

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： 厦门大学
队伍名称： 至善队
参赛队员： 吴铭鸿、叶增软、陈求兴
带队教师： 黄靖凯 、丘腾海

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

本文以第七届全国大学生智能车竞赛为背景，介绍了智能赛车控制系统的软硬件结构和开发流程。该比赛采用大赛组委会统一指定的由东莞市博思电子数码科技有限公司提供的 A 型车模，以 Freescale 半导体公司生产以 Kinetis60 32 位微控制器作为核心控制单元，通过 CCD 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，提取赛道两边的黑色边缘的中心线作为引导线，用于赛道路径判断；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 LABVIEW、MATLAB 仿真平台、小型无线模块、液晶屏模块、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，该系统设计方案可行。

关键词：智能车系统， Kinetis60 32 位单片机，CCD 摄像头，控制策略

目录

第一章	引言	1
2.1	智能汽车研究背景	1
2.2	飞思卡尔智能汽车大赛简介	1
第二章	智能车整体设计	3
2.1	系统结构框图	3
2.2	车模整体布局	3
2.3	车模结构参数	4
第三章	机械结构的安装与调整	7
3.1	舵机的固定方式	7
3.2	底盘设计	8
3.3	摄像头的安装	8
3.4	编码器的安装	10
3.5	前轮定位	11
3.6	后轮调整	13
第四章	系统硬件的设计及实现	14
4.1	硬件设计方案	14
4.2	CCD 摄像头	15
4.3	单片机最小系统板	16
4.4	稳压电源模块	17
4.4.1	3.3V 、5V 稳压电路	17
4.4.2	6V 稳压电路	18
4.4.3	12V 升压电路	19
4.5	视频同步分离电路	19
4.6	电机驱动电路设计	21
4.7	硬件电路部分小结	22

第五章	系统软件的设计与实现.....	23
5.1	系统软件流程图.....	23
5.2	图像采集与处理.....	24
5.2.1	赛道中心提取.....	26
5.2.2	赛道类型判断.....	26
5.3	舵机转向和速度调节的 PID 控制算法.....	27
5.3.1	经典的 PID 控制算法.....	27
5.3.2	经典 PID 算法在本智能车上的应用.....	29
5.4	调试环境软件说明.....	30
5.4.1	开发工具.....	30
5.4.2	上位机调试软件的设计.....	31
5.4.3	无线调试模块设计.....	32
第六章	总结与参数统计.....	33
6.1	模型车的主要技术参数.....	33
6.2	结论.....	33
参考文献	34
附录：程序源代码	35

第一章 引言

2.1 智能汽车研究背景

智能小车系统是迷你版的智能汽车，二者在信息提取，信息处理，控制策略及系统搭建上有很多相似之处，可以说智能小车系统将为智能汽车提供很好的试验和技术平台，推动智能汽车的发展。摄像头作为很多智能汽车的信息输入的传感器具有其他很多传感器所不具有的优势，如对物体的特征提取，视野宽广等，因此基于摄像头的智能小车系统的研究将推动智能汽车的发展。

智能汽车是未来汽车的发展方向，将在减少交通事故、发展自动化技术、提高舒适性等许多方面发挥很重要的作用；同时智能汽车是一个集通信技术，计算机技术，自动控制，信息融合技术，传感器技术等于一身的行业，它的发展势必促进其他行业的发展，在一定程度上代表了一个国家在自动化智能方面的水平[1]。汽车在走过的 100 多年的历史中，从没停止过智能化的步伐，进入 20 世纪 90 年代以来，随着汽车市场竞争激烈程度的日益加剧和智能运输系统（ITS）地兴起，国际上对于智能汽车及其相关技术的研究成为热门，一大批有实力有远见的大公司、大学和研究机构开展了这方面的研究。很多美国、日本和欧洲等国家都十分重视并积极发展智能车系统，并进行了相关实验，取得了很多成就。我国的相关研究也已经开展，清华大学成立了国内最早的研究智能汽车和智能交通的汽车研究所，在汽车导航、主动避撞、车载微机等方面进行了广泛而深入的研究，2000 年上海智能交通系统进入实质性实施阶段，国防科大研制出第四代无人驾驶汽车，西北工业大学、吉林交通大学、重庆大学等也展开了相关研究[2]。这一新兴学科正在吸引越来越多的研究机构和学者投入其中。

2.2 飞思卡尔智能汽车大赛简介

为加强大学生实践、创新能力和团队精神的培养，促进高等教育教学改革，受教育部高等教育司委托(教高司函[2005]201 号文，附件 1)，由教育部高等自动化专业教学指导分委员会（以下简称自动化分教指委）主办全国大学生智能汽车竞赛。该竞赛以智能汽车为研究对象的创意性科技竞赛，是面向全国大学生的一种具有探索性工程实践活动，是教育部倡导的大学生科技竞赛之一。该

竞赛以“立足培养，重在参与，鼓励探索，追求卓越”为指导思想，旨在促进高等学校素质教育，培养大学生的综合知识运用能力、基本工程实践能力和创新意识，激发大学生从事科学研究与探索的兴趣和潜能，倡导理论联系实际、求真务实的学风和团队协作的人文精神，为优秀人才的脱颖而出创造条件。

该竞赛由竞赛秘书处为各参赛队提供/购置规定范围内的标准硬软件技术平台，竞赛过程包括理论设计、实际制作、整车调试、现场比赛等环节，要求学生组成团队，协同工作，初步体会一个工程性的研究开发项目从设计到实现的全过程。该竞赛融科学性、趣味性和观赏性为一体，是以迅猛发展、前景广阔的汽车电子为背景，涵盖自动控制、模式识别、传感技术、电子、电气、计算机、机械与汽车等多学科专业的创意性比赛。该竞赛规则透明，评价标准客观、坚持公开、公平、公正的原则，保证竞赛向健康、普及，持续的方向发展。

该竞赛以飞思卡尔半导体公司为协办方，得到了教育部相关领导、飞思卡尔公司领导与各高校师生的高度评价，已发展成全国 30 个省市自治区近 300 所高校广泛参与的全国大学生智能汽车竞赛。2008 年起被教育部批准列入国家教学质量与教学改革工程资助项目中科技人文竞赛之一（教高函[2007]30 号文）。全国大学生智能汽车竞赛原则上由全国有自动化专业的高等学校（包括港、澳地区的高校）参赛。竞赛首先在各个分赛区进行报名、预赛，各分赛区的优胜队将参加全国总决赛。每届比赛根据参赛队伍和队员情况，分别设立光电组、摄像头组、创意组等多个赛题组别。每个学校可以根据竞赛规则选报不同组别的参赛队伍。全国大学生智能汽车竞赛组织运行模式贯彻“政府倡导、专家主办、学生主体、社会参与”的 16 字方针，充分调动各方面参与的积极性。全国大学生智能汽车竞赛一般在每年的 10 月份公布次年竞赛的题目和组织方式，并开始接受报名，次年的 3 月份进行相关技术培训，7 月份进行分赛区竞赛，8 月份进行全国总决赛。

第七届飞思卡尔杯大学生智能车竞赛共有摄像头、光电、电磁三个组别，由三人一队参加比赛。参赛选手须使用竞赛秘书处统一指定的竞赛车模套件，采用飞思卡尔半导体公司的 8 位、16 位、32 位微控制器作为核心控制单元，自主构思控制方案进行系统设计，包括传感器信号采集处理、电机驱动、转向舵机控制以及控制算法软件开发等。制作一个能够自主识别路线的智能车，在专门设计的跑道上自动识别道路行驶，以用时的多少来决定比赛的成绩。

第二章 智能车整体设计

2.1 系统结构框图

本队所设计的智能车是以飞思卡尔公司提供的 32 位微处理器 Kinetis60 为核心控制单元，通过 CCD 摄像头获取赛道路径，并对赛道两边的黑线进行提取得出赛道的中心，根据自身小车的姿态与赛道中心之间的偏差产生控制量，控制舵机进行转向。通过编码器实时反馈电机转速，并对小车进行速度控制。同时，我们也设计了一些调试模块以便对小车进行更加方便的调试。系统框图如图 2.1 所示：

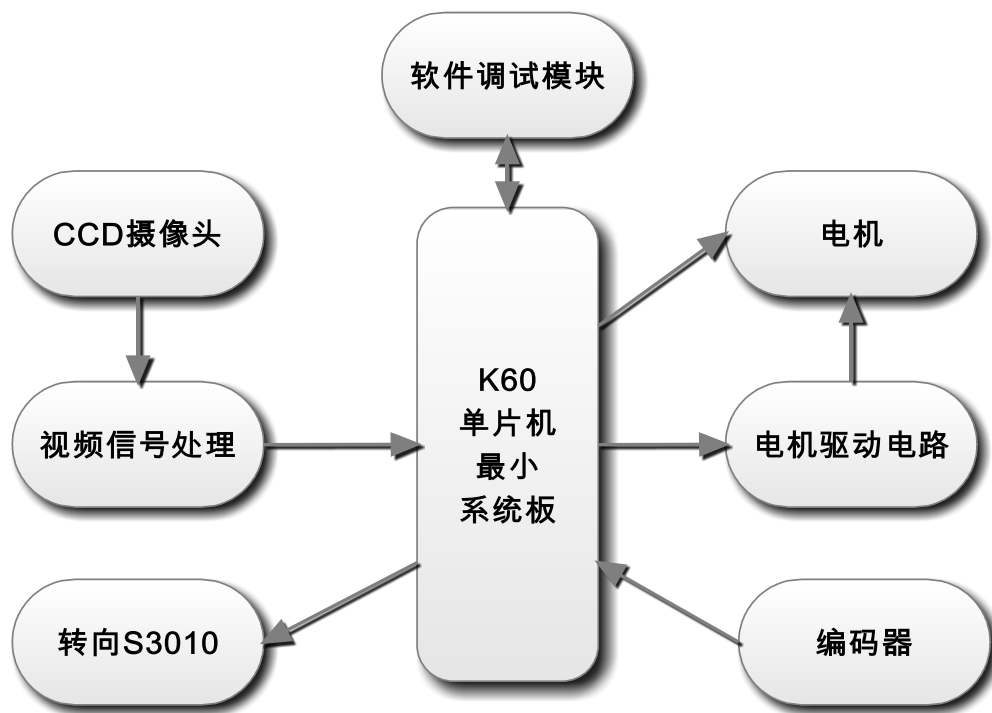


图 2.1 智能车系统框图

2.2 车模整体布局

飞思卡尔杯智能车在我国已经成功举办了六届，根据以往多届的经验，良好

稳定的图像信息是摄像头组智能车稳定行驶的基础，高效可靠的控制算法是智能车飞速行驶的重中之重，优秀的机械架构是智能车驰骋赛道的有力保障，优良的辅助调试系统是高效率调试的前提，而其它部分则相对次要。本次比赛摄像头组采用了前几届用到过的 A 车模，相比于上届的 C 车模，提速空间更大。在整个小车制作的过程中，我们注重对车模的机械改造及加工，力求智能车的重心较低，重量分布较为均匀，系统结构稳固，具有一定的防撞击防护；硬件电路方面，力求电路构成简单，稳定性高，传感器的安装可靠，外界对系统电路的干扰小。小车的整体布局如图 1-2-1 所示：

