

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: 浙江海洋学院东海科学技术学院

队伍名称: 超音速队

参赛队员: 周佐谕

沈 市

陈良园

带队教师: 刘娟意

李良儿

关于技术报告和研究论文使用授权的说明

本人完全了解第七届“飞思卡尔”杯全国大学生智能汽车竞赛关保留、使用技术报告和研究论文的规定，即：参赛作品著作权归参赛者本人，比赛组委会和飞思卡尔半导体公司可以在相关主页上收录并公开参赛作品的设计方案、技术报告以及参赛模型车的视频、图像资料，并将相关内容编纂收录在组委会出版论文集中。

参赛队员签名：_____

带队教师签名：_____

日 期：_____

摘 要

本文所设计的智能车以 MC9S12XS128 单片机为核心控制单元，使用 CCD 黑白摄像头检测赛道，通过区域搜索和灰度差值采集赛道两边的黑线，通过求平均的方式来确定中心引导线，计算与中心偏离的位置来区分不同的赛道，同时将偏离程度作为一种控制量，实现对舵机的 PID 方向控制，使用光电编码器检测得到的速度，结合赛道的信息，实现电机的 PID 速度控制。使用 LABVIEW 软件编写了上位机程序，使用无线蓝牙串口模块等工具来进行车模调试。通过大量的数据分析和测试，结果表明该系统设计方案可行。

关键字：智能车；MC9S12XS128；CCD，PID，LabVIEW

Abstract

Smart car is designed using MC9S12XS128 microcontroller as the core control unit and black and white CCD camera to detect circuit, which determined by averaging the position of the black line on both sides of track acquired by calculation of the gray level difference and the regional search. The deviation between the center line and calculation guide line is used to distinguish the different tracks at the same time the level of deviation as a control variable, on the direction of the servo PID control. The speed measured by encoders and the track information are as the control variable to control motor speed with PID. LABVIEW software is used to prepare a host computer program, using the wireless Bluetooth serial modules and other tools to finish Cars debugging. By a large amount of data analysis and test results show that the system design is feasible.

Key Words: Intelligent Car; MC9S12XS128; CCD camera; PID; LabVIEW;

目 录

摘 要.....	I
Abstract.....	II
引 言.....	1
第一章 智能车整体系统设计.....	2
1.1 系统概述.....	2
1.2 车模整体布局.....	2
第二章 智能车机械系统设计.....	4
2.1 舵机的安装.....	4
2.2 前轮的调整.....	4
2.3 底盘和重心的调整.....	5
2.4 摄像头的安装.....	5
2.5 编码器的安装.....	6
2.6 小结.....	7
第三章 智能车硬件系统设计.....	8
3.1 单片机模块.....	8
3.2 电源模块.....	9
3.3 图像采集模块.....	10
3.3.1 摄像头简介.....	10
3.3.2 视频信号分离电路.....	11
3.3.3 模数转换电路.....	12
3.4 电机驱动模块.....	14
3.5 调试模块.....	15
3.6 小结.....	15
第四章 智能车软件系统设计.....	16
4.1 系统总体框图.....	16
4.2 图像采集.....	16
4.2.1 摄像头工作原理.....	16
4.2.2 图像数据采集流程.....	17
4.3 图像处理.....	18
4.3.1 辨别左右黑线.....	18
4.3.2 提取左右黑线.....	19
4.4 赛道识别与特殊标记识别.....	24
4.4.1 起跑线识别模块.....	24
4.4.2 十字弯识别模块.....	25
4.5 PID 控制原理.....	26
4.5.1 位置式 PID.....	27
4.5.2 增量式 PID.....	27
4.6 舵机控制.....	28

4.6.1 舵机的工作原理	28
4.6.2 舵机的 PID 控制	28
4.7 电机控制	29
4.8 小结	31
第五章 系统开发及调试工具	32
5.1 开发工具	32
5.2 串口调试	32
5.3 无线调试模块	33
5.4 智能车上位机设计	34
5.5 小结	35
第六章 智能车的主要参数统计	36
结 论	37
参考文献	39
致 谢	40
附录 1 电路原理图	41
附录 2 程序源代码	42

引 言

汽车在走过的 100 多年的历史中,从未停止过智能化的脚步。进入 20 世纪 90 年代后,汽车市场竞争日益加剧,加上智能运输系统的兴起,国际上对于智能汽车相关技术的研究越来越感兴趣,一大批有实力有远见的大学、大公司和研究所在很早之前就开始了该方面的相关研究^[1]。日本、美国、欧洲等国家都非常重视并且积极开展智能车系统的研究工作,相关研究所进行了大量实地实验,取得了诸多成就。我国的研究工作在很早之前也已经开展,并且也取得了不少的成就。比如:清华大学远在 1980 年就成立了国内最早的汽车研究所,并且在汽车主动避撞、远程导航、车载微机等诸多方面进行了深入研究。2000 年智能交通系统进入实质性阶段,国防科技大学成功研制出第四代无人驾驶汽车^[1]。并且西北工业大学、吉林交通大学、重庆大学等诸多高校也展开了广泛研究。2011 年 7 月,一辆由国防科技大学自主研发的无人驾驶汽车,完成了全程长达 286 公里的高速无人驾驶实验,这项试验刷新了我国自主研发的无人车在复杂的交通路况下自动驾驶的新纪录。标志着我国无人车在识别复杂环境、行为决策和控制等方面实现了技术突破,并跻身到世界一流水平。

