

# 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：西南交通大学（峨眉校区）

队伍名称：**157-亡灵战车**

参赛队员：李旺竹

邓顺云

张  苏

带队教师：李  勇

## 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： 李旺竹  
邓怀云  
张苏  
带队教师签名： 黎松奇  
日 期： 2016.08.14

# 目录

摘要.....	V
第一章 引言.....	1
第二章 智能车系统总体设计.....	3
2.1 系统概述.....	3
2.2 整车布局.....	3
第三章 机械结构设计.....	4
3.1 舵机的安装.....	4
3.2 前轮机械结构的调整.....	4
3.2.1 车轴的调节.....	5
3.2.2 前轮的调节.....	5
3.3 摄像头的安装.....	7
3.4 编码器的安装.....	8
3.5 齿轮的调整.....	9
第四章 硬件系统设计.....	10
4.1 单片机最小系统.....	10
4.2 电源管理模块.....	10
4.3 电机驱动模块.....	13
4.4 图像采集模块.....	13
4.5 速度检测模块.....	14
第五章 软件系统设计.....	16
5.1 总体控制流程图.....	16
5.2 赛道中心线的提取及优化处理方法.....	16
5.2.1 普通赛道的处理方法.....	16

5.2.2 特殊赛道的处理方法.....	18
5.3 PID 控制算法介绍.....	21
5.3.1 位置式 PID.....	22
5.3.2 增量式 PID.....	22
5.4 舵机 PID 控制算法.....	23
5.5 速度 PID 控制算法.....	24
5.6 PID 参数整定.....	24
第六章 系统开发及调试工具.....	25
6.1 开发工具.....	25
6.2 上位机.....	26
6.3 按键调试模块与 OLED 液晶模块.....	26
第七章 模型车的主要技术参数.....	27
总 结.....	I
致 谢.....	II
参 考 文 献.....	III
附 录.....	IV

## 摘要

本文介绍了我队智能车的整体设计方案，该智能车属于摄像头组别，采用 B 型车模。本次设计的智能车系统以 MK60 为微控制器为核心控制单元，通过 CMOS 摄像头检测赛道信息，提取黑色边界线，用于赛道识别，从而使智能车在跑道黑线能以内用最快的速度正确行驶。该车通过用增量式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。同时为了提高模型车速度和稳定性，我队采用了自己设计的上位机从而为模型车参数的调节提供了方便。在比赛准备过程中通过分析模型车在不同算法下的实际情况，选择出实际效果最好的算法，并不断进行优化。

**关键词：**B 型摄像头智能、 MK60、 CMOS 摄像头、 PID 控制算法

## 第一章 引言

智能车竞赛最早始于韩国，2000 年韩国汉阳大学承办了第一届智能车竞赛，并由飞思卡尔公司(现在的恩智浦公司)赞助，全国每年大约有 100 余支大学生队伍参加比赛。智能车的设计制作，其专业知识设计控制、模式识别、传感技术、汽车电子、电器、计算机和机械等多个学科，对学生的知识融合和实践动手能力的培养，以及高等学校控制和汽车电子学科学术水平的提高，具有良好的产期推动作用。

教育部为了加强大学生实践、创新能力和团队精神的培养，在已举办全国大学生数学建模、电子设计、机械设计和结构设计四大竞赛的基础上，决定委托教育部高等学校自动化专业教学指导分委员会主办每年一度的全国大学生智能汽车竞赛。

在智能车的开发过程中，各参赛队伍需要自己改装竞赛车模，完成小车的设计与搭建。在此过程中需要学习嵌入式系统开发环境与在线编程方法等软件编程方式，同时还要学习画图软件的使用以及各个组成电路硬件方面的知识。飞思卡尔智能车竞赛到目前为止已经成功组织了十届，赛制、规则、组别不断进行改进，增加了比赛的观赏性、技术性和实用性。

智能车的开发与设计涉及到多个专业领域，这对于大学生综合素质的培养，知识面的拓展和分析问题解决问题的能力提高很有意义，并且有利于提高大学生的动手能力和创新能力。此外，制作这样一个高性能智能小车的过程，也是一个组员之间相互配合的过程，在这个过程中可提高参赛队员的交流合作能力。

## 第二章 智能车系统总体设计

### 2.1 系统概述

智能车系统整体结构如下所示：模型车设计制作的主要思路以及实现的技术方案说明。

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，输出到 MK60FX512 微控制器，进一步处理获得主要的赛道信息；通过 512 线编码器来检测车速，并采用 MK60FX512 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；驱动电机采用双闭环控制，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。

### 2.2 整车布局

为满足智能车系统在赛道上尽可能的快速稳定运行，智能车的整体布局是非常重要的。有效的算法需要搭配良好的机械结构特性才能有很好的效果。结合 B 车模的自身特性，我们的主要思路是：

- 1) 减轻车模重量，降低重心；
- 2) 车模重心靠近车模中心稍靠后的位置；
- 3) 电路板按功能模块化处理，主电路板在功能完备前提下精简；
- 4) 在降低重心的前提下尽量提高摄像头的高度。



图 2.1 整车布局图



## 第三章 机械结构设计

机械结构是智能小车的一个重要的基础，实际操作中发现是在速度提高到一个程度之后，其决定了小车性能的极限。因此通过学习对汽车有关的理论知识和转向结构的分析，找到了影响小车转向范围和效率的因素并进行了改进，使得小车的转向灵敏度得到提高，同时舵机效率也得到较大提高。

### 3.1 舵机的安装

舵机的安装方式有两种，一是卧式安装，二是直立式安装。卧式安装是车模默认安装方式，其优点是响应速度快、易于安放，可降低小车的重心，但是造成左右两边杆的长度不一样，这样会使舵机对长连杆轮子用的力要大些，导致舵机对左右两边转向响应时间不一样。而直立式安装把舵机架高了一些，增加了力臂，使得小车反应更加灵活，但与此同时也增大了阻力，减小了力距。因此，根据舵机性能和实际情况我们选择了直立式安装方式。



图 3.3.1 舵机的安装方式

## 3.2 前轮机械结构的调整

前轮对于整个车来说是十分重要的地方，直接决定了小车的方向控制能力。由于车模本身的机械结构存在一些不足，因此我们自己在车模上进行了一些调整。

### 3.2.1 车轴的调节

在研究车体模型时，我们发现：若车轴不完全固定，会有一定程度的间隙，导致前轮在自然的状态下会发生晃动，弯道转弯时发生转弯不对称等现象。在车轴的内测固定端加装垫片可以有效减轻这一误差的影响，但不能完全消除这一缺陷。

### 3.2.2 前轮的调节

对于汽车而言，要保持车辆直线行驶的稳定性的，使之转弯回正、转向轻便，必须确定车轮定位参数，包括主销后倾、主销内倾、前轮外倾和前轮前束。

#### 1) 主销后倾：

主销后倾角是指在纵向平面内主销轴线与地面垂直线之间的夹角。它在车辆转弯时会产生与车轮偏转方向相反的回正力矩，使车轮自动恢复到原来的中间位置上。所以，主销后倾角越大，车速越高，前轮自动回正的能力就越强，但是过大的回正力矩会使车辆转向沉重。通常主销后倾角值设定在  $1^{\circ}$  到  $3^{\circ}$ ，可通过垫片调整。



图 3.2.2.1 主销后倾角

#### 2) 主销内倾：

主销内倾角是指在横向平面内主销轴线与地面垂直线之间的夹角，它的作用也是使前轮自动回正。角度越大前轮自动回正的作用就越强，但转向时也就越费力轮胎磨损增大；反之，角度越小前轮自动回正的作用就越弱。通常汽车的主销内倾角不大于  $8^\circ$ 。

