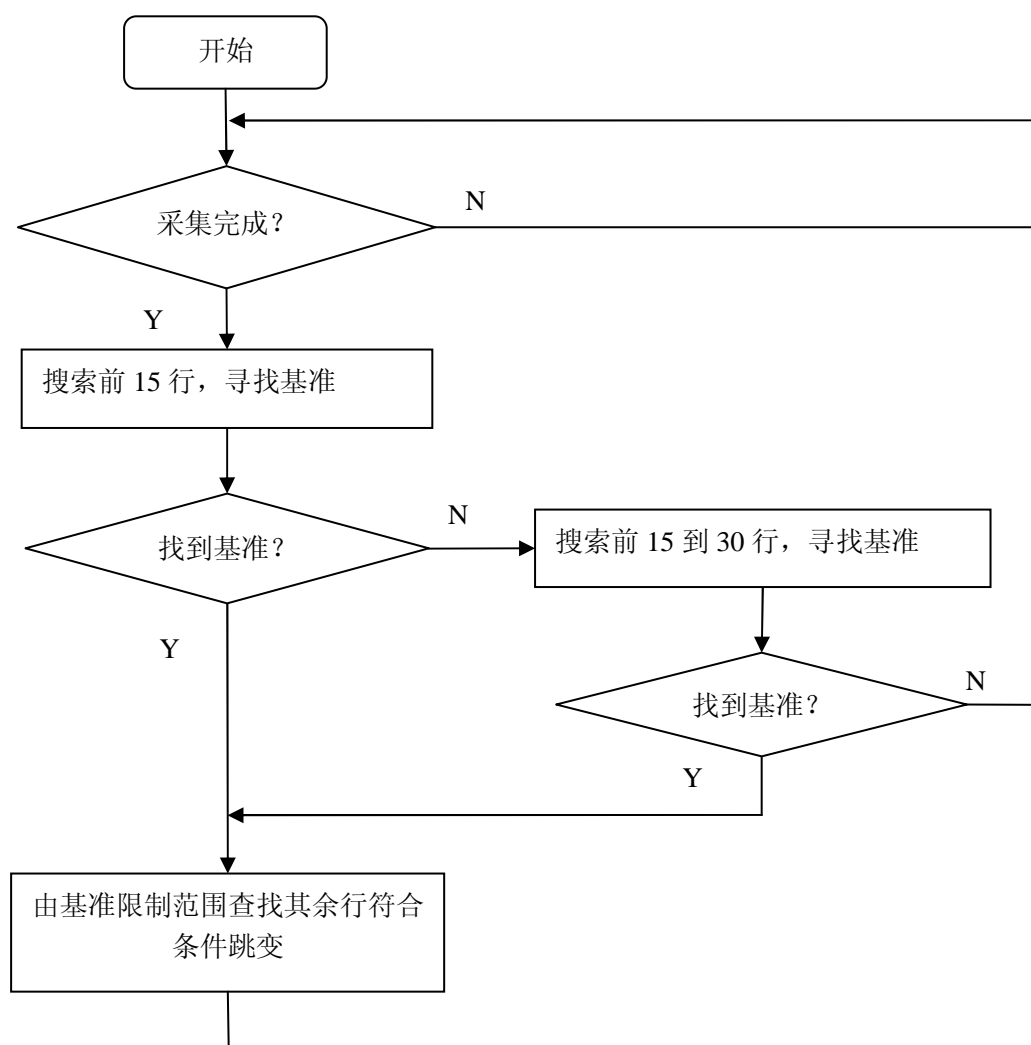


图 5.4.1 黑线提取算法流程图

### 5.4.2 单行跳变的提取

从摄像头读取的赛道信息是代表对应像素点的灰度值。其占 8 位，从 0 到 255 对应着纯黑到纯白的渐变。其灰度值越高对应的颜色越白。提取引导线的基本思想便是寻找白到黑的跳变。赛前不知道赛道背景，不能利用背景色，为提高寻找跳变的精度，本设计采取了从中心向两边寻找跳变的思路。一行的灰度图的示意图如图 5.4.2 所示。

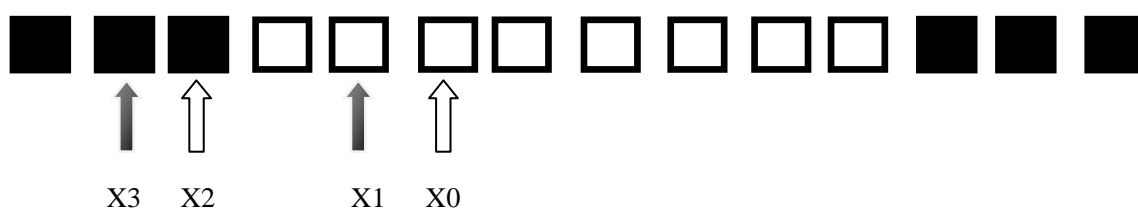


图 5.4.2 单行跳变示意图( $X_0$ 、 $X_1$ 、 $X_2$ 、 $X_3$  代表指示点的灰度值)

其搜索跳变的思想就是如果“ $X_1-X_2$ ”和“ $X_0-X_3$ ”的灰度值大于一给定阈值则认为是一跳变同时记录  $X_2$  的坐标。

### 5.4.3 黑线的完整提取

当利用单行提取的思想提取了一定长度的黑色引导线时，便可一次为基准，限定对应下一行或上一行的搜索范围再次利用单行提取搜索跳变的思想搜索跳变，从而较少了搜索范围，大大提高了程序的执行效率，减少了程序的执行时间。如图 5.4.3 所示实验证明能够较好的找到黑线。