无论是军事还是民用上,结合了传感技术和自动驾驶技术的智能车系统应用前景十分广阔。智能驾驶系统可完成恶劣环境下的行使任务,如能见度低,环境中存在有毒有害气体等。同时在减少交通事故、发展自动化技术、提高舒适性等方面具有重要作用。

“飞思卡尔”杯智能车比赛是教育部为了加强大学生实践、创新能力和团队精神的培养而举办的面向全国大学生的智能汽车比赛。“飞思卡尔”杯智能车大赛从 2006 年开始已经举办了六届。

智能车比赛中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,所涉及的专业知识包括控制、模式识别、传感技术、汽车电子、电气、计算机、机械等多个学科,智能小车系统将为智能汽车提供良好的技术试验平台,智能车比赛培养了大学生自主学习,不同学科之间知识的融会贯通、动手实践、探索研究等方面的能力,一定程度上推动着汽车电子技术、智能自动化技术等行业的发展。

第一章 智能车整体系统设计

1.1 系统概述

智能车系统以 MC9S12XS128 微控制器^[2]作为核心控制单元，选择动态性能较好的 CCD 视频传感器检测赛道信息，以获得较高质量的图像，并结合片外模数转化芯片 TLC5510 和视频分离芯片 LM1881 的配合使用来完成图像采集。利用跟踪算法进行黑线的提取，识别赛道。并通过光电编码器检测实时车速，使用 PID 控制算法调节转向舵机的角度和驱动电机的转速，并根据前方道路类型和当前车体的速度及位置误差，对舵机转角和驱动电机 PWM 波的占空比进行及时的调整，实现对智能车的运动方向和运动速度的闭环控制。系统按照功能主要划分为：电源模块、摄像头模块，视频处理模块、直流电机驱动模块、转向舵机模块、速度测量模块、辅助调试模块等。系统框图如下所示。

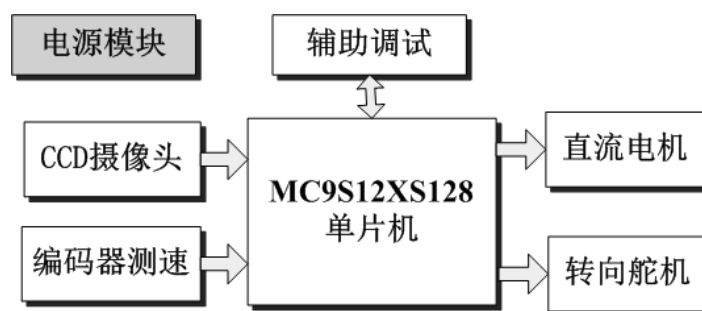


图1.1 系统框图

1.2 车模整体布局

总体而言，智能车系统可分为软件、硬件和机械三大块。软件是整个智能车的灵魂所在。优秀的控制算法，合理的架构和紧凑的语言是智能车竞赛比拼的关键之处。硬件可以说是整个智能车的大脑所在。硬件是灵魂所附着的栖息地，一个完善的硬件系统可以提供尽可能少的外界干扰。可以使软件系统运行在一个高稳定，低噪声的环境下，使车模更为稳定可靠。而机械是整个智能车的躯体所在。即使有时刻保持清醒的大脑和高执行力的灵魂，也必须要配置一具强壮的身躯，没有强壮的身躯，就算有健全的软件和硬件也无济于事。只有三者的完美结合才能将软件，硬件和机械各个分部分有效的链接在一起，将性能发挥至极致。

本次比赛摄像头组采用的是 A 型车模，布局时本着轻便，可靠的原则，尽量降低小车的重心，主板尽量低位放置，底盘也尽量放低，采用强度高，质量轻的碳纤维材料作为摄像头支架。小车的整体布局如下所示。



图1.2 车模前视图

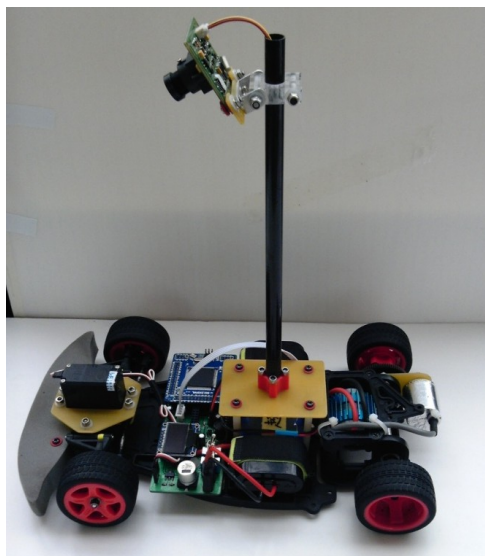


图1.3 车模侧视图

第二章 智能车机械系统设计

本次大赛指定的摄像头组车模为 A 车模, 车模型号 G768, 生产商为东莞博思电子数码科技有限公司。所使用的电机型号为 RS380-ST/3545。转向舵机型号为 FUTABA3010, 在 6V 工作电压下速度为 $0.16\text{sec}/60^\circ$, 扭矩为 $6.5\text{kg}/\text{cm}$, 外形参数为 $40 \times 20 \times 38\text{mm}$, 重量为 41g , 内部为尼龙齿轮。车模机械加工主要分为舵机支架、编码器支架和摄像头支架三个部分。将舵机进行卧式安装, 将光电编码器安装于后轮前方, 直接与电机齿轮咬合, 将摄像头竖直安装于车模中心位置。

