第十一届全国大学生“恩智浦”杯
智能汽车竞赛

技 术 报 告

学　　校：　洛阳理工学院

队伍名称：　洛理摄像头一队

参赛队员：　王鹏辉

　　张继春

　　李开创

带队教师：　葛运旺

　　姚惠林

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全国大学生“恩智浦”杯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：王明智 张维智 李开创

带队教师签名：

日　　期：2016.8.14

摘要

文中介绍了我们的智能车控制系统软硬件结构和开发流程，整个智能车涉及车模机械调整，传感器选择，信号处理电路设计，控制算法优化等许多方面。

本文设计的智能车系统以 MK10DN512VLQ10 微控制器为核心控制单元，基于 COMS 摄像头 OV7725 的图像采样获取赛道图像信息，提取赛道中心线，计算出小车与黑线间的位置偏差，采用 PD 方式对舵机转向进行反馈控制。使用 PID 控制算法调节驱动电机的转速，结合特定算法分析出前方赛道信息实现对模型车运动速度的闭环控制。

关键字：智能车，MK10DN512VLQ10，OV7725，PID

Abstract

This paper introduces the our smart car control system hardware and software structure and development process, the entire intelligent car involves the mechanical adjustment models, sensor selection, signal processing circuit design, control algorithm optimization and many other aspects.

In this paper, design of the intelligent vehicle system with MK10DN512VLQ10 micro controller as the core control unit, based on the COMS camera OV7725 image sampling circuit of image information, to extract the centerline of the track calculated between the car and black line position deviation, of the steering gear steering by way of PD feedback control.Using PID control algorithm to adjust the speed of the drive motor, combined with the specific algorithm analysis information out of the track ahead of model car speed closed-loop control.

KEYWORDS: INTELLIGENT CAR, MK10DN512VLQ10, OV7725, PID

目录

第一章 系统总体设计

1.1 引言

“恩智浦” 杯全国大学生智能汽车竞赛是由教育部高等学校自动化专业教学指导分委员会主办全国大学生智能汽车竞赛。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。智能车竞赛涉及自动控制、模糊识别、传感技术、电子、电气、计算机、机械与汽车等多个学科，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，激发了大学生的创新思维，对于其实践、创新能力和团队精神的培养具有十分最重要的价值。

随着科学技术的日新月异，智能控制的应用越来越广泛。智能车技术依托于智能控制，前景广阔且发展迅速。全国大学生“恩智浦”杯智能汽车竞赛包括光电组、摄像头组和电磁组等等，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方案。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，积累了丰富的实践经验，为以后的学习工作打下了坚实的基础。

1.2 系统概述

智能车系统主要包括三个大部分,分别为车模总体机械结构、硬件电路系统、软件算法。智能车系统的总体工作模式为:由 CMOS 摄像头 OV7725 拍摄赛道图像，输出硬件二值化后的黑白图像，将图像信息输入到 MK10DN512VLQ10 微控制器，进一步对主要的赛道信息作出判断，从而确定最优路径；通过光电编码器来检测车速，并采用 MK10DN512VLQ10 的输入捕捉功能进行脉冲计算获得速度和路程；

舵机、电机均采用 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。

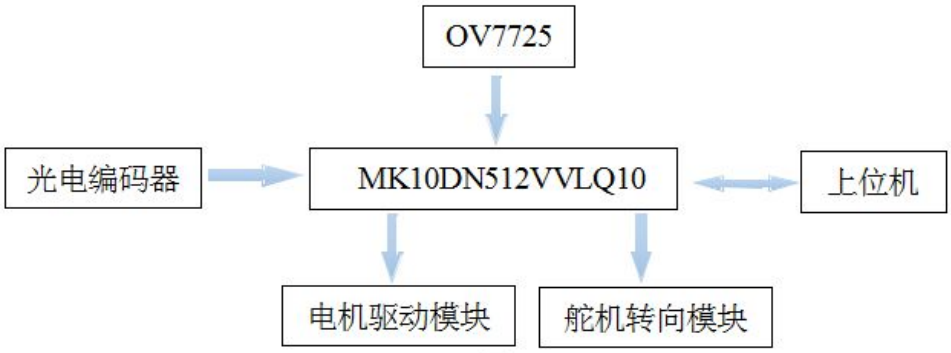


图 1.1

1.3 整车布局及机械系统设计

（1）利用碳纤杆做保险杠，避免碰撞前轮，从而保护舵机。

（2）舵机竖直放置，以提高舵机响应速度。

（3）用轻便坚固的碳纤杆作为摄像头杆的材料,下面用塑料底座固定，并用碳纤杆进行三角支撑，减小摄像头的晃动。

（4）摄像头安于电池的后方，有利于重心分布和盲区与前瞻的匹配。

图 1.2 整车布局图

为了提高智能车的竞技性能，必须了解其机械结构，并且对其结构上的不足之处加以改进。想要取得好的成绩，智能车底盘的优化和硬件设备的可靠性所占的比重不低于软件程序及其控制策略所占的比重。在比赛备战之初，我们就对该车模进行了详细的系统分析。今年的车模精度不是很高，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。

1.4 智能汽车前轮定位的调整

模型车通过四条轮胎与地面接触，两个后轮同轴受到限位，无法调整，与模型车的前进方向保持平行，因此要改变模型车与地面的接触方式，调试出利于模型车转向、直线的四轮定位，只能通过调整前轮各定位参数来实现。

B 型模型车可以调整的前轮参数有主销后倾角、主销内倾角、车轮前束，三个参数可以调整。

1.4.1 主销后倾角

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。

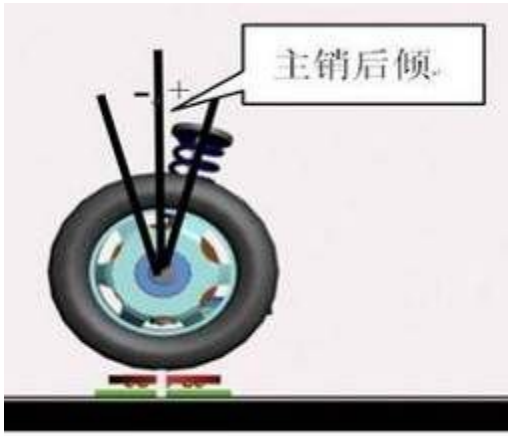


图1.3 主销后倾图

主销后倾角的存在使车轮转向轴线与赛道的交点在轮胎接地点的前方，可利用赛道对轮胎的阻力产生绕主销轴线的回正力矩，该力矩的方向正好与车轮偏转方向相反，使模型车保持直线行驶。后倾角越大，模型车的直线行驶性越好，转向后方向盘的回复性也越好，但过大的回正力矩也会使车辆转向沉重。通常主销后倾角值设定在 1° 到 3° 。B 型模型车的主销后倾角无法通过直接调整前桥结构实现改变，采用在前桥处增加垫片，可以适当的增加主销内倾角，有利于保持直线行驶、转向后回正。

1.4.2 主销内倾角

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。主销内倾角的作用，是使车轮在受外力偏离直线行驶时，前轮会在重力作用下自动回正。另外，主销内倾角还可减少前轮传至转向机构上的冲击，并使转向轻便；但内倾角越大，前轮自动回正的作用就越强，转向时越费力，轮胎磨损也更大增大。主销内倾的调整应该保持在一个合适的范围，“一般来说 0~8 度范围内皆可”。在实际的调整中，只要将角度调整为 5 度左右就会对于过弯性能有明显的改善。如果赛道比较滑，可以将这个角度再调节的大一些。在实际制作中，这个角度调节为 8 度左右。

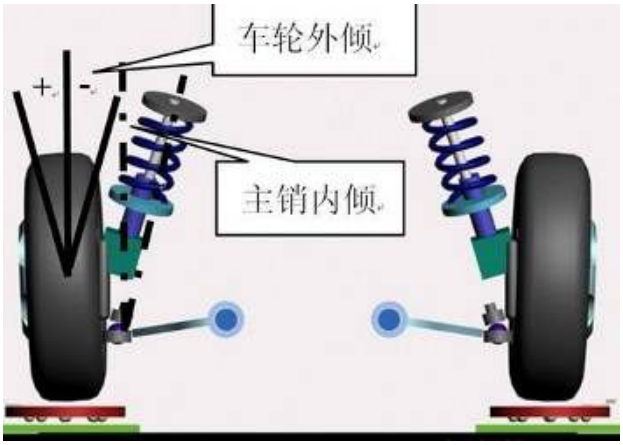


图1.4 主销内倾角

对于模型车，通过调整前桥的螺杆的长度可以改变主销内倾角的大小，由于过大的内倾角也会增大转向阻力，增加轮胎磨损，所以在调整时可以近似调整为0°~3°左右，不宜太大。主销内倾和主销后倾都有使汽车转向自动回正，保持直线行驶的功能。不同之处是主销内倾的回正与车速无关，主销后倾的回正与车速有关，因此高速时主销后倾的回正作用大，低速时主销内倾的回正作用大。