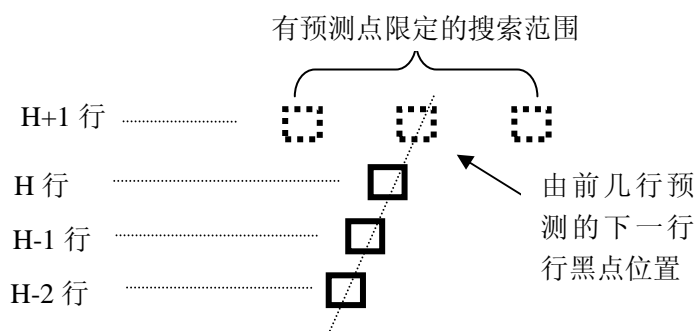


图 5.4.3 限定范围搜索示意图

## 5.5 黑线效果



图 5.5.1 真实直道

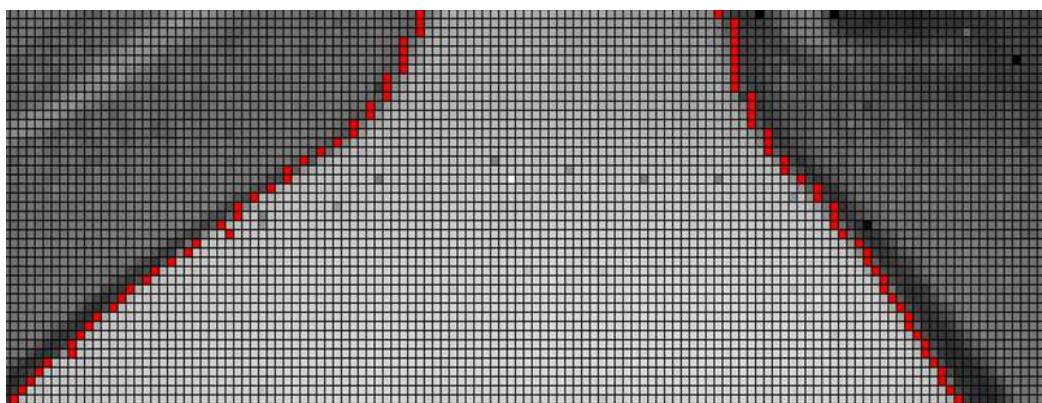


图 5.5.2 提取的直道

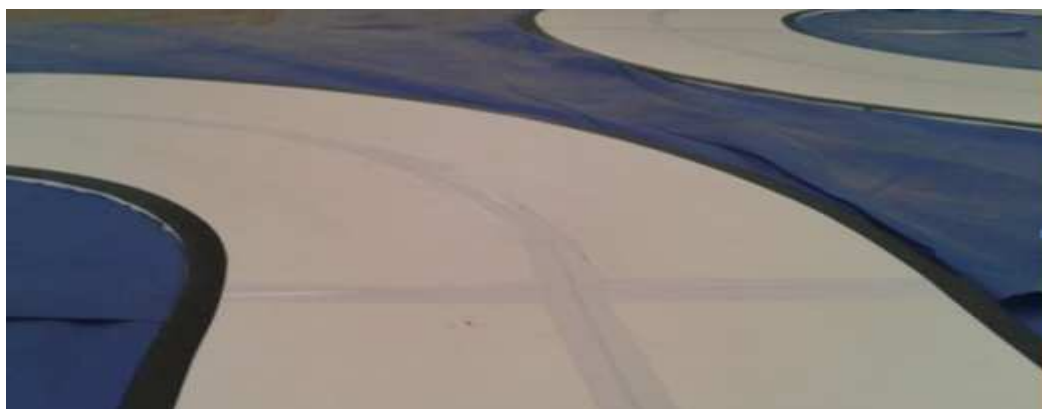


图 5.5.3 真实入弯

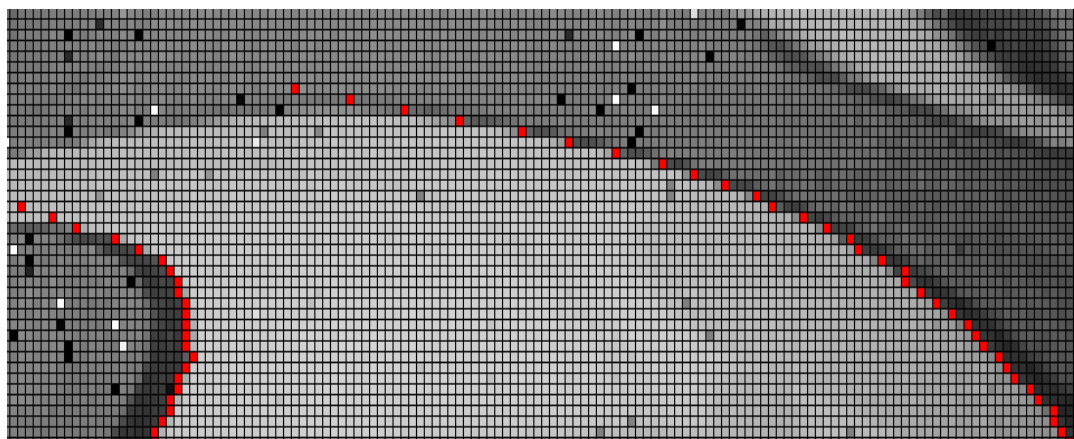


图 5.5.4 提取的黑线入弯



图 5.5.5 真实 S 弯

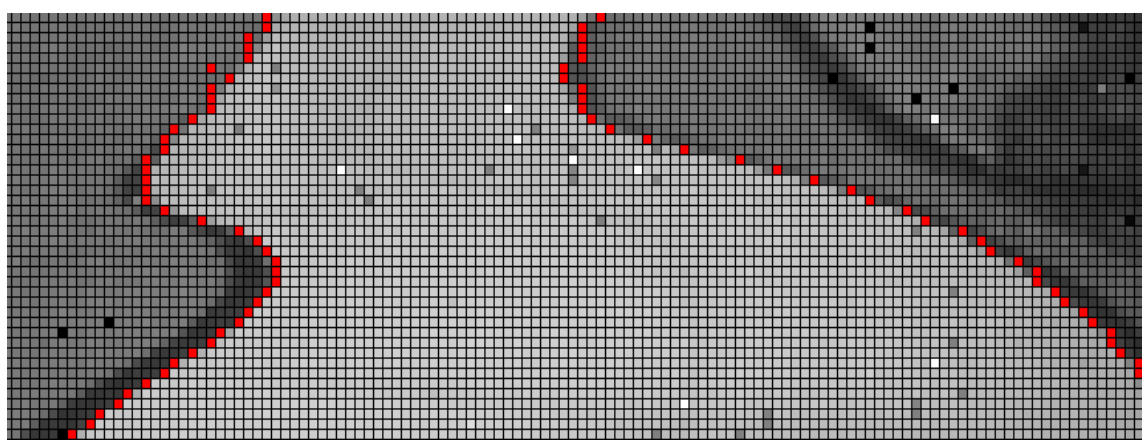


图 5.5.6 提取 S 弯

## 5.6 PID 算法简介

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<sup>[13]</sup>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。

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。

PID 控制算法包括模拟 PID 和数字 PID 两种控制算法。模拟 PID 调节具有原理简单、易于实现、鲁棒性强和适用面广等优点，在实际应用中，根据实际工作经验在线整定 PID 各参数，往往可以取得较为满意的控制效果。数字 PID 控制则以此为基础，与计算机的计算与逻辑功能结合起来，不但继承了模拟 PID 调节的优点，而且由于软件系统的灵活性，PID 算法可以得到修正而更加完善，使变得更加灵活多样，更能满足生产过程中提出的各种控制要求。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

### 5.6.1 位置式 PID

$$u(t) = K_p[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt}] \quad ①$$

式中，u(t) --控制量

KP--比例系数

Ti--积分时间常数



$T_d$ —微分时间常数