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。

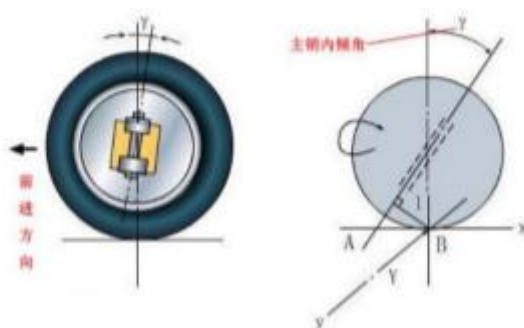


图 3.2.2.2 主销后倾角

### 3) 前轮前束:

俯视车轮，汽车的两个前轮旋转平面不平行，而是稍微带些角度，这种现象叫做前轮前束。车轮前束的作用是为了减轻或者抵消因前轮外倾角所造成的不良后果，二者相互协调，保证前轮在汽车行驶过程中滚动而无滑动。前轮前束一般为 0-12mm。

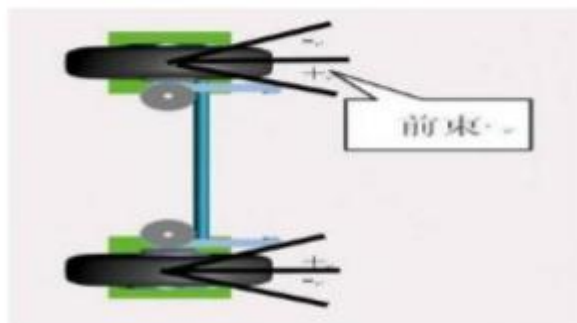


图 3.2.2.3 前轮前束

### 4) 前轮外倾

在汽车的横向平面内，前轮中心平面向外倾斜一个角度，称为前轮外倾角。前轮外倾角一方面可以使车轮接近垂直路面滚动而滑动减小转向阻力，使小车转向轻便；另一方面减少了轴承及其锁紧螺母的载荷，增加了使用寿命，提高了安全性。一般前轮外倾角为  $1^\circ$  左右，但对于有高速、急转向要求的车辆，前轮外倾角可减小甚至为负值。

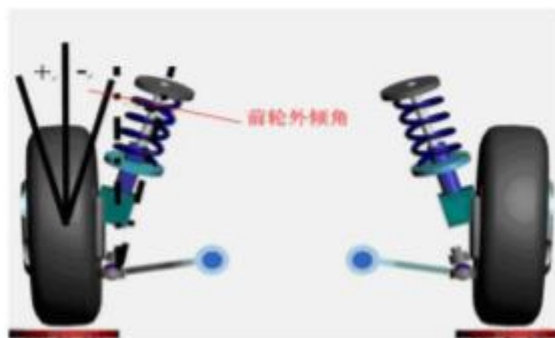


图 3.2.2.4 前轮外倾角

小结：对于 B 型车模而言，前轮参数的调节是通过前后两个连杆来调节，前面可以调节前轮的内倾或者外倾，而后面的连杆则是控制前轮的前束。当调节合适后可以固定，以免反复调节造成滑动。

### 3.3 摄像头的安装

摄像头一般采用碳素竿作为支柱，顶端固定传感器，尾端固定于车模底盘上，为了支柱稳固牢靠我们选择了不锈钢的金属底座。同时为减小抖动带来的数据采集误差，纵向扩大摄像头的视野范围，即减小摄像头对车体前面的盲区并尽量获得更大的前瞻，我们将摄像头置于中间。另外将摄像头置于舵机后方也是可以的。

摄像头的安装方式：



图 3-3 摄像头的安装方式

摄像头底座:



图 3-3 摄像头底座

### 3.4 编码器的安装

为了获取可靠的速度返回值，要求编码器安装必须稳固，因此，我们购买了合适的编码器支架，可以使编码器齿轮与电机齿轮很好的啮合。



图 3.4.1 编码器的安装方式

### 3.5 齿轮的调整

调整齿轮的时候，可以给车一个比较一般的速度，然后通过听齿轮转动的声音在微小的调节距离来使齿轮间的距离达到比较合适的位置。以不松动，无卡滞为佳。另外还要保证齿轮间咬合有足够的接触面积，以平行最佳。

## 第四章 硬件系统设计

在智能车系统中，硬件系统要以有效、稳定作为设计原则。对智能车来说，一个稳定可靠的硬件系统是顺利完成比赛的基础。为此，我们小组成员花费大量时间用于设计与改进硬件系统，先后制作了 2 次电路板，电路模块分为主板模块和驱动模块。最终，我们确定了一套稳定可靠的硬件系统。

### 4.1 单片机最小系统

我们采用了 MK60FX512VLQ15 型号的单片机。核心板主要用来处理传感器采集到的数据，同时输出控制信号，根据摄像头采集到的图像，二值化后的数据，求得黑线的实际位置，并根据 PID 算法对舵机和电机进行控制，同时它还控制速度参数的设定等工作，因此核心板是整个系统的控制中心。车模使用的最小系统板作为我们自行设计制作的最小系统板。为提高车模配件的适应性，我们此次制作的最小系统板与 MK60 最小系统板完全兼容，并能随时替换，大大提高了车模的可维修性。

### 4.2 电源管理模块

比赛规定：全部硬件电路的电源由 7.2v、2A/h、的可充电镍镉电池提供。但是在电路设计中各个模块所需要的额定电压与电流不一样，因此需要选择合适的稳压电路，将充电电池电压转换成各个模块所需要的电压。主电路板包括主电源开关、各级电压稳压电路、舵机控制电路、摄像头信息采集、液晶屏控制电路等电路。电源管理模块如下图所示：

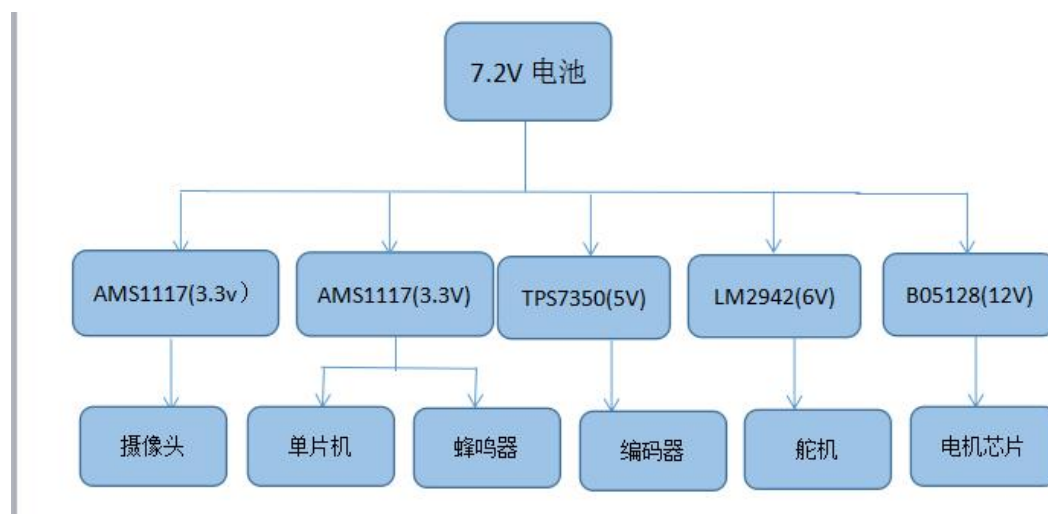


图 4.2.1 电源管理模块

为了避免各个模块之间的供电相互干扰，摄像头用一个 **AMS117** 单独供电。在电路设计中，考虑到电机驱动所引起的电源不稳定型（主要为瞬态脉冲），我们在电源输入端，各芯片电源引脚都加入了滤波电路和稳压电容。并且为了避免由于驱动电机转动时所引起的电磁干扰，在电路板设计中，在印制板上做了敷铜处理，将电路中的“地”与敷铜面相连接。