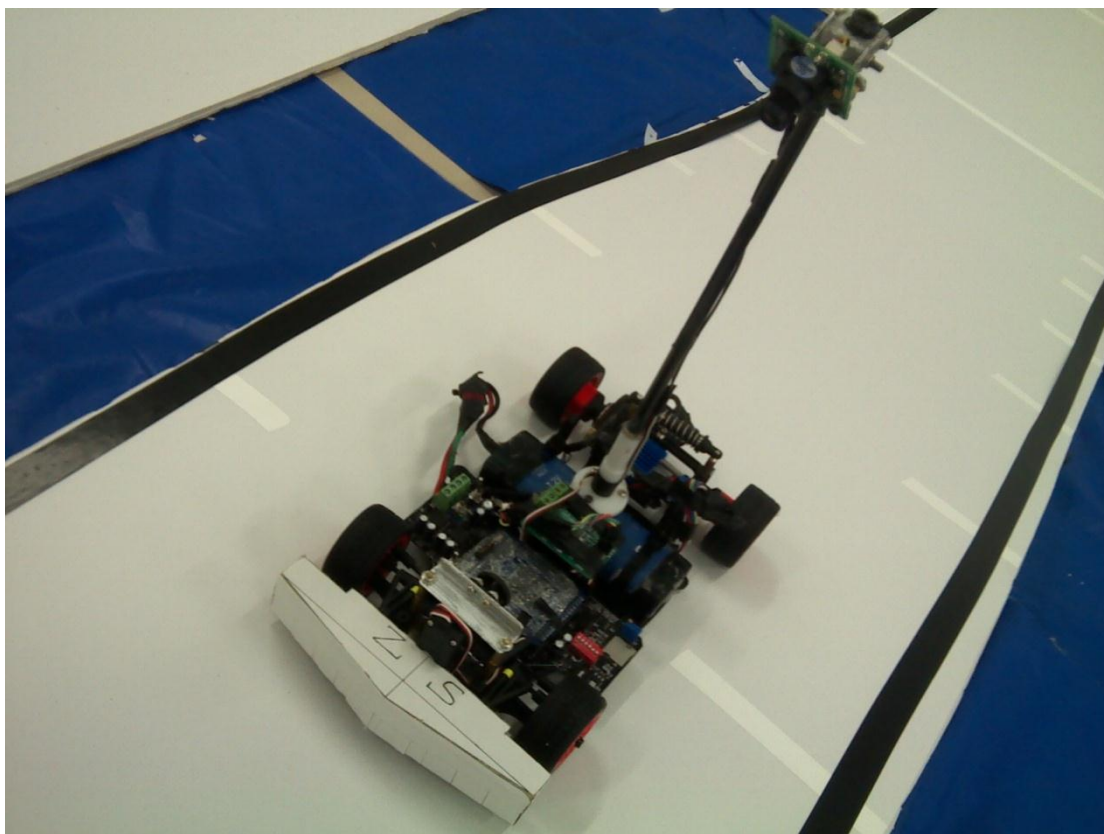


图 2.2 小车整体布局

2.3 车模结构参数

本次比赛摄像头组所用的车模为 A 型模型车，编号 03-302-2，尺寸为 $27 \times 16 \times 8\text{cm}$ ，有极好的调零装置，可以自由组装。A 型模型车马达的编号为 03-304-1，

DC 7.2V 380 马达，功率可达 26.5W，是模型车完成竞赛任务的有力保障。伺服器为编号 03-303-1，6V 时扭力达 6.5Kg.cm，动作速度 $0.16 \pm 0.02 \text{ sec}/60^\circ$ 。

第三章 机械结构的安装与调整

3.1 舵机的固定方式

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。

参考历届的技术报告，我们得知较为主流的舵机安装方式有直立式安装和倒置式安装，而这两种方式都能实现较好的控制。进一步考虑到舵机响应时间、稳定性以及虚位的诸多因素，我们最终选择竖直安装舵机，我们的舵机安装如图 3.1 和图 3.22 所示：

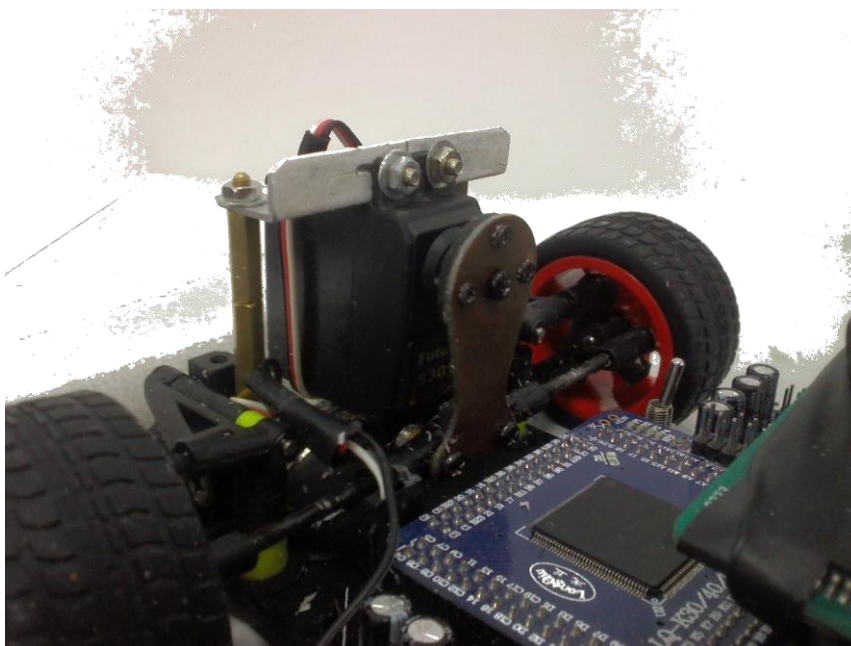


图 3.1 舵机安装方式 (①)

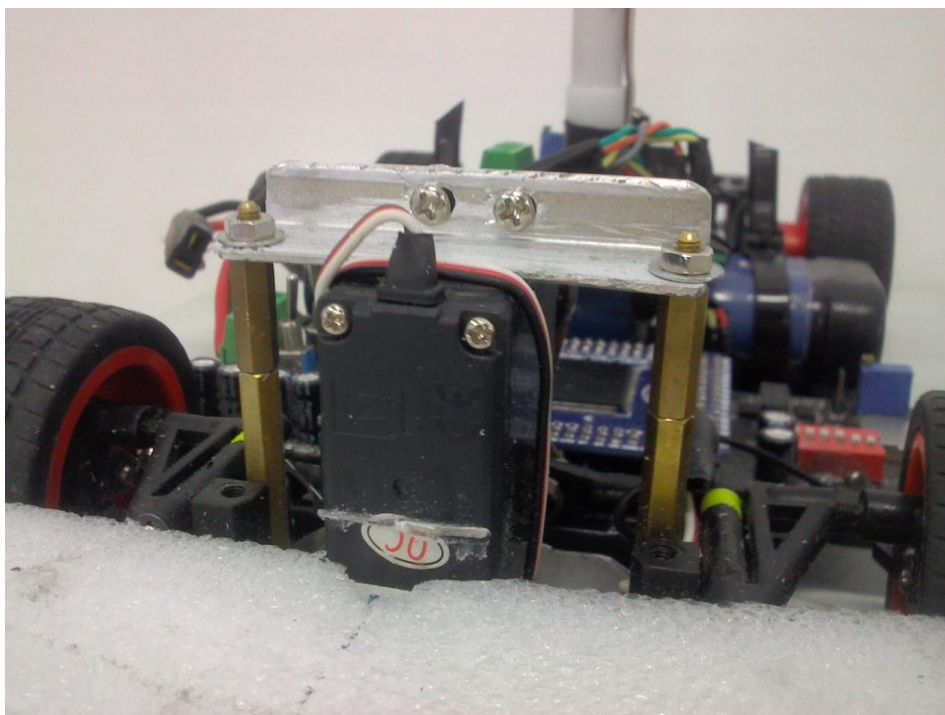


图 3.2 舵机安装方式 (②)

3.2 底盘设计

对于底盘，我们去掉了车模配有的悬挂。然后再适当调整了前后底盘的高度使车模整体的重心下降到合适的位置，即保证小车可以顺利过坡，且与赛道不会接触摩擦（用车模自带零配件即可调整底盘前后的高度）。我们知道合适的重心在小车过弯性能和小车速度这两个方面上起了很大的影响，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。

3.3 摄像头的安装

摄像头是整辆车的眼睛，摄像头的安装是最重要的。摄像头的安装要求使得摄像头位于整个车模的中心位置，而且高度要适合于图像的采集和处理。通过多组对比试验，我们最后决定使用单杆结构来固定摄像头，单杆使得整个支架十分简约，但也会带来摄像头易抖动的问题。这时，单杆与底盘之间的连接可靠性就显得尤为重要，如图 3.3 所示。为了严格的控制整辆车的重量和降低车重心，我们采用树脂碳杆作为摄像头的支架。单纯的碳杆刚性太强，遇到碰撞可

能会发生折断，树脂碳杆可以在保证低质量的同时，也保证了高强度。我们自己设计并制作了摄像头固定装置，该结构重量轻，可以方便的对摄像头进行调整和校正，如图 3.4 所示。

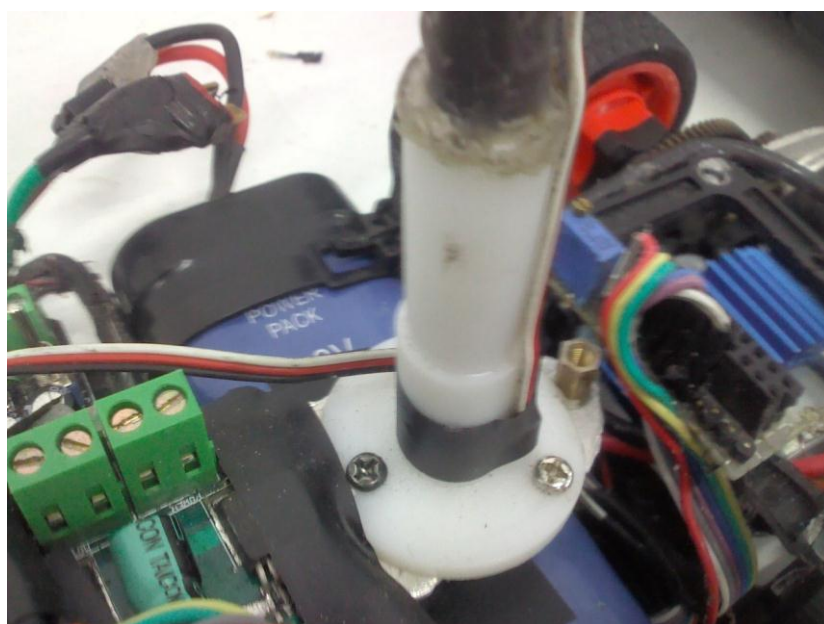


图 3.3 杆与底座的连接部分



图 3.4 摄像头的固定方式

另外，摄像头的高度也会影响其抖动，架的越高则越容易抖动。但为保证图像的稳定可靠，我们最终把摄像头架在了与地面的距离为 35cm 左右的位置。

3.4 编码器的安装

速度传感器一般可以选择对射式光栅或光电编码器。对射式光栅的重量轻，阻力小精度也高，然而光栅暴露在外界容易受到外界光线或粉尘等的影响，导致计数不准确；而光电编码器就不存在此类问题。所以最后我们选择了光电编码器，该编码器线数为 500 线，可以达到很高的精度，符合我们得要求。