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。

### 5.6.2 增量式 PID

由于位置式 PID 的缺点，便产生了增量式 PID<sup>[14]</sup> 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  $\Delta u(k)$ 。其是由位置式 PID 演变来的。

$$u(k) = u(k-1) + K_p [e(k) - e(k-1) + \frac{T}{T_i} * e(k) + \frac{T_d}{T} * (e(k) + e(k+2) - 2 * e(k-1))]; \quad (2)$$

增量式 PID 具有以下优点：

- (1) 由于计算机输出增量，所以误动作影响小，必要时可以用逻辑的方法去掉。
- (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。

## 5.7 转向控制算法

优秀的舵机控制算法，不仅能够保证智能车在低速下，能够按较优的路径行驶；在高速下也必须对赛道有一定的预判性，使舵机能够及时对赛道的变化产生相应。

### 5.7.1 赛车位置偏移计算

在控制智能车的算法中，开始是采用了车子给定某行相对赛道中心的偏移量作为控制量，来对舵机进行控制。当提取了图像中的黑色引导线，选定一边作为打向基准，那么相对于某给定行，智能车在行驶过程中的位置就跟车子处于正中心的位置，有了一定的偏差，此偏差量的大小与智能车偏离赛道中心的程度呈正相关。因此可以用偏差量对舵机进行控制。智能车的偏移量  $OFFSET\_X$  如图 5.7.1 所示。

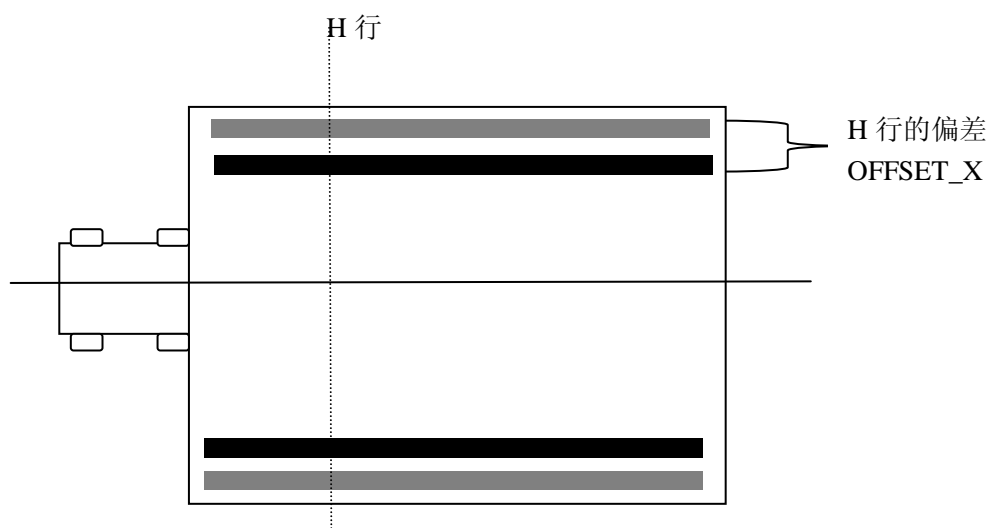


图 5.7.1 赛车位置 H 行偏移示意图（左基准为例）

### 5.7.2 转弯斜率计算

实践中发现如果仅通过偏移量对舵机进行控制，智能车在低速的情况下，能够选择较好的路径，但是在较高的速度下，很容易冲出赛道。其原因就是在舵机控制程序中仅利用单行的偏移量很难反映出赛道的变化规律，舵机延迟影响加重。经过多次试验后，对舵机的控制中加入了斜率控制来改善舵机打向的延迟现象。因为斜率可以在一定程度上反映出智能车下一步舵机打向，具体斜率算法如图 5.7.2 所示：