1.4.3 前轮约束

所谓前束是指两轮之间的后距离数值与前距离数值之差，前轮中心线与纵向中心线的夹角为前束角。前轮前束的作用是保证模型车的行驶性能，减少轮胎的磨损。前轮在滚动时，其惯性力自然将轮胎向内偏斜，如果前束适当，轮胎滚动时的偏斜方向就会抵消，轮胎内外侧磨损的现象会减少。前束的调整总是依据主销内倾的调整。只有主销内倾确定后才能确定合适的前轮前束与之配合。前轮前束的调整是方便的。主销内倾的调整由于要拧开螺丝钉，固定件又为塑料，所以频繁的调整容易引发滑丝现象。而前束不会，所以调整前束是最安全、方便的。前束在摩擦大的时候有明显的效果。但是一定不要太大，适当的放开一两圈就够了。

在模型车中，前轮前束是通过调整伺服电机带动的左右横拉杆实现的。主销在垂直方向的位置确定后，改变左右横拉杆的长度即可以改变前轮前束的大小。在实际的调整过程中，我们发现较小的前束，约束 0~2mm 可以减小转向阻力，使模型车转向更为轻便，但实际效果不是十分明显。调节合适的前轮前束在转向时有利过弯，还能提高减速性。将前轮前束调节成明显的内八字，运动阻力加大，

提高减速性能。由于阻力比不调节前束时增大，所以直线加速会变慢。 智能汽车采用稳定速度策略或者采用在直道高速弯道慢速的策略时，应该调节不同的前束。后一种策略可以适当加大前束。

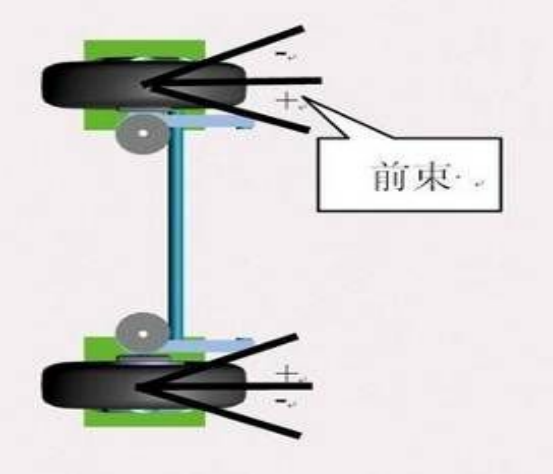


图 1.5 前束图

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

1.5 左右不对称问题的发现与解决

在调试的过程中，我们的小车遇到了一个比较严重的问题，就是在很长一段时间内，我们的车处于左右不对称的情况。也就是说，在左转弯，右转弯的时候，总是一个切弯，一个不切弯，这个问题困扰了我们相当长的时间。但是，凭借我们队员的坚持不懈的精神，在调节转向机构的内外倾角与增减垫片之后，又精确的调节舵机连杆的距离，做到多处高度对称，并且对左右轮胎做了处理之后，刚好解决了这个问题。

1.6 后轮差速片松紧度的调整

对于后轮差速，我们经过摸索总结以下经验：差速片外面的螺丝不能太松，太松容易导致直道摆动严重。但也不能太紧，太紧不利于转弯。是否合适是通过观察车在跑道上的状态来确定。

1.7 摄像头的安装

为了确保摄像头能稳定循迹而不晃动，摄像头底座的牢固性是关键。我们的方案是用塑料底座加碳纤杆做支撑对摄像头起稳定作用。由于车模底板硬度不够，我们用铝板作为摄像头的底部支撑，以减小摄像头的晃动。摄像头安装的位置与车模的前瞻量以及视野宽度也有直接关系，摄像头的安装位置低了，视域不够广阔，影响寻线的有效范围；安装的高了，整车系统会因重心抬高而稳定性变差。 所以摄像头安装的位置应同时考虑到机械性能的需要和图像的要求。摄像头传感器重量相对车架来说具有不可忽略的影响，从车模的性能上考虑，车模的重心越低越好。

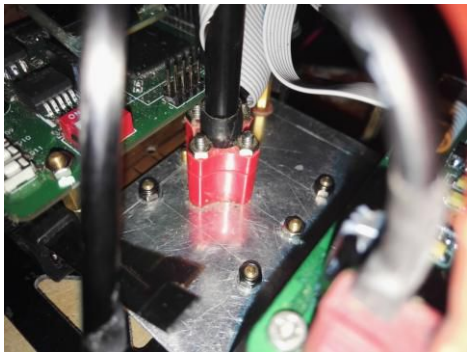


图 1.6(1) 摄像头底座

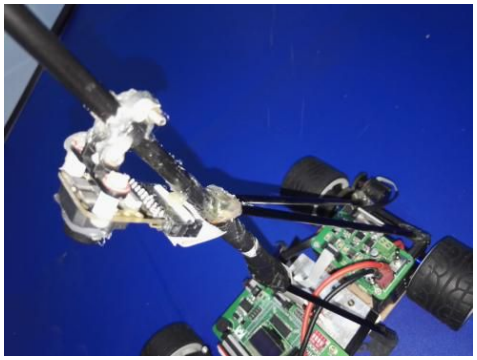


图 1.6(2) 摄像头安装

1.8 舵机安装

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。我们可以利用实际参数进行计算，得出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支架对其固定以及舵机连片(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，最终安装效果如图 1.7 所示。

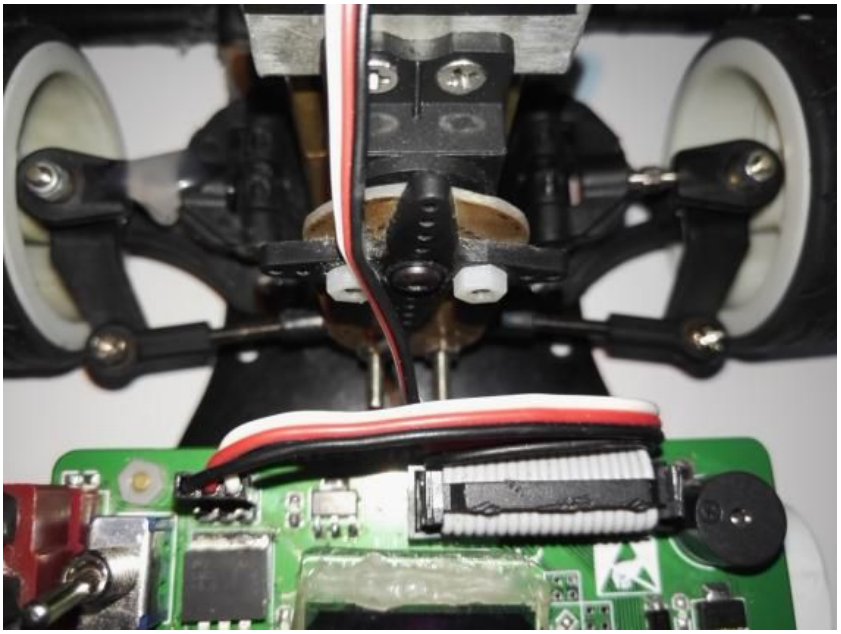


图 1.7 舵机安装方式

1.9 编码器的安装

选用编码器进行速度的测量。根据编码器的形状，我们自制了一个支架，速度传感器用螺钉通过支架固定在后轮支架上，这样固定好之后，就有了较高的稳定性。然后调节编码器齿轮，使其与差速齿轮紧密咬合，增大测速的精确性，但是咬合过紧也增大了摩擦，减小了对电机做功的利用率，影响小车的快速行驶，因此减小摩擦同时增强齿轮间的咬合是我们主要考虑的因素。如图 1.8 所示。

图 1.8 编码器的安装

1.10 本章小结

本章主要介绍了小车的安装制作和调试过程中机械方面的具体的问题。我们从开始比赛以来，一直坚持机械结构和算法是同样重要的原则，在更新算法的同时也在提高相应的机械和硬件结构来适应。上面的介绍是我们最稳定的小车版本的经验。而在刚开始做小车到现在，对机械结构方面我们做了许许多多各种各样的尝试和实验，收获了一些效果，但是目前这版是能够适应算法并且最稳定的机械结构系统。

第二章 硬件系统设计及实现

赛车共包括六大模块：MK10DN512VLQ10 主控模块、传感器模块、电源模块、电机驱动模块、速度检测模块和辅助调试模块。

各模块的作用：

MK10DN512VLQ10 主控模块：智能车系统以 MK10DN512VLQ10 为控制核心，将采集摄像头、编码器等传感器的信号，根据控制算法做出控制决策，驱动直流电机和舵机完成对智能汽车的控制并实现了单片机硬件的最优化设计和单片机资源的合理化使用。