在安装编码器的时候要保证有合适的齿轮咬合。咬合完美的原则是：两个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；传动部分要轻松、顺畅，容易转动。判断齿轮传动是否调整好的一个依据是，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，咬合过紧，或者两齿轮轴不平行，电机负载加大。调整好的齿轮传动噪音小，并且不会有碰撞类的杂音。图 2-4-1 为编码器的机械参数，根据这些参数我们自行设计和加工了轻巧的零件来使编码器与电机合适地咬合在一起，如图 3.5 所示。

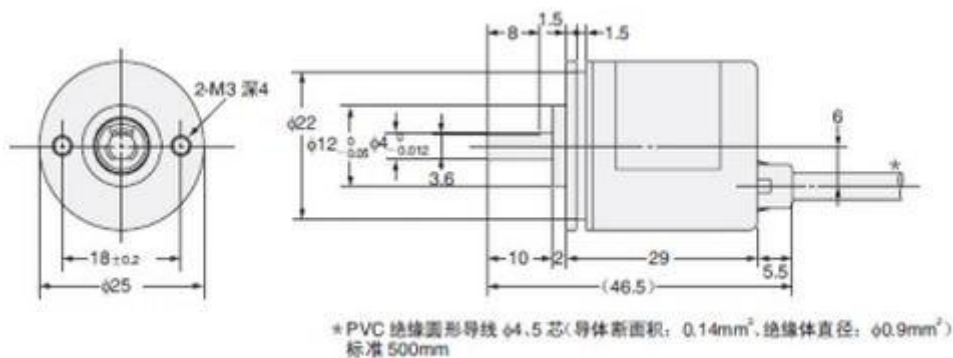


图 3.5 编码器的机械参数

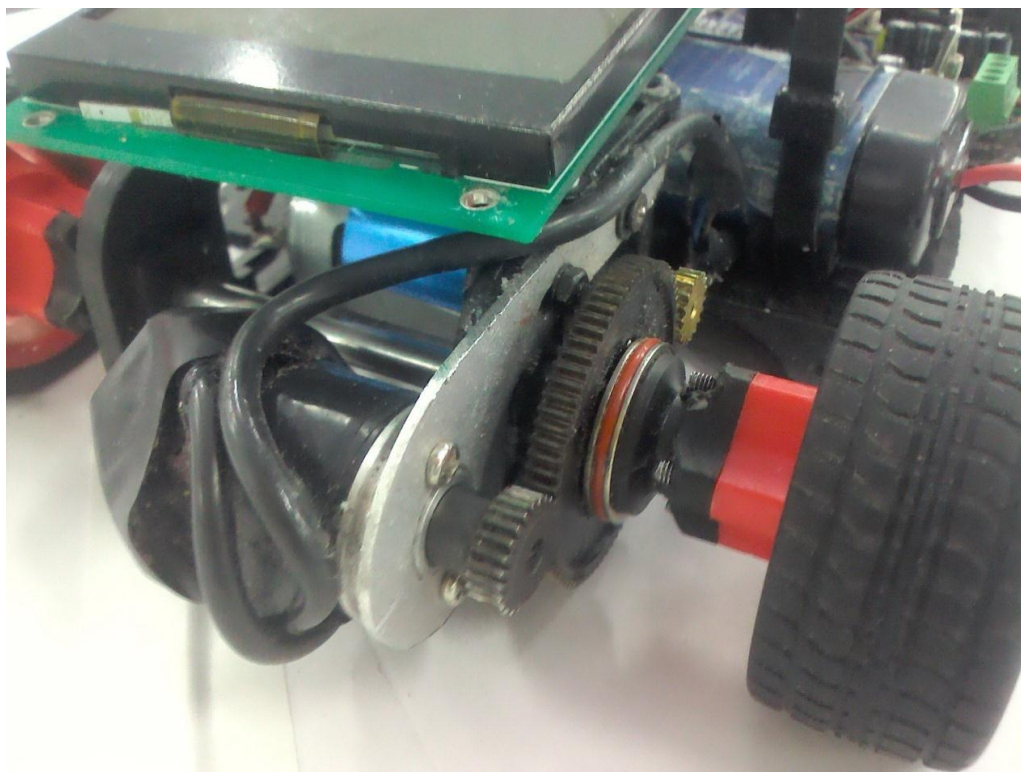


图 3.6 编码器的安装方式

3.5 前轮定位

小车的转向轮定位参数是很重要的因素，如果取得不恰当，那么将造成转向不灵活，效率低以及转向轮侧滑等问题，使得小车性能下降，加速轮胎的磨损，进而成为小车提速的瓶颈。

转向轮定位参数包括主销内倾角、主销后倾角、转向轮外倾角及转向轮前束。车轮定位的作用是使汽车保持稳定的直线行驶和转向轻便，并减少汽车在行驶中轮胎和转向机件的磨损。这其中最重要的就是转向轮外倾角和转向轮前束。

主销内倾是指主销装在前轴略向内倾斜的角度，它的作用是使前轮自动回正，如图 3.7。内倾角度越大前轮自动回正的作用就越强烈，但转向时也越费力，轮胎磨损增大；反之，角度越小前轮自动回正的作用就越弱。

主销后倾（Caster）是指主销装在前轴，上端略向后倾斜的角度，如图 3.6。它使车辆转弯时产生的离心力所形成的力矩方向与车轮偏转方向相反，迫使车

轮偏转后自动恢复到原来的中间位置上。由此主销后倾角越大，车速越高，前轮稳定性也愈好。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，因此高速时后倾的回正作用大，低速时内倾的回正作用大。

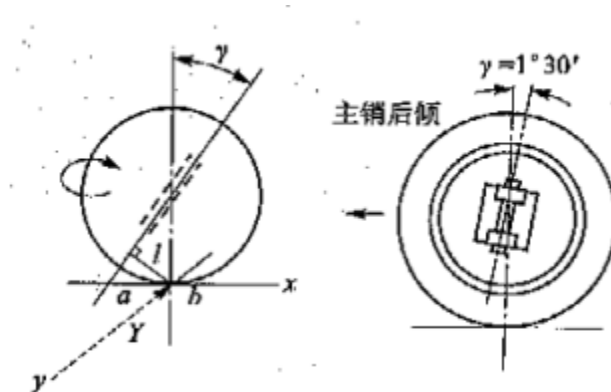


图 3.6 主销后倾角

车轮外倾角（Camber）是指从前放看前轴时，轮胎的中心平面不是垂直的，而是上面向外倾斜一个角度，如图 2-5-3。设置转向轮的外倾角是为了平衡和协调因为车重造成的前轮内倾倾向，使轮胎和路面呈垂直接触的最佳状态。

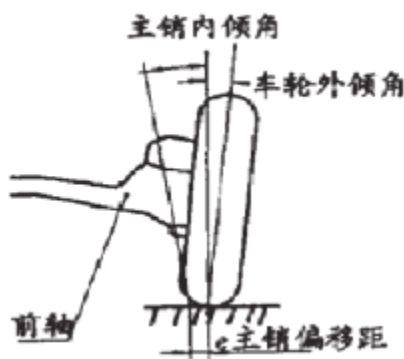


图 3.7 主销内倾角和车轮外倾角

转向轮前束 (Toe out) 是指同一轴两端车轮轮辋内侧轮廓线的水平直径的端点为等腰梯形的顶点，底边为车轮轴线。等腰梯形两底边长度之差为前束。如图 3.8 所示，当梯形前低边小于后底边时，前束为正 ($A < B$)，反之为负。车轮

的水平直径与纵向平面之间的夹角为前束角。正的前束角在车轮中心产生向内的侧向力，而正的外倾角在车轮中心产生向外的侧向力，因此前束角的作用是与外倾角协调，保持车轮做纯滚动和直线行驶，从而减少轮胎磨损，提高汽车的操纵稳定性。

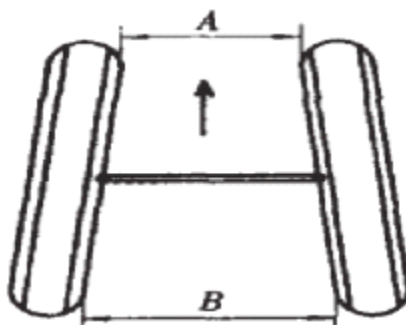


图 3.8 前束

3.6 后轮调整

车模后轮采用的电机由竞赛主办方提供，电机轴与后轮轴之间的传动比为 9:38（电机轴齿轮齿数为 18，后轮轴传动轮齿数为 76）。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。所以我们在电机安装过程中尽量使得传动齿轮轴保持平行，传动部分轻松、流畅，不存在卡壳或迟滞现象。

差速机构的作用是在车模转弯的时候，降低后轮与地面之间的滑动；并且还可以保证在轮胎抱死的情况下不会损害到电机。差速器的调整中要注意滚珠轮盘间的间隙，过松过紧都会使差速器性能降低，转弯时阻力小的车轮会打滑，从而影响车模的过弯性能。好的差速机构，在电机不转的情况下，右轮向前转过的角度与左轮向后转过的角度之间误差很小，不会有迟滞或者过转动情况发生。当车辆在正常的过弯行进中（假设：无转向不足亦无转向过度），此时 4 个轮子的转速（轮速）皆不相同，依序为：外侧前轮 > 外侧后轮 > 内侧前轮 > 内侧后轮。此次所使用车模配备的是后轮差速机构。差速器的特性是阻力越大的一

侧，驱动齿轮的转速越低；而阻力越小的一侧，驱动齿轮的转速越高，以此次使用的后轮差速器为例，在过弯时，因外侧前轮轮胎所遇的阻力较小，轮速便较高；而内侧前轮轮胎所遇的阻力较大，轮速便较低。后轮差速的调整主要是调整差速器中差速齿轮的咬合程度，差速的松紧与自己所要求的速度相匹配，已达到自己想要的状态。

第四章 系统硬件的设计及实现

4.1 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，并将数字地（DGND）与模拟地（AGND）用零欧姆电阻进行单点连接，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) 采用硬件实现图像的二值化，不仅提高了可靠性，也减小了 CPU 的负荷；

(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 100A 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。主板整体效果如图 4.1 所示：