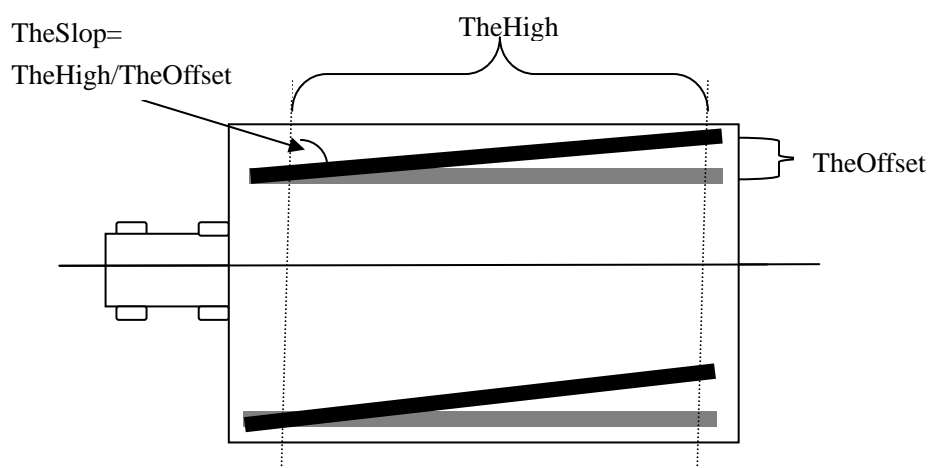


图 5.7.2 斜率说明示意图

### 5.7.3 转向控制算法

在算出车子某给定行的偏差，和给定两行之间的斜率的情况下，程序色好几中将偏移量和斜率量分别给以个增益量，这样就可以通过调节着两个增益量来调节器占的比重和相对舵机转向角的增益关系。实验证明通过调节着两个参数能够较好的实现，高速行驶下的切内到入弯道，是车子在高速下能够按照较好的路径行驶。如图 5.7.3 所示。

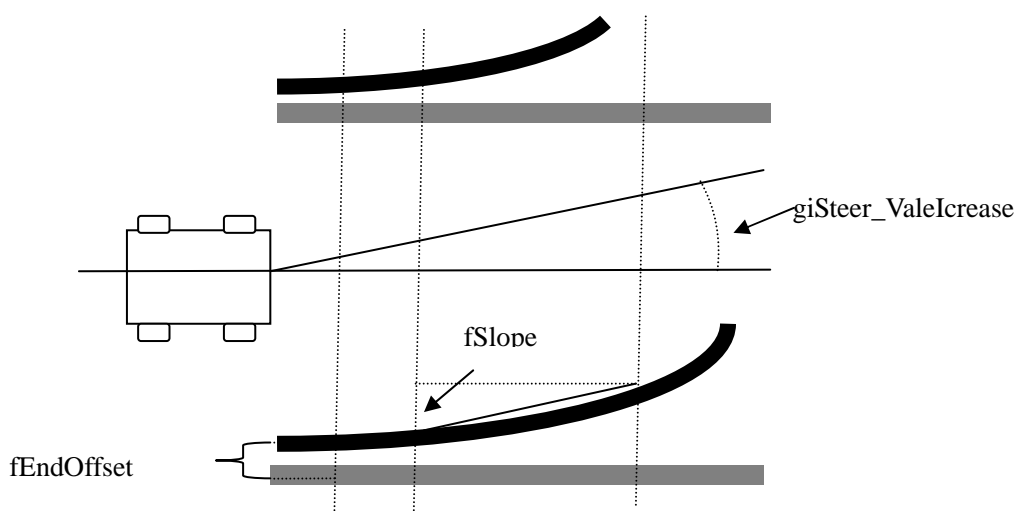


图 5.7.3 舵机控制示意图

其中 Sloped 代表给定两行斜率, fEndOffset 代表给定行的偏移, 对应舵机打向增量  $giSteer\_ValeIncrease = fKa * fEndOffset + fKb * fSlope$ ; fKa, fKb 为给定常数。

### 5.8 电机速度控制

对于电机的控制, 主要目的是实现: 在直道、小 S 等舵机打向值相对小的情况下, 车子能够高速行驶; 在入弯道的时候, 车子能够减速下来。对此本设计决定对电机采用增量型的 PID 控制算法, 提高车子对速度的响应能力, 因此需要做出赛道类型判断, 还需要设计 PID 算法。

对于电机控制再进行赛道类型判断, 主要的目的区别出直道, 小 S 等能够高速行驶的赛道类型和入弯、大 S 等需要低速行驶的赛道类型, 从而能够给不同的设定速度。通过观察发现了一个简单的方法: 在能高速行驶的赛道中, 提取到的黑线都能够达到 38 行以上, 否则应该低速行驶。

当实现了赛道类型的判断, 就可以根据不同的赛道类型给电机不同的速度。接下来就是需要设计 PID 控制程序, 由于增量式 PID 的优点, 本设计采用了增量式的 PID 控制程序。设计的过程中, 应该注意执行器输出的最大值, 和对偏差的保存。

## 第六章 开发工具及调试说明

拥有全面的开发调试工具，能够大大提高编程的效率，还能够观察智能车在行驶中的状态，从而大大降低了在智能车调试过程中的盲目性，提高了发现问题的效率。本设计在开发调试中用到了 Code Warrior4.7 开发软件，另外为了调试方便，本设计还采用了上位机、串口调试模块、SD 卡和外部状态指示单元等。