### 1) 5V 稳压电路

通过与其他几种相同功能的稳压芯片的对比，经综合考虑，我们选择了工作稳定纹波小的 **TPS7350**。

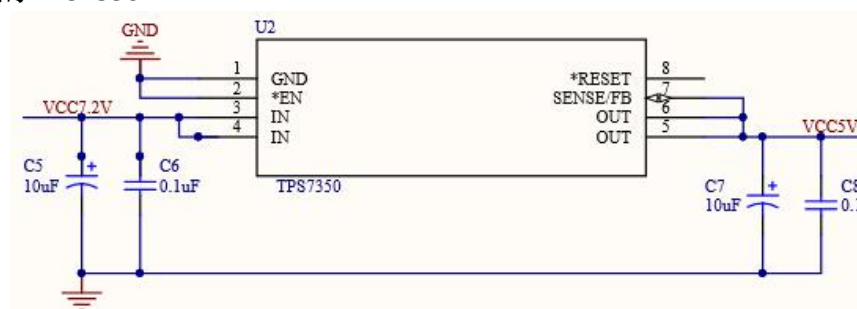


图 5.2.1 5V 稳压模块

### 2) 3.3V 稳压电路

**AMS1117** 型芯片价格便宜，在工作范围内工作稳定，为满足稳定的需要，我们选择了用一个芯片为摄像头进行单独供电。



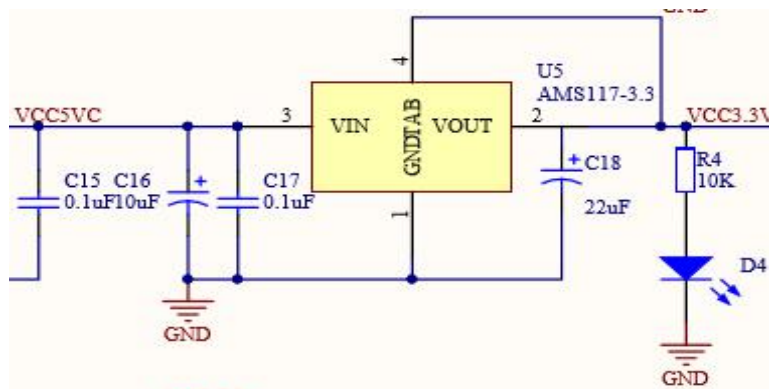


图 5.2.2 3.3V 稳压电路

## 3) 6V 稳压电路

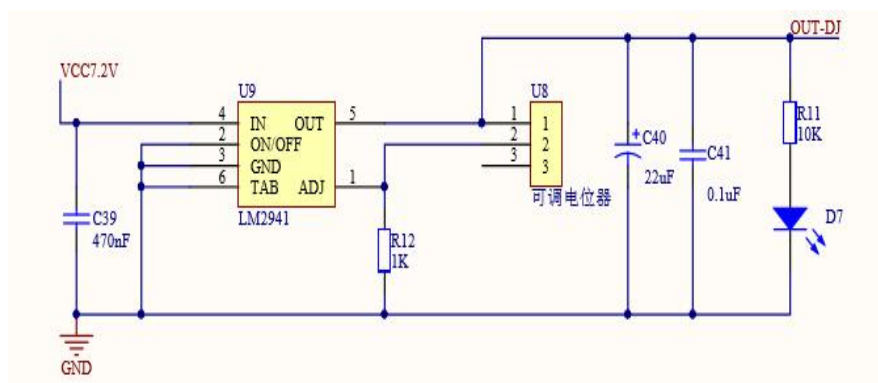


图 5.2.3 6V 稳压电路

## 4) 12V 稳压电路

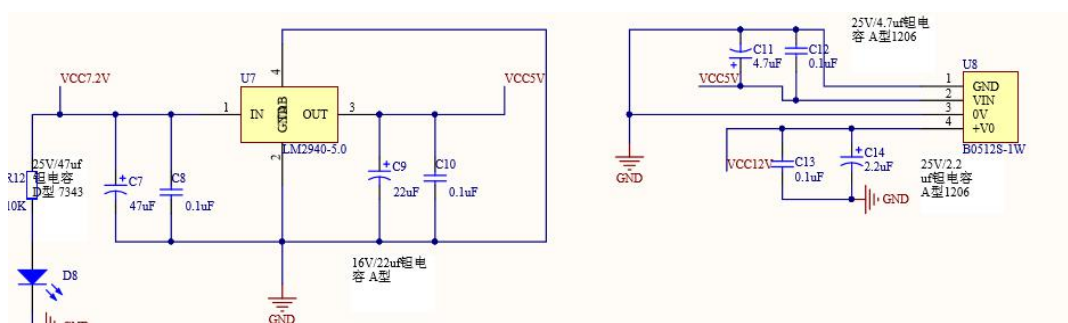


图 5.2.4 12V 稳压电路

### 4.3 电机驱动模块

今年摄像头模块采用的是 B 型的车模，该车模电机功率比较大，在实际工作中转速快，加减速明显，而采用传统的驱动芯片搭成的全桥驱动发热很严重。因此我们自行设计了电机驱动电路。电机驱动电路为一个分立元件制作的直流电动机可逆驱动电路。我们采用了 HIP4082 驱动芯片，由两个半桥电路构成 h 全桥驱动电路，从而驱动电机正反转，选用的是 N 沟道 MOS 管。这款芯片比较适合电机驱动，有过压过流过热的保护而且负载电流大的优点。驱动电路图如下图所示：

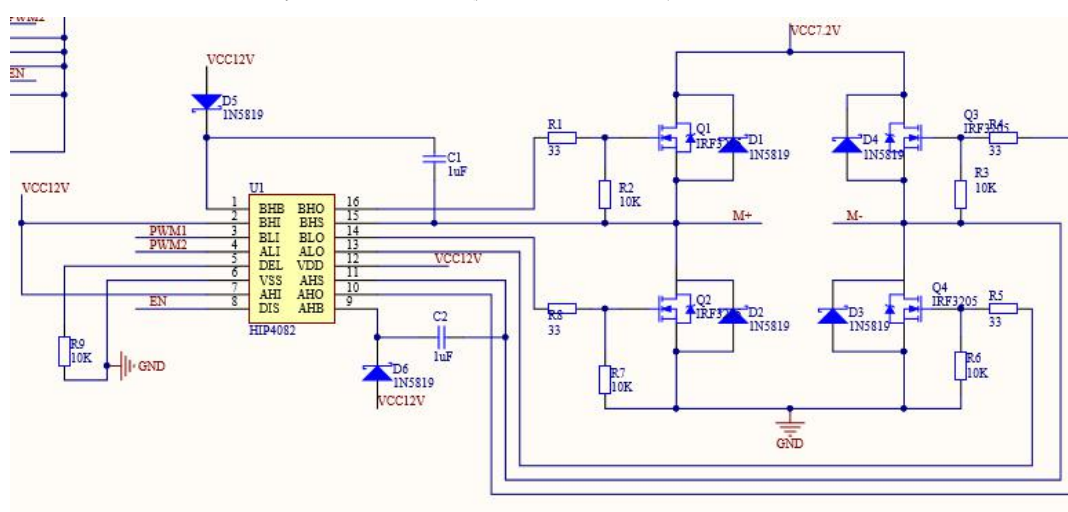


图 4.3.1 电机驱动电路

### 4.4 图像采集模块

CCD 摄像头具有对比度高、动态特性好的优点，在 12V 电压下工作，对于整个系统来说过于耗电。经过实验表明 CCD 对光线的要求较高，不能满足小车的适应性的要求，所以不选择 CCD 摄像头。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。因为小车队分辨率要求并不高，COMS 摄像头足够满足小车清晰度的要求，所以选用 COMS 摄像头。对于摄像头的选择，我们主要考虑它的视野大小和动态特性。