2.1 舵机的安装

将舵机采用卧式安装, 并加长舵机输出力臂。由于舵机转动存在时间延迟, 舵机的响应速度影响车模的转弯的灵敏性, 进而影响车模过弯的速度, 这里延长舵机臂至 55mm , 舵机只需转动很小的角度便可得到前轮较大的转角, 提高了舵机的响应速度, 使得转向灵敏。

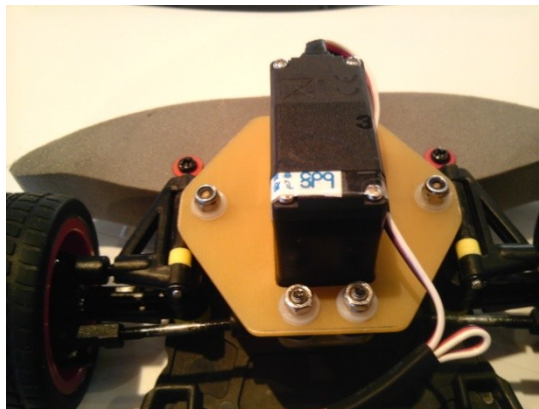


图2.1 舵机的安装

2.2 前轮的调整

参考往届的技术报告和实际的测试, 以转向轻便和减少轮胎磨损为原则, 调整前轮位置, 也就是调整主销内倾角、主销后倾角、前轮外倾角和前轮前束, 尽可能降低转向舵机负载, 保障车模的稳定性。

主销内倾角：前轮主销在车模水平面内向内倾斜的角度。虽然增大内倾角也可以增大回正的力矩，但同时车模转向的过程中，增加了赛车与路面的滑动，从而加速轮胎的磨损。

主销后倾角：前轮主销与前轮垂直中心线之间的夹角，也就是主销上端向后倾斜的角度。

前轮外倾角：前轮的上端向外倾斜的角度。若前面两个轮子呈现“V”字形则称正倾角，呈现“八”字则称负倾角。由于前轮外倾可以抵消由于车的重力使车轮向内倾斜的趋势，减少赛车机件的磨损与负重。

前轮前束：指前轮前端向内倾斜的程度。当两轮的前端距离小后端距离大时为内八字，前端距离大后端距离小为外八字^[3]。

根据实际的小车情况，经过反复的测试，最终选取合适的值，保证车模行使平稳，转向迅速轻便。

2.3 底盘和重心的调整

由赛车比赛可见，降低底盘的高度有利于提高速度，同时压低赛车的重心可防止高速过弯时翻车。对于车模，以顺利过坡为前提，尽可能降低底盘的高度。

重心尽量低且尽可能位于车模的中心位置。因为车身重心前移，有利于转向，但降低转向的灵敏度（因为大部分重量压在前轮，转向负载增大），同时降低后轮的抓地力；重心后移，不利于转向，但增大转向灵敏度，后轮抓地力也会增加。调整合适的车身重心，在过弯时，使其前后轮的侧向摩擦力大体相当，提高过弯性能和稳定性^[4]。

2.4 摄像头的安装

摄像头安装过高，视野较宽，但是由于呈梯形，远处的分辨率不足，而近处的盲区增加，实际意义不大，同时受到的干扰和抖动都变得很强烈。安装过低，视野变小，前瞻不足，而且容易反光。在过急弯时“丢线”现象变得严重。摄像头支架安装在模型车的前与后影响也很大，安装过于靠前，过弯时摄像头会抖得厉害，稳定性能较差。综合以上因素，经过实际的调试，确定了摄像头的安装位置和高度。将摄像头安装在模型车的中间稍微靠前一点，用碳纤维的

材料做支架，质轻且刚度较好，且抖动不强烈。安装的高度大概在距底板 30cm 的位置。为减少抖动，连接摄像头的单杆的固定异常重要，连接方式如下图所示。

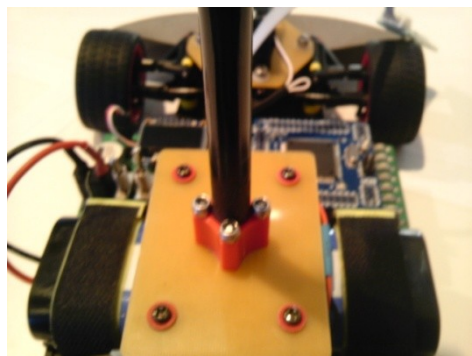


图2.2 摄像头单杆与底盘的连接方式

2.5 编码器的安装

编码器选择的是光电增量型的旋转编码器 E6A2-C 系列，该编码器线数为 500 线。编码器安装时齿轮通过一个大齿轮传动和电机齿轮直接咬合，减小误差，同时尽量靠近底板以降低重心位置，增强后轮抓地能力。编码器如图所示。



图2.3 光电增量型的旋转编码器

注意编码器与电机齿轮需较好的咬合，避免因车速过快而出现检测不准的现象。保持齿轮轴平行，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，不能有卡住或迟滞现象。如图 2.4 所示。

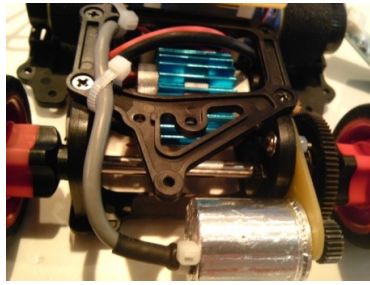


图2.4 编码器的安装

2.6 小结

机械方面的调整是相对比较繁琐的，需要耐心反复的调试，需要慢慢的摸索，只有合理完美的机械结构，才能成就优秀的车模，如前所述，机械结构就如同身躯，只有强健灵活的身躯才能完成大脑发出的各项命令。总而言之，机械部分在竞速中所占比重丝毫不亚于软硬件部分的比重，需要认真对待。