### 6.1 软件开发平台

此次智能车大赛的软件开发平台为 Code Warrior4.7 开发软件。CodeWarrior 的功能非常强大，可用于绝大部分单片机、嵌入式系统的开发。用户可在新建工程时将芯片的类库添加到集成环境开发环境中，工程文件一旦生成就是一个最小系统，用户无需再进行繁琐的初始化操作，就能直接在工程中添加所需的程序代码。包括集成开发环境 IDE、处理器专家库、全芯片仿真、可视化参数显示工具、项目工程管理器、C 交叉编译器、汇编器、链接器以及调试器。为软件的开发调试提供了极大的方便。其使用界面如图 6.1.1、6.1.2 所示：

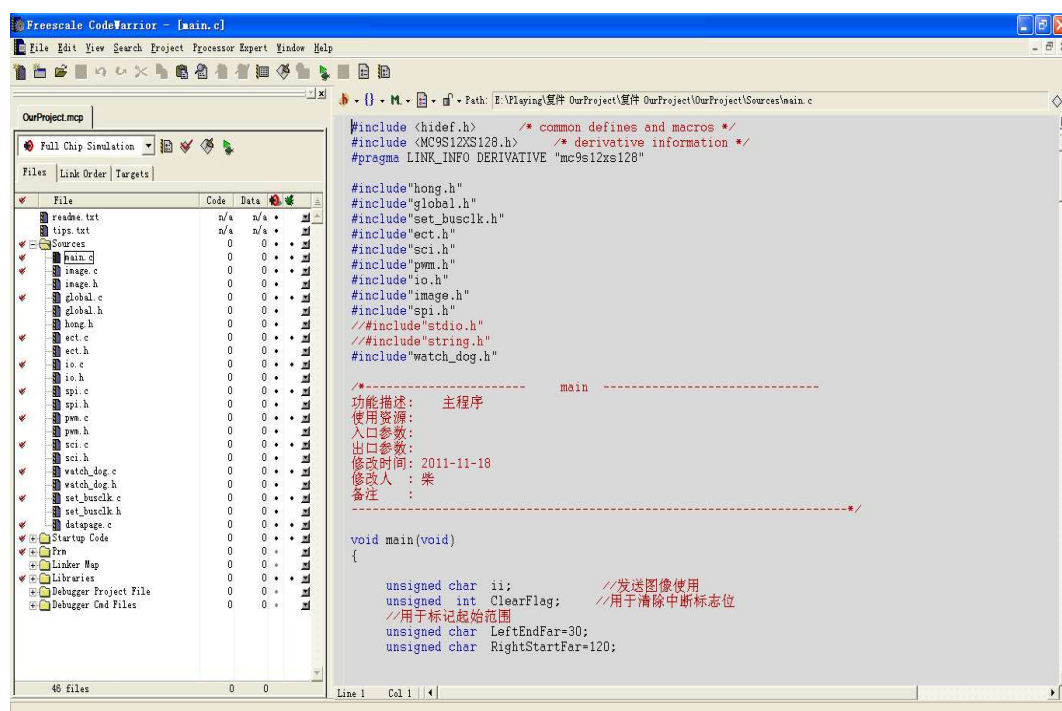


图 6.1.1 Codewarrior 开发界面

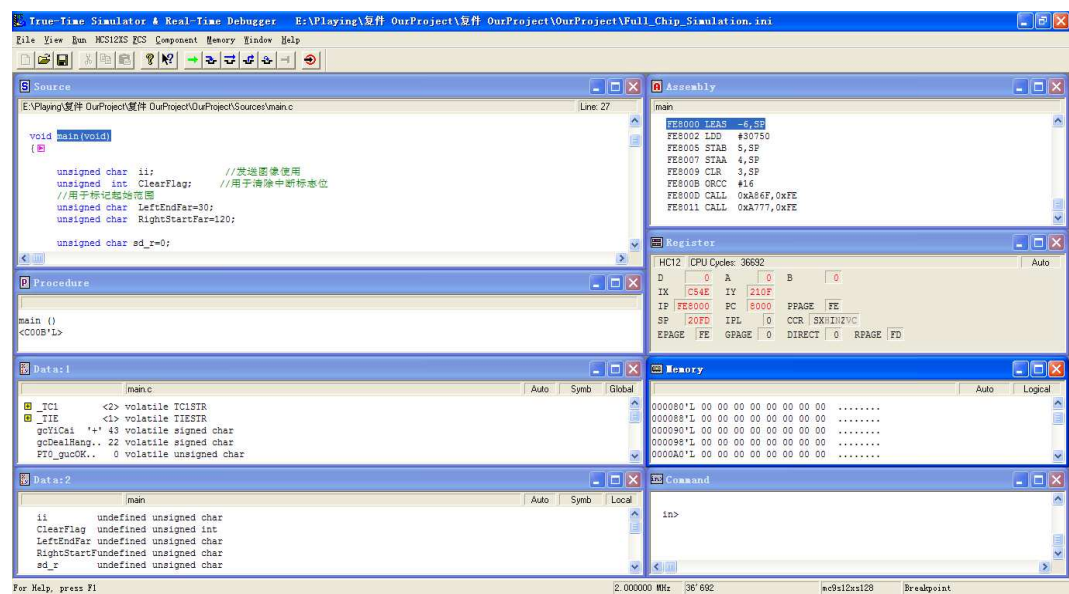


图 6.1.2 Codewarrior 调试界面

## 6.2 智能车上位机设计

### 6.2.1 图像上位机

由于采集的摄像头图像难于观察，于是本设计采用了专门针对摄像头调试的上位机，其将需要观察的数据，集成到了上位机上面，从而大大提高了调试的效率。其整体设计界面如图 5.2.1。