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 这几种。模拟摄像头主要有 OV5116, BF3003, MT9V136

这几种模型。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。

通过查阅资料我们发现 OV7725 和 PC1030N 型号的摄像头，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 OV7620 得到的信息类似。经对比，OV7725 和 PC1030N 型号的这两种摄像头图像效果更好，动态性能也更好。所以根据实际情况我们选用山外出厂的鹰眼 OV7725，自带硬件二值化电路，这样讲硬件二值化融合到摄像头上，使数据输出更加稳定，更有利于我们的算法处理。

#### 4.5 速度检测模块

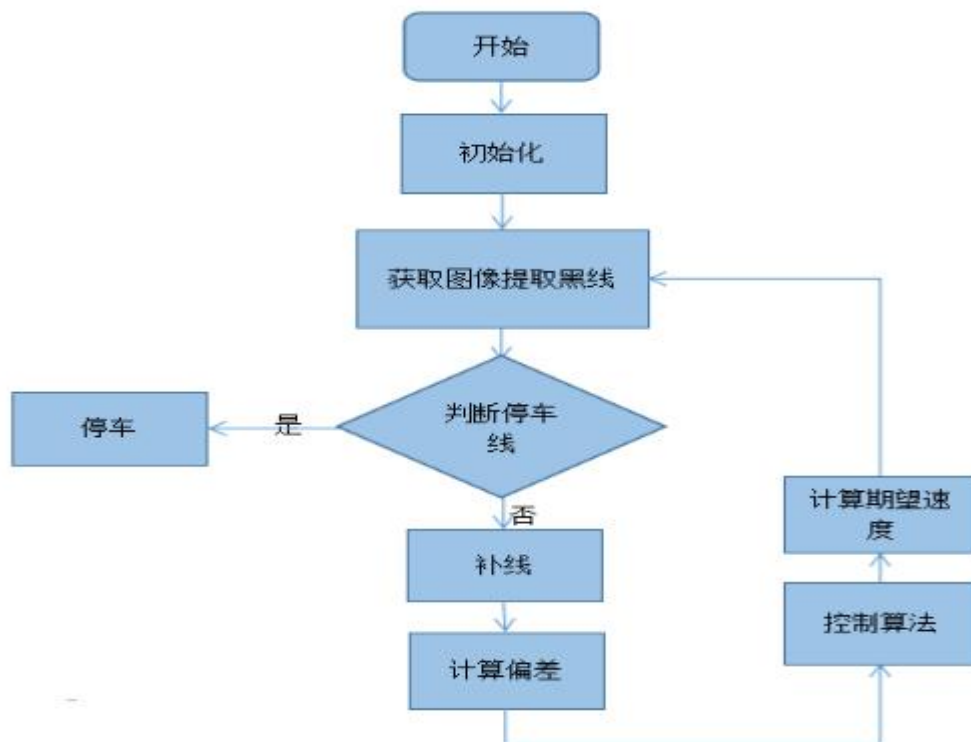
为了闭环控制，我们在车模上附加了编码器。经过机械和电路性能的考虑和挑选，我们选用龙邱 512 线的 3 相增量式编码器，这个编码器具有体积小，质量轻，稳定性高的特点，输出型号类型为 CMOS 信号，抗干扰能力强，不需上拉电阻直接和单片机连接，供电电压 3.3V 到 5V 可选，但价格稍微贵一点。编码器可输出编码器的运转方向，即不需要正交解码也可以完成方向识别，同时该编码器也可使用正交解码程序。如下图所示。



图 4.5.1 编码器实物图

## 第五章 软件系统设计

### 5.1 总体控制流程图



### 5.2 赛道中心线的提取及优化处理方法

#### 5.2.1 普通赛道的处理方法

基于行扫描的寻线方法：由于本次比赛赛道元素比较简单，考虑到摄像头采集的图像在远处不稳定。所以本次图像的处理基于行扫描。

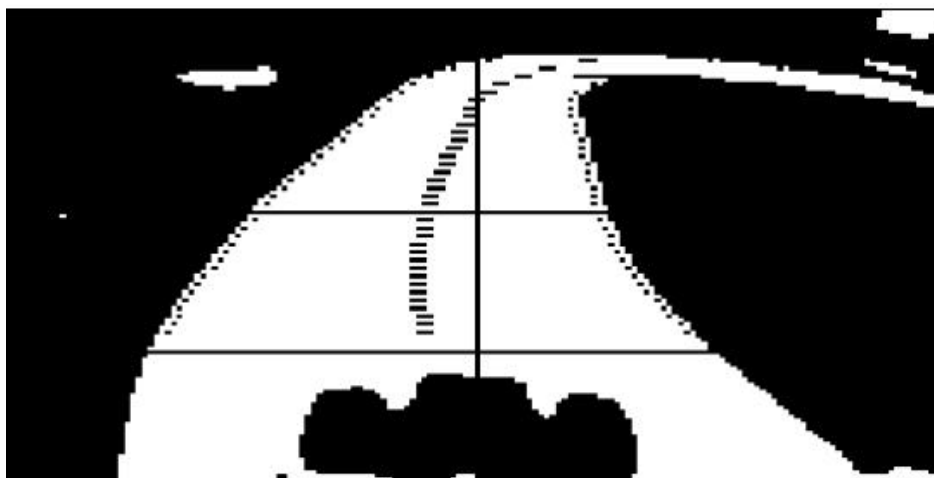


图 5.2.1 行扫描图

我们先从图的第一行也就是数组的最后一行开始(我们假定车是在赛道中间), 所以我们从图的中间向两边搜索寻找边界 `left` and `right`. 便可以确定第一行的 `mid`. 因为赛道是连续变化的, 所以不会出现跳变. 所以我们可以将本次确定的 `mid` 作为下一次搜索的基准。(经测试, 只要车是放在赛道上的第一行就能正确算出 `mid`)

算法流程:

1. 初始化 `list_left[]`(左边线列表), `list_right[]`(右边线列表)和 `list_mid[]`(中线列表);
2. 扫描第一行(图像近端)获取 `list_left[1]`, `list_right[1]`, `list_mid[1]`;
3. for line 2 to LINEMAX
4. BEGIN
5. if ( `IMG[line][ list_mid[line - 1] ] == black` )     BREAK;
6. 从 `list_mid[line-1]`处向两侧扫描得到 `list_left[line]`, `list_right[line]`,  
`list_mid[line]`;
7. END

最终我们可以得到三个表 `list_left[]`, `list_right[]`和 `list_mid[]`;

为了减少处理时间采用了隔行扫描。

### 5.2.2 特殊赛道的处理方法

#### 1) 十字

应对十字的策略是补线, 这里我们只对入十字时进行补线, 在十字前把车身摆正即可顺利通过十字。补线的关键是寻找拐点, 也就是斜率突变点。采用相邻点的偏差变化便能较好的判断出拐点。



图 5.2.2.1 十字补线

#### 2) 障碍

通过障碍本身的偏差来打角, 在速度较快的时候反应不过来。所以要对障碍特殊处理。判断障碍思路是识别出直道(因为障碍在直道上)和黑块。关于黑块的识别, 有黑块存在时, 有一边的边线发生了两次突变, 基于这点便可判断黑块。为了使小车看到障碍时提前打角, 便对中线进行了补线, 达到了预期的效果。