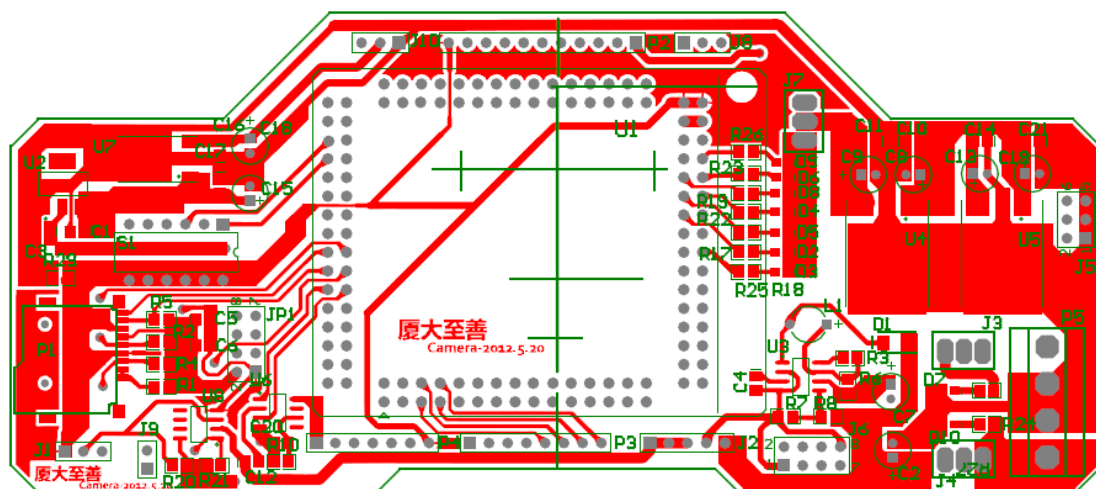


图 4.1 主板整体效果

4.2 CCD 摄像头

摄像头是智能小车系统信息提取关键，其信息输出信息的好坏将首先决定小车的性能。因此摄像头的选取必须慎重，既要保证图像质量好，满足后续处理和赛道识别的要求，又要考虑到单片机采集和处理的能力。对于由单片机组成的小系统来说，摄像头的分辨率不是越高越好。

因为这样只会徒增单片机的负担。摄像头由镜头、图像传感芯片和外围电路构成。而图像传感芯片是摄像头最重要的部分。摄像头按图像传感器可分为 CCD 图像传感器和 CMOS 图像传感器，CCD 图像传感器通常比 CMOS 图像传感器高 10 倍的感光度（ISO）。人眼能看到 1Lux 照度（满月的夜晚）以下的目标，CCD 图像传感器通常能看到比人眼略好，大约能看到在 0.1~3Lux 照度以下的目标，是 CMOS 图像传感器感光度的 3 到 10 倍。CMOS 图像传感器的感光度一般在 6Lux 到 15Lux 的范围内，CMOS 图像传感器有固定比 CCD 图像传感器高 10 倍的噪音，固定的图案噪音始终停留在屏幕上好像那就是一个图案，因为 CMOS 图像传感器在 10Lux 以下基本没用。但是 CMOS 图像传感器可以将所有逻辑和控制环都放在同一个硅芯片块上，可以使摄像头变得简单，因此 CMOS 图像传感器可以做得非常小、便于携带。同时 CMOS 图像传感器非常快速，比 CCD 图像传感器要快 10 到 100 倍，而且 CMOS 传感器不需要复杂的处理过程，可直接将图像半导体产生的电子转变成电压信号。

而根据摄像头输出信号的形式摄像头分为数字摄像头和模拟摄像头。数字摄像头是一种数字视频的输入设备，利用光电技术采集影像，而不像视频采集卡那样首先用模拟的采集工具采集影像，再通过专用的模数转换组件完成影像的输入。数字摄像头的优点是使用简单，输出即为数字信号。模拟摄像头多为 CCD 的，按不同档次分辨率不同。与数字摄像头同级的模拟摄像头一般有较高的分辨率，较好的实时性。模拟摄像头要与单片机相连必须先经过视频解码芯片进行解码，然后再交予单片机进行处理。

我们的主板上具有模拟视频输入接口，因此我们选择模拟摄像头。经过一些比较，我们选择了一款索尼模拟 CCD 摄像头。如图 4.2 所示：



图 4.2 索尼 CCD 摄像头

4.3 单片机最小系统板

Kinetis 系列微控制器是飞思卡尔公司于 2010 年下半年推出的基于 ARM Cortex-M4 内核的微控制器，是业内首款 Cortex-M4 内核芯片。Kinetis 系列微控制器采用了飞思卡尔 90 纳米薄膜存储器（TFS）闪存技术和 Flex 存储器功能

(可配置的内嵌 EEPROM)，支持超过 1000 万次的擦写。Kinetis 微控制器系列融合了最新的低功耗革新技术，具有高性能、高精度的混合信号能力，宽广的互连性，人机接口和安全外设。

4.4 稳压电源模块

稳定的电源对于一个控制系统来说至关重要，关系到系统能否正常工作，因此在设计智能车系统时为各个模块配置了合适、稳定的电源并且在电路设计上尽量避免不同电源和相同电源不同模块之间的干扰，保证了整个系统的稳定运行。

智能车系统的总的电源供应来自 7.2V 大容量镍镉电池，但是单片机 K60 最小系统 3.3V 的电源，SD 卡和无线调试模块需要 3.3V 低压，伺服电机工作电压范围在 4V 到 6V 之间，CCD 摄像头用 12V 供电，驱动电路需要 5V 的电源，直流电机需要 7.2V 电池直接供电，因此智能车电压调节电路框架如图 4.3 所示。

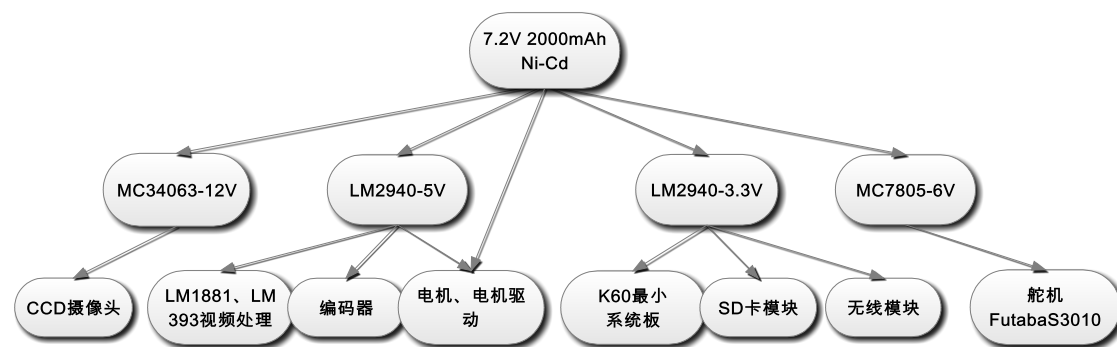


图 4.3 电压调节电路框架图

4.4.1 3.3V 、5V 稳压电路

SD 卡及无线模块所要求的标准电源电压为 3.3V，为此选用了低压差降压稳压芯片 LM2940 来为它们供电，在压差为 1V 时就可稳定输出 3.3V 电压，最大输出电流为 800mA，能够满足要求。

5V 电源模块用于为单片机系统、传感器模块和 LM1881 等芯片供电。经过比较，由于在电机驱动时电池压降较大为提高系统稳定性，必须使用低压差稳

压芯片。为此我们选用了低压差线性稳压芯片 LM2940-5.0V 为单片机和摄像头供电、LM1881 和传感器等供电，它们的纹波电压小，能对负载的变化迅速做出反应，适合为各个模块供电。电路如图 4.4 所示。

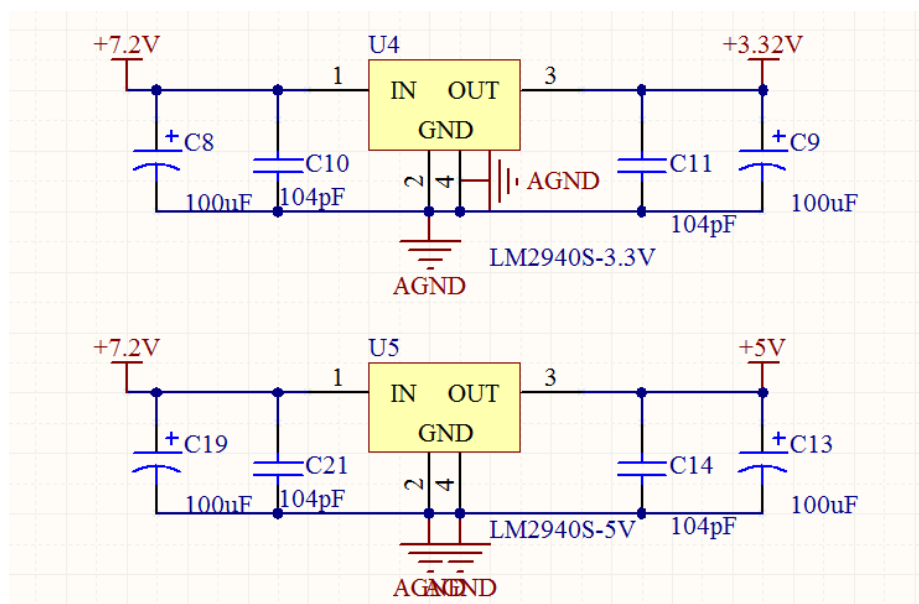


图 4.4 LM2940-3.3V、5V 稳压电路

4.4.2 6V 稳压电路

6V 供电电压用来给舵机供电。赛车调试过程中，发现若直接给舵机电池电压，电机驱动时对电源的电流冲击会使电源电压波动，导致舵机出现在给定打角时仍左右频繁摆动的现象，更严重的是这种方法会导致舵机烧掉的概率大大提高。经过比较 LM317 和 MC7805，发现 LM317 的压降要 2V 以上，而 MC7805 只需 1V 左右，而且 LM317 在大电流负载的情况下纹波较大，且它的输出阻抗较小，在舵机大摆角的情况下输出 6V 电压会被拉低。故最后我们选用了 MC7805 稳压芯片，完全能够满足舵机所需要的电压和电流，而且输出电压在 6V 左右可调以便满足不同要求。应用电路如图 3.7 所示。

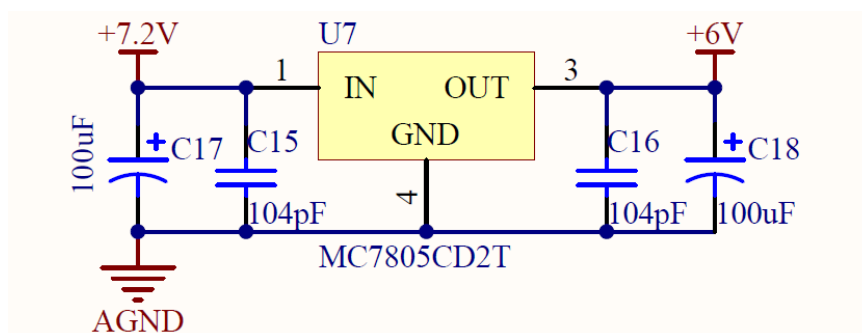


图 4.5 MC7805-6V 稳压电路

4.4.3 12V 升压电路

MC34063 升压扩流电路图，根据输出电压选择合适的扩流三极管，三极管要加散热片，本电路工作最低电压 4V，输出电压 12V，可以调整上拉电阻值改变输出电压。以下是 7.2V 到 12V 的升压电路，如图 4.6。

