

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： 上海电力学院

队伍名称： 电自二队

参赛队员： 周奇聪

黄明海

王琳博

带队教师： 杨 宁

黄福珍

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：周奇聪

黄明海

王琳博

带队教师签名：杨 宁

黄福珍

日 期：2012/8/13

摘 要

本文介绍了上海电力学院车模队伍在准备第七届“飞思卡尔”杯全国大学生智能车竞赛中设计的基于摄像头传感技术视觉引导的智能车控制系统。智能车的硬件平台采用带MC9S12XS128处理器的S12环境，软件平台为CodeWarrior IDE4.7 开发环境，车模采用大赛组委会统一提供的仿真车模。

文中介绍了智能车控制系统的设计开发流程，重点包括机械部分的调整和改进，硬件电路的设计、控制算法和策略优化、系统调试等多个方面。

我们采用了CMOS数字摄像头作为路径检测传感器获取道路信息，采用光电编码器作为测速传感器获取车模行驶速度以实现闭环控制。最终经过长时间的调试，我们制定了适合车模稳定行驶的策略，比较好的实现了车模稳定寻线行驶的任务。

目录

第一章 引言	1
第二章 智能车控制系统整体设计	2
2.1 系统结构图	2
2.2 智能车工作原理	3
2.3 智能车整体布局	3
第三章 智能车机械结构的改造	5
3.1 车模组装与改造	5
3.1.1 前轮定位调整	5
3.1.2 后轮差速的调整	6
3.1.3 舵机的安装与舵机力臂的调整	6
3.2 传感器的安装	7
3.3 编码器的安装	8
3.4 电路板的固定与安装	9
第四章 智能车硬件部分的设计	11
4.1 电源管理模块	11
4.2 路径检测模块	12
4.2.1 摄像头工作原理	12
4.2.2 摄像头选择	13
4.3 电机驱动模块	14
4.4 测速功能模块	14
第五章 智能车软件部分的设计	15
5.1 智能车软件结构	15
5.1.1 图像的采集与处理	16
5.1.2 图像二值化	16
5.1.3 路径的提取	17
5.2 程序主要算法介绍	17

5.2.1 部分路径类型的识别	17
5.2.2 舵机 PD 控制	19
5.2.3 电机的 PID 控制	21
第六章 智能车调试系统	23
6.1 CODEWARRIOR 开发工具简介	23
6.2 上位机系统	23
第七章 总结	26
7.1 车模的主要技术参数	26
7.2 小结与展望	26
参考文献	I
附录 A: 电路原理图	III
附录 B: 程序源代码	V

第一章 引言

随着“飞思卡尔”杯智能车竞赛在国内的普及，越来越多的队伍参与竞争，智能车比赛早已不再是简单的基本控制就可以较好完成的，需要的是对车模的精细化稳定控制，更需要一支队伍的合理分工与配合，体现出了团队精神的重要性。

智能车的制作与调试充满了困难与挑战，从一开始接触智能车的零起步，到学会使用单片机接收图像信息、处理信息，整体硬件电路的设计与焊接，以及期间元器件的选择与购买、车模机械结构的调整，到最后漫长的软件优化与调试。这期间无一不考验我们的动手能力和毅力，也锻炼了我们对于一些疑难问题的分析与解决能力，如单片机无故重启、控制系统内部一些细微的不易发现的漏洞和控制策略中对某些特殊路径适应性不好导致的概率性出错等。

本智能车以飞思卡尔公司的 16 位单片机 MC9S12XS128 为核心控制器，利用摄像头作为图像传感器采集路况信息，配合驱动电路和控制器件进行控制，以达到自主识别路径，控制车模高速稳定地在跑道上行驶。

本文以第七届飞思卡尔智能车竞赛为背景，行文结构如下：

第一章， 引言。

第二章， 智能车控制系统整体设计。

第三章， 智能车机械结构的改造。

第四章， 智能车硬件部分的设计。

第五章， 智能车软件部分的设计。

第六章， 智能车调试系统。

第七章， 总结。

第二章 智能车控制系统整体设计

智能车系统的制作要求是能够在按规则专门设计的跑道上自主识别行驶路径，并要求最快跑完全程而没有冲出跑道，即要求智能小车运行又快又稳。因此对于小车的控制系统来说稳定性和快速性是控制系统设计的两个重要指标。

2.1 系统结构图

本智能车系统主要包括以下模块：MC9S12XS128 单片机最小系统，CMOS 摄像头传感器模块，转向舵机模块，电机与驱动模块，速度反馈模块，调试工具以及电源模块为所有模块提供电能和参考电平。整个系统的大致框架如图 2.1 所示：

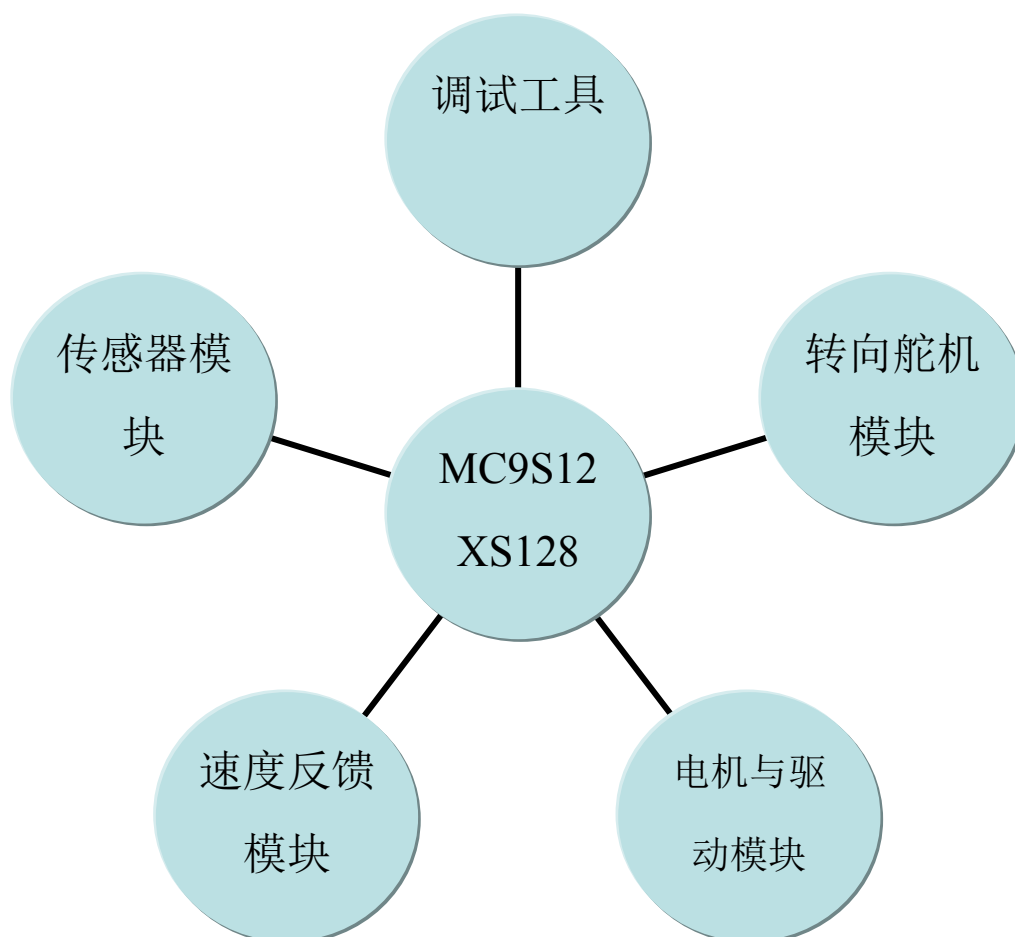


图 2.1 智能车系统结构图

2.2 智能车工作原理

基于视觉传感器的智能车工作原理大致如下：

通过摄像头传感器获取图像信息，由单片机采集接收到一定尺寸的灰度图像，经过二值化后将灰度图像转化为黑白二值图像，通过一定的算法搜寻引线作为前进路径参考，并通过 PWM 波控制舵机完成车模的转向，通过光电编码器反馈速度结合道路信息做出速度决策，并通过 PWM 波控制电机完成对车模的速度控制。

2.3 智能车整体布局

整体布局方面，我们本着简单、轻量的设计原则，通过降低摄像头高度、将主板贴底盘、卧式安装舵机等方式尽可能的降低车模重心。

制作完成后的小车整体布局如图 2.2 和图 2.3 所示。

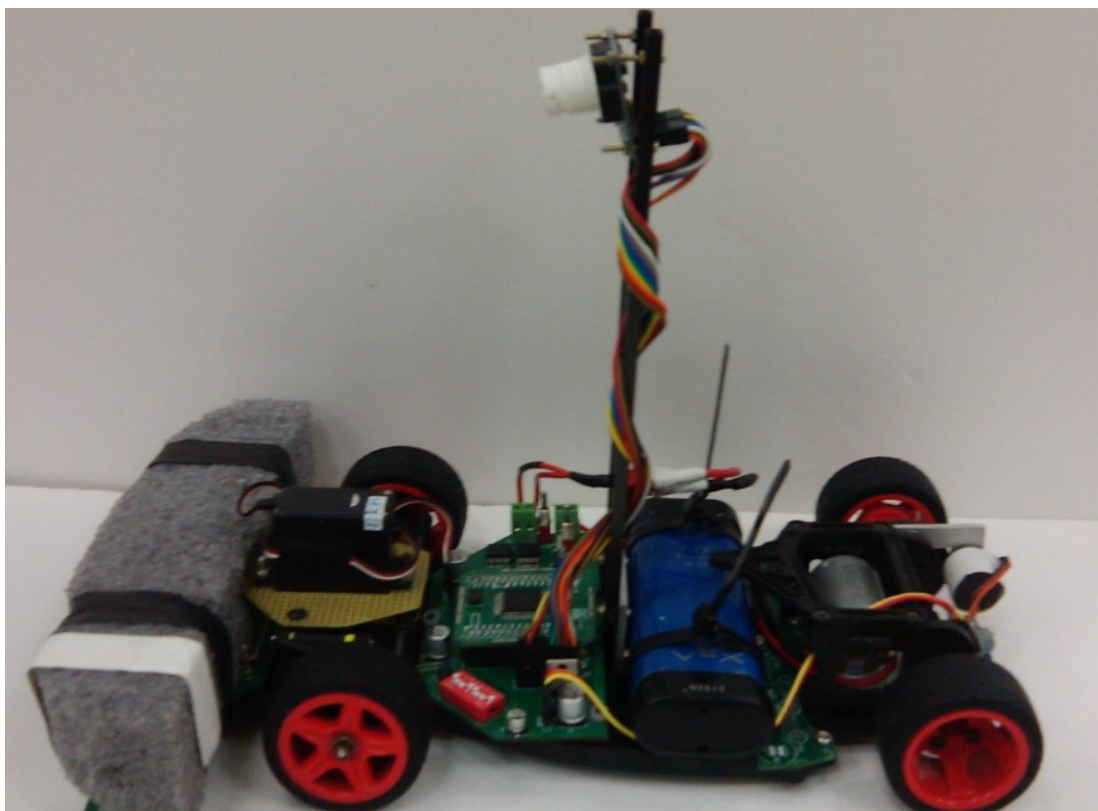


图 2.2 车模侧视图

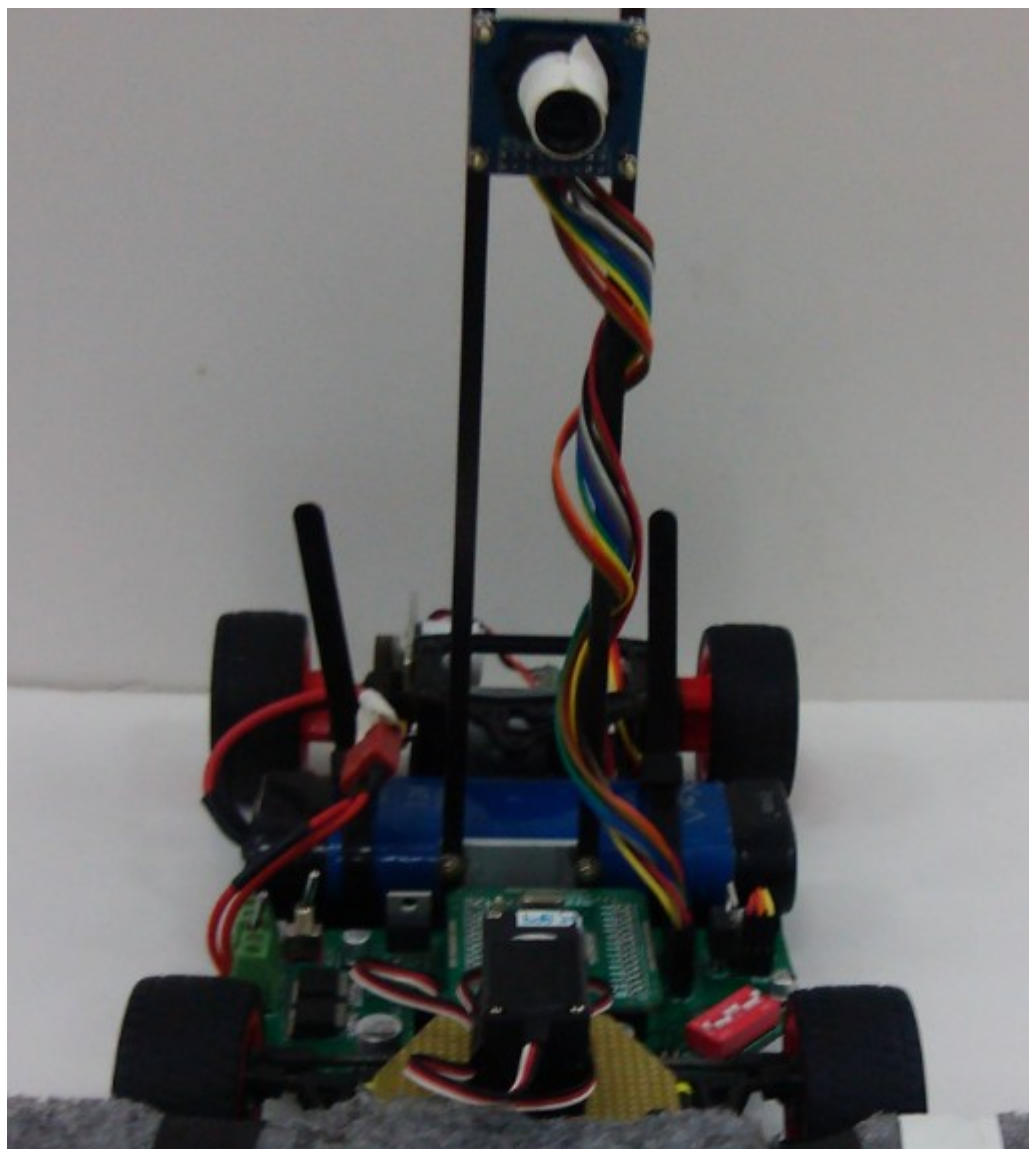


图 2.3 车模正视图

第三章 智能车机械结构的改造

在智能车比赛中，最主要的影响因素是速度和稳定性，车模的机械结构无疑是影响速度和稳定性的关键因素之一。因此我们在前几届的基础上对车模的机械结构做了大量的调整，使其达到较为理想的效果。