摄像头模块：可以通过一定的前瞻性，提前感知前方的赛道信息，为智能汽车做出决策提供必要的依据和充足的反应时间。

电源管理模块：为整个系统提供合适而又稳定的电源。

电机驱动模块:驱动直流电机和伺服电机完成智能汽车的加减速控制和转向控制。

速度检测模块：检测反馈智能汽车轮的转速，用于速度的闭环控制。

辅助调试模块：主要用于智能汽车系统的功能调试、赛车状态监控。

2.1 MK10DN512VLQ10 主控模块

MK10DN512VLQ10 是 K10 系列 MCU。Kinetis 系列微控制器是 Cortex-M4 系列的内核芯片。K10 内存空间可扩展，从 32 KB 闪存/ 8 KB RAM 到 1 MB 闪存 / 128 KB RAM，可选的 16 KB 缓存用于优化总线带宽和闪存执行性能。

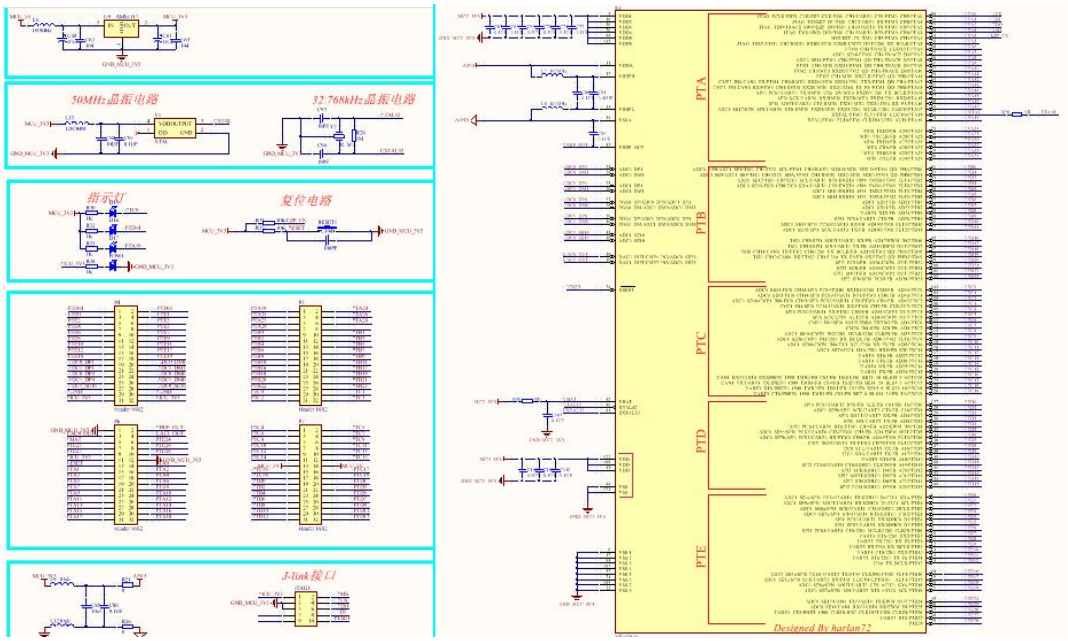
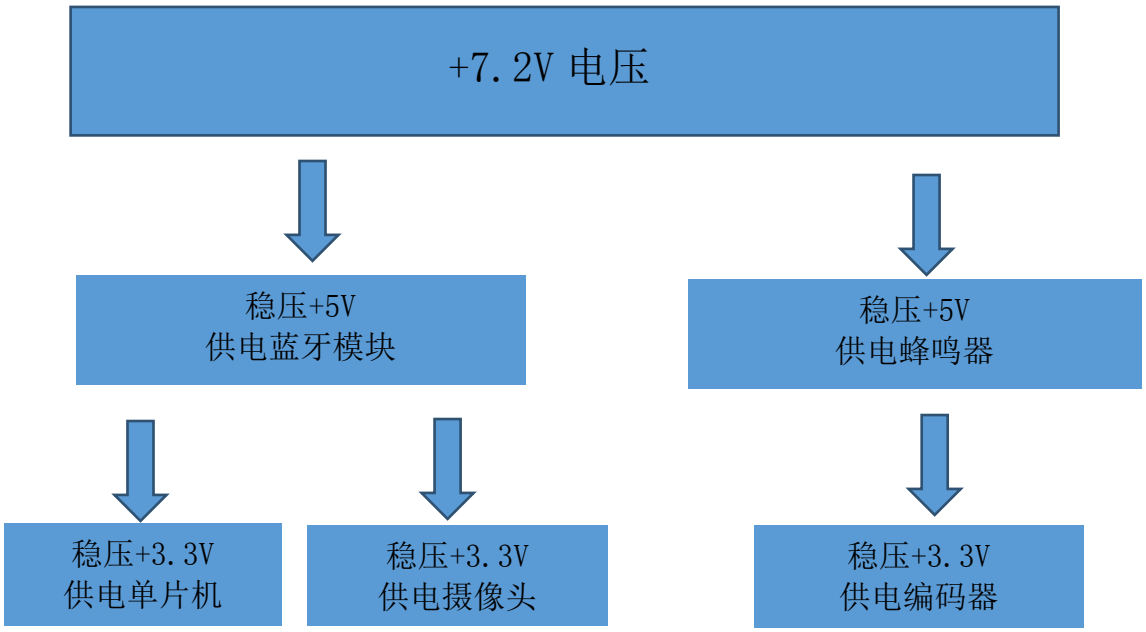


图 2.1 主控原理图

2.2 电源管理模块

本系统中电源稳压电路分别需要有+5V，+3.3V，+6V，+12V 供电。+3.3V 给单片机，摄像头,编码器；+5V 为蓝牙模块供电，蜂鸣器；+6V 给舵机供电；+12V 给电机驱动电路中 IR2104 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 LM2940。

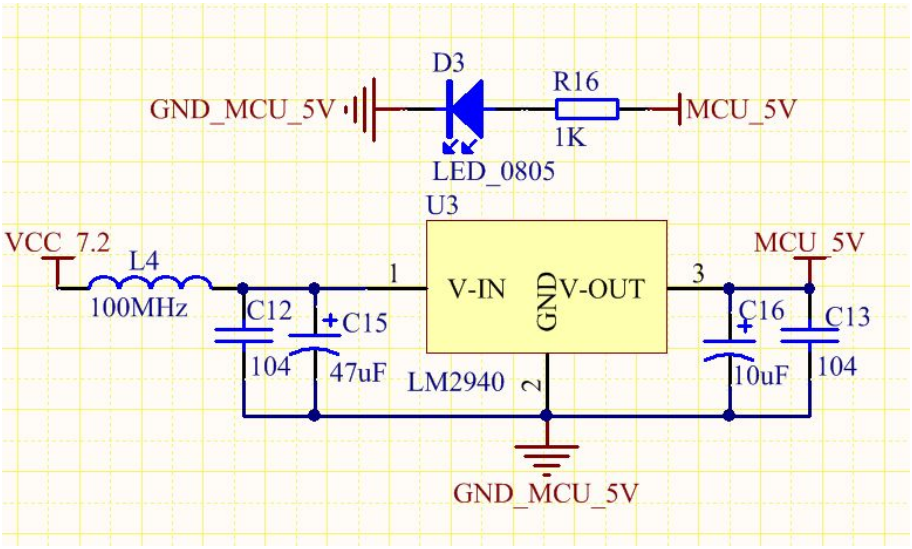


图 2.2　LM2940 原理图

舵机用 LM2941 进行供电，通过电阻值的合理调整，使用+6V 的电压给舵机供电。较高的电压可以提高舵机的响应速度，但过高电压容易导致舵机工作不稳定。舵机电源的稳压电路原理图如图 2.3 所示。

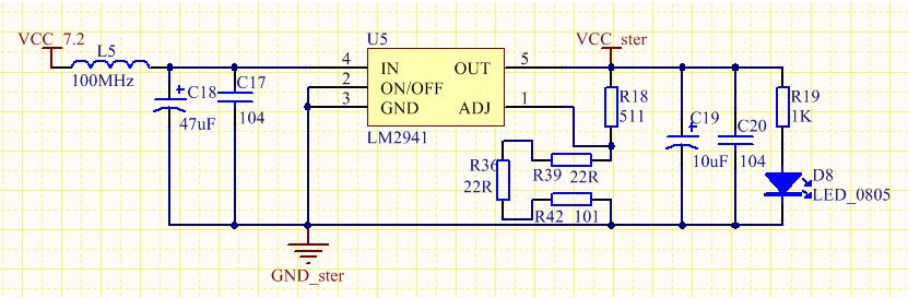


图 2.3　LM2941 原理图

2.3 摄像头模块

2.3.1 摄像头简介

摄像头分黑白和彩色两种，为达到寻线目的，只需提取画面的灰度信息，而不必提取其彩色信息，所以我们所使用的 CMOS 摄像头输出的信号为黑白视频信号。我们所选用的摄像头芯片为 OV7725，1/3 英寸数字式 CMOS 图像传感器 OV7725，