第三章 智能车硬件系统设计

硬件系统主要包括单片机模块、电源模块、图像采集模块、电机驱动模块、调试模块和各个接口电路。以可靠作为前提，尽量简洁，多数采用贴片封装的器件，和分辨率高但体积小的 LED 显示屏，尽可能缩减电路板体积和重量，使车模简洁可靠。PCB 图如下所示。电路原理图见附件 1。

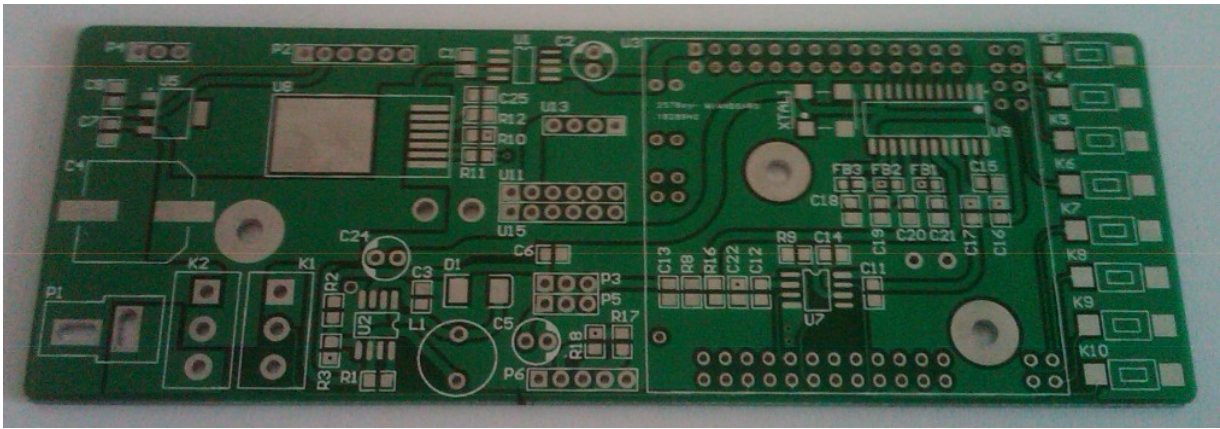


图3.1 主板PCB图

3.1 单片机模块

我们选用大赛推荐的 Freescale 16 位 S12 系列的 MC9S12X128 芯片作为控制核心。基于 S12 CPU 内核，总线频率最高可达 40MHz，并且具备完全的 CAN 功能，改进了中断处理能力。中断有 7 个优先级并且内核支持优先级的调度。单片机最小系统板使用 MC9S12XS128 单片机，112 引脚封装，为减少电路板空间，板上仅将本系统所用到的引脚引出，包括 PWM 接口、计数器接口、外部中断接口、若干普通 IO 接口等。还包括电源滤波电路、时钟电路、复位电路、串行通讯接口和 SPI 接口等。具体的接口配置如表 1 所示。

表1 单片机硬件资源端口分配明细

图像采模块	PA0-PA7 口用于数据接收,选取了 PT1 和 PT2 用于中断信号捕捉,完成数据的采集。
编码器控制	PT7 用于普通 I/O 口,控制编码器工作,记录编码器反馈的速度信息。
舵机控制	选取 PP5 输出 PWM 控制舵机。
电机驱动	利用 PK 口控制 74HC245,从而控制驱动芯片完成电机正反转。
辅助调试	PD 口用于显示屏的控制和 PS 口用于串口调试。
其他	PH 口用于按键操作口。

3.2 电源模块

智能车系统采用配发的标准的车模用的 7.2V 2000mAh Ni-cd 蓄电池进行供电,但各个模块所需要的电压不同,因此需要稳压电路进行电压调节。使用 TPS7350 进行稳压后,输出 5V 的电压供最小系统及其他芯片使用。经过升压电路得到的 12V 电压供摄像头模块使用。转换得到的 3.3V 供液晶显示屏所用,6.9V 电压供舵机使用。具体电源电路如下所示。

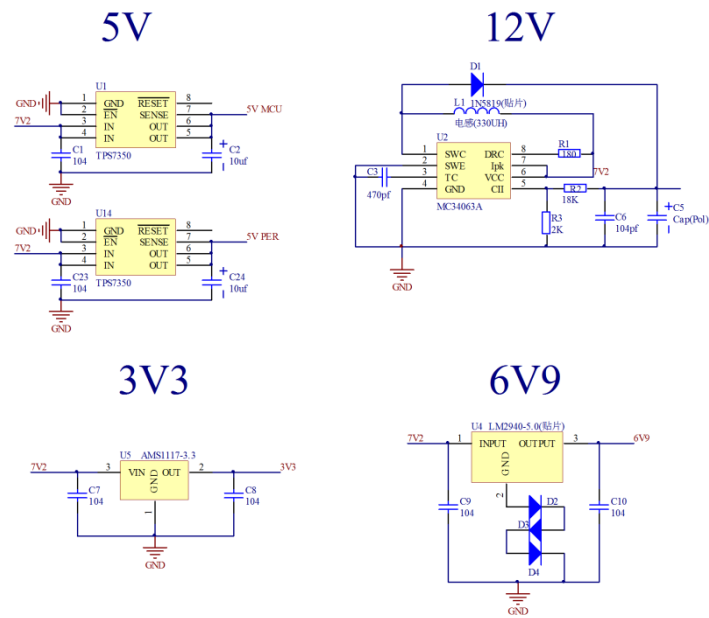


图3.2 电源电路图

3.3 图像采集模块

3.3.1 摄像头简介

一般来讲摄像头分为黑白摄像头和彩色摄像头两种类别。从赛道特点可知，单单想要达到提取黑线的目的，只需要画面中的灰度信息即可，所以在本次设计中采用了 SONY 的黑白 CCD 摄像头^[5]。