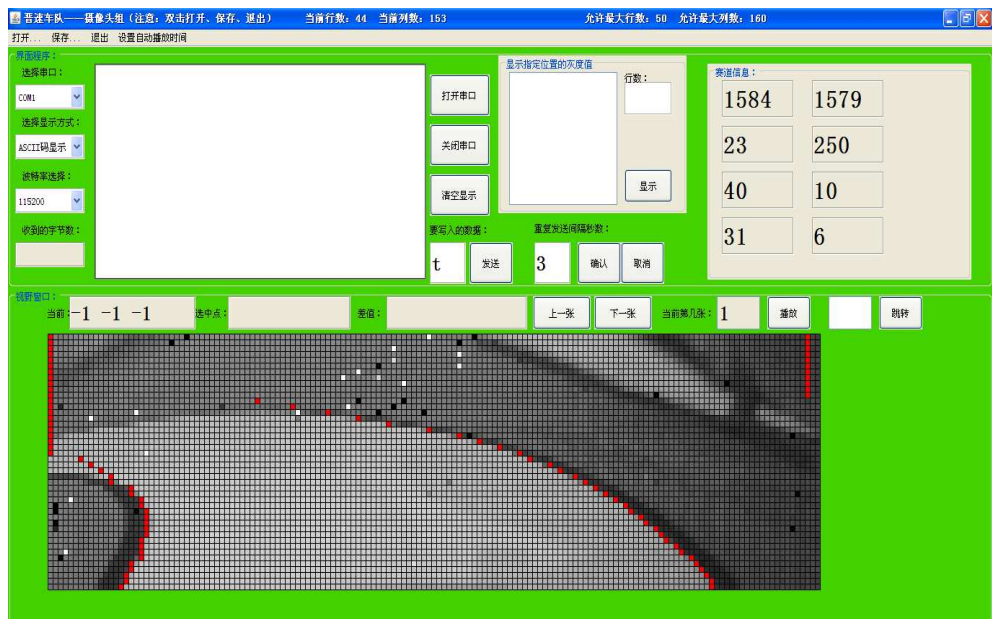


图 6.1.3 图像上位机

### 6.2.2 PID 参数整定上位机

PID 是速度调节中最为关键的部分，好的智能车能够实现快速的加减速，给定速度实现有较好的跟踪能力。为了更好的整定 PID 参数，本设计使用了绘制给定脉冲返回脉冲的上位机如图：



图 6.1.4 PID 整定上位机

## 第七章 模型车技术参数统计

表 7.1 模型车技术参数统计：

| 项目                | 参数                    |
|-------------------|-----------------------|
| 路径检测方法（赛题组）       | 摄像头组                  |
| 车模几何尺寸（长、宽、高）（毫米） | 245*162*280           |
| 车模轴距/轮距（毫米）       | 115                   |
| 车模平均电流（匀速行驶）（毫安）  | 800                   |
| 电路电容总量（微法）        | 1550                  |
| 传感器种类及个数          | CMOS 摄像头 1 个, 编码器 1 个 |
| 新增加伺服电机个数         | 0                     |
| 赛道信息检测空间精度（毫米）    | 16.1                  |
| 赛道信息检测频率（次/秒）     | 62.5HZ                |
| 车模重量（带有电池）（千克）    | 0.96                  |



## 第八章 总结

通过将近半年的努力，我们从零起点开始，完成了智能车的设计，实现了车子在较高速下的行驶。在这个过程中，我们不断的发现问题，解决问题。提高了我们解决问题，面对压力的能力，是我们能够更有效的调试智能车。

机械方面：在低速的情况下，机械方面的影响很小，但随着速度的提高，机械方面的影响不断扩大。如前轮的不正，会导致严重的抖动。前轮中心值出现偏差后，导致快速情况下车的甩尾。良好的机械性能，能够有效的提高智能车的转向能力和稳定性。

电路方面：电路的正常稳定，是整个智能车工作的基础。不稳定的电路，会导致智能车工作能力部分丧失，甚至完全丧失。当舵机电源供电偏低，直接导致舵机转向力不够，舵机延迟加重。PA口的不稳定，直接导致摄像头读取回来的灰度值错误。智能车无法正确行使。

软件方面：软件部分，是整个智能车的灵魂。当其他方面能够稳定工作的时候。软件算法，直接影响着车速。良好的算法，能够使智能车选择更好的路径。能够实现快速的加速减速。软件调试，是调试部分中最重要的部分。

到现在，我们的车子速度还不是很高，我们还发现了许多很多可以改进的地方。我们的参数、我们的速度控制方案并不够合理。还有我们的机械性能有待提高。虽然这些事情，都很微小，但是却成为制约我们车子车速的关键因素。相信在解决这些问题过程中，我们的车速还能有进一步的提高。

## 参考文献

- [1] 第六届“飞思卡尔”杯全国大学生智能车竞赛比赛规则. [Online], available: <http://www.smartcar.org.cn/web/index.asp>. 2008
- [2] MC9S12XS128 datasheet, 2010
- [3] 卓晴, 黄开胜, 邵贝贝 等编著. 学做智能车[M]. 北京: 北京航空航天大学出版社, 2001. 30
- [4] 夏路易, 石宗义. 电路原理图与电路板设计教程Protel 99SE[M]. 北京: 北京希望电子出版社, 2002
- [6] 阎石. 数字电子电路基础[M]. 北京: 高等教育出版社, 2005
- [7] 孙同景, 陈贵友. Freescale 9S12 十六位单片机原理及嵌入式开发技术[M]. 北京: 机械工业出版社, 2008
- [8] 朱广, 周程亮. CCD 图像传感器在路径识别系统中的应用[J]. 工业控制计算机. 2009. (3): 50-51
- [9] 吴斌华, 黄卫华, 程磊, 等. 基于黑白摄像头的智能小车路径识别系统设计[J]. 电子技术应用. 2007. (3): 80-82
- [10] 潘新民, 王燕芳. 微型计算机控制技术[M]. 北京: 高等教育出版社, 2001
- [11] 贾翔宇, 季庆庸, 丁芳. 前馈-改进PID 算法在智能车控制上的应用[J]. 计算机与信息技术, 2008, (12): 28-30
- [12] 阮毅, 陈伯时. 电力拖动自动控制系统[M]. 北京: 机械工业出版社, 2010