总有效像素单元为664(水平方向)×492(垂直方向)像素;内置10位双通道A/D转换器,输出8位图像数据;具有自动增益和自动白平衡控制,能进行亮度、对比度、饱和度、γ校正等多种调节功能;其视频时序产生电路可产生行同步、场同步、混合视频同步等多种同步信号和像素时钟等多种时序信号;5V电源供电,工作时功耗<120mW,待机时功耗<10μW。可应用于数码相机、电脑摄像头、可视电话、第三代网络摄像机、手机、智能型安全系统、汽车倒车雷达、玩具，以及工业、医疗等多种用途。OV7620是1/3”CMOS彩色/黑白图像传感器。它支持连续和隔行两种扫描方式，VGA与QVGA两种图像格式；最高像素为664×492，帧速率为30fps；数据格式包括YUV、YCrCb、RGB三种，能够满足一般图像采集系统的要求。

2.3.2 摄像头的工作原理

摄像头的工作原理是：按一定的分辨率，扫描的方式采集图像上的点，当扫描到某点时,就通过图像传感芯片将该点处图像的灰度转换成与灰度一一对应的电压值,然后将此电压值通过视频信号端输出。摄像头连续地扫描图像上的一行，则输出就是一段连续的电压信号,电压信号的高低起伏反映了该行图像的灰度变化。当扫描完一行，视频信号端就输出一个低于最低视频信号电压的电平（如0.3V)，并保持一段时间。这相当于，紧接着每行图像信号之后会有一个电压“凹槽”，此“凹槽”叫做行同步脉冲，它是扫描换行的标志。然后，开始扫描新的一行，如此下去，直到扫描完该场的视频信号，接着会出现一段场消隐区。该区中有若干个复合消隐脉冲，其中有个远宽于（即持续时间远长于）其它的消隐脉冲，称为场同步脉冲，它是扫描换场的标志。场同步脉冲标志着新的一场的到来，不过，场消隐区恰好跨在上一场的结尾和下一场的开始部分，得等场消隐区过去，下一场的视频信号才真正到来。

2.3.3 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，因为小车对图像的清晰度，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。

OV7725，BF3003 比较，OV7725 具有成像稳定，技术成熟等优点，于是我们选用了摄像头 OV7725。经试验证明，虽然 OV7620 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，能够实现动态调节，但是 OV7620 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 OV7725，所以我们选用了鹰眼的硬件二值化摄像头 OV7725 来采集图像，效果颇好。

OV7725 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 640 480，有效光敏面为 312×215 像素，电源是 3.3 V(可兼容 5V)，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 30 帧，电视扫描线为 625 线，奇场在前，偶场在后。

我们采用集成摄像头模块，其优良的性能能很好地满足要求，如图 2.4（1）

图 2.4（1）摄像头的安装

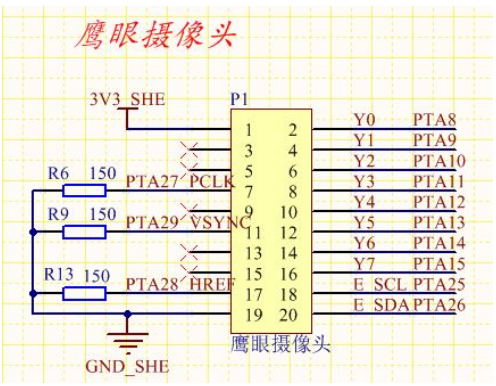


图 2.4（2）摄像头原理图

2.4 电机驱动模块

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。

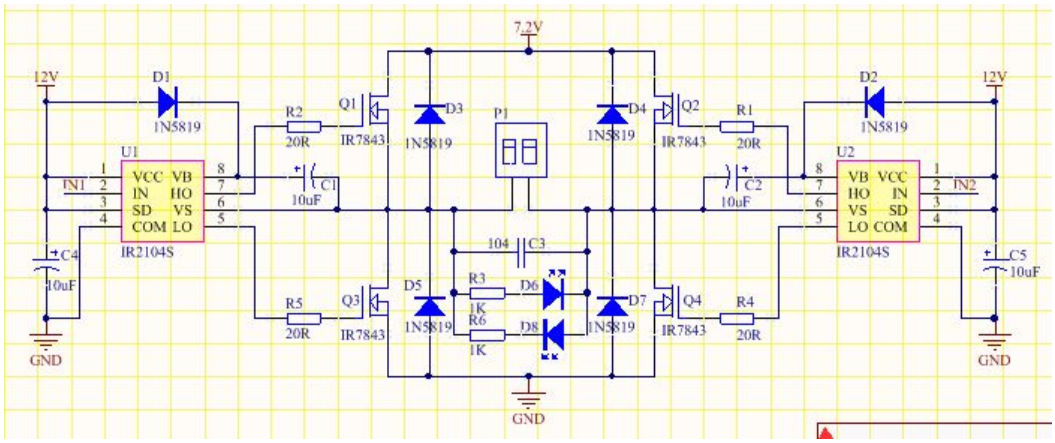


图 2.5 电机驱动模块原理图

2.5 测速模块

本小车使用编码器由 AM1117 为其提供 3.3V 工作电压。处理器通过读取编码器脉冲数来实现小车速度的检测，通过读取编码器旋转方向脚的高低电平来检测电机的正反转。

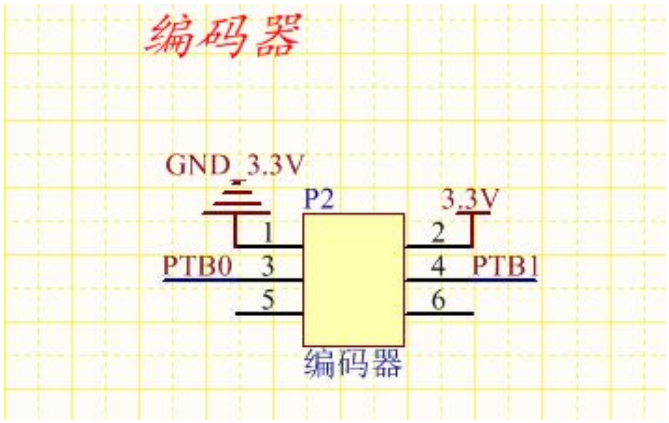


图 2.6 编码器原理图

2.6 辅助调试模块

2.6.1 无线调试蓝牙模块及蓝牙上位机

蓝牙信号的收发采用蓝牙模块实现，具有片内数字无线处理器 DRP（Digital Radio Processor）、数控振荡器，片内射频收发开关切换，内置 ARM7 嵌入式处理器等。接收信号时，收发开关置为收状态，射频信号从天线接收后，经过蓝牙收发器直接传输到基带信号处理器。基带信号处理包括下变频和采样，采用零中频结构。数字信号存储在 RAM（容量为 32KB）中，供 ARM7 处理器调用和处理，ARM7 将处理后的数据从编码接口输出到其他设备，信号发过程是信号收的逆过程，此外，还包括时钟和电源管理模块以及多个通用 I/O 口，供不同的外设使用。主机接口可以提供双工的通用串口，可以方便地和 PC 机的 RS232 通信。 除此之外，我们还编写了蓝牙上位机，直接可以通过蓝牙将图像发到手机上，方便实时地看图像，省得每次都要用电脑发串口才能看图像。

图 2.7 无线模块

2.6.2 按键、拨码开关、OLED 调试

为了减轻负担方便现场调试我们加入了按键、拨码开关、OLED。大大的节省了我们时间，避免了经常用电脑下程序的麻烦。调节按键、拨码开关用 OLED 来显示各个参数。也方便我们观看图像和各项参数。