本部分包括车模前轮定位调整，后轮差速调整，光电编码器的安装，舵机安装，舵机输出臂的制作与传感器的安装等。

3.1 车模组装与改造

3.1.1 前轮定位调整

车模过弯时，转向舵机的负载会因为车轮转向角度增大而增大。为了尽可能降低转向舵机负载，我们对前轮定位进行了调整。前轮定位的作用是保障汽车直线行驶的稳定性，转向轻便和减少轮胎的磨损。前轮定位参数主要包括：主销后倾角、主销内倾角、前轮外倾角和前轮前束。

主销后倾角是主销轴线与地面垂直线在汽车纵向平面内的夹角，能在车子偏移的时候产生一回正力矩，纠正车轮的偏转。通常后倾角值设定在 1~3 度。

主销内倾角是主销轴线与地面垂直线在汽车横向断面内的夹角，它也有使车轮自动回正的作用，小车转向时路面作用在后轮上的阻力力矩也会减小，从而可减小转弯时驾驶员施加在转向盘上的力，使转向操纵轻便，通常其调整范围为 0~10 度。

前轮外倾角是汽车横向平面与车轮平面的法线与地面垂线之间的夹角，它可以增加零件的使用寿命，提高汽车的安全性，一般为 1° 左右。

汽车两轮前边缘距离与后边缘距离之差称为前轮前束，一般前束的值为 0~12mm，它是为了消除车轮外倾带来的不良后果而调整的。

前轮调整效果图如图 3.1 所示。

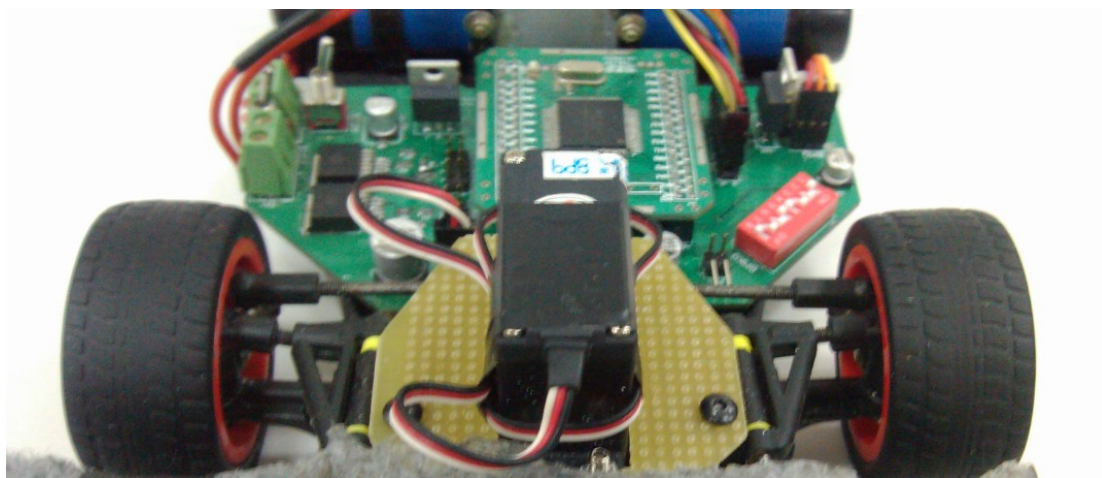


图 3.1 车模前轮调整效果图

3.1.2 后轮差速的调整

车模后轮的差速对转弯时的影响很大，由于 A 车模是机械差速，因此差速的调整也成为影响车模速度的重要因素。差速太松会导致后轮空转，传动效果不好，容易发生侧滑现象；太紧又达不到差速的效果，使转弯性能下降。通过长期调试我们发现，当差速调整在车模全力加速恰好不打滑空转时比较理想，即没有太紧导致转弯性能下降，也不至于太松导致加减速时齿轮空转。

此外，我们通过添加润滑油的手段，改进差速装置，提高了差速的效果，进而也提高了小车的转向性能。

3.1.3 舵机的安装与舵机力臂的调整

舵机安装在车模的前方，舵机自身的重量对车模的重心有一定的影响，对前轮的抓地力也有不可忽略的作用，我们经过多次比较舵机的直立安装与卧式安装后，我们选择了舵机卧式安装，进一步降低了舵机的重心，但是带来了一定的问题，如舵机打角会存在较大的虚位等。对于舵机的安装，我们将继续试验和研究下去。

组委会规定使用的 S3010 舵机的响应速度对于 S12 单片机的处理速度来说存在很大的延时，为了提高舵机的响应速度，我们对舵机的力臂进行了调整。在相同的舵机转速条件下，转向连杆在舵机一端的连接点离舵机轴心距离越远，转向轮转向变化越快，本车模中我们将舵机臂加长到 32mm，通过用自制的转向盘代替舵机上的曲柄来增大舵机的上连接点到舵机中心的距离，增加了输出转

动力矩，使得前轮在转向时更加灵敏。舵机的安装如图 3.2 所示。

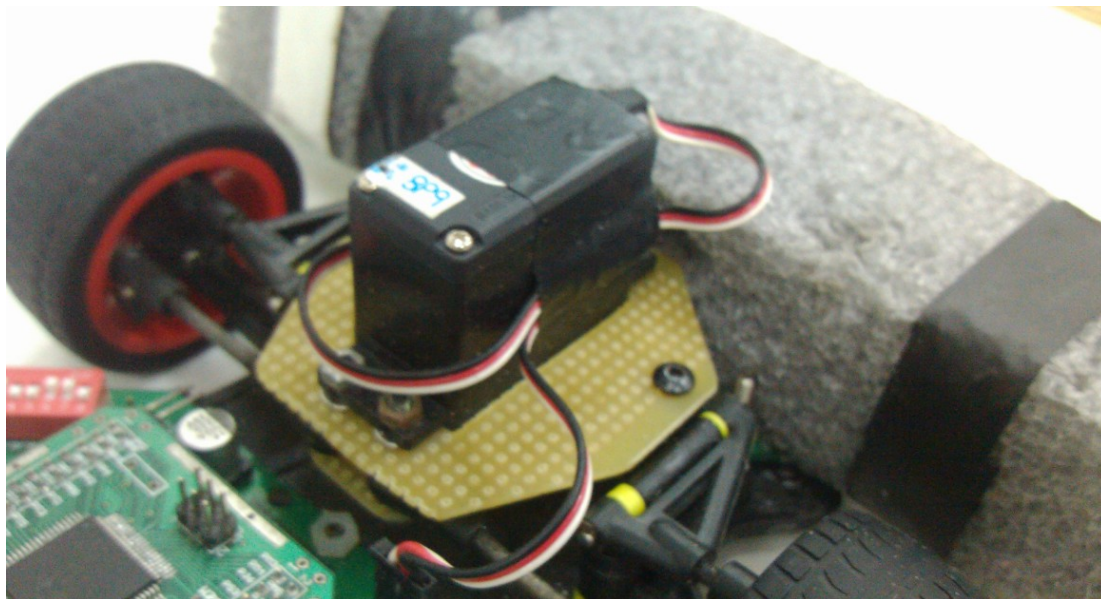


图 3.2 舵机的安装效果图

3.2 传感器的安装

摄像头组对于摄像头的安装一直是影响整个车体性能的一个重要因素，由于需要保证摄像头能够获取足够的前瞻，同时又不能使摄像头对车模重心带来较大的负面影响，而且在车模高速行驶时亦需要保证摄像头的稳定从而能获取稳定的信息，因此在摄像头的安装上我们也是煞费苦心，综合了摄像头对车模重心高低、位置前后以及摄像头自身受后轮抖动的影响等多方面因素，经过多次的尝试，最终选择了较低的位置安装摄像头，安装后摄像头中心高度约 19.5cm。

对于摄像头的支撑，我们选用了两根 4mm*5mm 的方形碳素杆进行双杆固定，将摄像头立在车模的近中心位置，双杆下方用角铝连接固定在底盘上，使得摄像头的支架有了相当的刚度。以保证在尽量减少摄像头盲区的基础上保证四轮定位支撑的受力均匀从而确保轮子抓地力的稳定。

摄像头上方的固定则用双杆打孔，用细螺丝连接摄像头的 PCB，使得摄像头可以在稳稳的固定的同时也可通过调节螺丝上螺母的位置来移动摄像头，从而调整摄像头的视角。

摄像头的安装如图 3.3 和图 3.4 所示：

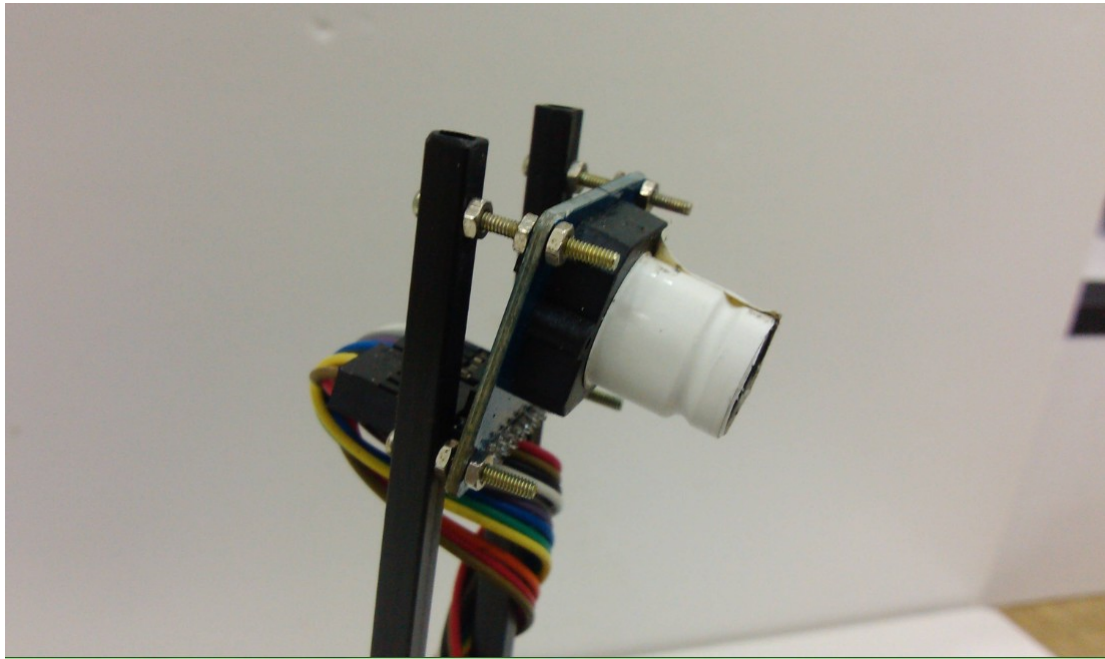


图 3.3 摄像头固定斜视图

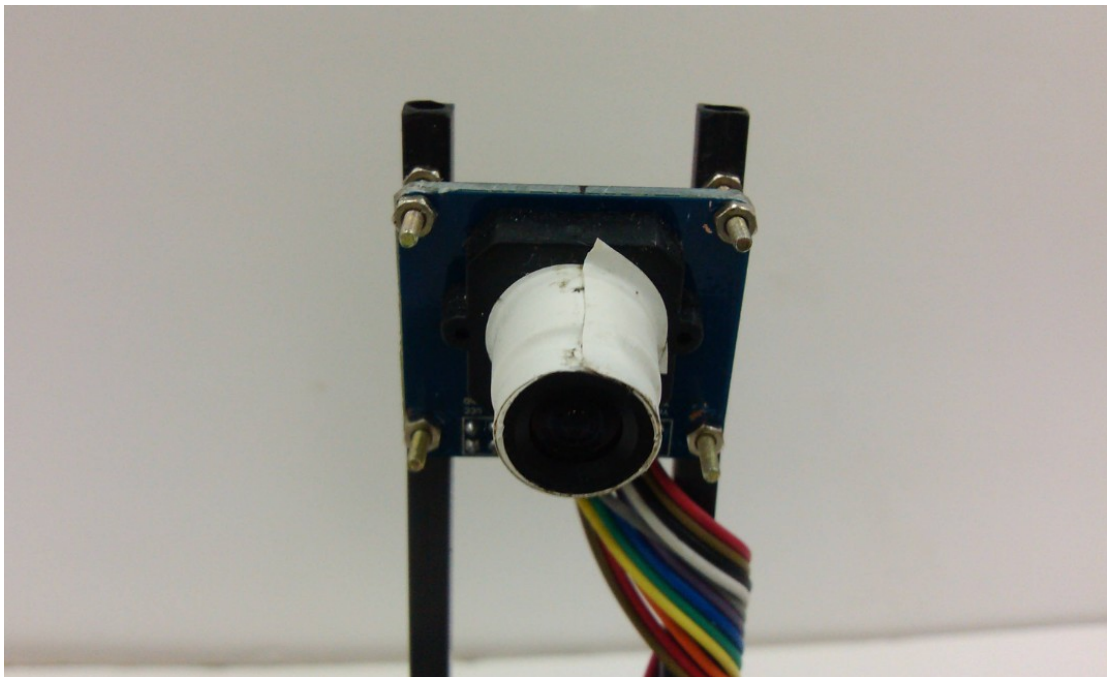


图 3.4 摄像头固定正视图

3.3 编码器的安装

光电编码器，是一种通过光电转换将输出轴上的机械几何位移量转换成脉

冲或数字量的传感器，这是目前应用最多的传感器。由于光电码盘与电动机同轴，电动机旋转时，光栅盘与电动机同速旋转，经发光二极管等电子元件组成的检测装置检测输出若干脉冲信号，通过计算每秒光电编码器输出脉冲的个数就能反映当前电动机的转速。