在现实生活中，我们所看到的黑白摄像头大体上分为两类：

1：单板摄像头



图3.3 单板摄像头

单板摄像头主要由图像传感芯片、相应的外围电路和镜头构成。图像传感芯片可以说是摄像头的核心，但该芯片也要配以合适的外围电路才能工作。将芯片和外围电路制作在一块电路板上，就称为单板摄像头。

2：摄像头



图3.4 摄像头

若给单板摄像头配上镜头、外壳、引线和接头，就构成了通常所见的摄像头，如聊天用的摄像头。

单板摄像头通常有三个接线端口：

- 1: 电源端 VCC;
- 2: 地端 GND;
- 3: 视频信号端 VIDEO;

CCD 摄像头的电源一般为 12V，而视频信号电压一般在 0.5V-2V 之间。

3.3.2 视频信号分离电路

CCD 摄像头引出的 VIDEO 信号波形如下所示：

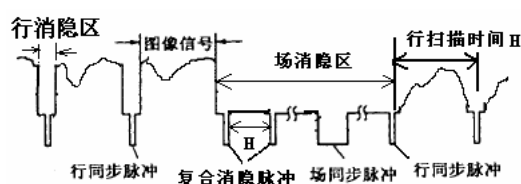


图3.5 VIDEO波形

VIDEO 输出的将是一段连续的电压波形，所以对 VIDEO 信号不做特殊处理的话是无法解决以下两个问题：

- 1: 如何区分一场中各行；
- 2: 如何区分各场；

只有正确的区分出各行与各场，才能将正确的图像数据存放在正确的位置上。所以为了能够正确的区分开这些行场信号，在此，我们对比两种较为普遍使用的方案。

- 1: 使用单片机的片内 AD^[6]提取信号。

因为行场同步信号的电平远远低于摄像头数据信号的电平，所以可以设定一个电平阈值^[6]，用 AD 采样到的电压信号与该电平阈值进行比较，以此来确定行场同步信号。

- 2: 使用外围芯片提取信号。

选用特定的芯片，只要该芯片能够提取出行场同步信号以供单片机控制之用即可。

考虑到单片机捕捉脉冲的速度有限，并且为了减轻其处理负担，所以在本次设计中采用了第 2 种方案进行信号的提取。

通过翻阅资料,最终决定使用视频信号分离芯片 LM1881。LM1881 视频信号分离芯片就可以从摄像头的 VIDEO 信号中提取出行场同步脉冲信号,并将它们转换成 TTL 电平^[6]。这样就可以直接连接到单片机的 I/O 口作控制之用。

利用 LM1881 芯片所搭建的视频信号分离模块原理图如下所示:

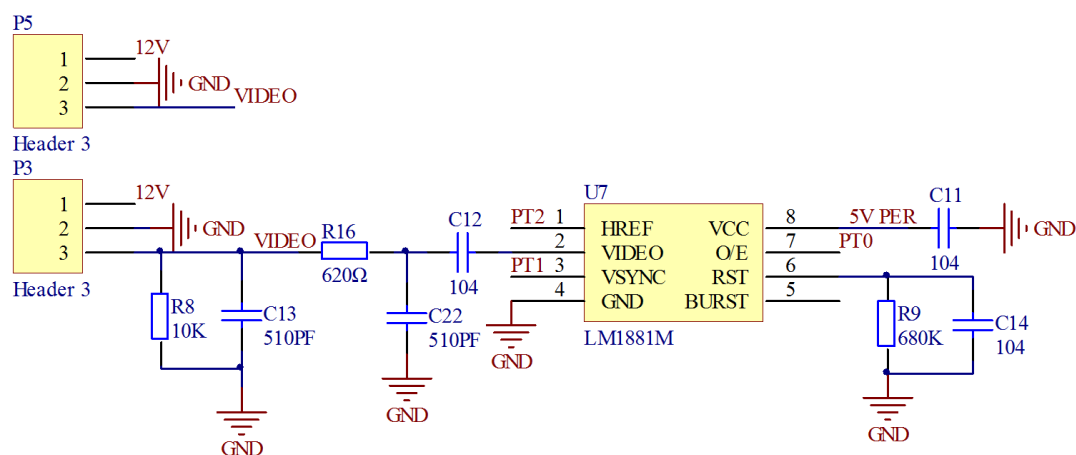


图3.6 视频分离模块原理图

3.3.3 模数转换电路

采集图像时，行场信号只是起到指示的作用，并没有出现真正的数据，其真正的数据是由我们的 ADC^[6]模块来提供的。ADC 模块的主要功能是：对 CCD 摄像头输出的 VIDEO 信号进行模数转换。MC9S12XS128 单片机内部的 ADC 最大的转换速率仅为 2M。为了提高图像的分辨率，所以最终选用了 TI 公司生产的高速率模数转换芯片 TLC5510。TLC5510 是美国 TI 公司生产的新型模数转换^[5]器件。它能并行输出 8 位高阻抗数据，并且最大能提供 20MSPS 的采样速率。