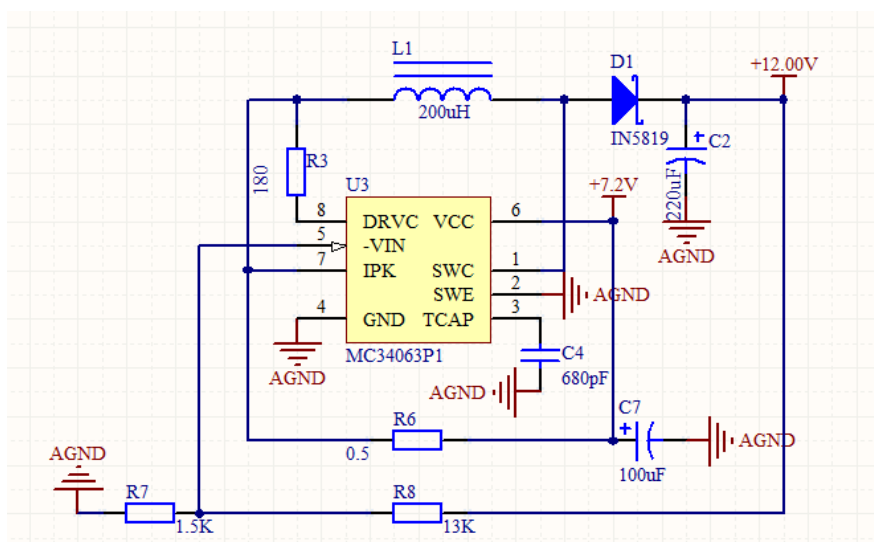


图 4.6 12V 升压电路

4.5 视频同步分离电路

我们用了 LM1881 视频分离芯片对视频的行同步和场同步进行了分离提取。而后进行了视频信号的二值化处理。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、场消隐信号等。若单单依靠单片

机进行处理效率不高，最好借助于视频分离电路对信号间的逻辑关系进行更好把握。LM1881 是最合适的选择。LM1881 的电路原理图如图 4.7。视频信号由 2 号引脚进入芯片，由 1 引脚给出行同步信号提供给单片机作为行中断。3 引脚提供场同步信号，提供给单片机作为场中断信号。

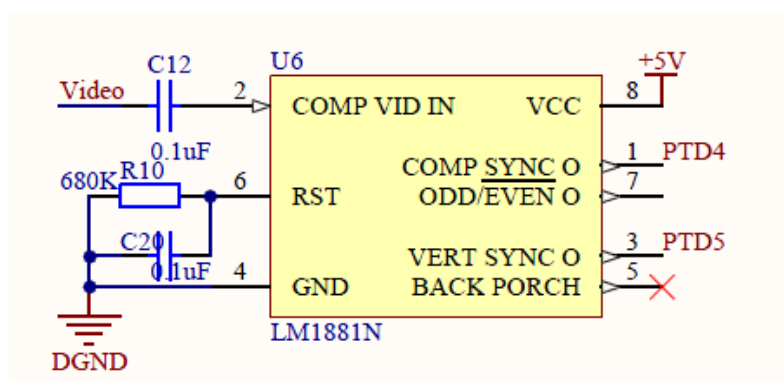


图 4.7 LM1881 电路

了对视频信号进行更高效的处理，我们设计了硬件二值化处理电路。采用了比较器 393 实现图像的二值化，电路如下：

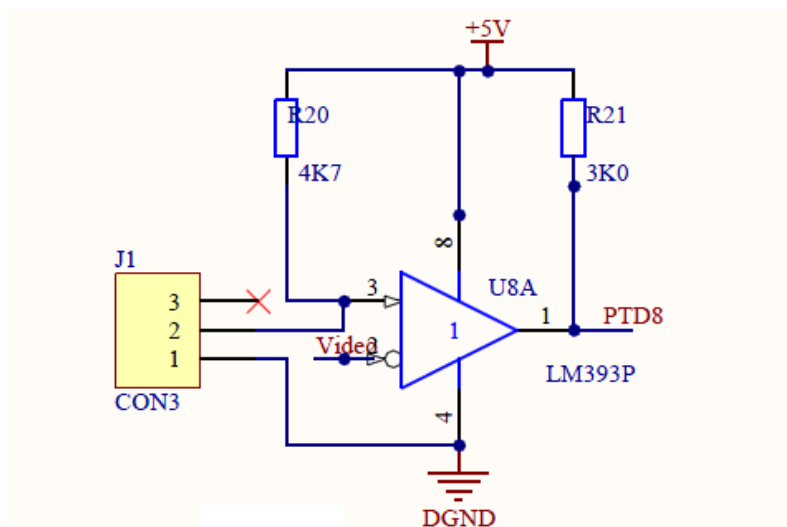


图 4.8 比较器电路

4.6 电机驱动电路设计

驱动电路对于竞速比赛的重要性是不言而喻的，较好的加速与制动能力对小车平均速度的提高有很大帮助。电机的速度与施加的电压成正比，输出转矩则与电流成正比。对直流电机的控制是一个挑战，因为必须在工作期间改变直流电机的速度。一般的，直流电机高效运行的最常见方法是施加一个 PWM（脉宽调制）方波，其通-断比率对应于所需速度。电机起到一个低通滤波器作用，将 PWM 信号转换为有效直流电平。PWM 驱动信号很常用，因为使用微处理器的控制器很容易产生 PWM 信号。虽然用精确的脉冲宽度可以调节电机的速度，实际应用中的 PWM 频率却是可变的，应对其进行优化，以防止电机颤抖，发出耳朵听得到的噪声。如要使直流电机反转，必须转换电机中电流的方向。一般电机驱动都用的是经典的 H 桥控制电路，有些芯片内部都集成了 H 桥电路，并且还集成了内部驱动，保护也做得不错，像 SGS 公司的 L298，飞思卡尔的 MC33886 和 MC33887 等，但是这些芯片的内部电阻较大，发热严重，驱动能力有限，经测试并不是十分理想。一般情况下，我们使用 MOSFET 驱动 IR2104 和 MOSFET 管 IRLR7843 搭建成 H 桥电路作为电机驱动。这种电路的优点是内阻很小，理论计算值在毫欧级别。IRLR7843 的最小内部只有 3.3 毫欧姆，最大允许连续源极电流 161A，超额符合要求；IR2104 外接自举电路，输出驱动电压在 10 到 20V，具有刹车使能功能，外接电路简单。缺点是需要外接 12V 电路，电路保护相对较差，电路稳定性相对较低，体积比较大。

针对第七届使用的电机，对驱动电路的驱动相对较低，而对稳定性和重量因素比较敏感的特点，我们选用在智能车中广泛使用的半桥驱动芯片 BTS7960。BTS7960 是应用于电机驱动的大电流半桥高集成芯片，它带有一个 P 沟道的高边 MOSFET、一个 N 沟道的低边 MOSFET 和一个驱动 IC。BTS7960 通态电阻典型值为 16 毫欧，驱动电流 43A。从数据上看 BTS7960 的驱动能力低于我们之前应用的 H 桥驱动电路，但驱动目前的电机已经绰绰有余，因此综合稳定性等因素，我们选用这款芯片搭建电机驱动电路。驱动电路模块原理图如下：

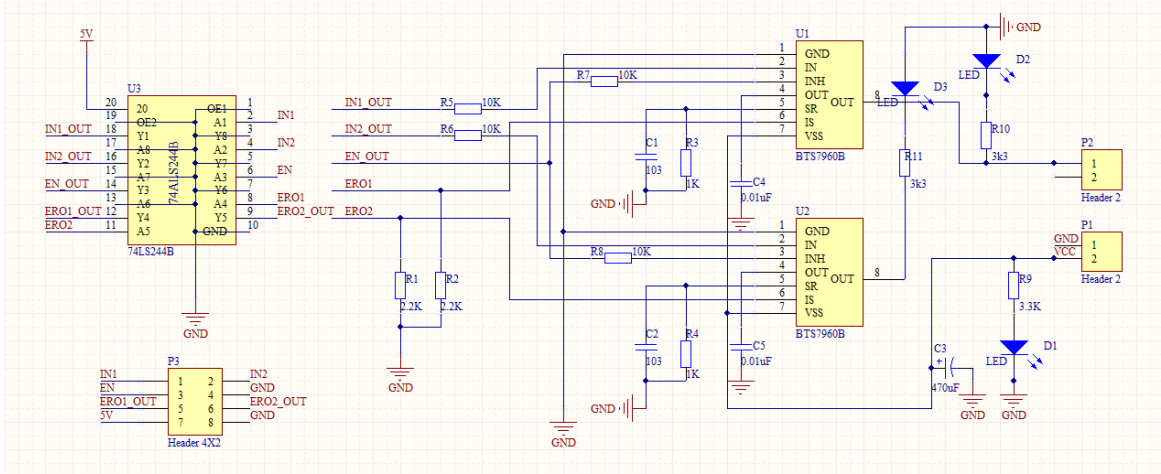


图 4.9 驱动电路原理图

驱动模块电路 PCB 和印制电路板如下：

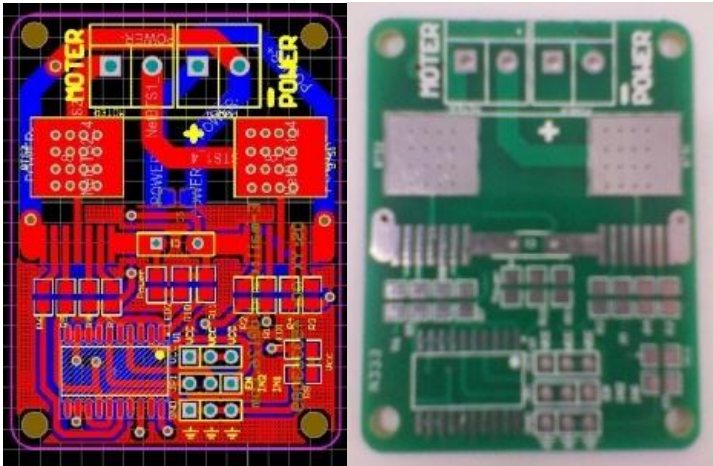


图 4.10 驱动电路原理图

4.7 硬件电路部分小结

硬件电路的关键是稳定，智能车系统的正常工作即是以硬件的稳定为前提的。而整个电路系统中，电源部分的稳定性亦至关重要，只有各个电源供应稳定可靠纹波小才能保证其它电路模块的正常工作，尤其是单片机最小系统。稳定的电路板为后续的工作奠定了坚实的基础。在电机转速急减急加或舵机打大角度时，电池电压会产生大的波动，此时若 5V 稳不住单片机就会出现复位等问题。