经过一定的比较，我们延续上一届的选择，使用体积较小的微型 OME-N 型 157 线编码器，此编码器符合我们的对精度的要求，安装效果如图 3.5 所示。



图 3.5 编码器安装效果图

3.4 电路板的固定与安装

为了使整个车子看起来轻巧灵活，结构简单，我们放弃了原来单独做驱动板的方法，在精简了主板电路后将电机的驱动模块集成在主板上，并使主板贴在车模前方底盘上，安装效果图如图 3.6 所示。

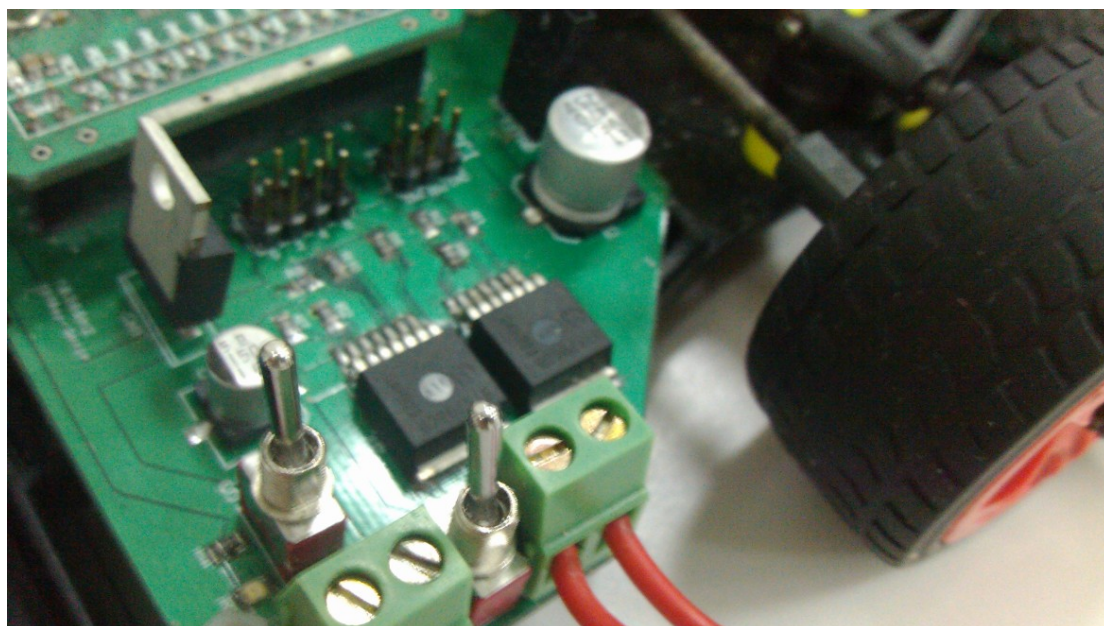


图 3.6 主板及驱动模块的安装效果图

第四章 智能车硬件部分的设计

智能车控制系统从硬件上分为电源模块、传感器模块、直流电机驱动模块、转向舵机驱动模块和单片机模块。为了简化驱动电路，我们将驱动电路集成在主板上，以减少驱动电路板对车模重心的影响。

4.1 电源管理模块

可靠性是整个硬件模块的第一要求，电源模块的可靠性决定了整个系统能否稳定运行。为了保证整个系统运行的可靠性，单片机的供电采用 LM2940CT 低压差 5V 稳压管，该电源芯片最大输出电流为 3A，压差在最大输出电流时也不会超过 500mV。它具有低压差电压、瞬态响应快的优点，是小功率微处理器理想的配套件。内部电流限制及发热限制可以防止可能造成结温过高的过载现象。因此，MCU 和测速模块以及传感器的电源采用它供应，而舵机和电机则直接使用电源电压供电。

整个电源管理模块如图 4.1 所示。

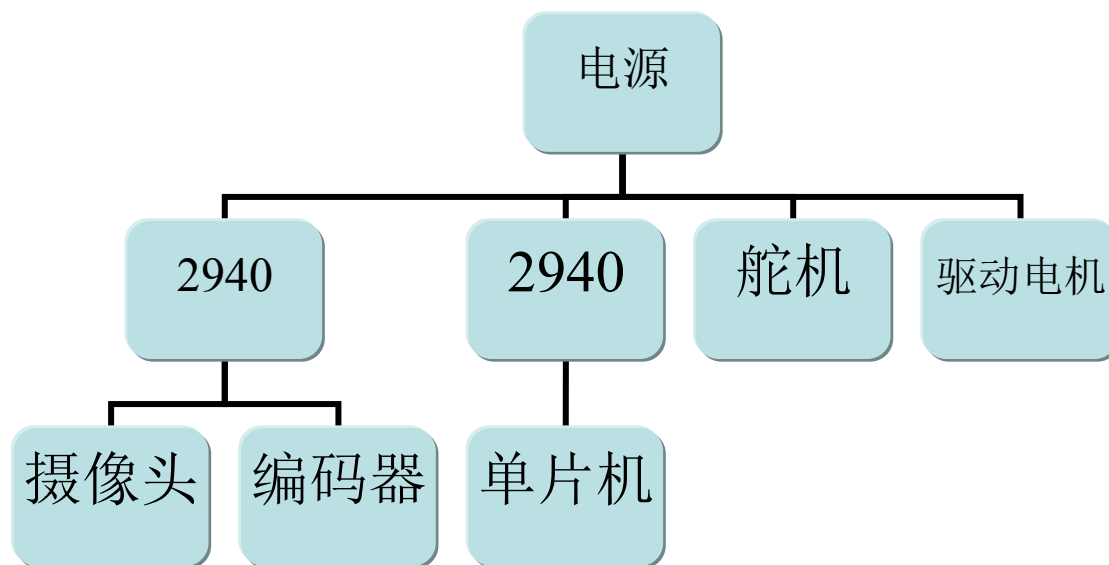


图 4.1 电源管理模块框架

为了使单片机的供电稳定可靠，我们采用了两片 2940 芯片，一片单独供给 MCU 工作，一片供给其他。

考虑到电机的理想空载转速与电枢电压成正比关系，设计时将电池的电压不经过稳压电路而直接加在电机驱动模块上，这样保证了驱动电机速度具有较大的可调范围，且使电路设计简化。同理，舵机也直接采用电池供电。

主板的原理图见附录 A。PCB 效果图见图 4.3。

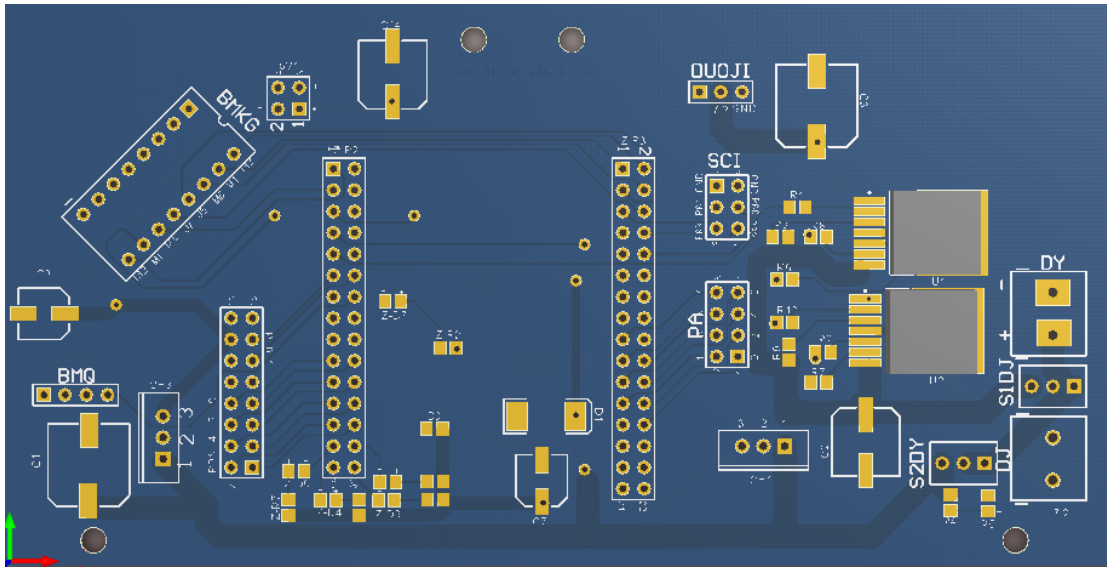


图 4.3 主板 PCB 效果图

4.2 路径检测模块

传感器是车模前进的眼睛，车模寻线行驶源自于传感器对路径的稳定检测。因此，保证传感器稳定的运行是车模能够稳定完成比赛的根本。对于路径检测模块，我们将从摄像头工作的原理和摄像头的选择两个方面来介绍。

4.2.1 摄像头工作原理

摄像头分黑白和彩色两种，根据赛道特点可知，为达到寻线目的，只需提取画面的灰度信息，而不必提取其色彩信息，所以本系统中采用的是黑白摄像头。

摄像头主要由镜头、图像传感芯片和外围电路构成。摄像头的工作原理是：按一定的分辨率，以隔行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。具体而言（参见图 4.4），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，该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（胖零模式，非 0 V，比 0 V 稍高一点），并保持一段时间。这样相当于，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，跳过一行后（因为摄像头是隔行扫描的），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。摄像头每秒扫描 30 幅图像，每幅又分奇、偶两场，先奇场后偶场，故每秒扫描 60 场图像。奇场时只扫描图像中的奇数行，偶场时则只扫描偶数行。

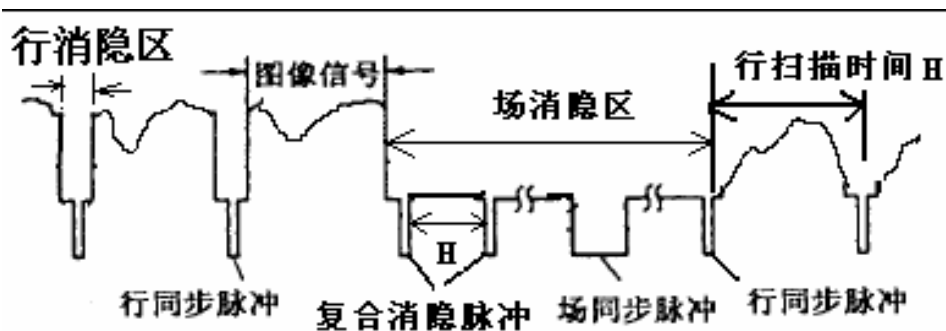


图 4.4 摄像头视频信号

4.2.2 摄像头选择

在选择摄像头时，我们参照上届比赛的经验和前几届比赛各参赛队的摄像头选择，对目前可选的各种摄像头进行了比较和实验。目前市面上常见的摄像头主要有 CMOS 和 CCD 两种。下面对两种摄像头进行了比较：

1、CMOS 重量轻，外围电路简单，低功耗，可 5V 供电，但是动态响应性能差，车子跑起来之后采集到的图像会有一定的模糊，给下一步图像处理带来麻烦。

2、CCD 重量稍重于 CMOS，外围电路相对复杂。需用 12V 供电，功耗大。但动态性能优于 CMOS，车子动态中采集到的图像效果优于 CMOS。

最终，考虑到赛场环境对比性好，CMOS 摄像头可以满足要求，且我们没有将太多精力花在处理 CCD 摄像头的外围电路，也为了进一步简化硬件电路，做到简洁、可靠，我们最终选定了 COMS 数字摄像头作为主传感器。

我们选择的摄像头见图 3.3。

4.3 电机驱动模块

前几届我们一直使用 MC33886 芯片+H 桥驱动电路来驱动电机，虽然可以较大能力驱动电机，但是驱动电路板占了一定的体积和重量，因此这一次我们选择了较为主流的两片 BTS7960 并联驱动电机，并将其集成在主板上，从而进一步简化了电机驱动的硬件模块，并降低了车模重心。

驱动模块见图 3.6。

4.4 测速功能模块

本设计的测速功能采用光电式编码器，它具有测量准确度高、响应速度快、可靠性高、结构简单、测速准确度高以及使用寿命长等优点。光电式编码器的工作原理就是利用光码盘上透光与不透光区域交替变换，在码盘的另一侧形成光脉冲。该光脉冲照射在光电敏感元件上产生与其相对应的电脉冲，即可完成速度测量功能。