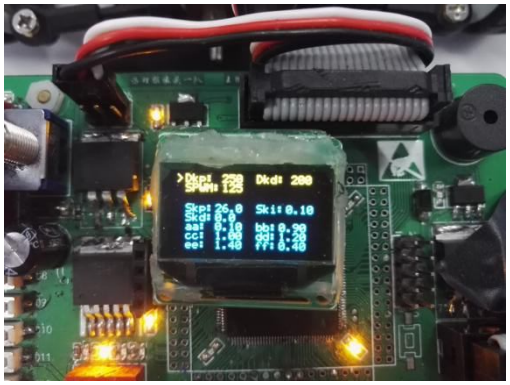


图 2.8 OLED 显示屏

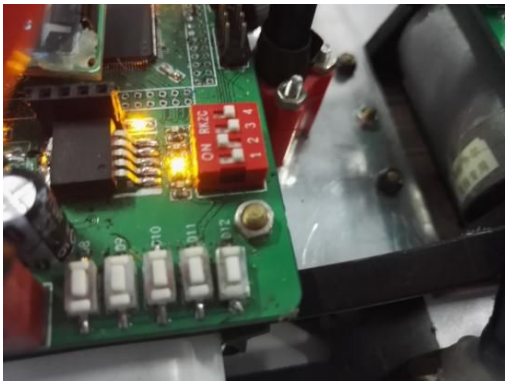


图 2.9 按键及拨码开关

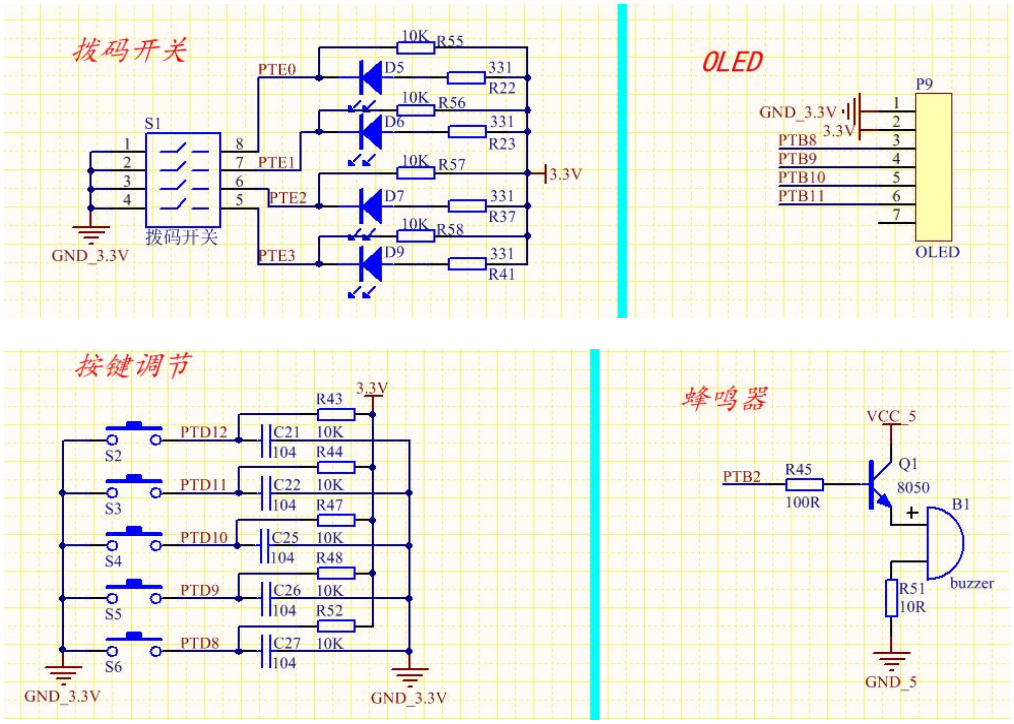


图 2.10 辅助调试模块原理图

第三章 软件系统设计

3.1 软件设计的总体思路

用摄像头采集原始图像，我们用的是 60 行 160 列的图像，像素高能更好的采集出赛道的信息，硬件二值化得到我们想要的图像。寻找出有效的起始行，找到起始行后小范围内寻找左右边线，寻不到的边缘线进行补线，后算出中线，进而来控制舵机打角，速度用编码器有周期的反馈，来控制直道、弯道的速度。

3.2 图像采集

主要用鹰眼 ov7725 摄像头、野火硬件二值化摄像头，速率可达 150 帧每秒，去噪点能力极强，二值化效果非常理想。

主要的采集思路分为以下四点：

1. 需要采集图像时，开场中断

2. 场中断来了，初始化 DMA 传输，并启动 DMA 传输

3. 每个 PCLK 上升沿来了都触发 DMA 传输，把摄像头输出的值读取到内存数组里。当触发 n 次（n=图像像素数目）后就停止 DMA 传输。

4.DMA 停止传输时触发中断，中断里关闭场中断，图像采集完毕。或者等待下一个场中断来临才关闭场中断，标记图像采集完毕

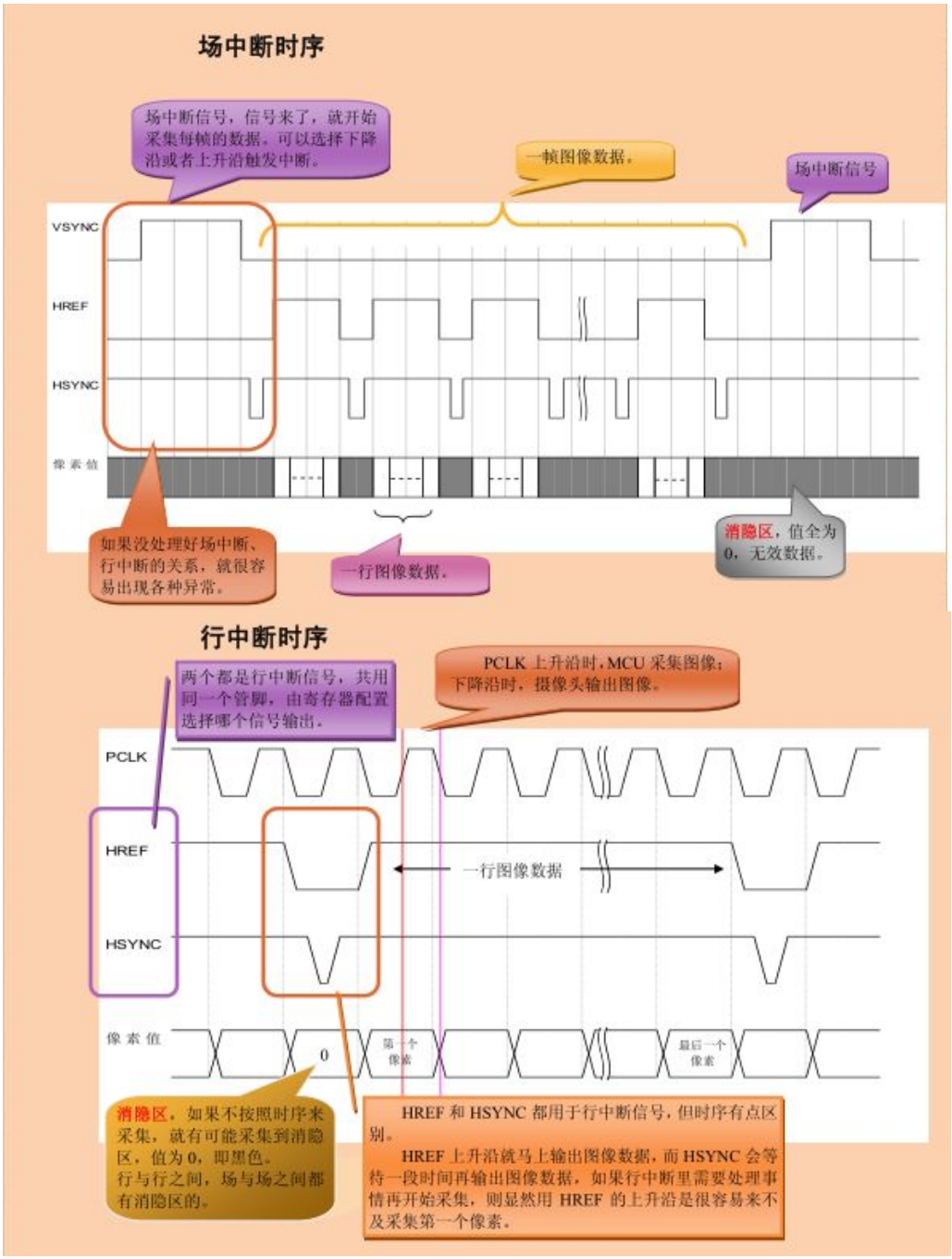


图 3.1 摄像头采集图像时序图

3.3 边线提取与补线

用摄像头采集原始图像，我们用的是 60 行 160 列的图像，像素高能更好的

采集出赛道的信息，硬件二值化之后得到我们需要的图像。寻找出有效的起始行，最后一行从中间向俩边大范围寻线，找到跳变点后记录下来，在最后一行的基础上增加小范围的列数寻找倒数第二行的跳变点，以此类推找到左右边线。寻不到边缘线以下几行同样的趋势进行补线处理，让每一场图像都寻到左右边缘线，根据左右边缘线算出中线。下面图像为原始图像和处理后的图像对比。

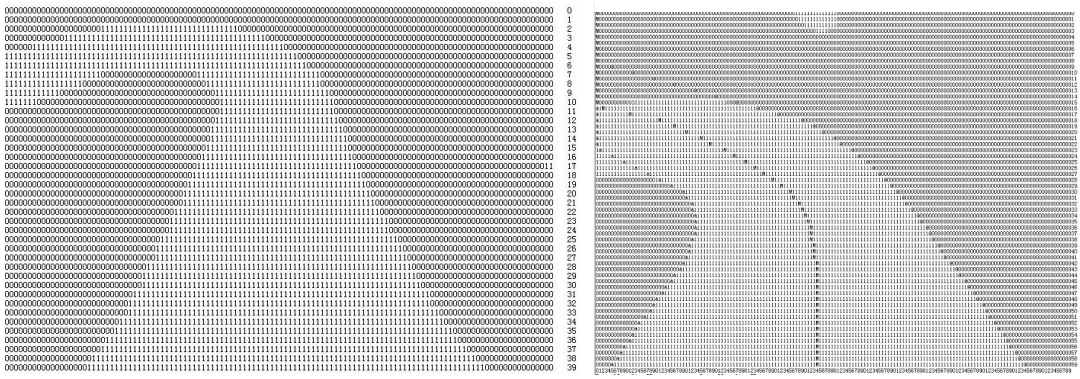


图 3.2（1）原始图像　　图 3.2（2）处理后图像

3.4 PID 控制算法介绍

PID 控制是工业过程控制中历史最悠久，生命力最强的控制方式。这主要是因为这种控制方式具有直观、实现简单和鲁棒性能好等一系列的优点。PID 控制主要有三部分组成，比例、积分、微分。