## 附录

## 附录 A 源程序

```
/*----- setting_PLL -----*/
函数名称: 系统时钟设置函数
功能描述: 对系统时钟进行设置
入口参数: 无
出口参数: 无
使用资源:
备 注: 芯片上用的晶振是 16M
-----*/
void Setting_PLL( void )
{
    CLKSEL = 0x00;
    PLLCTL_PLLON = 1;
    SYN_R_VCOFRQ = 2;
    SYN_R_SYNDIV=8;
    //SYN_R_SYNDIV = 9;
    REFDV_REFFRQ = 1;
    REFDV_REFDIV = 1;

    POSTDIV = 0;
    _asm(nop);
    _asm(nop);
    while (!(CRGFLG_LOCK==1)); // 等待稳定
    CLKSEL|=0x80;
}
/*----- setting_ECT -----*/
/*函数名称: 输入捕捉和脉冲捕捉 */
/*功能描述: */
/*入口参数: 无 */
/*出口参数: 无 */
/*使用资源: */
/*备 注: */
-----*/
void Setting_ECT(void)
{
    TIOS_IOS0=0; //用作输入捕捉
    TIOS_IOS1=0; //用作输入捕捉
```

```
TIOS_IOS2=0;
TIOS_IOS3=1;
TIOS_IOS4=1;
TIOS_IOS5=1;
```

```
TCTL4_EDG0A=1;
TCTL4_EDG0B=0;
TCTL4_EDG1A=0;
TCTL4_EDG1B=1;
TCTL4_EDG2A=0;
TCTL4_EDG2B=0;
```

```
TCTL2_OM3=0;
TCTL2_OL3=0;
TCTL1_OM4=0;
TCTL1_OL4=0;
TCTL1_OM5=0;
TCTL1_OL5=0;
TSCR1=0x90;
TSCR2=0x07;
```

```
TIE_C0I=0;
TIE_C1I=0;
TIE_C2I=0;
TIE_C3I=0;
TIE_C4I=0;
TIE_C5I=0;
```

```
//初始化脉冲累加器
PACTL=0x50;    //PT7 上升沿脉冲计数
PACNT =0x00;    //累加器置初值
```

```
    }
/*----- Sci0_init -----
功能描述:
入口参数: 无
出口参数: 无
使用资源:
备    注:
-----*/
```

```
void Sci0_init(void)
{
    unsigned char t;
    SCI0SR2_AMAP=0;

    SCI0BD=0x01d4;          //设置波特率为 9600

    SCI0CR1=0x00;
    t = SCI0DRL;
    t = SCI0SR1;
    SCI0CR2=0x0c;
}

/*----- Sci0_Tx1 -----
函数功能:
入口参数:
出口参数:
使用资源:
备    注:
-----*/
void Sci0_Tx1(unsigned char text)    //SCI0
{
    while (!(SCI0SR1&0x80));
    SCI0DRL=text;
}

/*----- Sci0_Txn -----
函数名   :
功能描述:
使用资源:
入口参数:
出口参数:
修改时间:
修改人   :
备注     :
-----*/
void Sci0_Txn(unsigned int n,unsigned char ch[])
{
    unsigned int i;
    unsigned char *p;
```

```
p=ch;
for (i=0; i<n; i++)
{
    Sci0_Tx1( *(p++) );
}
}

void main(void)
{

    unsigned char  ii;           //发送图像使用
    unsigned  int  ClearFlag;    //用于清除中断标志位
    //用于标记起始范围
    unsigned char  LeftEndFar=30;
    unsigned char  RightStartFar=120;

    unsigned char sd_r=0;

    DisableInterrupts;

    Setting_PLL( ); //锁相环的初始化
    Io_init( );
    //Sci0_init( );    //串口 0 的初始化
    Sci1_init( );
    Setting_ECT( ); //定时计数器的初始
    Pwm_init ( );    //PWM 初始化
    //Dog_init();

    //-----
    gvReadKey();    //读取方案

    /****SD 卡初始 ****
    #ifdef  SD_debug    //SD 卡的开始 其中包括复位和初始化
        sd_r=SD_MMC_SPI_Start();
        if(sd_r==1)while(1);
    #endif
    /****sd 结束****

    ClearFlag=TC1;    //清楚场标志位
    Enable_PT1_chang; //开启场中断
    Enable_PT4;
```

```
EnableInterrupts;    //开启中中断

while(1)
{
    if(gcYiCai<gcDealHang)
    {
        if(gcDealHang>FEN_DUAN)
        {
            gvTB_AllLine(gcDealHang);
            gcDealHang--;

        }
        else
        {
            gvTB_LeftNear(gcDealHang);
            gvTB_RightNear(gcDealHang);
            gcDealHang--;
        }
    }

    if(gcDealHang== -1 && PT0_gucOK==1) //如果图像采集完成
    {
        // Feed_dog();//喂狗
        //-----控制程序-----
        //提取黑线内边沿
        gvFindLeft();
        gvFindRight();
        gvCkeckCross();
        //电机赛道控制
        gvMotor_type();

        // Feed_dog();//喂狗

        gvSteerControl();
        gvMotorControl();
        if(gucDelayTime>2 ) { gvCkeck_StartLine();} //检测起跑线

        PT0_gucOK=0; //保证只每次控制一次
    }
}
```

```
        ClearFlag=TC1;//清楚场中断标志
        Enable_PT1_chang;//使能场中断

    }//if(PT0_gucOK==1)结束
} //while(1)语句结束
}
/*-----SteerControl()-----
功能描述: 舵机控制程序
入口参数:
出口参数:
使用资源:
备    注:
-----*/
void gvSteerControl(void)
{
    unsigned char  ii=0;//用于循环
    int  temp=0;

    static  float   sfSteer_err=0;//上上次偏差
    static  float   sfSteer_pre=0;//上次偏差

    const float fKd=0;//PD 运算中的 D 参数

    float  fKa=gstKey.fKa;
    unsigned char  ucKb=0;
    float  fQuanLow=0;

    unsigned char ucFar=0;
    unsigned char ucRef=gstKey.ucRef;
    unsigned char ucLow=gstKey.ucLow;
    unsigned char ucEnd=gstKey.ucEnd;

    float fSlope=0;

    float fEndOffset=0;

    //正式处理
    switch(gucSteer_DaoType)
    {
```



```

//---长道修正判断-----
case LEFT
{
    ucKb=gstKey.ucLKb;
    fQuanLow=gstKey.fQuanLow;
    /

    //计算斜率
    fSlope=1.0 * ucFar/17 * (gstJinSu.zhengti.left[ucLow] -
gstJinSu.zhengti.left[ucFar] - ucDaoLeft_a[ucLow] + ucDaoLeft_a[ucFar])/(ucFar-ucLow);

    //计算底部偏差
    fEndOffset=1.1*(ucDaoLeft_a[ucEnd] -
gstJinSu.zhengti.left[ucEnd]);

    sfSteer_err=fKa * fEndOffset + ucKb * 30.0/ucFar*fSlope;
    sfSteer_err=fKa * fEndOffset + ucKb * fSlope;

    giSteer_ValeIcrease = (sfSteer_err + fKd*(sfSteer_err -
sfSteer_pre));

    giSteer_ValeNow = (int)giSteer_ValeIcrease + D_MID_LIMT;
    break;
}
case RIGHT:
{
    ucKb=gstKey.ucLKb;
    fQuanLow=gstKey.fQuanLow;

    fSlope=1.0 * ucFar/17 * (gstJinSu.zhengti.right[ucLow] -
gstJinSu.zhengti.right[ucFar] -ucDaoRight_a[ucLow] + ucDaoRight_a[ucFar])/(ucFar - ucLow);

    fEndOffset=1.1*(ucDaoRight_a[ucEnd] -
gstJinSu.zhengti.right[ucEnd]);
    sfSteer_err=fKa * fEndOffset + ucKb * fSlope;

    giSteer_ValeIcrease = (sfSteer_err + fKd*(sfSteer_err -
sfSteer_pre));