第五章 智能车软件部分的设计

5.1 智能车软件结构

对于智能车来说，讲究的是智能控制。因此算法是其核心部分，相当于人的思维，负责按预先定好的流程处理传感器采集过来的数据。

软件系统流程图如图 5.1 所示：

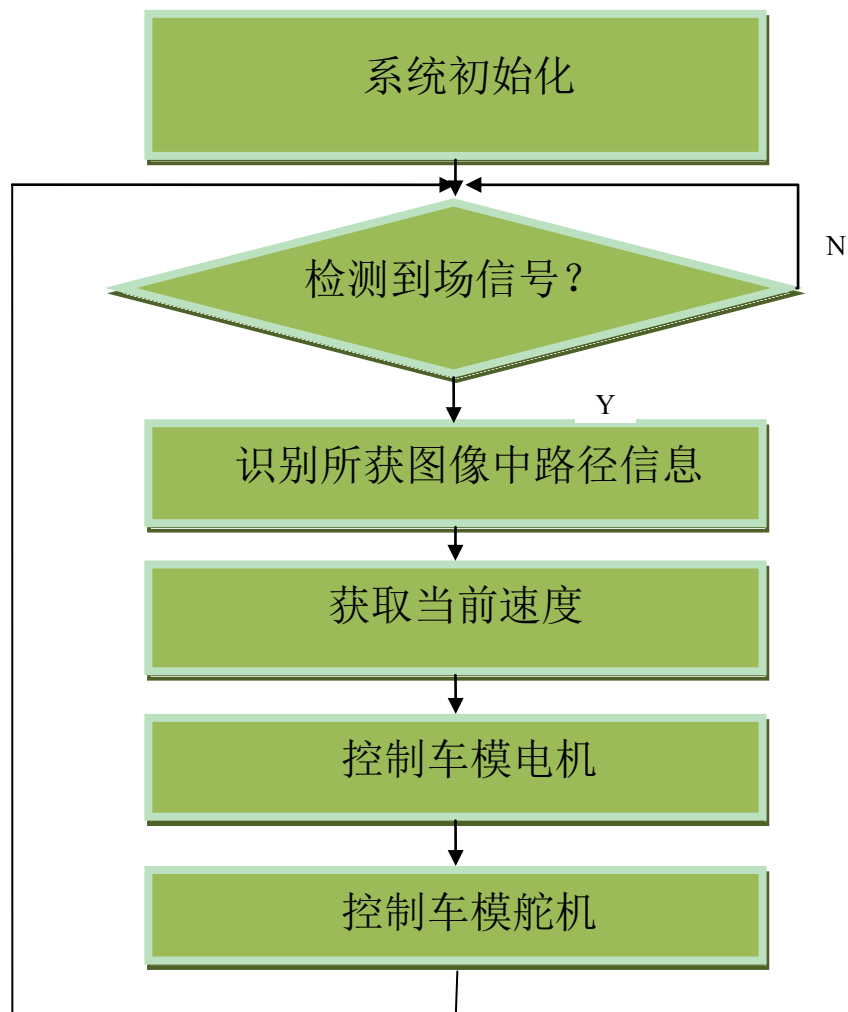


图 5.1 软件系统流程图

5.1.1 图像的采集与处理

由于摄像头传送的数据量很大，而且传输的数据很快，单片机在有限的总线频率下无法采集所有的数据，所以必须对摄像头传送的信息进行优化处理。

由于摄像头的分辨率都较高，所以一幅图像扫描的行数和每行的点数（列数）都比较多。但对于飞思卡尔小车比赛，不需要把整幅图像的所有信息都采集进来。考虑到赛道宽度为 45cm，两边导引黑线宽度为 2.5cm，所以只要采集能够分辨出导引黑线的行数和足够的点数就可以了。最终，我们在 MCU 超频的条件下，加大采集的行数与列数，使得每幅图像采集到 36 行，每行 160 个点，大大提高了图像的纵向分辨率。

由于 XS128 内部只有 8K 的内存，无法放下较大的图像，因此无法完整的将一幅 36*160 图像二值化。针对这样的问题，我们曾经想到过使用外部存储器，可以获取分辨率更高的图像，但读取外部存储器的图像本身就有一场图像的延时，也不利于硬件的简化。最终我们放弃了使用外部存储器，转而采用动态处理的方法，即二值化一行数据提取其有效信息后释放，最终只获取处理后的有效信息，这样就避免了将一整幅图像都保存的空间不足的问题。

5.1.2 图像二值化

图像二值化一直是摄像头的一个重点也是难点，只有合理的阈值分割才能较好的将所获取的灰度图像二值化处理，从而尽可能简单高效的提取有效信息。常用的二值化方法有硬件二值化和软件二值化两大类。

由于单片机采集到的图像本身较大，二值化会大大占用单片机处理时间。硬件二值化由于是在片外完成图像的二值处理，因此可以大大减轻单片机的负担，从而有更多的时间进行后期处理。但是由于此方法采取固定的阈值，而现场的光线与环境是时刻变化的，因此硬件二值化对环境的适应能力较差。鉴于这一点我们放弃了硬件二值化的探索，选择了对环境适应性更好、二值化处理效果更优的软件二值化。

单片机采集到的图像本身较大，软件二值化需要对图像的每一个点都进行处理，如何合理的选取阈值成为了难点。为此我们尝试了多种方法，如双峰法、部分比较法等。最终我们选择了使用累加灰度值后平均的方式选取阈值，避免了双峰法存在的由于某个点出现极端情况导致的处理失败的问题。此方法的思

想是：将采集到的一行数据的灰度值全部累加，然后计算其均值，并在此基础上做一定的调整(我们调试的结果是减小一定值)，以此作为本行数据的阈值，进行二值化。

5.1.3 路径的提取

由于这一届比赛的赛道黑色引导线铺设在道路两边，因此我们调整了路径提取的扫描方式。改为从中间向两边扫描。引导线的确认与提取我们依然采取了前几届采用的连续两点为黑点的黑白跳变。从而在取得了一定程度的滤波效果上能够准确找到引导线的基础上。

此后，我们在这个基础上又引入了跟踪边缘的提取方法，即在上一行提取到黑色引导线信息的列数上在其左右一定范围内搜寻，从而更进一步减轻了单片机的负担，更快速有效的提取了路径信息。

寻到两边的黑线后，记录两边黑线的坐标，将其坐标除以 2 即可获取拟合中心线坐标，并以此作为车模前进的绝对坐标。

5.2 程序主要算法介绍

5.2.1 部分路径类型的识别

■ 十字路口

由于跑道引线铺设位置的改变，寻线扫描方式发生改变，拟合的中心线不再是以前绝对连续的黑线，而一定程度上受车子位置的影响。尤其是新赛道的十字路口，黑线断续了很长一段距离。对于摄像头组来说，十字路口的改变是一个新的挑战。

对于十字路口的处理，由于是新类型赛道使用的第一年，相信很多优秀的队伍都会有很好的算法。经过很长时间的调试，我们最终选取了补线的方式，通过对扫描提取的引线的修补确保拟合中心线不会在十字路口有较大的偏差。从而确保车子能顺利通过十字路口。

补线的思想：当跟踪边缘算法在前一行黑色引导线坐标左右一定的范围内未搜寻到引线时，置这一行的引线坐标同上一行坐标。

经过一段时间的微调，我们使用这种方法可以较好的通过十字路口，但是通过时会有一定的向左或向右抖的倾向，一定程度上影响了车子的速度。所以这种方法还有待于完善，我们也会在今后的过程中继续改进这个地方。

十字路口的补线效果见图 5.2。

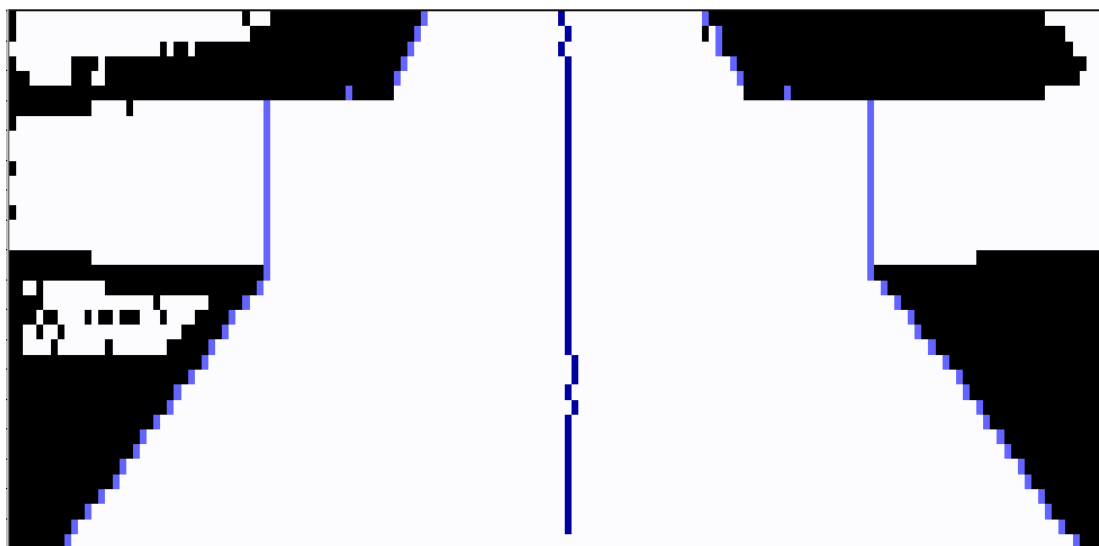


图 5.2 十字路口补线效果图

■ 坡道

坡道一直是很多高速小车的棘手问题。而今年在新规则的情况下，不难发现，黑色引导线的间距能够体现出赛道的视觉宽度。

由于摄像头自身扫描图像的梯形失真特点，使得图像中的赛道由于距离车的远近不同而呈现出不同的宽度。而恰恰当小车将要上坡时，坡道处的赛道距小车摄像头较近，因而在图像上会体现出赛道宽度变得很宽，而形态上是标准的直道，但是图像上的赛道宽度却会明显比正常直道的宽度宽，因此可以利用这一点选取一两行参考行，比较其在直道形态下的宽度即可判断此时是坡道还是普通直道。

坡道的效果图见 5.3。普通直道效果图见 5.4。



图 5.3 坡道处理效果图

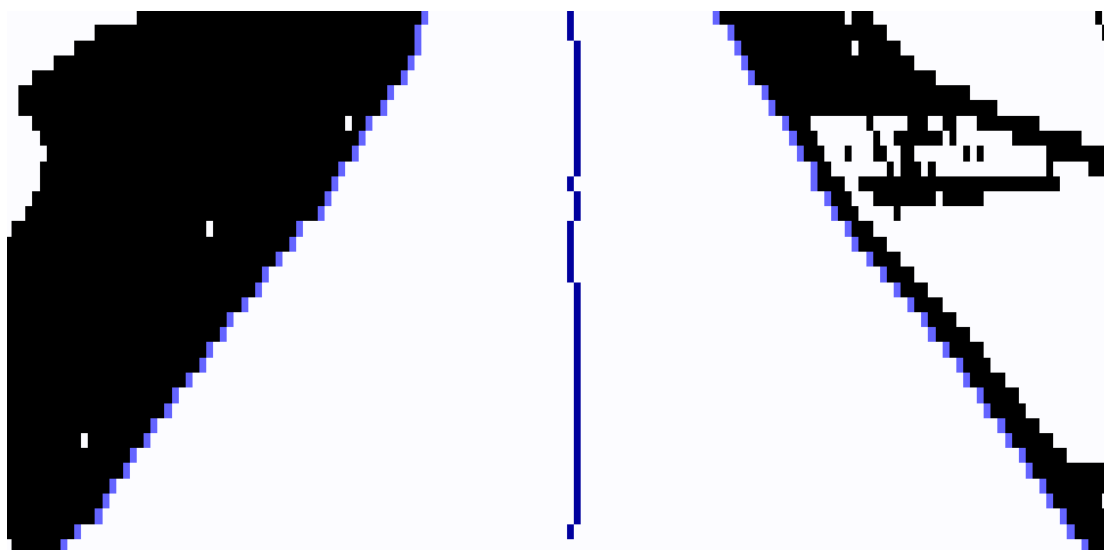


图 5.4 普通直道处理效果图

综上图可知，两者之间还是有较大区别的，坡道的特点显而易见。

5.2.2 舵机 PD 控制

通过处理后的图像中心的拟合中心线较好的体现出了赛道的路径情况。通过计算拟合中线在参考行的坐标与图像绝对中点坐标的偏差可以得到一个偏移距离。此偏移距离可以反映车身在赛道的位置。因此可以通过建立此偏移距离与舵机打角 PWM 波的函数关系来实现对舵机的控制。

经过实践之后我们发现,如果 PWM 波占空比与偏移距离用简单的线性关系虽然能够简单的实现车模的基本打角,完成车模的转弯,但会带来一定的弊端:如果比例系数较大弯道会有比较灵敏的表现,但直道会发生抖动而不稳定;比例系数太小直道比较稳定,而弯道则会因转角过小发生冲出赛道的现象。

基于此,我们建立了一个非线性的数据对应关系:当小车偏移较小时比例系数比较小来使直道行驶稳定,而偏移变大时加大比例系数来增大弯道的灵敏度。同时加入了微分控制,确保车模在连续转弯时有一定良好的动态性能。