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。偏差一旦产生，调节器立即产生控制作用使被控量朝着减小偏差的方向变化，控制作用的强弱取决于 KP。当仅有比例控制时系统输出存在稳态误差（Steady-state error）。

为了消除稳态误差，引入积分控制。积分项对误差取决于时间的积分，随着时间的增加，积分项会增大。这样，即便误差很小，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。

为了预测预测误差变化的趋势，引入微分的控制器，这样就能够提前使抑制误差的控制作用等于零，甚至为负值，从而避免了被控量的严重超调。PID 控制框图如图 3.3 所示。

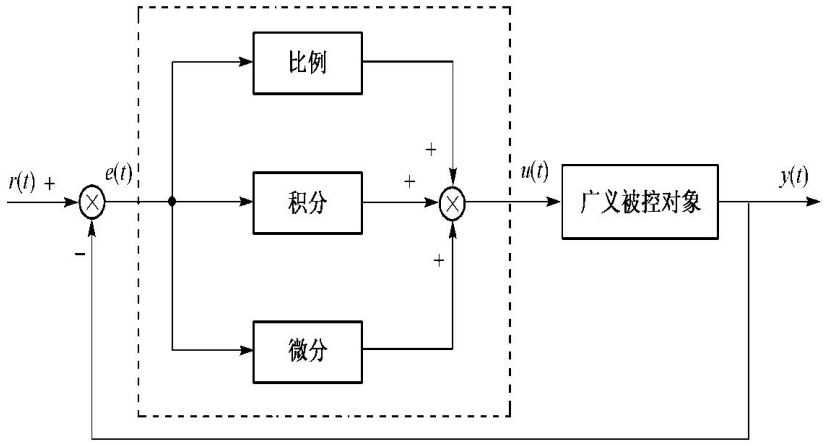


图 3.3 PID 控制框图

对应的误差传递函数为：

$$\frac{U(s)}{E(s)}=K_p(1+\frac{1}{T_i}+T_d)$$ （公式一）

式中，Kp 为比例增益；Ti 为积分时间常数；Td 为微分时间常数；U(s)为控制量；E(s) 为被控量与设定值 R(s)的偏差。

时域表达式为：

$$u(t)=K_{\mathrm{p}}\left[e(t)+\frac{1}{T_{\mathrm{i}}}\int_0^t e(t)\mathrm{d}t+T_{\mathrm{d}}\frac{\mathrm{d}e(t)}{\mathrm{d}t}\right]$$ （公式二）

在单片机中，我们仅能对数字信号处理，即数字 PID 控制。将上式离散化，得

$$u(k)=K_p\left\{e(k)+\frac{T}{T_i}\sum_{j=0}^{k}e(j)+\frac{T_d}{T}[e(k)-e(k-1)]\right\}$$ （公式二）

3.4.1 位置式 PID

直接利用上述离散化公式计算，框图如右图所示。由于积分项（Pi）是将所有采集值偏差相加，在一段时间后会很浪费单片机资源。对其稍加改进，得到增量型 PID 算法。

3.4.2 增量式 PID

根据式二得第 k-1 个采样周期的控制量为

$$u(k-1)=K_p\left\{e(k-1)+\frac{T}{T_i}\sum_{j=0}^{k-1}e(j)+\frac{T_d}{T}[e(k-1)-e(k-2)]\right\}$$ （公式三）

式二减式三得

$$u(k)=K_p[e(k)-e(k-1)]+K_i*e(k)=K_d[e(k)-2e(k-1)+e(k-2)]$$ （公式四)

由此，第 k 个采样时刻实际控制量为$u(k)=u(k-1)+\Delta u(k)$,为方便书写，写为

$$\Delta u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)$$ （公式五）

其中，$q_0 = K_p(1 + \frac{T}{T_i} + \frac{T_d}{T})$，$q_1 = -K_p(1 + \frac{2T_d}{T})$，$q_2 = K_p \frac{T_d}{T}$。

由上可知，利用三个历史数据，递推使用，即可完成 PID 控制量。

3.4.3 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。

3.5 转向舵机的 PID 控制算法

舵机采用位置式 PID 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，经过一段时间的调试发现，只用 PD 两个参数便可以达到较稳定控制效果，分析后发现确实可以将 I 项置零，因为对于舵机，需要的是快速的反应，迅速的较准确的打角，即便存在一些静态误差也没有大的问题，通过调整 PD 两个参数，小车在直道也不会震荡。经过试验，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。

3.6 驱动电机的 PID 控制算法

速度 PID 采用增量式控制算法，事实上，对于速度控制 PID 算法，尝试了位置式和增量式两种，在 B 车上的表现并没有很大的差别，还尝试了 PID 和 bang-bang 结合的控制方法，bang-bang 控制的优点就是最大程度加快系统的调节速度，PID 控制进行精细调节，因为调试时间较短，调节死区和参数的选择没

有达到最佳，最终选用了增量式 PID 算法，无论选用哪一种，都需要特殊处理，均不能简单的采用经典的 PID 控制，需要加入积分限幅和积分分离处理，最终用的增量式 PID 还用到了前馈处理，为了提高电机的反应速度。

3.7 舵机打角控制

我们根据采集的 60 行图像进行了加权处理，因为远处和近处的行与中间的行相比不重要。我们把 60 行分成了 6 段处理。通过实际中线与图像中线的偏差，找到适合自己小车的数分配，用试凑法确定权值和舵机的 PD 值。（一定要有舵机保护，限制舵机的最大和最小值）。

部分代码如下：

```
Sum=0;
    for(temp_x = 0;temp_x < 60;temp_x++)
  {
    if(temp_x<13)
      Sum += aa*(float)(Middle[temp_x]- center);
  //差值不加权平均沿着中线   90~180
    else if(temp_x<17)
      Sum += bb*(float)(Middle[temp_x]- center);
    else if(temp_x<22)
      Sum += cc*(float)(Middle[temp_x]- center);
 //差值不加权平均沿着中线跑     40~90
    else if(temp_x<30)
      Sum += dd*(float)(Middle[temp_x]- center);            //       15~40
    else if(temp_x<39)
      Sum += ee*(float)(Middle[temp_x]- center);
    else
```

```
            Sum += ff*(float)(Middle[temp_x]- center);
        }
        Aver = Sum/60;
        referlast = refer;
refer    = Aver;
Corner = (int)((TurnKd*10 * (refer - referlast)) + (TurnKp * refer));
        value    = middle_limit +Corner    ;
        if(value < left_limit)    value = left_limit;          //舵机转角保护
        else if(value > right_limit)    value = right_limit; //舵机转角保护
        ftm_pwm_duty(FTM0, FTM_CH6, value);
         //设置 FTM0_CH6 占空比为 10/FTM0_PRECISON
```

3.8 速度闭环控制

PID 控制策略其结构简单，稳定性好，可靠性高，并且易于实现。

在本方案中，使用试凑法来确定控制器的比例、积分和微分参数。试凑法是通过闭环试验，观察系统响应曲线，根据各控制参数对系统响应的大致影响，反复试凑参数，以达到满意的响应，最后确定 PID 控制参数。试凑不是盲目的，而是在控制理论指导下进行的。我们利用虚拟示波器输出图像，通过图像变化趋势，来确定合适的参数。在控制理论中已获得如下定性知识：

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 Ti，Ti 越小，积分作用就越强。反之 Ti 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。