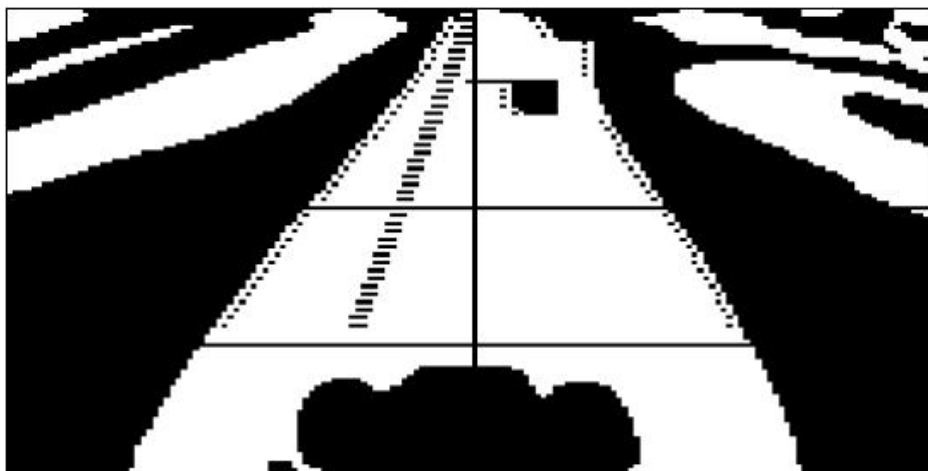


图 5.2.2.2 障碍处理图像

### 3) 停车线

纵向扫描便能较为准确的判断出停车线。

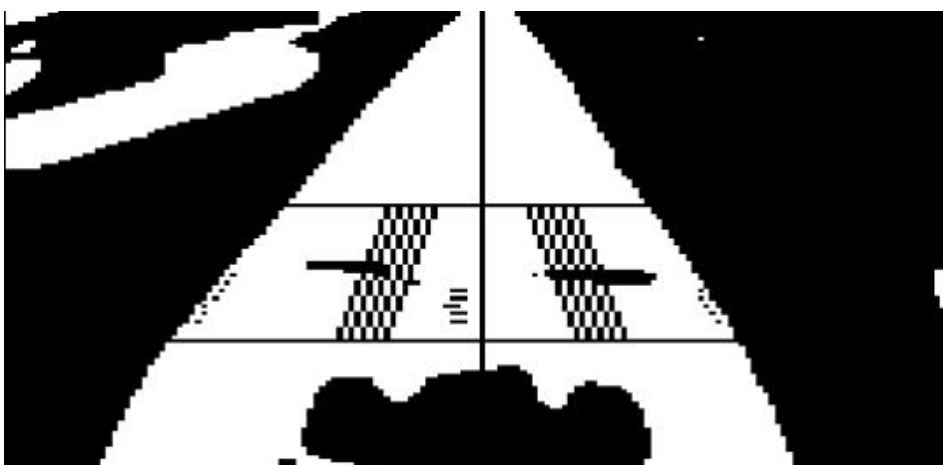


图 5.2.2.3 起跑线扫描图像

### 4) 浅尝透视变换与路劲优化

通过透视变换, 可以消除图像上的畸变, 还原图像, 从人的视角来处理。通过几个点的定标即可计算出变换参数。

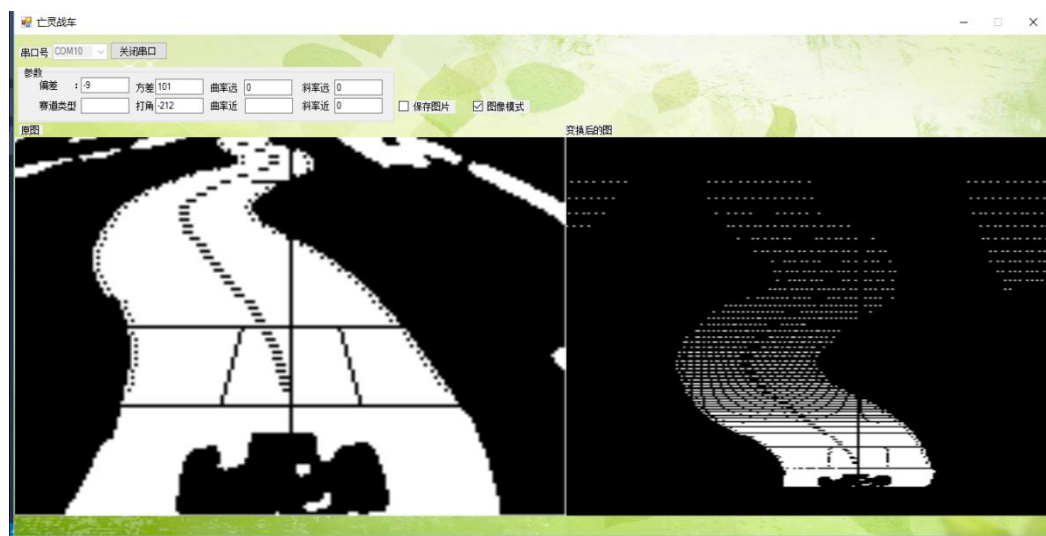


图 5.2.2.4 透视变换的图像

透视变换的意义在于路劲的优化, 参考 K1999 算法, K1999 是通过曲率滤波来进行路劲的优化, 可是对于单片机, 曲率的计算无疑是硬伤。所以采用线性滤波近似优化。

但考虑到本次比赛有路肩, 路径优化的意义并不大。但是也值得去研究研究。



图 5.2.2.5 路径优化图像



## 5.3 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 5.3.1 所示。

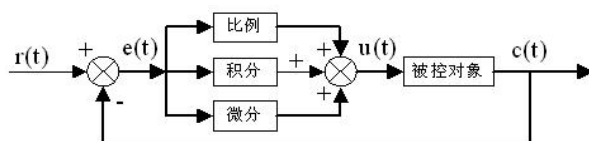


图 5.3.1 PID 控制器原理框图

在计算机控制系统中，使用的是数字 PID 控制器，控制规律为：

$$e(k) = r(k) - c(k) \quad (\text{公式 5.1})$$

$$u(k) = K_p \left\{ e(k) + \frac{T}{T_i} \sum_{j=0}^k e(j) + \frac{T_d}{T} [e(k) - e(k-1)] \right\} \quad (\text{公式 5.2})$$

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；

$K_p$ ——比例系数；

$T_i$ ——积分时间常数；

$T_d$ ——微分时间常数；

$T$ ——采样周期。

简单说来，PID 控制器各校正控制方法的作用如下：

**比例控制：**比例控制是一种最简单的控制方式。其控制器的输出与输入误差信号成比例关系。比例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，但当仅有比例控制时系统输出存在稳态误差（Steady-state error）。

**积分控制：**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，时间越大，积分作用越弱，反之则越强。即就算误差很小，积分项也会随着时间的增加而加大，它推动控制器的输出增大使稳态误差进一步减小，直到等于零。

**微分控制：**能够预测误差变化的趋势，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。

### 5.3.1 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(5.2)为位置式 PID 控制算法。

位置式 PID 控制算法的特点是：简单直接，但由于是全量输出，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。

### 5.3.2 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(5.2)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(5.3)可得式(5.4)。 $u(k-1) = K_p \{e(k-1) + \frac{T}{T_i} \sum_{j=0}^{k-1} e(j) + \frac{T_D}{T} [e(k-1) - e(k-2)]\}$

(公式 5.3)

$$\begin{aligned} \Delta u(k) &= K_p \{[e(k) - e(k-1)] + \frac{T}{T_i} e(k) + \frac{T_D}{T} [e(k) - 2e(k-1) + e(k-2)]\} \\ &= K_p \Delta e(k) + K_I e(k) + K_D [\Delta e(k) - \Delta e(k-1)] \end{aligned} \quad \text{(公式 5.4)}$$

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(5.4)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  $K_P$ 、 $T_I$ 、 $T_D$ ,只要使用前后三次测量值的偏差,即可由式(5.4)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,所以有误差动作时影响小,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。