实践证明通过这样的非线性的算法配合简单有效的 PD 控制,保证了小车平稳流畅的通过各种赛道情况。

舵机转角与偏移距离曲线如图 5.5 所示(其中横坐标为转向参考行偏移距离,总纵坐标为对应的相对打角值)。

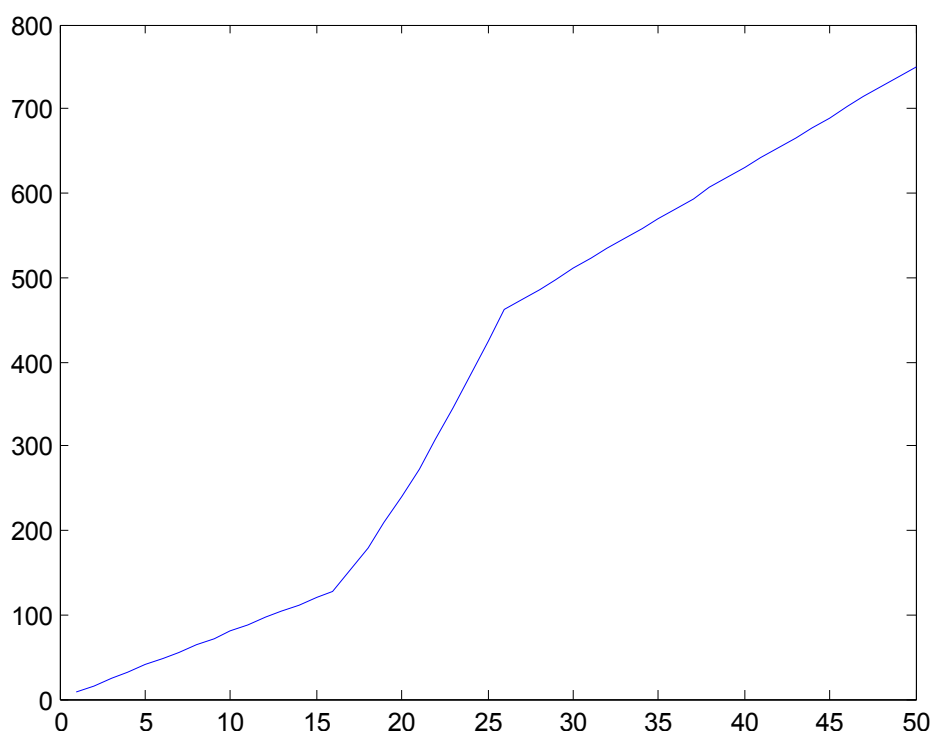


图 5.5 舵机转角与偏移距离曲线

5.2.3 电机的PID控制

智能车比赛拼的就是速度，因此如何让小车以最快的速度而又不发生冲出赛道的完成比赛是我们最终的目的。速度的控制则是程序的重点，经过长期的调试，我们发现，小车能否以较快速度完成比赛取决于能否实现直道入弯有效的刹车和入弯后灵敏的速度调节，在控制算法的选择上我们选择了经典的PID控制。

速度的PID控制分位置式和增量式两种。这两种PID算法的表达式也深为大家所知。

常规的PID控制算法有三种控制参数组成，即比例、积分和微分。比例系数 K_p 对系统的稳定性、超调量和相应速度起着主要影响。增大 K_p 可以提高响应速度，但会影响系统的稳定性。一般来说，当偏差较大时可以增大 K_p ，当偏差较小时可以减小 K_p 。积分可以提高系统的无差度。微分可以增强系统的动态响应性能，对动态调节过程影响很大，增大 K_d 有利于减小超调，但会增大调节时间。

■PID控制算法的公式：

$$U = K_p * e + K_i \int_0^t e dt + K_d * e$$

PID参数的选取在稳定性和响应速度之间存在矛盾，为了获取满意的系统性能，在控制中应根据系统的动态特征，采取变P控制，以增强系统的适应性。对于小车系统而言，利用P分量就已经得到很好的控制效果了。

■增量式PID控制算法公式：

$$\begin{aligned} \Delta U_n &= U_n - U_{n-1} \\ &= K_p * (e_n - e_{n-1}) + K_i * e_n + K_d * (e_n - 2 * e_{n-1} + e_{n-2}) \end{aligned}$$

位置式PID和增量式PID的比较。

位置式PID：具有直观、易理解、易操作的优点，静态误差易消除。但由于位置式PID是全量输出，一旦某一次有计算错误，将造成比较严重的后果。

增量式PID：具有一定的信号锁存性能，由于是增量输出，出现计算错误时对系统的误操作影响小。使用起来较为不易。

我们对两种 PID 算法都进行了一定时间的调试，由于增量式一直未能调试出效果较为理想的参数，最终我们放弃了增量式 PID 算法参数的调试，选择了位置式 PID 算法。经过相当长时间的调试，我们最终选择了一组比较合适的参数。

我们将速度的理想差值处理成近非线性曲线（其中横坐标为偏移量，纵坐标为对应的速度减少量）见图 5.6。

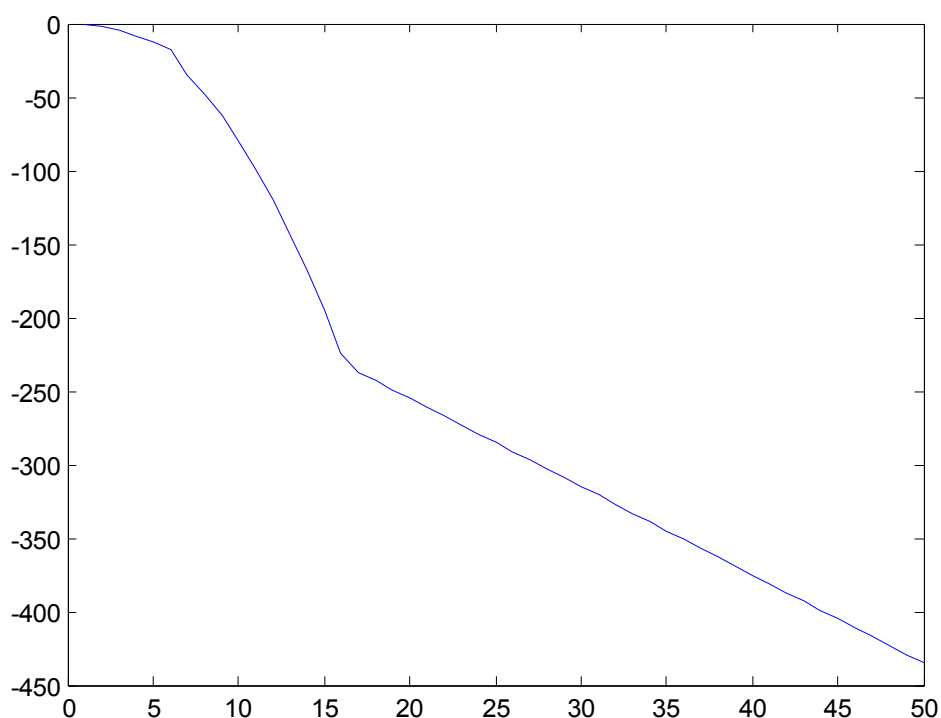


图 5.6 速度变化量曲线

第六章 智能车调试系统

6.1 Codewarrior 开发工具简介

此次智能车大赛的软件开发平台为Metroworks 公司的Code Warrior 4.7, Codewarrior 是由Metrowerks 公司提供的专门面向Freescale 所有MCU 与DSP 嵌入式应用开发的软件工具。其中包括集成开发环境IDE、处理器专家、全芯片仿真、可视化参数显示工具、项目工程管理、C交叉编译器、汇编器、链接器以及调试器。用户可以将芯片的类库添加到集成环境开发环境中生成就是一个最小系统, 就能直接在新建工程时添加所需的程序代码。图6.1为Codewarrior用户操作界面。

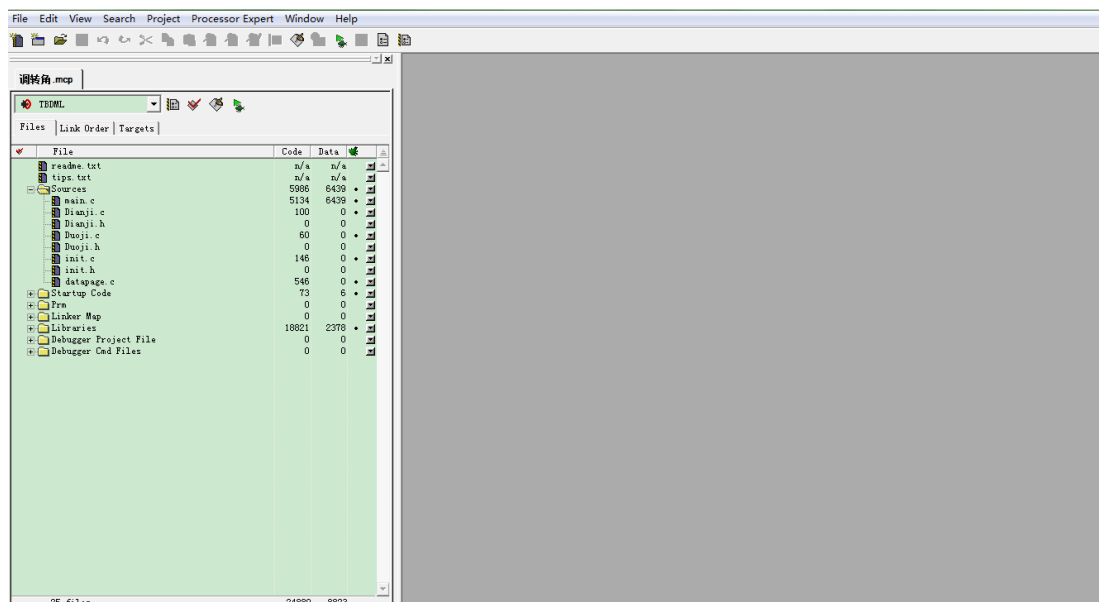


图 6.1 Codewarrior 用户操作界面

在用户源程序编译连接生成.abs 文件之后, 可借助 CodeWarrior 4.7 自带的 hiwave.exe 进行调试工作, 如监视寄存器状态、修改指针、设置断点、实时查看源程序各数据的变化等, 最后先擦除 flash 中的源程序, 之后再将 abs 文件烧入 flash 中。

6.2 上位机系统

对于摄像头组来说，能否准确获取图像信息是车模能否高速、稳定的完成比赛的关键，因此，为了直观的看到车模所获取的信息，我们自己用 labview 软件编写了上位机，通过上位机观察到了车模正常工作时所获取的信息，为准确识别各种赛道类型提供了基础工具，极大的方便了今后的调试。图 6.2 为我们设计的上位机操作界面。