合理的使用滤波电容可以减少此类的影响，使电压稳定。

我们的电路中既包含模拟电路又包含数字电路，为提高系统稳定性，电路中区分了模拟地与数字地。为保证电路板的稳定性和可靠性，我们将设计好的 PCB 文件送入工厂打样。各个电路板之间通过排针排母或杜邦线相连，保证了整体电路系统的稳定性。

第五章 系统软件的设计与实现

高效稳定的软件程序是智能车平稳快速寻线的基础。我们设计的智能车使用 SONY 黑白 CCD 摄像头来获取赛道信息，图像采集处理就成了整个软件的核心内容。而单片机处理速度有限，如何高效的提取所需的赛道信息和利用所剩不多的单片机资源进行控制成为了我们研究的核心内容。经过不断的尝试和改进，在图像信息采集方面，最终我们成功的把计算融入单片机采集图像的间隙中，充分利用了单片机的资源。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，使在寻线中的智能车达到了稳定快速的效果。

5.1 系统软件流程图

软件设计部分主要包括：图像采集、图像处理、赛道判断、舵机打角、电机速度控制以及速度反馈处理，系统流程图如图 5.1 所示。

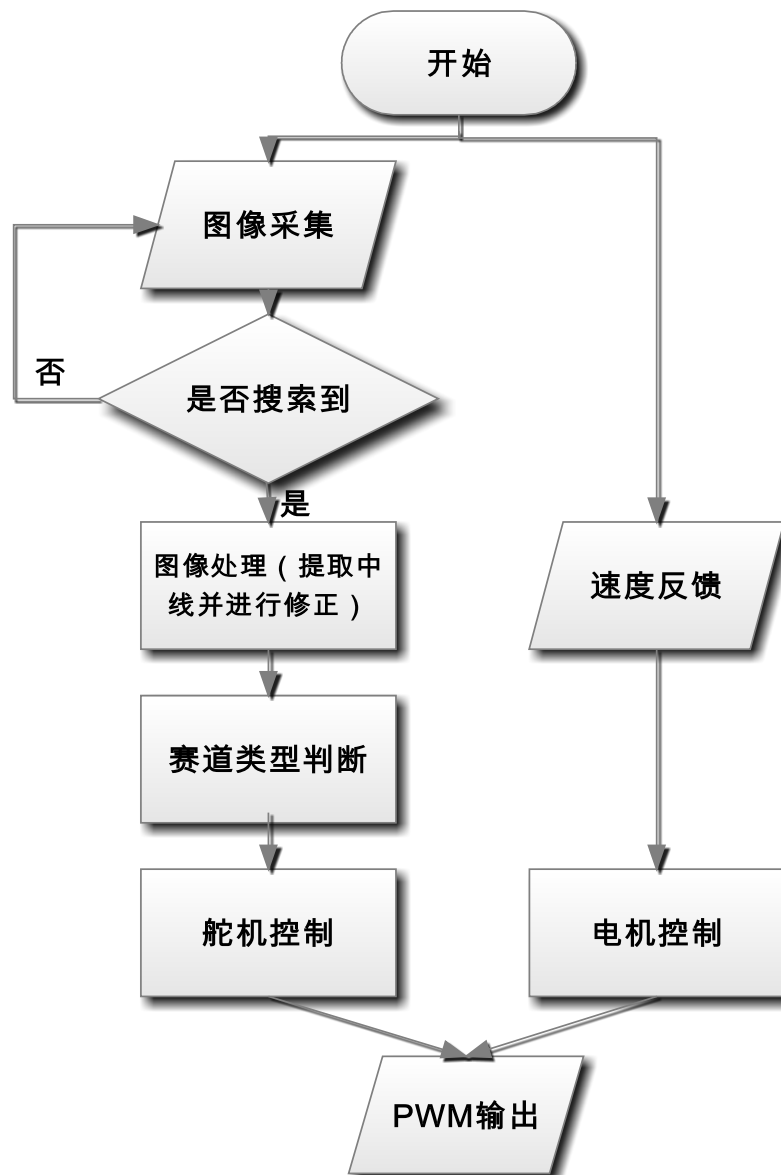


图 5.1 驱动电路原理图

5.2 图像采集与处理

图像采集部分主要有摄像头、视频分离芯片 LM1881。摄像头主要由镜头、CCD 图像传感器、图像信号形成电路、同步信号发生器、预中放、CCD 驱动器 等外围电路组成。LM1881 用于分离视频信号中的行同步脉冲和奇偶场同步脉

冲以供单片机处理。

对于图像信号的处理，我们的思路是一场图像隔几行采集一次，采集行之间的间隔来进行图像处理。仔细分析可以发现，对于智能车系统而言，不需要逐行采集图像，间隔若干行采集就可较好地获取赛道的信息。那么图像采集的过程中会存在一些空闲时间，我们可以利用这些时间来对图像进行处理。这种方法的优势在于：1) 图像处理嵌入到图像采集的过程中，有效解决了单片机处理图像信号能力不足的缺陷。2) 图像采集的时间可以更多，可以充分获取赛道的信息。同时也能够为智能车的控制算法预留出更多的时间。如图 5.2 所示。

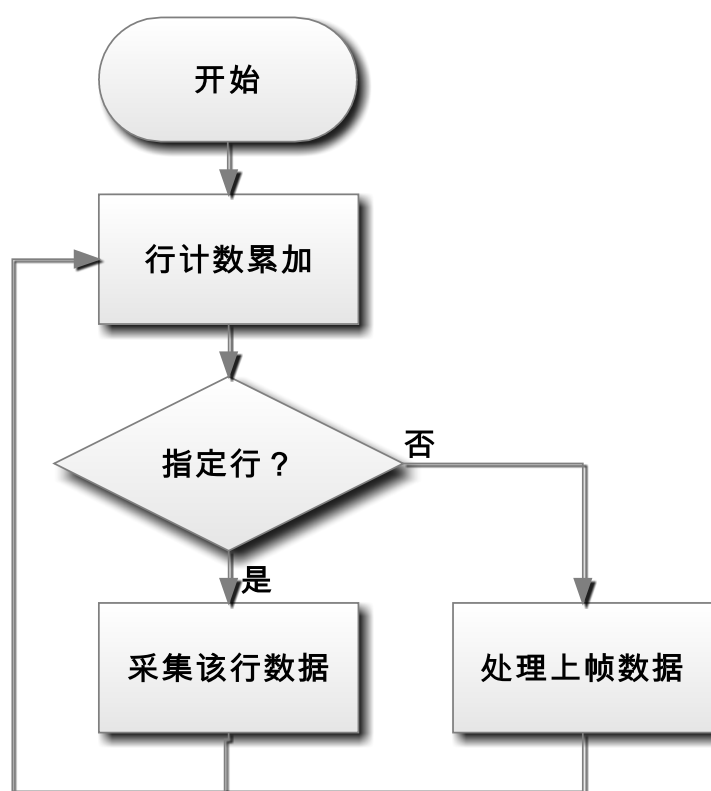


图 5.2 驱动电路原理图

为了使采集到的图像更加便于对小车的控制，我们对近处的图像采集的更加密集一些，远处的图像采集的更加疏松，这样也方便对起跑线的判断。采集到的图像用液晶来显示出来，如图 5.3 所示。



图 5.3 液晶模块

5.2.1 赛道中心提取

今年的赛道与往届最大的不同是，寻迹线不是在赛道的中心而改在了赛道的两边，这对于摄像头组来说基本是没有太大影响。相反地，由于赛道的底色基本都是深色，这样一来对赛道信息的提取变得更加的简单。我们采用的是简单的边缘检测法来提取赛道的边界，再加上一些的去噪，取得到两条边界基本都不会出错。提取完两条赛道的边界以后，我们用求平均的方法来求得赛道的中心，若是弯道，则对中心进行一定的补偿校正。这样一来，对于小车的控制就可以更多的参考历届小车的成熟的控制方式。

5.2.2 赛道类型判断

我们根据提取到的赛道中心来进行对赛道类型的判断，主要分为直道、小 S、弯道、180°弯。赛道类型的判断基本很难做到非常的准确，我们是根据多种方法结合起来进行判断的，最主要是根据赛道中心的一个增长趋势来进行判断的。我们 S_left、S_right 和 S_Straight 来分别记录赛道往左、往右和保持不变的行数。当 S_left 或 S_right 中的一个较大，其余的两个较小则判断为弯道；当 S_left 或 S_right 中的一个较大，其余的两个较小且有效行较小则判断为180°弯；当 S_left、S_right 和 S_Straight 基本差不多，则判断为小 S；当 S_Straight 较大，其余的较小，则判断为直道。

5.3 舵机转向和速度调节的 PID 控制算法

5.3.1 经典的 PID 控制算法

PID 控制是工程实际中应用最为广泛的调节器控制规律。问世至今 70 多年来，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。单位反馈的 PID 控制原理框图如图 5.4:

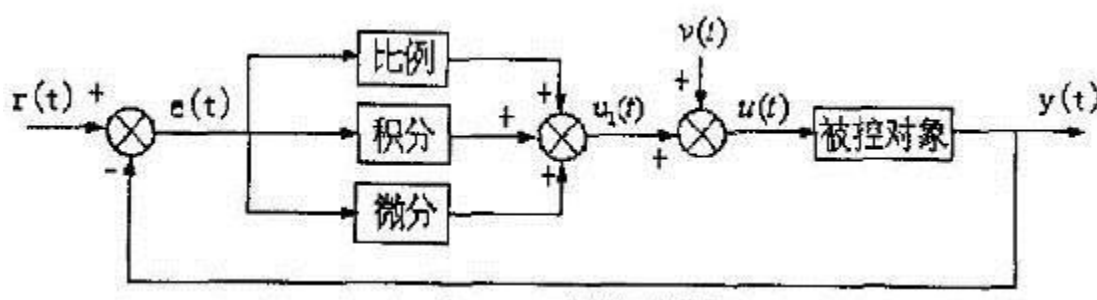


图 5.4 PID 控制器工作原理

单位反馈 e 代表理想输入与实际输出的误差,这个误差信号被送到控制器,控制器算出误差信号的积分值和微分值,并将它们与原误差信号进行线性组合,得到

输出量 u 。

$$u = k_p e + k_i \int e dt + k_d \frac{de}{dt} \quad \text{公式 1}$$

其中, k_p 、 k_i 、 k_d 分别称为比例系数、积分系数、微分系数。 u 接着被送到了执行机构,这样就获得了新的输出信号 u ,这个新的输出信号被再次送到感应器以发现新的误差信号,这个过程就这样周而复始地进行。

PID 各个参数作用基本介绍:

比例调节 (P) 作用: 是按比例反应系统的偏差,系统一旦出现了偏差,比例调节立即产生调节作用用以减少偏差。比例作用大,可以加快调节,减少误差,但是过大的比例,使系统的稳定性下降,甚至造成系统的不稳定。

积分调节 (I) 作用: 是使系统消除稳态误差,提高无差度。因为有误差,

积分调节就进行，直至无差，积分调节停止，积分调节输出一常值。积分作用的强弱取决与积分时间常数 T_i ， T_i 越小，积分作用就越强。反之 T_i 大则积分作用弱，加入积分调节可使系统稳定性下降，动态响应变慢。积分作用常与另两种调节规律结合，组成 PI 调节器或 PID 调节器。