我们还加入了分离积分 PID 控制算法和抗积分饱和 PID 控制算法,使小车能够快速的响应。达到快速加速和迅速减速的目的。

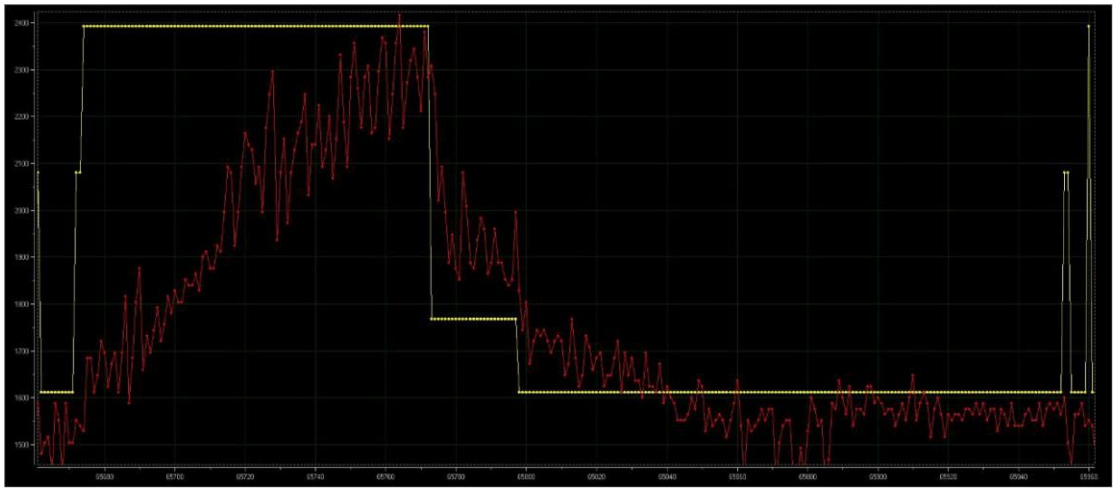


图 3.4 实际速度与理论速度的波形（黄线为理论速度，红线为实际速度）

3.9 软件程序总结

1.准确的寻找出左右边线，来求出中线。

2.通过实际中线与图像中线的偏差，找到适合自己小车的数分配，用试凑法确定权值和舵机的 PD 值。

3.速度控制 PID 选取和参数的整定使车快速的响应所给的理论速度。

4.使速度控制完美配合好舵机的打角。

第四章 智能车调试说明

4.1 软件开发工具

程序的开发是在组委会提供的 IAR Embedded Workbence 下进行的，包括源程序的编写、编译和链接，并最终生成可执行文件。 IAR C 语言开发软件 由 IAR 公司开发的第三方 MSP430 开发环境，是一个新的专业化集成开发环境，用来编辑，编译和调试 Windows 9x/NT/2000/XP 环境下的 MSP430 应用程序。还包含一个汇编器和一个仿真器。

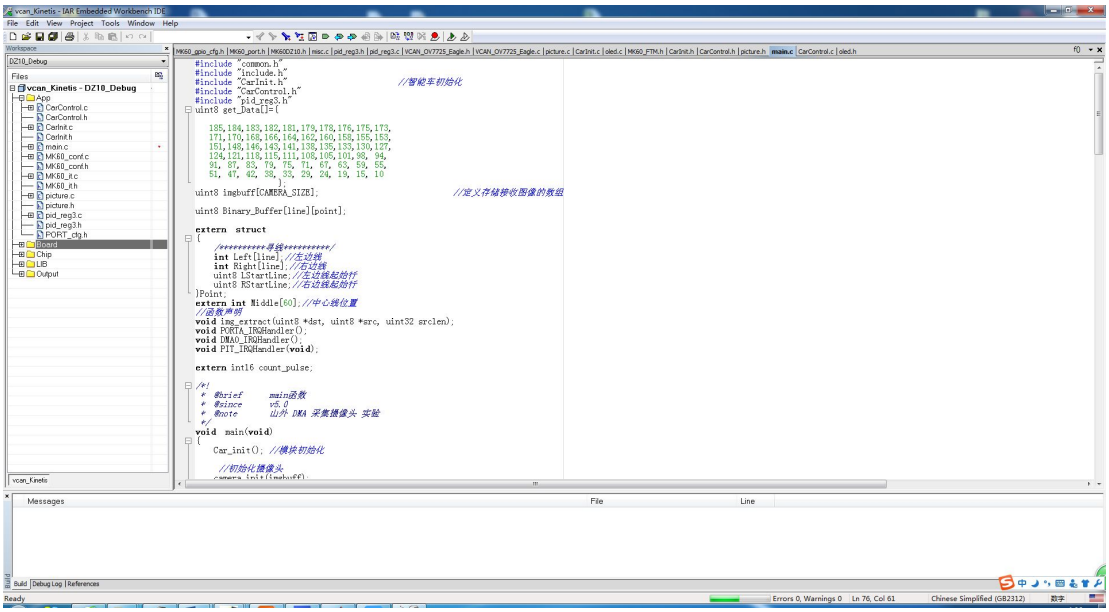


图 4.1 IAR 工作界面

4.2 上位机调试软件的设计

上位机是指可以直接发出操控命令的计算机，一般是 PC/host computer/master computer/upper computer, 屏幕上显示各种信号变化(液压，水位，温度等)。我们通过用串口调试助手来显示图像，找出我们在图像处理上的不足。

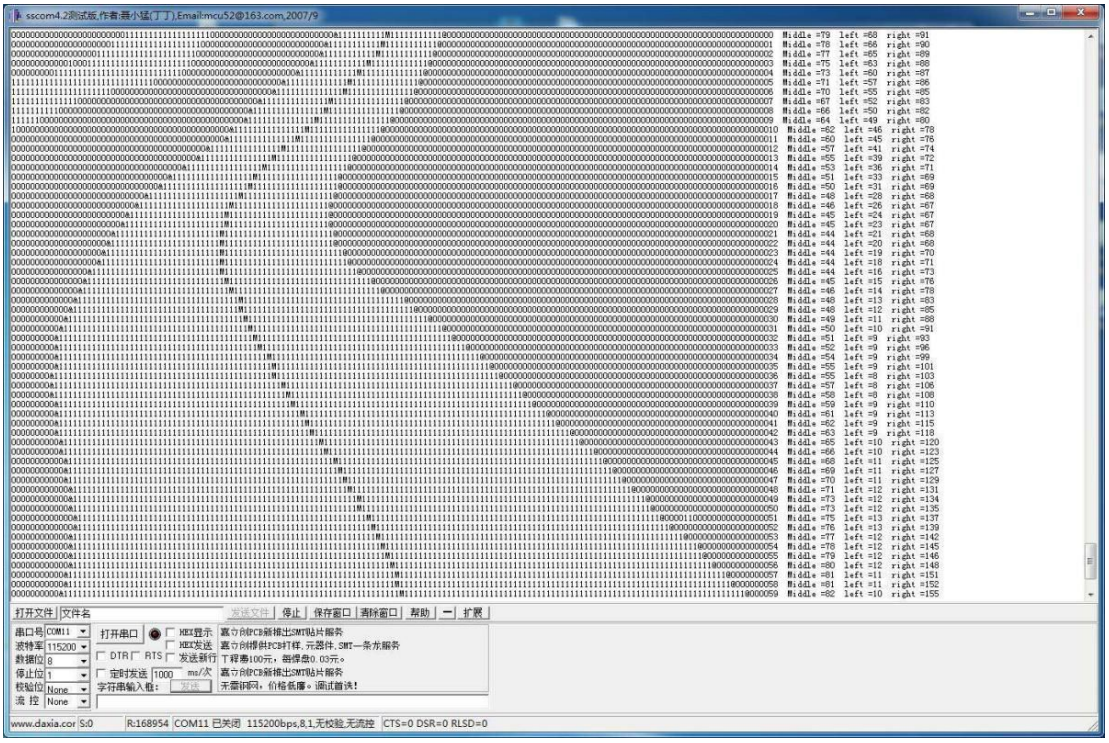


图 4.2 串口调试助手界面

第五章 模型车的主要技术参数

赛车基本参数	长	30cm
	宽	18cm
	高	35cm
车重		1321g
功耗	空载	10W
	带载	大于 12W
电容总容量		1700uF
传感器	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）	3/150cm
	精度(近/远)	2/12.5mm
	频率	150Hz

第六章 致谢

自报名参加“恩智浦”杯智能汽车竞赛以来，我们小组成员从查找资料、设计机构、组装车模、编写程序一步一步的进行，最后终于完成了最初目标，定下了现在这个设计方案。

在此份技术报告中，我们主要介绍了准备比赛时的基本思路，包括机械、电路以及最重要的控制算法的创新思想。在机械结构方面，我们分析了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。

在为本次大赛制作智能车期间，我们遇到过很多问题，从最初的传感器选型与方案确定，到后来的软硬件联合调试。在解决一个个问题之后，我们发现，我们技术上在不断成长，思想上不断成熟。而在这过程中，离不开学校，老师和同学的支持。首先，我们要感谢学校对这次比赛的重视，感谢学校教务处对我们比赛的大力支持。没有他们的支持，我们的车模绝对上不了赛场。其次，我们要感谢指导老师的悉心教导。没有他们在思想上的指导和整体的规划安排，我们很难有今天的成果。最后，我们要感谢同实验室的其他同学，感谢我们同时工作在这么和谐的实验室。感谢他们在赛道制作和其它方面的帮助。

现在，面对即将到来的大赛，在历时近十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) 田书林等．电子测量技术．北京．机械工业出版社．2012.
(2) Mark I. Montrose．电磁兼容的印制电路板设计．北京．机械工业出版社．2012.
(3) 余志生．汽车理论．北京．机械工业出版社．2012.
(4) 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.
(5) 阎石．数字电子技术基础［M］．北京：高等教育出版社，2000.
(6) 谭浩强著．C 程序设计．北京：清华大学出版社，2003.
(7) 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.
(8) 夏克俭．数据结构及算法［M］．北京：国防工业出版社， 2001.
(9) 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.
(10) 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.
(11) 刘金琨.先进 PID 控制 MATLAB 仿真（第 3 版）.北京：电子工业出版社，2011.3