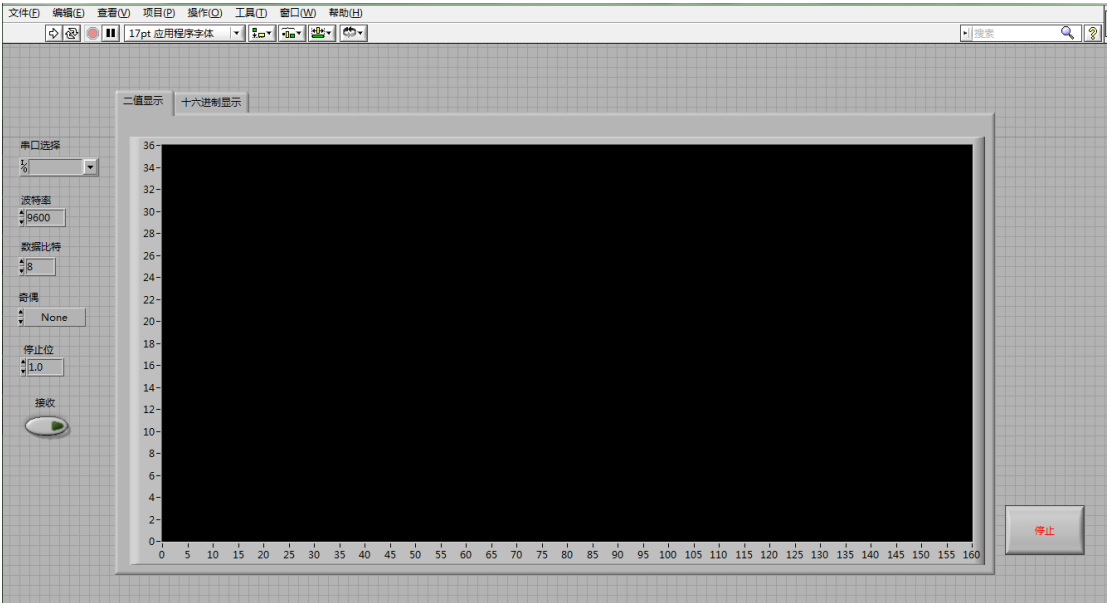


图 6.2 上位机图像信息界面

图 6.3、6.4、6.5 为小车在静止状态下在某些类型赛道上获取的图像信息。

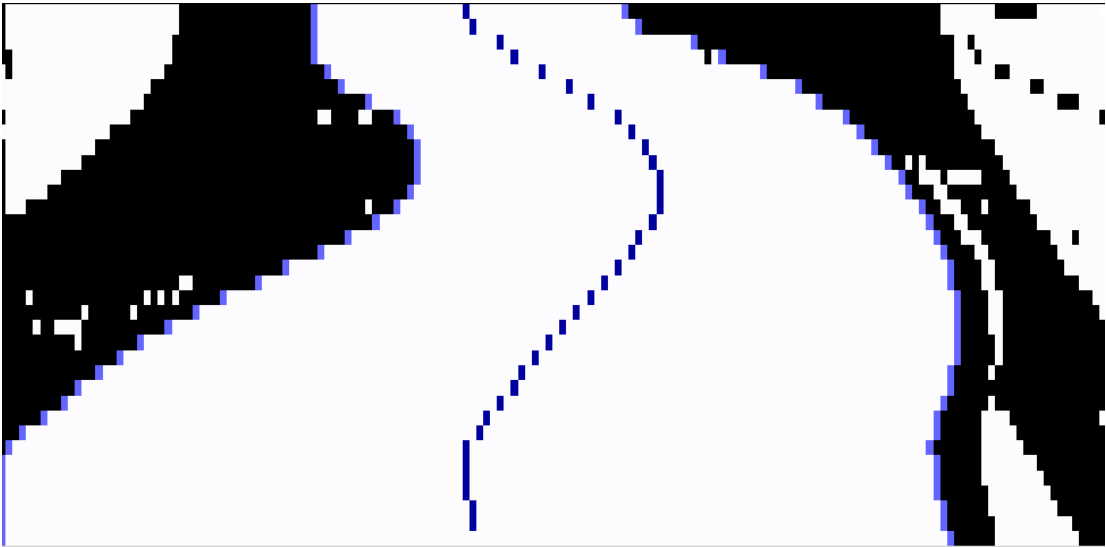


图 6.3 S 弯处理效果图

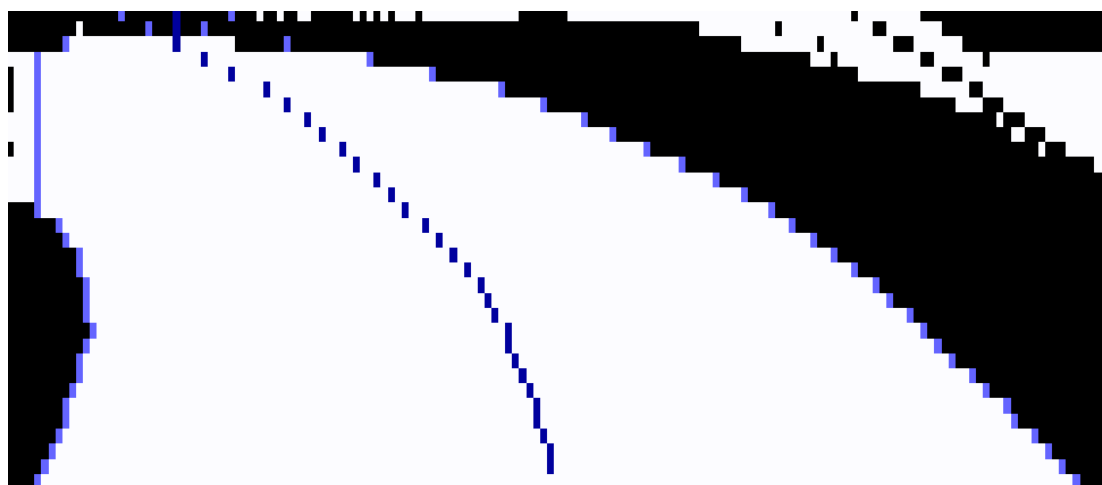


图 6.4 90 度急弯处理效果图

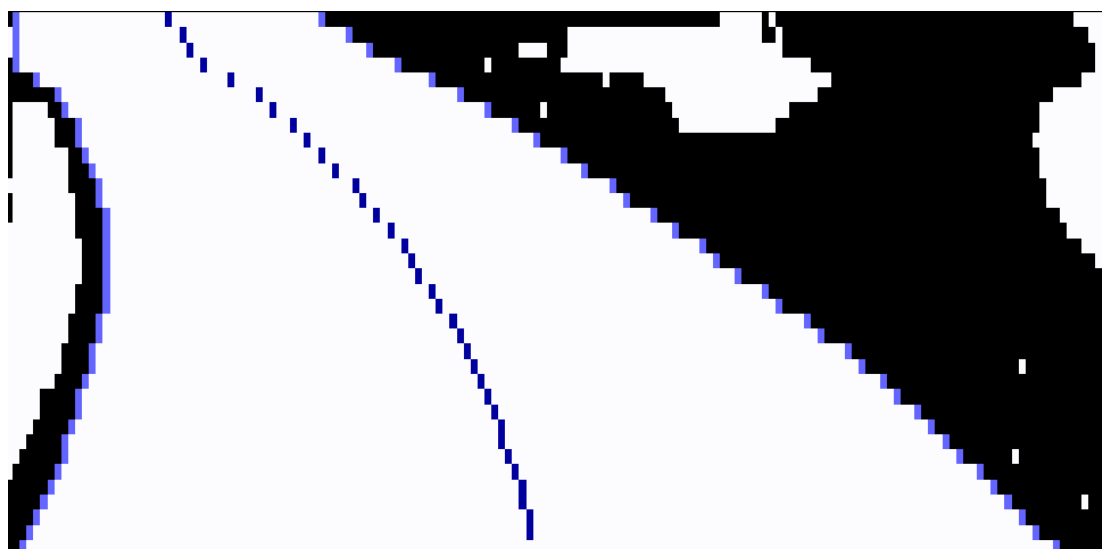


图 6.5 长弧处理效果图

第七章 总结

7.1 车模的主要技术参数

表 7-1 车模的主要参数

项目	参数
车模几何尺寸 长、宽、高 (mm)	290 190 230
车模轴距、前轮距、后轮距 (mm)	240 140 140
车模总体重量 (kg)	0.8 (不含电池)
电容总容量 (uf)	1515.12
传感器种类与个数	COMS 数字摄像头*1、光电编码器*1 红外发射接收管*3
除了车模原有的驱动电机、舵机之外伺服电机个数	0
赛道信息检测精度 (mm)	3
赛道信息检测频率 (Hz)	60

7.2 小结与展望

首次听闻飞思卡尔是在大一时，我们的专业负责人也就是我们的指导老师

杨老师为我们介绍这个比赛，记得当时看到小车在白色的跑道上循着一根黑线报，很是惊讶，以前对小车的概念一直停留在儿时玩的遥控车和看的动画片里。一直都觉得模型车不靠遥控的话要在左右都有护栏的跑到里靠着车四周的导轮来维持方向。突然见到能够全自动的在赛道上跑的小车让我产生了浓厚的兴趣。到了大三后，我主动找到了杨老师毛遂自荐成功的加入了车队。

之后我和队友们开始查阅大量资料，了解车模的工作原理，了解传感器的工作原理，并动手拆装车模，对车模的机械部分和硬件组成有了深入的了解。由于今年较大范围的改动跑道参数，为我们带来了较大的挑战的同时也给了我们新的机遇。而我们也从初入队伍时的几乎一窍不通到渐渐明白了单片机、传感器的工作原理，明白了硬件电路的设计原理和电机驱动原理等等。

真正着手调智能车是今年开学来，也就是 2 月底开始。经过几天接连的奋斗，很快在 3 月初我们成功的实现了小车的处女航，使得小车可以以较低的匀速寻迹完成一圈，这给了我们极大的信心。在这之后，我们有了目标和方向，一步一步完成控制算法的优化、扩展，硬件电路的自主设计和优化以及机械部分的改造等。

在这近 7 个月的备赛过程中，我们经历了太多的困难和挑战，小车的状态几乎是以波浪式的曲线向上，刚获取了一些进步没多久就突然发生一些故障导致小车瘫痪，中间甚至由于各项硬件的老化、损坏以及单片机的重启复位等因素耽搁了一个半月之久，但在我们各个队员相互鼓励下，我们不轻易向任何困难低头，最终整体对车模的硬件电路和机械结构更新换代，并成功解决了单片机重启复位等问题，从而使小车的硬件基础得以稳定，为我们进一步调试软件提供了充分的保障。

很开心能挺进全国决赛，取得这样的成绩也是相当不易的。赛场上激烈的斗争，紧张的气氛让我们深深感受到每个队伍、每个队员都在为做好智能车努力着，不管结果如何，大家都是有收获的。通过区域赛我们看到了自己与其他优秀的队伍相比还有很大的差距。为了能在全国总决赛当中也取得好成绩，我们的队员回校后马上投入紧张的调试当中。在这过程中有多少个不眠夜大家心里都清楚，但是没人会在意。

历时将近半年的时间，我们完成了对智能赛车的制作和改进。并经过长期

的调试获得了一组不错的参数。但由于我们花了大量的时间在尝试不同取线方式、赛道类型识别和速度控制上，虽然确保小车可以稳定的以 2.7m/s 的速度行驶，但由于转角参数调试的不理想，使得小车不能很流畅的行驶，损失了不少的时间。同时我们也没有很好的优化、发展上位机系统，不能很好的跟踪小车的速度，这使得我们的调试效率大大下降，在有限的时间内很难较好的完成小车的转向优化和速度优化，使得我们很久都没能成功的将速度稳定在 3m/s 以上。但我们相信，经过我们的不断积累、发展和优化，相信在第八届、第九届或是以后的比赛中，我们一定能够跻身全国强队之列。

参加这个比赛以来，在场地、经费等方面得到了学校和学院的大力支持，感谢比赛组委会能组织这样一项很有意义的比赛，让我在大学里学到了我意想不到电路终身受益的知识和实践。同时，我们还特别感激我们学校和学院里一直支持和关注智能车比赛的领导以及老师。正是有了这个展示的平台和大家的支持，才让我们有机会提高自己。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辛勤汗水的结晶，凝聚着我们小组每个人的心血和智慧，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] 卓晴, 黄开胜, 邵贝贝. 学做智能车——挑战“飞思卡尔”杯[M]. 北京: 北京航空航天大学出版社, 2007
- [2] 王威. HCS12 微控制器原理及应用. 北京: 北京航空航天大学出版社, 2007
- [3] BDM_for_S12_TBDML_用户手册 V3.1 中国·北京 清华大学 Freescale MCU/DSP 应用开发研究中心 2006. 12.
- [4] 林锐. 高质量 C++/C 编程指南.
- [5] 谭浩强, C 语言程序设计[M]. 清华大学出版社, 2006. 1
- [6] 张宏建, 蒙建波, 自动检测技术与装置[M]. 化学工业出版社 2004. 7