微分调节（D）作用：微分作用反映系统偏差信号的变化率，具有预见性，能预见偏差变化的趋势，因此能产生超前的控制作用，在偏差还没有形成之前，已被微分调节作用消除。因此，可以改善系统的动态性能。在微分时间选择合适情况下，可以减少超调，减少调节时间。微分作用对噪声干扰有放大作用，因此过强的加微分调节，对系统抗干扰不利。此外，微分反应的是变化率，而当输入没有变化时，微分作用输出为零。微分作用不能单独使用，需要与另外两种调节规律相结合，组成 PD 或 PID 控制器。

对连续系统中的积分项和微分项在计算机上的实现，是将上式转换成差分方程，由此实现数字 PID 调节器。

位置式 PID 控制算法

用矩形数值积分代替上式中的积分项，对导数项用后向差分逼近，得到数字 PID 控制器的基本算式（位置算式）：

$$u_n = k_p(e_n + \frac{1}{T_i} \sum_{k=1}^n e_k T + T_d \frac{e_n - e_{n-1}}{T}) \quad \text{公式 2}$$

其中 T 是采样时间， k_p 、 T_i 、 T_d 为三个待调参数，我们在实际代码实现算法时，处理成以下形式：

$$\text{PreU} = K_p * \text{error} + K_i * \text{Integral} + K_d * \text{derror} \quad \text{公式 3}$$

增量式 PID 控制算法

对位置式加以变换，可以得到 PID 算法的另一种实现形式（增量式）：

$$\Delta u_n = u_n - u_{n-1} = k_p \left[(e_n - e_{n-1}) + \frac{1}{T_i} e_n + \frac{T_d}{T} (e_n - 2e_{n-1} + e_{n-2}) \right] \quad \text{公式 4}$$

我们在实际代码实现时，处理成以下形式：

$$\text{PreU} += (K_p * \text{derror} + K_i * \text{error} + K_d * \text{dderror}) \quad \text{公式 5}$$

这种算法用来控制步进电机特别方便，对直流电机也可以采用，其实如果控制有更高的要求或者干扰因素较多，我们可以对 PID 算法做各种改进，比如

用梯形法做数值积分以提高精度，将差分改成一阶数字滤波等等，在实际调车的过程中，我们确实遇到过由于自制码盘采样得到的脉冲上升下降沿不够陡，使得速度采样出现不稳定和失真，但由于这些附加处理比较耗费代码的运行时间，出于代码效率和实际效果的比较，我们没有采用这些改进的方案，另外可以考虑加反向器来整波形得到较为理想的方波。

运用 PID 控制的关键是调整三个比例系数，即参数整定。PID 整定的方法有两大类：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。由于智能车整个系统是机电高耦合的分布参数系统，并且要考虑赛道具体环境，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不断修正，模型参数变化较频繁，可操作性不强；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，我们采用了这种方法，同时，我们先后实验了几种动态改变 PID 参数的控制方法。

5.3.2 经典 PID 算法在本智能车上的应用

舵机控制。在小车跑动中，因为不需要考虑小车之前走过的路线，所以，我们舍弃了 I 控制，将小车舵机的 PID 控制简化成 PD 控制。我们对舵机的控制采用了位置式 PID 算法，根据往届的资料和自己的测试，将图象经过算法处理后得到的赛道中心位置和对应的舵机 PID 参照角度处理成一次线性关系。另外，为了过弯更加的顺畅，我们将 K_p 值设为与偏移量成二次行数的关系。

速度控制。我们对速度的控制采用了增量式 PID 算法，我们在速度控制中采取的基本策略是弯道降速，直道提速。速度控制策略的好坏就决定小车取得能否优秀成绩，如何给速度是关键，什么样的赛道配合什么样的速度。比如过弯时要提前减速使赛车在入弯前达到一个安全速度以免冲出赛道，过 S 弯时要同一般弯道和直道区分开，经过实际测试我们将速度给定处理成二次曲线，这样达到的实际效果还是比较理想的。电机闭环控制 PID 控制程序设计如下：

speed 为编码器定时测到的脉冲数

//偏差计算和传递

EN2 = EN1;

```
EN1 = EN0;

EN0 = sv - speed;

//计算增量

dPWMN    =    (int)((Kp_DianJiN)*(float)(EN0    -    EN1)    +
Ki_DianJiN*(float)(EN0) +

//限幅

if(PWMN > 5000)

{

    PWMN = 5000;

}

if(PWMN < 0)

{

    PWMN = 0;

}
```

5.4 调试环境软件说明

5.4.1 开发工具

软件开发工具选用的是 Embedded Workbench for ARM。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。它为用户提供一个易学和具有最大 量代码继承能力的开发环境,以及对大多数和特殊目标的支持。嵌入式 IAR Embedded Workbench 有效提高用户的工作效率,通过 IAR 工具,可以大大节省软件调试时间。

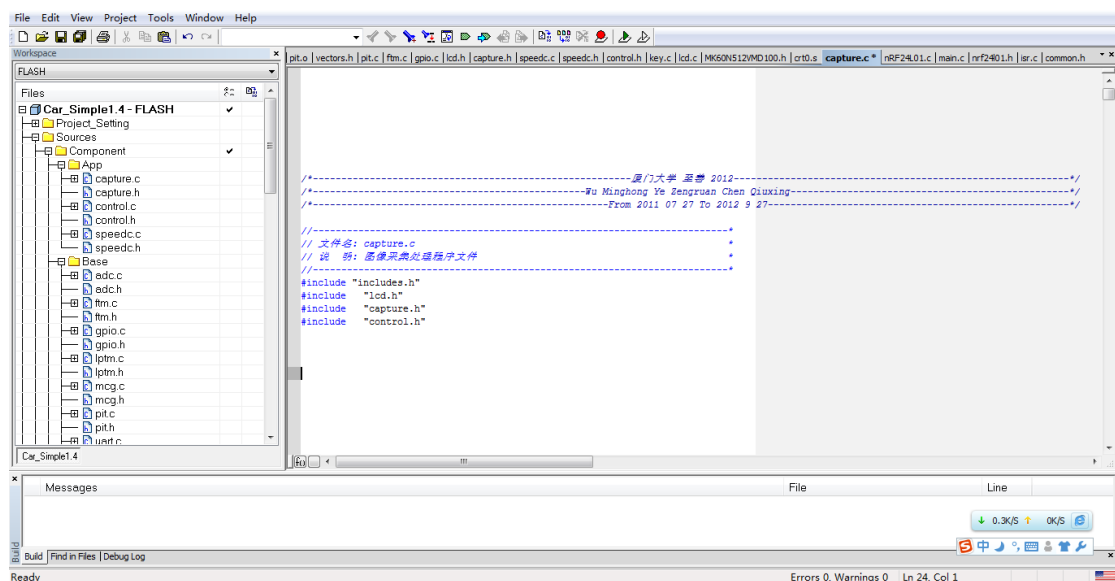


图 5.5 软件开发工具

5.4.2 上位机调试软件的设计

为了方便实时地监视小车在行进过程中的各种参数，我们特意用 Labview 编写了一个上位机，并利用无线模块（下面有介绍）将车子的实时数据传输到电脑上，这样，大大提高了调试车子的效率。



图 5.6 Labview 调试软件

5.4.3 无线调试模块设计

NRF24L01 是 NORDIC 公司最近生产的一款无线通信通信芯片，采用 FSK 调制，内部集成 NORDIC 自己的 Enhanced Short Burst 协议。可以实现点对点或是 1 对 6 的无线通信。线通信速度可以达到 2M (bps)。NORDIC 公司提供通信模块的 GERBER 文件，可以直接加工生产。嵌入式工程师或是单片机爱好者只需要为单片机系统预留 5 个 GPIO，1 个中断输入引脚，就可以很容易实现无线通信的功能，非常适合用来为 MCU 系统构建无线通信功能。

同时，我也利用无线模块，设计出一个简易的遥控器。在平时调试过程中，遥控器对于保护车子起到了很重要的作用。以下是无线模块和遥控器图片：

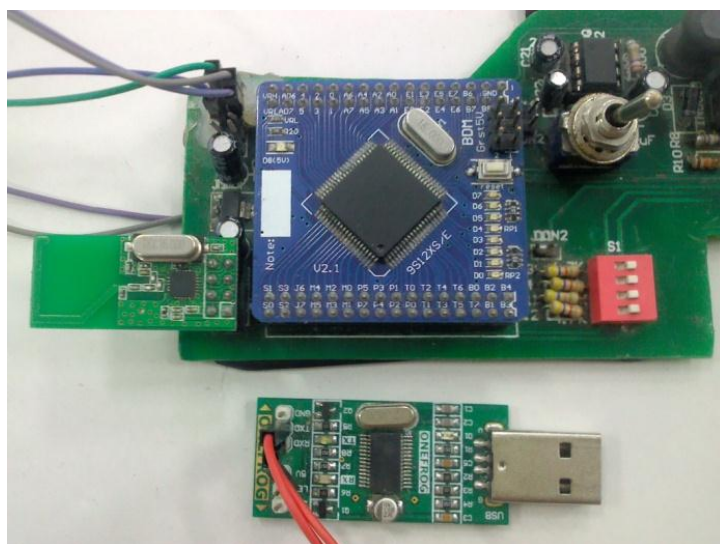


图 5.7 无线调试模块

第六章 总结与参数统计

6.1 模型车的主要技术参数

赛车基本参数			
长： 28.5cm		宽： 18.5cm	
高： 35cm		车重： 1200g	
功耗			
空载： 3W		带载： 大于4W	
电容总容量： 1700uF			
传感器			
光电编码器	1个	CCD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/远）： 5/130cm		精度(近/远)： 2/12.1mm	
频率： 50Hz			

表 6.1 模型车的主要技术参数

6.2 结论

我们从认识、报名参加智能车，到自己动手编写程序、调整机械、焊接电路让车子跑起来，历时接近一年。这中间也花费了大量的精力、心思，也得到了学校、学院、老师和师兄们的支持，在此向他们表示感谢。

精彩激烈的赛区赛才刚刚落下帷幕，全国赛就已经吹响集结号。面对即将到来的大赛，也许我们的知识还不够丰富，考虑问题也不够全面，但是，我们小组也是经历过华南赛区赛的考验的，相信这一年来，我们的辛勤与付出，会让我们在比赛中收获到我们最美好的记忆。