## 5.4 舵机 PID 控制算法

在舵机控制算法方面,增量式和位置式我们都采用过,经过实际测试,后来选择了增量式来闭环控制舵机的转向。舵机采用增量式 PD 控制算法,将每场图像所选控制行的偏差平均值作为自变量,构建一次函数,我们没有采用动态 PD 算法,而是均使用定值,采用最保守的策略,确保稳定性。另外因为 D 项会影响车子打角的预判性,和偏差的变化率联系在一起,在 P 项的调整达到一定的程度后, D 项的调整也会影响车子的稳定性和路径,需要细细调试。这样的算法可以适应大部分赛道,但大小 S 弯并不能最小抖动的通过,为了应对这两种赛道,加入了偏差加权平均算法,单独处理,基本可以保证直线通过,最大程度减少耗时。

## 5.5 速度 PID 控制算法

速度控制我们采用了双闭环的控制方法，即当期望转速与当前转速相差很大的时候，转速调节达到饱和，在电流调节器的作用下电动机的电流达到一个定值，驱动力矩也为一个定值。当期望转速与当前转速相差较小时，转速调节器发挥作用，系统在转速调节器与电流调节器的共同作用下达到稳定值。控制器为转速、电流双闭环控制器，转速环为外环，电流环为内环。转速环输出值经过一个限幅环节后，作为参考电流值输入电流调节器，电流调节器的输出即为加在电动机两端的控制电压。

使用双闭环后能更加快速准确的控制电机。软件上为两个 PID 串联调节，电流环为内环，转速环为外环。电流环 P 控制，速度环 PD 控制。

## 5.6 PID 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，经过理论计算来确定参数；二是工程整定法，它主要依赖工程经验，常用的方法有试凑法，临界比例度法等等。

对于智能车的 PID 整定，我们采用工程整定法，因为智能车是一个动态系统，没有一个确定的数学模型。具体的整定步骤为，先调节 P 直至微微超调，再调节 I，最后调节 D，加快系统响应。

## 第六章 系统开发及调试工具

### 6.1 开发工具

程序开发工具采用的是 IAR Embedded Workbench IDE。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。它能够为单片机提供与之配套的应用程序开发模块比较方便。相比较其他的 ARM 开发环境，IAR EWARM 具有入门容易、使用方便和代码紧凑等优点。

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。



图 6.1.1 使用 IAR Embedded Workbench IDE 开发智能车

## 6.2 上位机

为了方便我们看摄像头采集的图像，以及一些参数的设定，我们使用 C# 自行计了上位机。

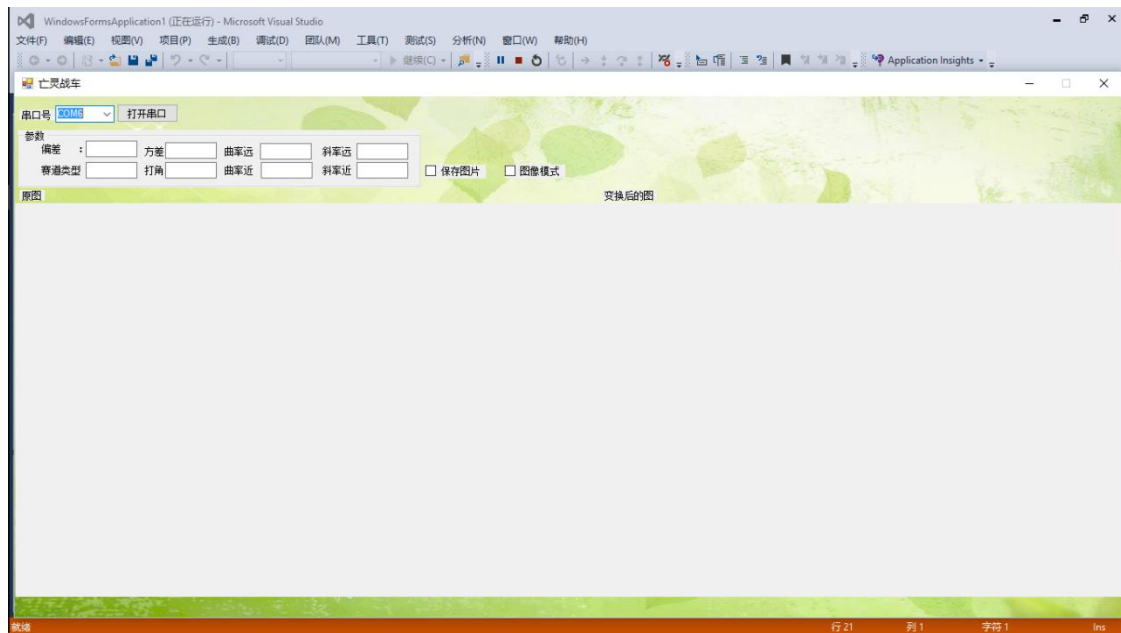


图 6.2.1 上位机

## 6.3 按键调试模块与 OLED 液晶模块

当需要调试参数时，如果一直在程序里面改动，下载需要一些时间。而通过按键液晶显示可以对一些可变参数进行设定，这可以节约调试时间，尤其是在赛前调试参数极其方便。

## 第七章 模型车的主要技术参数

车模主要参数名称	参数
重量	1400g
长度	290mm
宽度	180mm
高度	32mm
车模轴距	100mm
除了车模原有的驱动电机、舵机之外伺服电机个数	0 个
电路电容总量	1156.4uF
赛道信息检测频率	66.67(次/秒)

## 总结

通过将近半年的智能车制作，我们学到了许多。在机械结构上尝试了大量方案，进行了各种改装，最后确定了小车的整体布局。硬件制作上自行设计了，主板和电机驱动。软件上，在图像处理方面考虑到整体的稳定性，采用了基于行扫描的扫线方式；舵机控制尝试过直接计算角度来打角，最终还是采用 PD 来控制；电机控制最初采用转速环，但 PID 参数一直没整定好，后面驱动上加入电流采样模块对电机进行了电流转速双闭环，得到了较好的效果。对于小车来说，任何环节都相当的重要。

在做车的过程中，有过兴奋，有过迷茫。当小车在赛道上飞奔时，我们的内心是愉悦的；当程序上，机械上，硬件上遇到问题时我们也会苦恼。经常和队友熬夜奋战，焊板子，装车，闭环.....印象深刻的是校赛前，和队友熬夜焊板子搭车，新板子又出现了一些不能解释的问题，最终还是解决了这些问题。在智能车比赛中收获的不仅是小车，更多的是那份浓浓的友情和情怀。我们觉得小车就是自己的孩子，小车的每一次成长让我们感到开心。我们感谢恩智浦公司提供了如此一个好 的平台来让我们展现自我，小车在赛道上的驰骋，无疑也是我们那颗年轻的心的驰骋！

因为时间仓促，水平有限，有很多疏漏或者错误，也很遗憾没有更加 细致的写出整个设计方案，衷心希望老师对我们的工作提出宝贵的意见，给我 们指点，让我们在今后的工作学习中获取更多的进步。