附录 A: 电路原理图

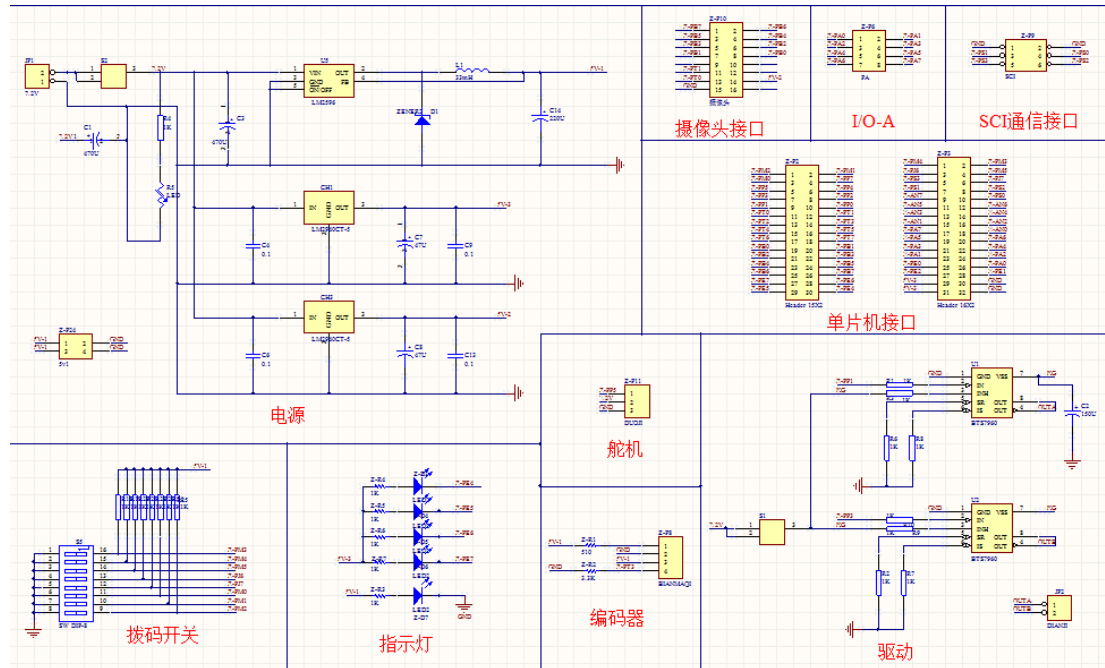


图 A-1 电路原理图总图

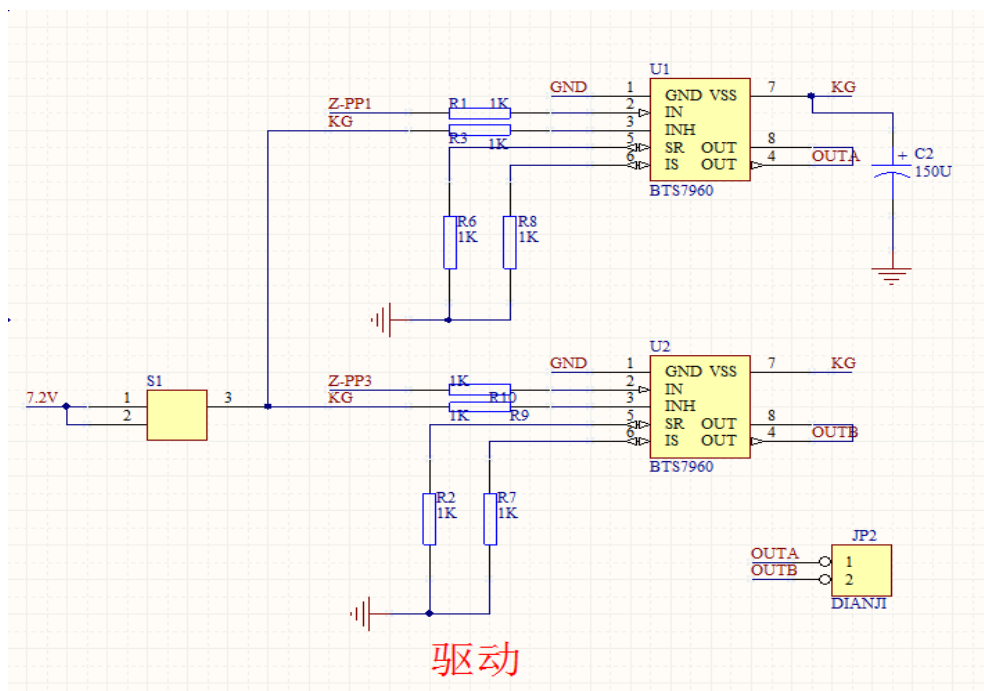


图 A-2 驱动部分原理图

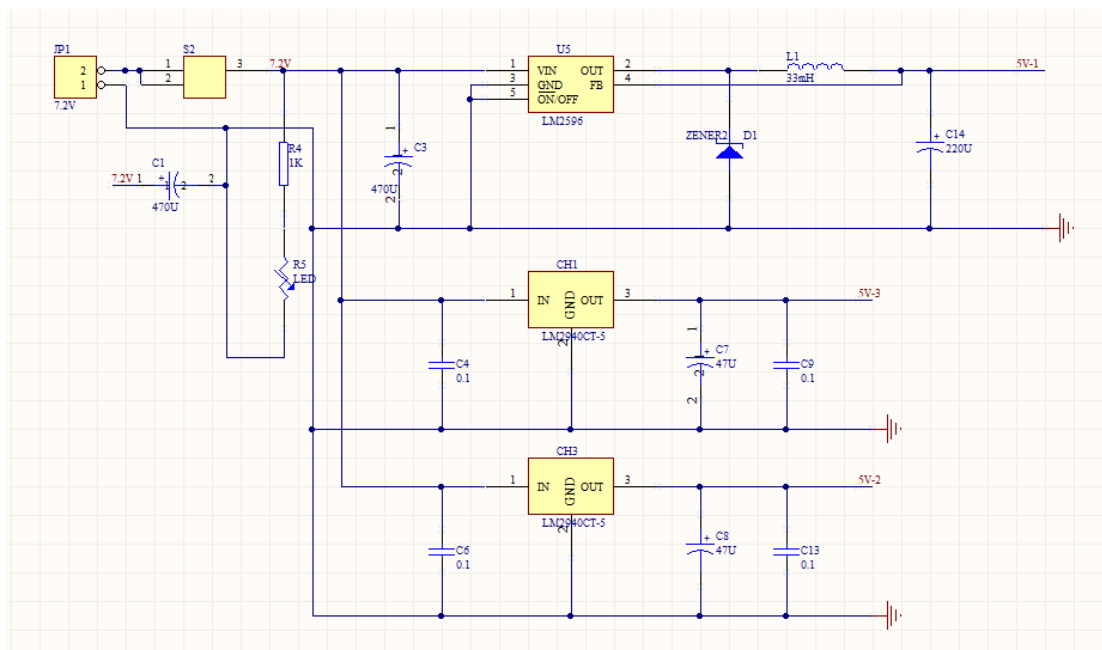


图 A-3 电源部分原理图

附录 B：程序源代码

```
#include <hidef.h>          /* common defines and macros */
#include <MC9S12XS128.h>     /* derivative information */
#include "Init.h"
#include "Duoji.h"
#include "Dianji.h"

#pragma LINK_INFO DERIVATIVE "mc9s12xs128"

#define row 36
#define line 160
#define middle_line 80
#define middle_angle 5405
#define angle_cal(a,b,c,d,e) a*b+c*d+e
#define full_angle 700
#define xiankuan_min 60

#pragma DATA_SEG DEFAULT

unsigned char huidu[row][line],shuju[line],hangshu=0,h_count,hang,lie;
        int hd[row],width[row],tolerance[row],chazhi[row],mid[row];
unsigned int max_effect_row;
unsigned int left_lose_count,right_lose_count,power_lose_count;
unsigned int sum_hd;
        int key_a,key_b;
        int old_key_b,d_key_b;
```

```

        int key_near,left_full_angle,right_full_angle;

unsigned char boundary=0;
unsigned char black_found;
unsigned char power_lose,lose_flag;
unsigned char flag0=0,flag1=1;
unsigned char LED_flag=0,LED_delay=0;
unsigned char test_flag=0;
unsigned char start_flag,start_count;
unsigned char stop_flag=0,stop_count=0;
        char h,h1,h2,h3,red;
unsigned char startline_enable;
unsigned char start2=0;
unsigned char speed_changeable;
unsigned char test;
unsigned int test_count,test_value;
unsigned int old_angle;
unsigned int KP_duoji,KD_duoji=30;
unsigned int angle,angle_plus;
        int change_angle;
        float K=8.38,KP_dianji,KI_dianji,KD_dianji;
        int v,speed,old_speed,true_speed;
        int d_speed=0,od_speed=0;
        int u,pwm_u;
        int v_desire,speed_max;
        int speed_full=4000;

```



```

        int speed_err;
unsigned int brake;
unsigned char brake_count,brake_flag;
unsigned char brake1,brake2;
unsigned int brake1_count;
unsigned char straight_brake_count;
unsigned int d_brake;
unsigned char brake_over=0;
unsigned char brake_wait=0;
unsigned char long_curve_limit=0;
unsigned char parten;
unsigned char little_hill=0,hill_count=0;
unsigned char hill_brake,hill_acc=0;
unsigned char curve=0;
unsigned char little_s=0,little_s_flag=0;
unsigned char straight=0,straight_flag=0;
unsigned char long_count,long_straight=0;
unsigned char long_curve=0;

        int mean_tolerance;
unsigned int S_right=0,S_left=0,S_straight=0;

        int S_count=0;
unsigned int C_right=0,C_left=0;

        int C_count=0;

        int straight_max;

#pragma CODE_SEG NON_BANKED

```

```
#pragma TRAP_PROC
```

```
void interrupt 11 isr3(void) //每一个下降沿捕捉一次脉冲
```

```
{  
    TFLG1_C3F =1;  
    v++;  
}
```

```
void interrupt 9 isr1(void) //场中断，下降沿捕捉
```

```
{  
    TFLG1_C1F=1;  
    hang=0;  
    hangshu=0;  
    flag0=1;  
    TIE_C0I=1;  
}
```

```
void interrupt 8 isr0(void) //行中断，上升沿捕捉
```

```
{  
  
    TFLG1_C0F=1;  
  
    if(!hangshu)  
    {  
        if(h_count<=22)  
            hangshu=3;
```

```

else if(h_count<=32)
hangshu=2;
else
hangshu=1;

if(h_count!=0)
{
lie=9;
while(lie--);

for(lie=line;lie>0;lie--)
huidu[hang][lie]=PORTB;
hang++;
}
h_count++;
}
else
hangshu--;

if(hang>=row)
{
TIE_C0I=0;
h_count=0;
}

```

```
}
```

```
void interrupt 15 isr7(void)
```

```
{
```

```
    TFLG1_C7F=1;
```

```
    TC7=TCNT+8000; //设置输出中断时间间隔【可改】
```

```
    speed=v;
```

```
    v=0;
```

```
    if(speed>560) //速度大于阈值，则保持前一刻速度【可改】
```

```
        speed=old_speed;
```

```
        old_speed=speed;
```

```
        true_speed=(int)(K*speed);
```

```
}
```

```
unsigned int absolute(int i) //取绝对值函数
```

```
{
```

```
    if(i<0)
```

```
        i=-i;
```

```
    return i;
```

```
}
```

```
void delay(int i) //延时
```

```

{
    int j;
    while(i)
    {
        j=8000;
        while(j)j--;
        i--;
    }
}

void startline_detect()
{
    if(start_flag && startline_enable && !little_hill)    //起跑线检测
    {
        h1=PORTA_PA1;
        h2=PORTA_PA3;
        h3=PORTA_PA5;
        h=h1+h2+h3;
        if(h>=2)stop_flag=1;
    }

    if(start_flag && startline_enable && !little_hill && start2 && !stop_flag
    && change_angle>=-90 && change_angle<=90 && mean_tolerance>=13 &&
    max_effect_row>=28)
    {
        h1=PORTA_PA1;

```

```

        h2=PORTA_PA3;
        h3=PORTA_PA5;
        if(h1==0 && h2==1 && h3==0)stop_flag=1;
    }
}

```

```

int count_below(int a[],int s,int j)                //统计函数 1
{
    int i=2;
    int t=0;
    for(;i<=s;i++)
        if(a[i]<=j && a[i]>=-j)
            t++;

    return t;
}

```

```

int count_above(int a[],int s,int j)                //统计函数 2
{
    int i=2;
    int t=0;
    for(;i<=s;i++)
        if(a[i]>=j || a[i]<=-j)
            t++;

    return t;
}

```

```
}
```

```
void main(void)
```

```
{
```

```
    int cc[row],dd[row];           //动态变量
```

```
    register int i,j;
```

```
    int t,s,m;
```

```
    int *c,*d;
```

```
    /* put your own code here */
```

```
    /*各种初始化*/
```

```
    init_PLL();
```

```
    init_Duoji();
```

```
    init_Dianji();
```

```
    init_Capture0RTI();
```

```
    init_Capture1RTI();
```

```
    init_Capture3RTI();
```

```
    init_7RTI();
```

```
    /*端口初始化*/
```

```
    DDRA=0x55;
```

```
    DDRB=0x00;
```

```
    DDRE=0xf0;
```

```
    DDRM_DDRM2=0;
```

```
    DDRM_DDRM1=0;
```

```

DDRM_DDRM0=0;
DDRJ_DDRJ7=0;
DDRJ_DDRJ6=0;
DDRM_DDRM5=0;
DDRM_DDRM4=0;
DDRM_DDRM3=0;

/*数据初始化*/
cc[0]=middle_line/2;
dd[0]=line-middle_line/2;
c=cc;
d=dd;
PORTE=0xf0;

/*****档位*****/

    m=0;
    m|=PTM_PTM3;
    if(m==1)
        startline_enable=1;
    else
        startline_enable=0;
    m=0;
    m|=PTM_PTM4;
    m|=PTM_PTM5<<1;
    if(m==3)
    {

```



```

        test=0;
        long_curve_limit=1;
    }
else if(m==2)
    {
        test=1;
        test_value=176;
    }
else if(m==1)
    {
        test=1;
        test_value=58;
    }
else
    {
        test=0;
        long_curve_limit=0;
        start2=1;
    }

m=0;
m|=PTJ_PTJ6;
m|=PTJ_PTJ7<<1;
if(m==3)
    angle_plus=2;
else if(m==2)

```

```

        angle_plus=1;
else if(m==1)
        angle_plus=0;
else
        angle_plus=-1;

m=0;
m|=PTM_PTM0;
m|=PTM_PTM1<<1;
m|=PTM_PTM2<<2;

if(m==7)
{
        v_desire=2200;
        angle=21;
}
else if(m==6)
{
        v_desire=2100;
        angle=20;
}
else if(m==5)
{
        v_desire=2000;
        angle=20;
}

```

```
else if(m==4)
{
    v_desire=1900;
    angle=19;
}
else if(m==3)
{
    v_desire=1800;
    angle=18;
}
else if(m==2)
{
    v_desire=1600;
    angle=16;
}
else if(m==1)
{
    v_desire=1250;
    angle=14;
}
else
{
    boundary=1;
    v_desire=980;
    angle=13;
}
```

```

/*****档位*****/

speed_max=2300;
angle+=angle_plus;

EnableInterrupts;
delay(6000);

for(;;)
{

if(flag0==1 && flag1==1)
{
    flag0=0;
    flag1=0;
    left_lose_count=0;
    right_lose_count=0;

    hd[0]=155;
    s=6;
    /*****阈值分割*****/
    for(i=1;i<row;i+=s)
    {
        sum_hd=0;
        for(j=0;j<line;j++)
            sum_hd+=huidu[i][j];
    }
}
}

```

```

    hd[i]=sum_hd/(line+11);
    if(i<=6)
        hd[i]-=38;
    else if(i<=12)
        hd[i]-=32;
    else if(i<=18)
        hd[i]-=26;
    else if(i<=24)
        hd[i]-=10;
    else if(i<=30)
        hd[i]-=9;
    else hd[i]-=8;

hd[i]=hd[i]>105?hd[i]:105;

    for(j=1;j<s;j++)
        hd[i+j]=hd[i];

    }

/*****閾值分割*****/

/*****取中线*****/

```

```

/*****近场部分*****/
for(i=0;i<=7;i++)
{
    for(j=0;j<line;j++)
    {
        if(huidu[i][j]<=hd[i])shuju[j]=11;
        else shuju[j]=88;
    }
    if(i==0)
    {
        for(j=(c+i)+10;j>0;j--)
        {
            if(shuju[j]==11 && shuju[j-1]==11)
            {
                *(c+i)=j;
                break;
            }
            else *(c+i)=1;
        }
        for(j=(d+i)-10;j<line;j++)
        {
            if(shuju[j]==11 && shuju[j+1]==11)
            {
                *(d+i)=j;
                break;
            }
        }
    }
}

```

```

        else *(d+i)=line-1;
    }

    if(*(c+i)>middle_line+50)
        *(c+i)=middle_line+50;

    if(*(d+i)<middle_line-50)
        *(d+i)=middle_line-50;

    width[i]=*(d+i)-*(c+i);
    mid[i]=(*(d+i)+*(c+i))>>1;
}
else
{
    t=*(c+i-1)+7;
    m=t-14;
    if(m<=1)m=1;
    for(j=t;j>=m;j--)
        if(shuju[j]==11 && shuju[j-1]==11)
        {
            black_found=1;
            break;
        }else black_found=0;

    if(black_found)
        *(c+i)=j;
}