参考文献

- [1]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
- [2]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
- [3]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
- [4]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
- [5]Protel DXP 电路设计与制版实用教程. 2 版李小坚[等]编著. 人民邮电出版社 2009
- [6]卓晴, 王璿, 王磊. 基于面阵CCD 的赛道参数检测方法. 电子产品世界, 2006(4):141-143
- [7]胡海峰, 史忠科, 徐德文. 智能汽车发展研究[A]. 西安: 西北工业大学, 2004
- [8]贺亮亮, 谢剑桥, 周沈刚. 第六届“飞思卡尔”杯全国大学生智能车大赛北京科技大学
摄像头一队技术报告[R]. 北京科技大学 2011
- [9]龚路, 王浩, 黄炫圭. 第六届全国大学生“飞思卡尔”杯智能汽车竞赛上海交通大学Cyber++
队技术报告[R]. 上海交通大学. 2009
- [10]薛定宇, 陈阳泉. 基于MATLAB/Simulink的系统仿真技术与应用 [M]. 北京: 清华大
学出版社, 2002.
- [11]陆科宇, 刘伯阳, 周发亮. 第六届全国大学生“飞思卡尔”杯智能汽车竞赛国防科技大学
摄像铁军一师技术报告[R]. 国防科技大学. 2009
- [12]陈伯时. 电力拖动自动控制系统——运动控制系统. 机械工业出版社, 2005
- [13] OmniVision Serial Camera Control Bus (SCCB) Functional Specification V2.1. OmniVision
Technologies, Inc. 26th February 2003
- [14]于海生, 潘松峰, 于培仁等. 微型计算机控制技术 [M]. 北京: 清华大学出版社, 1999
- [15] Tinku Acharya等. 数字图像处理——原理与应用. 清华大学出版社, 2007

附录：程序源代码

[illegible]

```
void Delay_us(unsigned int n)
{
    for(;n>0;n--);
    for(;n>0;n--);
    for(;n>0;n--);
    for(;n>0;n--);
    for(;n>0;n--);
}
```

```
void Delay_ms(unsigned int n)
{
    for(;n>0;n--)
    {
        Delay_us(7980);
    }
}
```

//最基本的读函数

```
uchar SPI_RW(uchar byte)
{
    uchar bit_ctr;
    for(bit_ctr=0;bit_ctr<8;bit_ctr++) // output 8-bit
    {
        if(byte & 0x80) // output 'byte', MSB to MOSI
            MOSI_OUT=0;///
        else
            MOSI_OUT=1;///
        byte = (byte << 1); // shift next bit into MSB..
        SCK_OUT = 1; // Set SCK high..
        byte |= MISO_IN; // capture current MISO bit
        SCK_OUT = 0; // ..then set SCK low again
    }
    return(byte); // return read byte
}
```

//寄存器 读 函数

```
uchar SPI_Read(uchar reg)
{
    uchar reg_val;
    CSN_OUT = 0; // CSN low, initialize SPI communication...
```



```

SPI_RW(reg); // Select register to read from..
reg_val = SPI_RW(0); // ..then read register value
CSN_OUT = 1; // CSN high, terminate SPI communication
return(reg_val); // return register value
}

```

//寄存器 批量读 函数

```

uchar SPI_Read_Buf(uchar reg, uchar *pBuf, uchar bytes)
{
uchar status,byte_ctr;
CSN_OUT = 0; // Set CSN low, init SPI transaction
status = SPI_RW(reg); // Select register to write to and read status byte
for(byte_ctr=0;byte_ctr<bytes;byte_ctr++)
pBuf[byte_ctr] = SPI_RW(0); // Perform SPI_RW to read byte from nRF24L01
CSN_OUT = 1; // Set CSN high again
return(status); // return nRF24L01 status byte
}

```

//寄存器 批量写 函数

```

uchar SPI_Write_Buf(uchar reg, uchar *pBuf, uchar bytes)
{
uchar status,byte_ctr;
CSN_OUT = 0; // Set CSN low, init SPI transaction
status = SPI_RW(reg); // Select register to write to and read status byte
Delay_us(1);
for(byte_ctr=0; byte_ctr<bytes; byte_ctr++) // then write all byte in buffer(*pBuf)
SPI_RW(*pBuf++);
CSN_OUT = 1; // Set CSN high again
return(status); // return nRF24L01 status byte
}

```

//接收模式

```

void RX_Mode(void)
{
    CE_OUT = 0;
    SPI_Write_Buf(WRITE_REG + RX_ADDR_P0, TX_ADDRESS,
TX_ADR_WIDTH); // 接收设备接收通道 0 使用和发送设备相同的发送地址
    SPI_RW_Reg(WRITE_REG + EN_AA, 0x01); // 使能接收通道 0 自动应答
    SPI_RW_Reg(WRITE_REG + EN_RXADDR, 0x01); // 使能接收

```

通道 0

```

    SPI_RW_Reg(WRITE_REG + RF_CH, 40);           // 选择射频通道
    道 0x40
    SPI_RW_Reg(WRITE_REG + RX_PW_P0, TX_PLOAD_WIDTH); // 接收
    通道 0 选择和发送通道相同有效数据宽度
    SPI_RW_Reg(WRITE_REG + RF_SETUP, 0x07);       // 数据传输率
    1Mbps, 发射功率 0dBm, 低噪声放大器增益
    SPI_RW_Reg(WRITE_REG + CONFIG, 0x0f);         // CRC 使能,
    16 位 CRC 校验, 上电, 接收模式
    CE_OUT = 1;                                   // 拉
    高 CE 启动接收设备
}

```

//发送模式

//void TX_Mode(uchar * BUF)

void TX_Mode()

```

{
    CE_OUT = 0;
    SPI_Write_Buf(WRITE_REG + TX_ADDR, TX_ADDRESS,
    TX_ADR_WIDTH); // 写入发送地址
    SPI_Write_Buf(WRITE_REG + RX_ADDR_P0, TX_ADDRESS,
    TX_ADR_WIDTH); // 为了应答接收设备, 接收通道 0 地址和发送地址相同
    SPI_RW_Reg(WRITE_REG + EN_AA, 0x01); // 使能接收通道 0 自动
    应答
    SPI_RW_Reg(WRITE_REG + EN_RXADDR, 0x01); // 使能接收通道 0

    SPI_RW_Reg(WRITE_REG + SETUP_AW, 0x03);

    SPI_RW_Reg(WRITE_REG + SETUP_RETR, 0x0a); // 自动重发延时等待
    250us+86us, 自动重发 10 次
    SPI_RW_Reg(WRITE_REG + RF_CH, 40); // 选择射频通道 0x40
    SPI_RW_Reg(WRITE_REG + RF_SETUP, 0x07); // 数据传输率 2Mbps,
    发射功率 0dBm, 低噪声放大器增益
    SPI_RW_Reg(WRITE_REG + CONFIG, 0x0e); // CRC 使能, 16 位 CRC
    校验, 上电

    // SPI_Write_Buf(WR_TX_PLOAD, BUF, TX_PLOAD_WIDTH);
    // 写数据包到 TX FIFO

    CE_OUT = 1;
}

```

```

}

void TX_Data(uchar *BUF)
{
    SPI_Write_Buf(WR_TX_PLOAD,      BUF,      TX_PLOAD_WIDTH);
    // 写数据包到 TX FIFO
    //Delay_us(100);    //  //
    while(IRQ_IN);
    status=SPI_RW_Reg(WRITE_REG+STATUS,0xff);
}

while(1)
{
    PORTD_PCR4                |=                PORT_PCR_IRQC(9);
    //开行中断
    while(1)
    {
        if( (GPIOC_PDIR&0X10)==0x10)
        {
            if( (system_time > 4000) && ( (GPIOC_PDIR&0X80)==0x80))    //第
4 个拨码开关
            {
                check_end_line=1;    //开始检测起跑线，终点线
            }
            else
            {
                check_end_line=0;    //不用检测起跑线，终点线
                End_Flag = 0;//不停车
                StartDelay=0;
            }

            if(SPEED_PID_FLAG    ==    1    &&    start_flag    !=    0    )
            //等待延时结束和软启动结束信号，启动 PID //控制速度

            //if(start_flag != 0 )                                //等待延时结束和软启动结束信
            号，启动 PID //控制速度
            {
                if(Rx_stop == 0)
                {

```

```
        ZMotorPWM=0;
        FMotorPWM=0;
    }
    else if(Rx_stop == 2) //退车
    {
        SteerPWM=Center;
        ZMotorPWM=0;
        FMotorPWM=1700;
    }
    else if(Rx_stop == 1)
    {
        if( (GPIOC_PDIR&0X80)!=0x80 )//|| system_time < 1600)
        {
            End_Flag = 0;
            StartDelay=0;
            StartFlag = 0;
        } //清除停车标志，无需停车

        if(End_Flag != 0)
        {
            SV_Mc = 0;
            SV_McN = 0;
            if(speed > 10)
            {
                if(end_line_stop_delay == 0)
                {
                    end_line_stop_delay = system_time;
                }

                if( (end_line_stop_delay != 0) && (system_time -
end_line_stop_delay > (260 - speed)/100*10/2) )
                {
                    StopControl_end_line(3 - speed/10);
                }
            }
            else
            {
                STC = 0;
                FMotorPWM = 0;//滑行
            }
        }
    }
}
```

```
        ZMotorPWM = 0;
    }
}
else
{
    if(speed_class == 1)
    {
        SpeedControl_PID_XLL_ZS1();
    }
    else if(speed_class == 2)
    {
        SpeedControl_PID_XLL_ZS2();
    }
    else if(speed_class == 3)
    {
        SpeedControl_PID_XLL_ZS3();
    }
    else if(speed_class == 4)
    {
        SpeedControl_PID_XLL_ZS4();
    }
    //以下的 4 档带定时
    else if(speed_class == 5)
    {
        SpeedControl_PID_XLL_ZS1();
    }
    else if(speed_class == 6)
    {
        SpeedControl_PID_XLL_ZS2();
    }
    else if(speed_class == 7)
    {
        SpeedControl_PID_XLL_ZS3();
    }
    else if(speed_class == 8)
    {
        SpeedControl_PID_XLL_ZS4();
    }
}
```

```
    }
    SPEED_PID_FLAG=0;
}

Get_Speed_Class();      //读取控制档，控制方案

if(start_flag == 0 && StartFlag == 0)
{
    if(speed_class <= 4) //前 4 档(1,2,3,4)无定时
    {
        GO_GO();
//等待延时结束和软启动结束信号，启动 PID
    }
    else
    {
        GO_GO_PIT();      //后面 4 档（5,6,7,8）带定时
    }
}

if((process_count > save_count) )

//如果处理行大于存储行，就开始提取黑线，因为是倒序存储，所以也要倒序处理：从近处开始存储和处理 start_flag=1;&& (start_flag != 0)
{
    Process();

//light_change(Light_Run_PORT,Light_Run4);//指示灯的亮、暗状态切换
}
}
else
{
    Get_Speed_Class();      //读取控制档，控制方案
    check_end_line=0;
    end_line_stop_delay = 0;
    End_Flag = 0;//不停车
    Rx_stop = 1;
    StartDelay=0;
    SaiDaoType = 0;
```

```
stop_control_flag=0;
stop_time=0;
stop_delay_time=0;
StartFlag = 0;
system_time = 0;
//系统时间初始化清零，3 秒启动
start_flag=0;
//回正&停车
SteerPWM=Center;
ZMotorPWM=0;
FMotorPWM=0;
//开始键盘检测
JianPan_Use();
//清楚和初始化变量
Clear_and_Set_Value( );

//uart_speed();//有线串口发送速度值，调 PID
}
}
//while 2
}
//while 1
}
```