## 致 谢

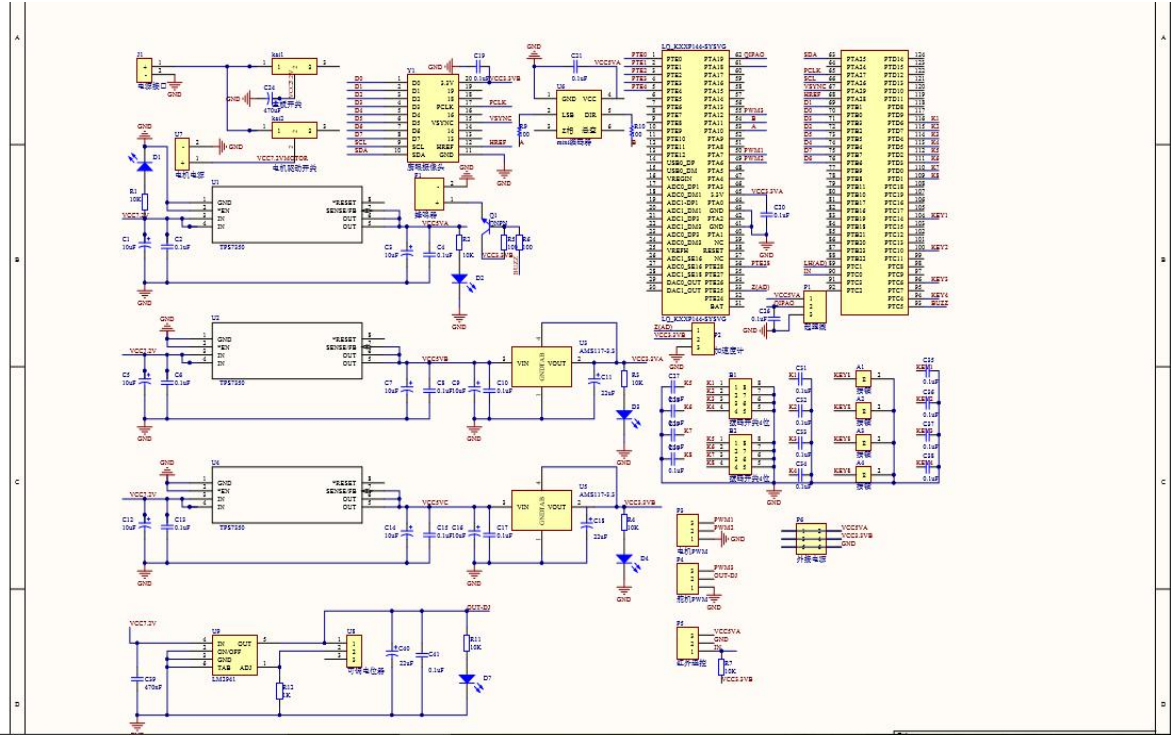
最后，借书写技术报告的机会向帮助过我们的老师、同学，组委会的老师，还有队内的兄弟姐妹，表示深深地谢意！向校内的车队——157 车队致谢！向没有机会出现在队员名单中的幕后英雄致敬！

## 参 考 文 献

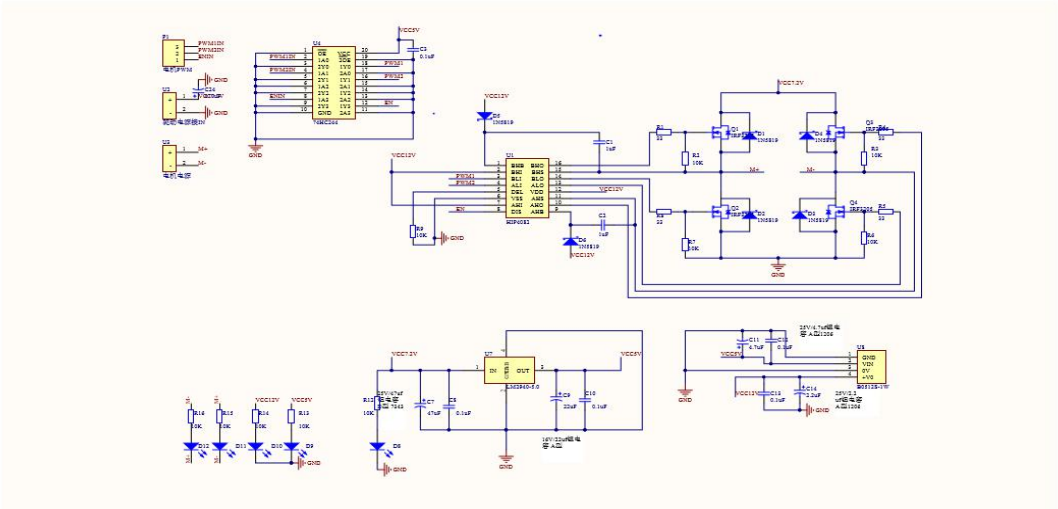
- [1] 蔡述庭 “飞思卡尔”杯智能车竞赛设计与实践—基于 S12XS 和 Kinetis K1 北京航空航天大学出版社, 2012
- [2] 卓晴, 黄开胜, 邵贝贝. 学做智能车[M]. 北京北航出版社, 2007. 3
- [3] 阎石. 数字电子技术基础[M]. 北京: 高等教育出版社, 1998.
- [4] 雷霏霖, 梁志毅. 基于 CMOS 传感器 OV7620 采集系统设计[J]. 电子测量技术, 2008
- [5] Stephen Prata. C primer plus.人民邮电出版社; 2005 年 2 月 1 日
- [6] Mark Allen Weiss. 数据结构与算法分析:C 语言描述. 机械工业出版社; 2004 年 1 月 1 日
- [7] 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.
- [8] Charles L.Phillips,John M.Parr. Feedback Control Systems.科学出版社, 2012.
- [9] 张文春. 汽车理论[M]. 北京. 机械工业出版社. 2005
- [10] 侯虹. 采用模糊 PID 控制律的舵机系统设计[J]. 航空兵器, 2006, 2(1):7-9.
- [11] 孙浩, 程磊, 黄卫华, 等. 基于 HCS12 的小车智能控制系统设计[J]. 单片机与嵌入式系统应用, 2007, 03(16):51-54.
- [12] 闫裴 黄志辉 罗峰 基于双闭环 PID 的电动智能车防滑控制研究 2009 中国汽车工程学会年会论文集

附录：

附录 A：主控板电路图



附录 B：驱动板电路图



## 附录 C: 程序源代码

```
/*  
    基本的寻线  
*/  
/*  ch : 0 动态中点; ch : 1 静态中点 */  
int8 static Get_Point(int8 ch)  
{  
    int16 mid;  
    int8 first = 0;  
    int8 lost  = 0;  
    int8 num   = 40;  
    int8 line  = First_Line;  
    int8 step  = Search_Step;  
    while ( num-- )  
    {  
        if ( first == 0 )  
        {  
            if ( img[line*CAMERA_W + CAMERA_W/2] == 1 )  
            {  
                mid = search_double(line, CAMERA_W/2);  
                first = 1;  
            }  
        }  
        else  
        {  
            if ( img[line*CAMERA_W + CAMERA_W/4] == 1 )
```

---

```
        {
            mid = search_double(line, CAMERA_W/4);
            first = 1;
        }
    else if ( img[line*CAMERA_W + CAMERA_W*3/4] == 1 )
    {
        mid = search_double(line, CAMERA_W*3/4);
        first = 1;
    }
    else
        lost++;
    }
}
else
{
    if ( ch == 1 )
        mid = search_double(line, CAMERA_W/2);
    else
        mid = search_double(line, mid);
    if ( mid == -1 )
    {
        return 1;
    }
}
valid_line = line;
line -= step;
```

```
        if ( line <= 0 )return  1;

    }

    if ( lost >= 1 )    return  0;

    else                return  1;

}

int16 static search_double(int line, int begin)
{
    uint8 left, right, flag_l, flag_r;

    uint8 easy_line;

    left = right = begin;

    flag_l = flag_r = 0;

    if ( img[line*CAMERA_W + begin] == 0 )

        return -1;

    // 找出左右边界

    while ( !flag_r || !flag_l )

    {

        if ( left == 1 )

            flag_l = 1;

        if ( right == CAMERA_W - 2 )

            flag_r = 1;

        if ( !flag_l )

        {
```