```

```

        else *(c+i)=1;

t=*(d+i-1)-6;
m=t+14;
if(m>=line-1)m=line-1;
for(j=t;j<=m;j++)
    if(shuju[j]==11 && shuju[j+1]==11)
    {
        black_found=1;
        break;
    } else black_found=0;

    if(black_found)
        *(d+i)=j;
        else *(d+i)=line;

width[i]=*(d+i)-*(c+i);
mid[i]=(*(d+i)+*(c+i))>>1;
chazhi[i]=mid[i]-middle_line;
tolerance[i]=mid[i]-mid[0];

}

startline_detect();

}

```



```
/******近场部分******/
```

```
/******中场部分******/
```

```
for(i=8;i<=27;i++)
{
    for(j=0;j<line;j++)
    {
        if(huidu[i][j]<=hd[i])shuju[j]=11;
        else shuju[j]=88;
    }
    t=(c+i-1)+8;
    m=t-16;
    if(m<=0)m=1;
    for(j=t;j>=m;j--)
        if(shuju[j]==11 && shuju[j-1]==11)
        {
            black_found=1;
            break;
        } else black_found=0;

    if(black_found)
        *(c+i)=j;
    else
    {
```

```

        *(c+i)=*(c+i-1);
        left_lose_count++;
    }

t=*(d+i-1)-8;
m=t+16;
if(m>=line)m=line-1;
for(j=t;j<=m;j++)
    if(shuju[j]==11 && shuju[j+1]==11)
    {
        black_found=1;
        break;
    }else black_found=0;

if(black_found)
    *(d+i)=j;
else
    {
        *(d+i)=*(d+i-1);
        right_lose_count++;
    }

width[i]=*(d+i)-*(c+i);

if(width[i]>=(xiankuan_min-i))
{

```

```

max_effect_row=i;
mid[i]=(*(c+i)+*(d+i))>>1;
s=mid[i]-mid[i-1];
if( absolute(s)>(4+i/5) )
{
    mid[i]=mid[i-1];
}
else
{
    chazhi[i]=mid[i]-middle_line;
    tolerance[i]=mid[i]-mid[0];
}

}

else

    power_lose++;                //丢失或可靠性不足行计数

key_near=tolerance[7];

startline_detect();

}

power_lose_count=power_lose;

if(power_lose>=9)                //本场图像丢失或可靠性不足
{

```

```

        lose_flag=1;
    }
    else
        lose_flag=0;

    power_lose=0;

    /*****中场部分*****/

    /*****远场部分*****/
    if(s<3 || !lose_flag)
    {
        for(i=28;i<row;i++)
        {
            for(j=0;j<line;j++)
            {
                if(huidu[i][j]<=hd[i])shuju[j]=11;
                else shuju[j]=88;
            }

            t=(c+i-1)+11;
            m=t-22;
            if(m<=0)m=1;
            for(j=t;j>=m;j--)
                if(shuju[j]==11 && shuju[j-1]==11)
                    {

```

```

        black_found=1;
        break;
    }else black_found=0;

    if(black_found)
        *(c+i)=j;
    else
    {
        *(c+i)=*(c+i-1);
        left_lose_count++;
    }
t=*(d+i-1)-11;
m=t+22;
if(m>=line)m=line-1;
for(j=t;j<=m;j++)
    if(shuju[j]==11 && shuju[j+1]==11)
    {
        black_found=1;
        break;
    }else black_found=0;

    if(black_found)
        *(d+i)=j;
    else
    {
        *(d+i)=*(d+i-1);

```

```

        right_lose_count++;
    }
    width[i]=*(d+i)-*(c+i);

    if(width[i]>=30)
    {
        mid[i]=*(c+i)+*(d+i)>>1;
        chazhi[i]=mid[i]-middle_line;
        tolerance[i]=mid[i]-mid[0];
        max_effect_row=i;
    }else mid[i]=middle_line;

    }
    startline_detect();
}

/*****远场部分*****/

/*****取中线*****/

/*****          选          取          参          考          值          *****/
*****/

/****车身位置判断****/

if(key_near<=-3)left_full_angle=1;
else if(key_near>=3)right_full_angle=1;

```

```

else
{
    left_full_angle=0;
    right_full_angle=0;
}

if(!lose_flag)
{

/*****赛道判断*****/

/****基本赛道类型****/
mean_tolerance=count_below(tolerance,max_effect_row,3);

if(mean_tolerance>=19 && absolute(tolerance[angle])<=7)
    curve=0;
    else
        curve++;

if(curve>=3)
    straight=0;
    else straight++;
if(straight>=3)
{
    straight_flag++;
    straight_brake_count++;
}

```

```

    }

    else straight_flag=0;

    if(mean_tolerance>=25 && width[22]>=92 && width[23]>=91
        && left_lose_count<=1 && right_lose_count<=1 && !little_hill) //坡道
        little_hill=1;

    if(mean_tolerance>=29) //长直道
        long_count++;
        else long_count=0;

    if( (long_count>=3 || straight_flag>=10) && (left_lose_count<3 ||
right_lose_count<3) )
        long_straight=1;

    S_straight=S_left=S_right=0;

    for(i=2;i<=max_effect_row;i++)
    {
        if(mid[i-1]-mid[i]>1)
            S_left++;
        else if(mid[i]-mid[i-1]>1)
            S_right++;
    }

```



```

else
{
    S_straight++;
    if(i<27)
        straight_max=i;
}
}
S_count=S_right-S_left;

//单向弯道
if( (S_right<=2 || S_left<=2) && absolute(S_count)>=10)
    parten=1;
else
    if(    S_left>=6    &&    S_right>=6    &&    S_straight>=14    &&
max_effect_row>=31
        &&    count_above(chazhi,33,29)<=1    &&    left_lose_count<=1    &&
right_lose_count<=1 )
    {
        little_s++;
        if(little_s>=2)
            parten=2;
    }
    else
        parten=0;

```

```

C_count=C_right=C_left=0;
for(i=2;i<=max_effect_row;i++)           //长弧判断
{
    if(mid[i-1]>mid[i])
        C_left++;
    else if(mid[i]>mid[i-1])
        C_right++;
}
C_count=C_right-C_left;

if( absolute(C_count)>=19 && count_below(tolerance,max_effect_row,30)>=26
    && count_below(tolerance,max_effect_row,5)<=18 ) //长弧
{
    long_curve=1;
    key_b=chazhi[angle+8];
}
else
{
    long_curve=0;
    if(parten==2)
        key_b=chazhi[angle+2];
    else key_b=chazhi[angle];
}

if(key_b>=45)key_b=45;
else if(key_b<=-45)key_b=-45;

```

```

        d_key_b=key_b-old_key_b;
        key_a=absolute(key_b);

if( !(parten==2) )
{
    if(key_a<=6)
        speed_err=(key_a*key_a)/2;
    else if(key_a<=15)
        speed_err=(key_a-2)*key_a;
    else
        speed_err=6*key_a+115;
} else
    speed_err=180;

    speed_err=speed_err>330?330:speed_err;

/*****赛道判断*****/
}

if(!stop_flag)
{
    /*****电 机 控 制*****/
    *****/
    if(!lose_flag)
    {

```

```

    if(brake_over)
        pwm_u=1000;
    else
        pwm_u=v_desire-speed_err;

        if(boundary && long_straight)pwm_u-=80;
        else if(long_curve && long_curve_limit)
            pwm_u-=150;
            else if(pwm_u>speed_max)pwm_u=speed_max;
    }
else
    pwm_u=v_desire-400;

d_speed=pwm_u-true_speed;
i=d_speed-od_speed;

if(key_a<=18)
    KP_dianji=1.3;
    else KP_dianji=1.8;

    KD_dianji=0.8;

if(d_speed<-150 && !little_hill)
{

```

```

    KP_dianji=2.3;
    KD_dianji=1.0;
}

u=(int)( KP_dianji*d_speed+KD_dianji*i );
    pwm_u+=u;

if(long_straight)
    if( count_above(tolerance,34,9)>=3 && !boundary)
    {
        brake1=1; //长直道入弯刹
车
        long_straight=0;
    }

if(brake1)
{

    if(v_desire>=1700)
        pwm_u=0;
    else pwm_u/=2;

    if( S_straight>=14 && count_below(tolerance,angle+7,5)>=13 &&
parten==1 && absolute(tolerance[25])>=10

```

```

        && count_above(tolerance,max_effect_row,26)>=3 &&
speed>(v_desire/12) && !boundary && !brake_flag)
    {
        brake_flag=1;
    }
}

else if( S_straight>=14 && count_below(tolerance,angle+7,5)>=14 &&
parten==1 && absolute(tolerance[24])>=10
        && count_above(tolerance,max_effect_row,27)>=3 &&
speed>(v_desire/11) && !boundary && !brake_flag )
    {
        brake_flag=1;
        d_brake=speed;
    }

else if( long_curve && (left_lose_count>3||right_lose_count>3) &&
speed>(v_desire/11)
        && !boundary && !brake_flag && start_flag )

    {
        brake_flag=1;
        brake2=1;
    }

```

/***/刹车***/

```

if(brake_flag)
{
    if(speed<v_desire/19 || brake_count>4)
    {
        brake=0;
        brake1=brake2=brake_flag=0;
        brake_over=1;
        straight_brake_count=0;
    }
    else
    {

        if(brake1)
        {
            brake=v_desire*3-2800;
        }
        else if(brake2)
        {
            pwm_u/=4;
            brake=v_desire;
        }
        else
        {
            pwm_u=(v_desire/6+straight_brake_count*50);
            brake=straight_brake_count*(d_brake/6);
        }
    }
}

```

```

    }
}
else
    brake=0;
if(start_count<35)brake=0;
else if(hill_brake && v_desire>=2000)pwm_u/=2;

if(    (pwm_u>=speed_full    ||    start_count<34    ||    hill_acc)
&& !boundary)pwm_u=speed_full;
else if(pwm_u<=0 || brake1)pwm_u=0;

PWMDTY01=pwm_u;
PWMDTY23=brake;

/*****          电          机          控          制
*****/
}else
{
    stop_count++;
    if(stop_count<=2)
    {
        PWMDTY01=4000;
        PWMDTY23=0;
    }
else

```



```

{

    if(speed<60 || stop_count>=120)
    {
        PWMDTY01=0;
        PWMDTY23=0;
    }
    else
    {
        PWMDTY01=0;
        PWMDTY23=v_desire+100;
    }
}
}

```

```

/***** 舵 机 控 制 *****/
*****/

```

```

if(!lose_flag)
{
    i=d_key_b;

    if(i>7)i=7;
    else if(i<-7)i=-7;
}

```

```

if(v_desire>=2000)

```

```

{
    s=8;
    m=12;
    t=150;
}
else if(v_desire>=1600)
{
    s=5;
    m=8;
    t=260;
} else
{
    s=3;
    m=5;
    t=320;
}

if(key_a<=13)
{
    KP_duoji=(2*key_a)/s;
    t=0;
}
else if(key_a<=24)
{
    KP_duoji=key_a-m;
    t=0;
}

```

```

    }
    else
    {
        KP_duoji=key_a/4;
        if(key_b<0)
            t=-t;
    }

    KP_duoji=KP_duoji<4?4:KP_duoji;

    if(parten==1 && !brake_flag && v_desire>=2000)KP_duoji+=3;
        else if(brake_flag) KP_duoji+=(true_speed/220);

    change_angle=angle_cal(KP_duoji,key_b,KD_duoji,i,t);
    if(stop_flag)
        change_angle/=2;
    else if(parten==2)
        change_angle/=4;
        else if( count_above(chazhi,33,29)<=1 && (left_lose_count<3
|| right_lose_count<3)
&& !boundary && start_flag &&
count_above(chazhi,max_effect_row,5)>=5)
            change_angle=(2*change_angle)/3;

    i=middle_angle+change_angle;

```

```

        old_key_b=key_b;
    }
    else if(left_full_angle)i=middle_angle-690;
        else if(right_full_angle)i=middle_angle+690;
            else i=old_angle;

        if(i-full_angle>middle_angle)i=middle_angle+full_angle;
        else if(i+full_angle<middle_angle)i=middle_angle-full_angle;

    PWMDTY45=i;
    old_angle=i;

    /*****                               舵           机           控           制
    *****/

    /*****出发标志*****/
    if(!start_flag)
    {

        if(speed>20)start_count++;

        if(start_count>=100)
            start_flag=1;
    }

```

```

    }
    if(start_flag && test)
        test_count++;
    if(test_count>test_value)
        stop_flag=1;

    /***处理标志***/

    //速度差值存储
    od_speed=d_speed;

    if(key_a>=20 && long_straight)                //长直道入缓弯不刹车标志
        long_straight=0;

    if(brake_over)                                //刹车后限速处理
    {
        brake_count=0;
        speed_full=1200+brake_wait*270;
        brake_wait++;
    }else speed_full=4000;

    if(brake_wait>=12)
    {

```

```

    brake_wait=0;
    brake_over=0;
}

if(brake_flag || brake2)
    brake_count++;
    else brake_count=0;

if(brake1)
    brake1_count++;

if(brake1_count>=18 || speed<(v_desire/19) )
    brake1=0;

if(little_hill)
{
    hill_count++;
    if(hill_count<=17)
        hill_acc=1;
    else if( hill_count>=18 && hill_count<=(55-v_desire/100) )
    {
        hill_brake=1;
        hill_acc=0;
    }
    else hill_brake=0;
}

```

```

    if(boundary && hill_count>=140)
        little_hill=0;
    else if( hill_count>=(75-v_desire/100) )
        little_hill=0;

    curve=curve>30?30:curve;
    straight_flag=straight_flag>30?30:straight_flag;
    straight=straight>30?30:straight;
    stop_count=stop_count>150?150:stop_count;
    long_count=long_count>30?30:long_count;
    straight_brake_count=straight_brake_count>60?60:straight_brake_count;

    flag1=1;
    /**处理标志***/

    }

    startline_detect();

    }/* wait forever */

    /* please make sure that you never leave this function */
}

```