SONY 的黑白 CCD，拍摄其全白和全黑时的 VIDEO 波形如下图所示：

全黑时的波形:



图3.7 VIDEO波形图

全白时的波形：

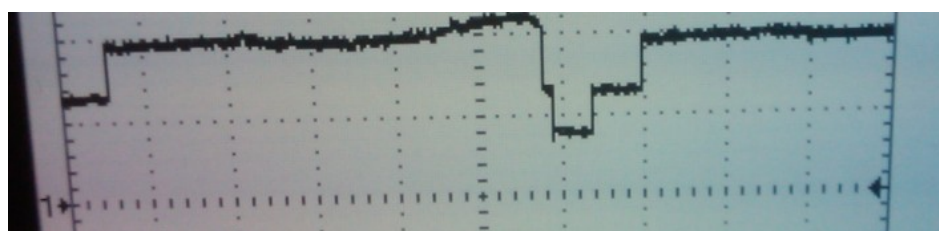


图3.8 VIDEO波形图

从两者的比较中可以发现：

全黑的电压大致为 1V，全白的电压大致为 2V，所以白和黑的压差大致为 1V 左右。所以最理想的情况为参考的波峰电压为 2V，参考的波谷电压为 1V，把 1V 的压差进行 0-255 的分级。

但是 TLC5510I 的满刻度的基准电压为 2V，并且为了安全起见，最终决定使用 0.6V 的参考波谷电压，2.6V 的参考波峰电压，把 0-255 分配在 0.6V 到 2.6V 之间。

最终通过实际测试证明，该设计基本满足采集图像的需要，较为合理。

使用 TLC5510 芯片搭建的 ADC 模块原理图如下所示：

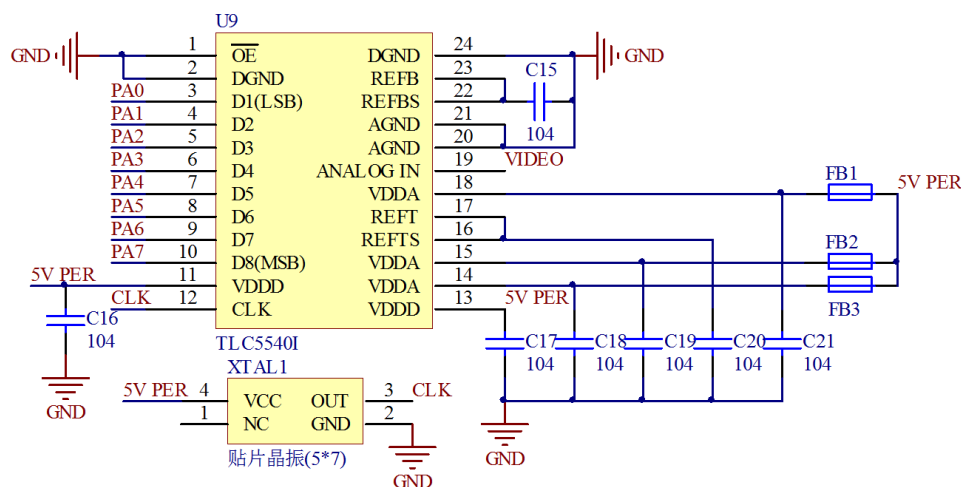


图3.9 ADC模块原理图

3.4 电机驱动模块

单片机输出数字信号，输出电流很小，无法直接驱动直流电动机，因此需要使用驱动电路。为了能让直流电动机在四个象限内工作，需要使用H桥电路。因此，选用BTS7960B芯片进行电机驱动。一片BTS7960B是一个半桥，使用两片BTS7960B芯片便可组成一个H桥驱动电机。电机驱动电路如下所示。

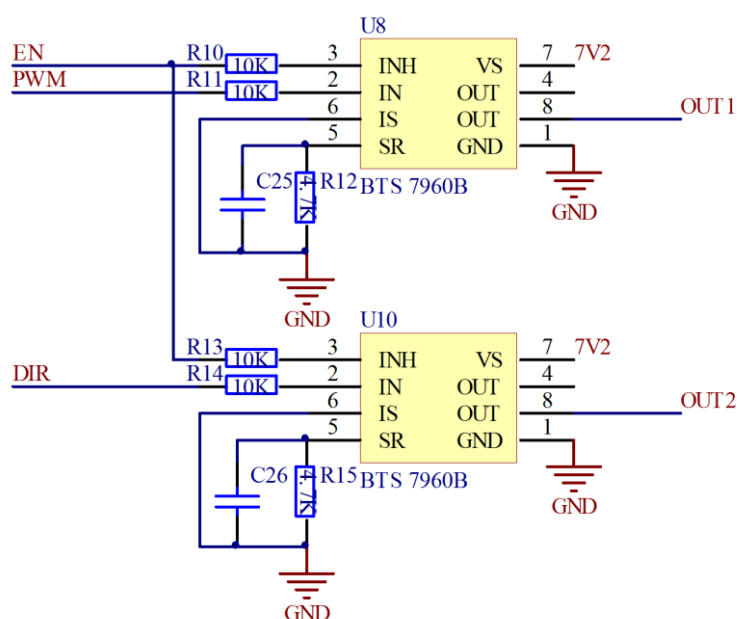


图3.10 电机驱动模块电路

3.5 调试模块

调试电路选用常用的“显示屏+按键”模式。显示屏采用龙丘 OLED 模块，而按键使用贴片式封装。如此便缩减电路板的尺寸，可在一定程度上减轻车模的重量，使车模更加简洁美观。