---

```
        if ( img[line*CAMERA_W + left] == 1 && img[line*CAMERA_W
+ left - 1] == 0 && img[line*CAMERA_W + left - 2] == 0 )
            left--, flag_l = 1;
        else
            left--;
    }
    if ( !flag_r )
    {
        if ( img[line*CAMERA_W + right] == 1 && img[line*CAMERA_W
+ right + 1] == 0 && img[line*CAMERA_W + right + 2] == 0 )
            right++, flag_r = 1;
        else
            right++;
    }
}

easy_line = ( First_Line - line ) / Search_Step;

if ( right - left <= 5 )    return -1;

if ( left == 1 )
{
    if ( first_left == 0 )
        first_left = easy_line;
    lost_left++;
}
```

```
if ( right == CAMERA_W - 2 )
{
    if ( first_right == 0 )
        first_right = easy_line;
    lost_right++;
}
```

```
// 找到左右边界, 更新列表
list_right[easy_line] = right;
list_left [easy_line] = left ;
list_width[easy_line] = right - left;
```

//寻找突变点 C 弯时出现 ( 记录第一次发生突变的行数 ): 特点 有效行必然小于直道

```
if ( break_left == 0 && list_left[easy_line - 1] - list_left[easy_line] >= 5 )
{
    break_left = easy_line;
}
if ( break_right == 0 && list_right[easy_line] - list_right[easy_line - 1] >= 5 )
{
    break_right = easy_line;
}
```

```
// 宽度过于大就认为两边丢先了( 具体测测 )
if ( right - left >= CAMERA_W - 15 )
    cnt_allwhite++;
```



```

        return (right + left)/2;
    }

    /*
    十字处理
    */

    Is_Inflection_Left();
    Is_Inflection_Right();

    if ( Inflection_Left_Point && Inflection_Right_Point && Inflection_Left_Point
    <= 35 && Inflection_Right_Point <= 35 && (Inflection_Left_Point >= 4 &&
    Inflection_Right_Point >= 4) && ABS(Inflection_Left_Point-Inflection_Right_Point) <
    20 )
    {
        flag_cross = 1;

        GetSlope(MAX(1, Inflection_Left_Point-12), Inflection_Left_Point-2,
list_left, &line_k, &line_b);

        for ( i = Inflection_Left_Point - 2; i <= 39; i++ )
        {
            list_left[i] = MAX((int16)(i*line_k + line_b), 0);
        }

        GetSlope(MAX(1, Inflection_Right_Point-12), Inflection_Right_Point-2,
list_right, &line_k, &line_b);

        for ( i = Inflection_Right_Point - 2; i <= 39; i++ )

```

```
        {
            list_right[i] = MIN((int16)(i*line_k + line_b), CAMERA_W - 2);
        }
    }
    else if ( Inflection_Left_Point && lost_right >= 5 && cnt_allwhite == 0 &&
Inflection_Left_Point <= 30 )
    {
        flag_cross = 1;
        GetSlope(MAX(1, Inflection_Left_Point-12), Inflection_Left_Point-1,
list_left, &line_k, &line_b);
        for ( i = Inflection_Left_Point - 2; i <= 39; i++ )
        {
            list_left[i] = MAX((int16)(i*line_k + line_b), 0);
        }
    }
    else if ( Inflection_Right_Point && lost_left >= 5 && cnt_allwhite == 0 &&
Inflection_Right_Point <= 30 )
    {
        flag_cross = 1;
        GetSlope(MAX(1, Inflection_Right_Point-12), Inflection_Right_Point-1,
list_right, &line_k, &line_b);
        for ( i = Inflection_Right_Point - 2; i <= 39; i++ )
        {
            list_right[i] = MIN((int16)(i*line_k + line_b), CAMERA_W - 2);
        }
    }
```

```
else if ( cnt_allwhite > 2 )
{
    flag_cross_in = 1;
    flag_cross = 1;
    Get_Point(1);
    img_line = (First_Line - valid_line) / Search_Step;
}

/*
    障碍处理
*/

void Solve_Block(void)
{
    int8 i;
    int8 flag_begin = 0;
    int8 cnt = 0;
    int8 line;
    int8 cnt_black = 0;
    int8 ss;
    Width_Hop_Begin = Width_Hop_End = 0;
    for ( i = 1; i <= 33; i++ )
    {
        if ( !flag_begin && list_width[i] - list_width[i+1] > 10 )
        {
            Width_Hop_Begin = i+1;
            flag_begin = 1;
            continue;
        }
    }
}
```

```
    }
    if ( flag_begin && ABS(list_width[i+1] - list_width[i]) < 5 )
    {
        cnt++;
    }
    else if ( flag_begin && list_width[i+1] - list_width[i] > 5 )
    {
        Width_Hop_End = i;
        break;
    }
}
if ( Width_Hop_Begin != 0 && Width_Hop_End != 0 && cnt >= 2 )
{
    line = First_Line - Search_Step*(Width_Hop_End + Width_Hop_Begin)/2;
    if ( list_mid[(Width_Hop_End + Width_Hop_Begin)/2] >
list_mid[Width_Hop_Begin-1] )
    {
        ss = list_left[(Width_Hop_End + Width_Hop_Begin)/2] - 1;
        while ( !img[line*CAMERA_W + ss] && cnt_black++ < 60 ) ss--;
        if (cnt_black >= 60 || ss < list_left[Width_Hop_Begin - 1])    return;
    }
    else
    {
        ss = list_right[(Width_Hop_End + Width_Hop_Begin)/2] + 1;
        while ( !img[line*CAMERA_W + ss] && cnt_black++ < 60 ) ss++;
        if (cnt_black >= 60 || ss > list_right[Width_Hop_Begin - 1])    return;
```

```

    }

    flag_block = 1;

    GetSlope(Width_Hop_Begin, Width_Hop_End, list_mid, &line_k,
&line_b);

    if ( line_b > 80 ) line_b = line_b+2;
    else                line_b = line_b-2;

    for ( i = 0; i <= img_line; i++ )
    {
        list_mid[i] = (uint8)(i*line_k + line_b);
    }
}

}

/*
    K1999 算法
*/

void Optimize_Route(int16 step)
{
    double C1, C2, C, DERTA;

    int16 new_y_up, new_y_down, new_x;

    K1999_point = 0; // 初始化

    // 第一步, 取点(梯度为 10)最多 20 个点

    for ( int i = Transform_start_line - 5; i > Transform_end_line + 5; i = i - step )
    {
        K1999_map[K1999_point++] = i;

        K1999_mid[i] = Transform_mid_list[i];
    }
}

```

```
}  
  
// 第二步, 分析  
  
int cnt_or = 10;  
while ( cnt_or-- )  
{  
    for ( int i = 2; i < K1999_point-2; i++ )  
    {  
        C1 = Get_C(i-1);  
        C2 = Get_C(i+1);  
        C  = Get_C(i);  
        new_x = K1999_map[i];  
  
        // 暴力寻找 i 点  
        // 路漫漫其修远兮, 吾将上下而求索 是左右而求索!! ^v^  
        new_y_up = new_y_down = K1999_mid[new_x];  
  
        DERTA = abs_double( C - (C1+C2)/2 );  
  
        int cnt = 5;  
        while ( cnt-- )  
        {  
            new_y_up++; //get y  
            C = Get_C_check(new_x, new_y_up, i);  
            if ( abs_double( C - (C1+C2)/2 ) < DERTA ) // updata  
            {
```

---

```
        K1999_mid[new_x] = new_y_up;
        DERTA = abs_double(C - (C1 + C2) / 2);
    }
    new_y_down--; //get y
    C = Get_C_check(new_x, new_y_down, i);
    if (abs_double(C - (C1 + C2) / 2) < DERTA) // updata
    {
        K1999_mid[new_x] = new_y_down;
        DERTA = abs_double(C - (C1 + C2) / 2);
    }
}
}
}
}
```