```

```
        giSteer_ValeNow = (int)giSteer_ValeIncrease + D_MID_LIMT;
        break;
    }

    ///---错误-----
    case TYPE_ERROR:          //如果舵机值大于了一定值就打死 否者就
保持
    {
        if(giSteer_ValeLast>D_MID_LIMT+Steer_ChangeMax)
        {
            giSteer_ValeNow=D_LEFT_LIMT;
        }
        else if(giSteer_ValeLast<D_MID_LIMT - Steer_ChangeMax)
        {
            giSteer_ValeNow=D_RIGHT_LIMT;
        }
        else
            giSteer_ValeNow=giSteer_ValeLast;

        break;
    }
    ///---其他-----
    default:                  //如果舵机值大于了一定值就打死 否者就保
持
    {
        if(giSteer_ValeLast>D_MID_LIMT+Steer_ChangeMax)
        {
            giSteer_ValeNow=D_LEFT_LIMT;
        }
        else if(giSteer_ValeLast<D_MID_LIMT - Steer_ChangeMax)
        {
            giSteer_ValeNow=D_RIGHT_LIMT;
        }
        else
            giSteer_ValeNow=giSteer_ValeLast;

        break;
    }
} //swtich()结束
```

```

//限幅值 防止超界
if(giSteer_ValeNow<D_RIGHT_LIMT) giSteer_ValeNow=D_RIGHT_LIMT;
else if(giSteer_ValeNow>D_LEFT_LIMT) giSteer_ValeNow=D_LEFT_LIMT;

sfSteer_pre=sfSteer_err; //保存误差

giSteer_ValeLast=giSteer_ValeNow; //保存舵机值
PWMDTY23=giSteer_ValeNow; //舵机赋值
}
/*-----gvPID_Control(int set_speed)-----
功能描述: 进行不同方案设置
入口参数:
出口参数:
使用资源:
备 注:
-----*/

void svPID_Control(int speed_set)
{
    const unsigned char PID_yuzhi=8; //小于八位为 pid 大于为匀速
    const unsigned char Dead_Line=1;
    const unsigned char Dead_Change=10;

    int err_fab=0;
    float speed_increase=0;
    /*******PID 参数*****
    const float Kp=20.0;
    const float Ki=3.3;
    const float Kd=2.0;
    /*******
    if(gucMotor_key<PID_yuzhi) // 匀速
    {
        giSpeed_now=speed_set;
    }

    else if(gucDelayTime>1)//pid 控制
    {
        //PID 控制
        giSpeed_err_now=speed_set-giSpeed_back;

```

```
speed_increase=Kp*(giSpeed_err_now-giSpeed_err_last)+Ki*(giSpeed_err_now)+Kd*(giSpeed_err_now-2.0*giSpeed_err_last+giSpeed_err_pre) ;//PID 公式
giSpeed_now=giSpeed_last+(int)speed_increase;

giSpeed_err_pre=giSpeed_err_last;
giSpeed_err_last=giSpeed_err_now;
//防止越界
if(giSpeed_now>1000)    giSpeed_now=1000;
else if(giSpeed_now<0)  giSpeed_now=0;

// giSpeed_back=0;  //清楚 PID
}
else giSpeed_now=500;

}
```