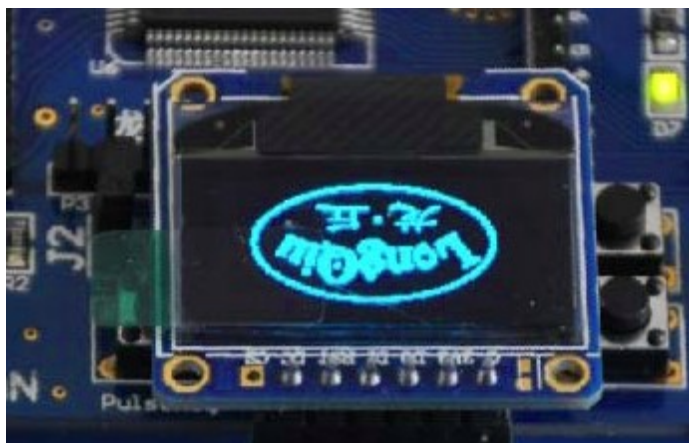


图3.11 OLED模块

3.6 小结

硬件系统设计的首要任务是可靠，在此基础上追求简洁和美观。整个系统中电源部分电路的稳定性尤为重要，电源部分的不稳定容易造成车模行使中出现单片机最小系统复位等异常现象，为了使电源输出能够稳定且纹波小，可以适当的加入滤波电容。另外选用了 74HC245 芯片，在最小系统和驱动之间充当隔离的作用，进一步保证最小系统能够稳定工作。

第四章 智能车软件系统设计

4.1 系统总体框图

智能车系统的机械和硬件系统完成以后，而车模仍无法行使，因为缺乏灵魂。想在比赛中获得佳绩，高效率、高稳定性的软件系统是关键。当然随着智能车竞赛的发展，最终优良的算法将是智能车比拼中至关重要的一个项目。软件系统主要由图像采集、图像处理、路径识别、方向和速度控制五大部分构成，如下图所示。当然还有一些辅助模块，包括 LED 显示模块、按键模块、无线蓝牙收发模块等未在图中画出。此智能车系统通过 CCD 摄像头采集赛道信息，通过光电编码器检测速度，使用局部搜索和求平均值的方式识别赛道信息，使用 PID 算法完成电机和舵机的控制。

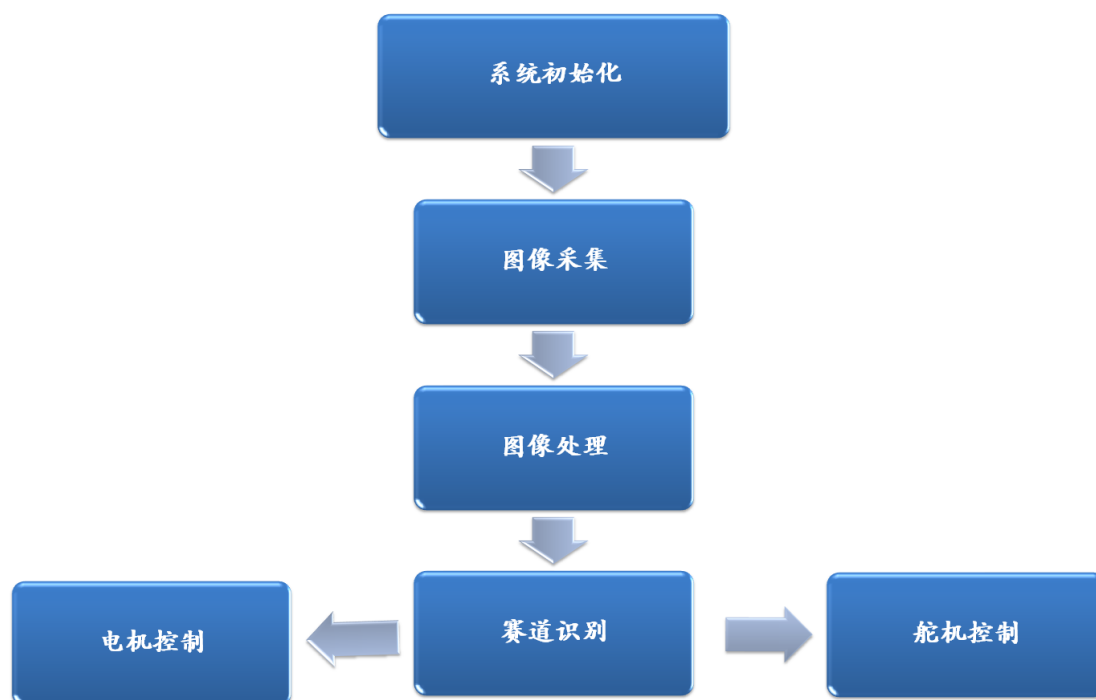


图 4.1 软件系统总体框图

4.2 图像采集

4.2.1 摄像头工作原理

1: 扫描图像过程

它将按照规定的分辨率采集摄像头图像上的点。当扫描到某个像素点时，就通过图像传感芯片将该像素点的灰度值转换成与灰度值相对应的电压值，然后通过 VIDEO 信号线输出该电压值。所以，当摄像头连续扫描图像上的某一行，则输出的就是一段连续的电压信号，并且该电压信号的高低起伏程度就代表了该行图像的灰度变化程度^[7]。