附录：程序源代码

```
#include "common.h"
#include "include.h"
#include "CarInit.h"                                    //智能车初始化
#include "CarControl.h"
#include "pid_reg3.h"
uint8 get_Data[]={

    185,184,183,182,181,179,178,176,175,173,
    171,170,168,166,164,162,160,158,155,153,
    151,148,146,143,141,138,135,133,130,127,
    124,121,118,115,111,108,105,101,98, 94,
    91, 87, 83, 79, 75, 71, 67, 63, 59, 55,
    51, 47, 42, 38, 33, 29, 24, 19, 15, 10
                        };
uint8 imgbuff[CAMERA_SIZE];        //定义存储接收图像的数组
uint8 Binary_Buffer[line][point];

extern int Printf_Fdb;
extern int Printf_Ref;
extern int Printf_speed_pwm;
extern int Printf_Corner;
extern uint8 Stop;

extern   struct
{
     /**********寻线**********/
     int Left[line];//左边线
     int Right[line];//右边线
     uint8 LStartLine;//左边线起始行
     uint8 RStartLine;//右边线起始行
}Point;
extern int Middle[60];//中心线位置
//函数声明
void img_extract(uint8 *dst, uint8 *src, uint32 srclen);
void PORTA_IRQHandler();
void DMA0_IRQHandler();
void PIT_IRQHandler(void);
void Printf_Picture(void);
//示波器子函数
void VisualScope(void);
unsigned short CRC_CHECK(unsigned char *Buf, unsigned char CRC_CNT);
void sendware(uint8 *wareaddr, uint32 waresize);

extern int16 count_pulse;
/*!
 * *************主函数
```

```
 */
void    main(void)
{
    Car_init(); //模块初始化
      //初始化摄像头
    camera_init(imgbuff);
      //配置中断服务函数
    set_vector_handler(PORTA_VECTORn ,PORTA_IRQHandler);
      //设置 LPTMR 的中断服务函数为 PORTA_IRQHandler
    set_vector_handler(DMA0_VECTORn ,DMA0_IRQHandler);
      //设置 LPTMR 的中断服务函数为 PORTA_IRQHandler

      //定时器中断
    pit_init_ms(PIT0, 5);
      //初始化 PIT0，定时时间为： 1000ms

    ftm_quad_init (FTM1);
    set_vector_handler(PIT0_VECTORn ,PIT_IRQHandler);
       //设置 PIT0 的中断服务函数为 PIT_IRQHandler
    enable_irq (PIT0_IRQn);
       //使能 PIT0 中断

    while(1)
    {
        camera_get_img();
       //摄像头获取图像
        img_extract((uint8 *)Binary_Buffer, imgbuff,CAMERA_SIZE);
       //解压
        Edge();
        Midline();              //补线处理
        Car_position();      //舵机打角

        if(BUT3 == 0)
          Printf_Picture();      //串口发送图像到上位机
        if(BUT2 == 0)
          VisualScope();
    }
}
/*!
 * *************解压图像
 */
void img_extract(uint8 *dst, uint8 *src, uint32 srclen)
{
    uint8 colour[2] = {1, 0}; //0 和 1 分别对应的颜色
    //注：山外的摄像头 0 表示 白色，1 表示 黑色
    uint8 tmpsrc;
    uint8 mm=0,nn=0;
    uint8 src_H;
    while(srclen --)
    {
```

```
            tmpsrc = *src++;
            src_H = srclen/40;//一行 160 20 个字节一个发送 8 个字节

               if(src_H==get_Data[mm])
               {

           //*dst++ = colour[ (tmpsrc >> 7 ) & 0x01 ];
              *dst++ = colour[ (tmpsrc >> 6 ) & 0x01 ];
           //   *dst++ = colour[ (tmpsrc >> 5 ) & 0x01 ];
              *dst++ = colour[ (tmpsrc >> 4 ) & 0x01 ];
           //   *dst++ = colour[ (tmpsrc >> 3 ) & 0x01 ];
              *dst++ = colour[ (tmpsrc >> 2 ) & 0x01 ];
           //   *dst++ = colour[ (tmpsrc >> 1 ) & 0x01 ];
              *dst++ = colour[ (tmpsrc >> 0 ) & 0x01 ];//采集 120 列
                  nn++;
               }
                  if(nn==40)
                  {
                     mm++;
                     nn=0;
                  if(mm==60)
                       mm=0;
                  }
        }
}
/*!
 *   @brief          PORTA 中断服务函数
 *   @since          v5.0
 */
void PORTA_IRQHandler()
{     unsigned int h,w;
      extern uint8 x,y;
      uint8    n;       //引脚号
      uint32 flag;
      while(!PORTA_ISFR);
      flag = PORTA_ISFR;
      PORTA_ISFR    = ~0;                          //清中断标志位
      n =29;                                       //场中断
      if(flag & (1 << n))                          //PTA6 触发中断
      {
            camera_vsync();//鹰眼 ov7725 场中断服务函数  采集
      }
}
/*!
 *   @brief          DMA0 中断服务函数
 *   @since          v5.0
 */
void DMA0_IRQHandler()
{
      camera_dma();
}
```

```
/*!
 *   @brief          PIT0 中断服务函数
 *   @since          v5.0
 */
void PIT_IRQHandler(void)
{

      if(PIT_TFLG(PIT0) == 1 )             //判断是否 PIT0 进入中断
      {
                  /****编码器***********/
         count_pulse =    ftm_quad_get(FTM1);     //保存脉冲计数器计算值
         ftm_quad_clean(FTM1);
         //清空脉冲计数器计算值（开始新的计数）

         PIT_Flag_Clear(PIT0);//清中断标志位
         if(count_pulse>0)
         count_pulse=0;
         if(count_pulse<0)
         {
            count_pulse=-count_pulse;
         }
         speed_control();

//          printf("Point.RStartLine=%d",count_pulse);
//          printf("\r\n");
//          OLEDshowfive(97,1,count_pulse);
//          OLEDshowthree(0,6,max);
      }
}
/*!
 ****************************虚拟示波器
 */
unsigned short CRC_CHECK(unsigned char *Buf, unsigned char CRC_CNT)
{
            unsigned short CRC_Temp;
            unsigned char i,j;
            CRC_Temp = 0xffff;

            for (i=0;i<CRC_CNT; i++){
                        CRC_Temp ^= Buf[i];
                        for (j=0;j<8;j++) {
                                    if (CRC_Temp & 0x01)
                                                CRC_Temp = (CRC_Temp >>1 ) ^ 0xa001;
                                    else
                                                CRC_Temp = CRC_Temp >> 1;
                        }
            }
            return(CRC_Temp);
}
```

```
void VisualScope(void)
{
      unsigned char temp[10];
      unsigned int data[4];
      unsigned int crc;
      int i= 0;

         data[0]= (int)Printf_Fdb;
         data[1]= (int)Printf_Ref;
         data[2]= (int)Printf_speed_pwm;
//         data[0]= (int)Printfspeed_pwm;
//         data[1]= (int)Printfspeed_set;
//         data[2]= (int)RightRealValue*100;
//         data[3]= (int)RightRealValue;


      temp[0]=data[0];
      temp[1]=data[0]>>8;
      temp[2]=data[1];
      temp[3]=data[1]>>8;
      temp[4]=data[2];
      temp[5]=data[2]>>8;
      temp[6]=data[3];
      temp[7]=data[3]>>8;

      crc = CRC_CHECK(temp,8);
      temp[8]=crc;
      temp[9]=crc>>8;
        for(i=0;i<10;i++)
      {
//          UART_WriteByte(HW_UART5,temp[i]);
                  uart_putchar (UART3, temp[i]);    //发送字节'A'
      }
  }
/*!
 * ****************************山外多功能调试助手
 */
void sendware(uint8 *wareaddr, uint32 waresize)//虚拟示波器
{
   #define CMD 3
   uint8 cmdf[2] = {CMD, ~CMD};       //开头命令
   uint8 cmdr[2] = {~CMD, CMD};       //结尾命令
   uart_putbuff(UART3, cmdf, sizeof(cmdf));       //先发送命令
   uart_putbuff(UART3,wareaddr, waresize);         //再发送图像
   uart_putbuff(UART3, cmdr, sizeof(cmdr));       //再发送命令
}
/*!
 * ****************************上位机发送图像
 */
void Printf_Picture(void)
{
```

```
    unsigned int h,w;
    extern uint8 x,y;

        printf("\r\n");
        printf("                1               1               1               ");
        printf("V               1               1               1               ");
        printf("A               1               1               1               ");
        printf("V               1               1               1               ");
        printf("\r\n");

        for(h=0;h<60;h++)
        {

            for(w=0;w<160;w++)
             {
              if(w==Middle[h])
                        printf("M");

              else if(w==Point.Right[h+2])
                        printf("@");
              else if(w==Point.Left[h+2])
                        printf("&");
              else
                        printf("%d",Binary_Buffer[h][w]);

             }
              printf("%d",h);

             printf("    Middle =%d",Middle[h]);
             printf("    left =%d",Point.Left[h+2]);
             printf("    right =%d",Point.Right[h+2]);

                  printf("\r\n");
        }
        printf("Point.LStartLine=%d",y);
        printf("                        Point.RStartLine=%d",x);

}
```