2: 换行或者换场过程

当扫描完上一行，准备扫描下一行时，从上一行末尾到下一行开头的转换过程，称为换行操作。在换行的时间段里 VIDEO 端将输出一个低于最低视频信号电压并且保持一段时间的电平，该电平就称为行同步脉冲，它是扫描换行的标志。

同理，当扫描完上一场，准备扫描下一场时，从上一场末尾到下一场开头的转换过程，称为换场操作。在换场的时间段里 VIDEO 端将输出若干个复合消隐脉冲，其中有个远宽于其它的消隐脉冲，该脉冲就称为场同步脉冲，它是扫描换场的标志。

4.2.2 图像数据采集流程

场图像的分辨率大约为 320 行×582 列，但是在一般情况下我们不会采集所有行的数据，原因如下：1) 单片机的内存有限。MC9S12XS128 的 RAM 仅为 8K，在放置完局部变量、全局变量和静态变量之后，RAM 资源便所剩无几了，所以不可能存储各行的图像数据。2) 相邻行的图像数据基本一致。经图像数据分析发现，相邻行的图像数据相差无几，所以没必要将每行的数据都存放起来。所以在图像采集过程中，一般都采取隔行采集的方案，做法为：隔 5 行采集一行。所以最终把 250 行的图像数据存放在一个二维数组中，比如 `image[50][100]`；

上一个采集行与下一个采集行之间是有一段较大的时间间隔的，足够我们完成一定量的图像处理任务，主程序流程如下：

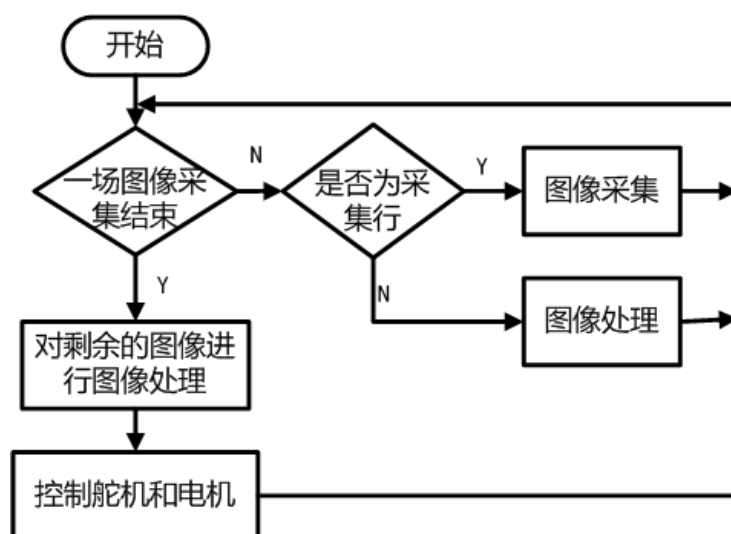


图 4.1 程序流程图

在 16ms 的图像采集过程中就完成了绝大部分的图像处理任务，而 4ms 时间内只要完成极小一部分的图像处理和舵机控制。所以选用这样的图像处理时机，使得在一场图像周期内即能完成图像采集和处理又能完成舵机和电机控制成为可能。

4.3 图像处理

4.3.1 辨别左右黑线

第七届“飞思卡尔”杯全国大学生汽车智能车竞赛跑道的材料和尺寸图如下所示。今年的赛道不同往年，赛道分为了左右黑线。白色的 KT 板两边贴有黑色的胶带，两者的距离为 45cm，并且地板的颜色是不确定的。

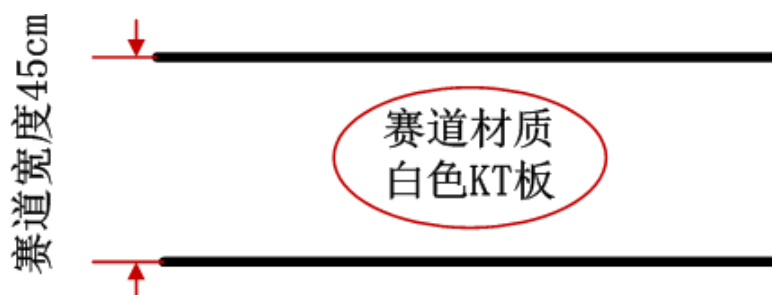


图4.2 跑道材料、尺寸图

辨别右黑线步骤如下：

1、设定搜寻范围。为什么要确定搜寻范围，关于这点，会在下文中着重分析，此处略过。

2、从左往右搜寻右黑线。因为在比赛细则中组委会明确提出：地板的颜色是未知的。如果我们采取从右往左搜寻右黑线的方案，是从地板开始搜寻右黑线，必定会受到地板颜色的干扰，而使用从左往右搜寻右黑线的方案，则是从KT板开始搜寻右黑线，滤除了地板颜色的干扰。所以这点是非常关键的。

3、辨别黑线。从 CCD 摄像头^[5]返回的图像数据中可以明显发现：白色的灰度值较高，黑色的灰度值较低。所以我们可以利用相邻两点灰度值做差来辨别黑线。当差值较大的时候，则很有可能是黑线，但也有一定的可能是噪点，所以有必要在程序段中额外添加黑线宽度这个信息来滤除噪点。

以下是辨别右黑线程序的流程图，辨别左黑线程序的流程图类似。

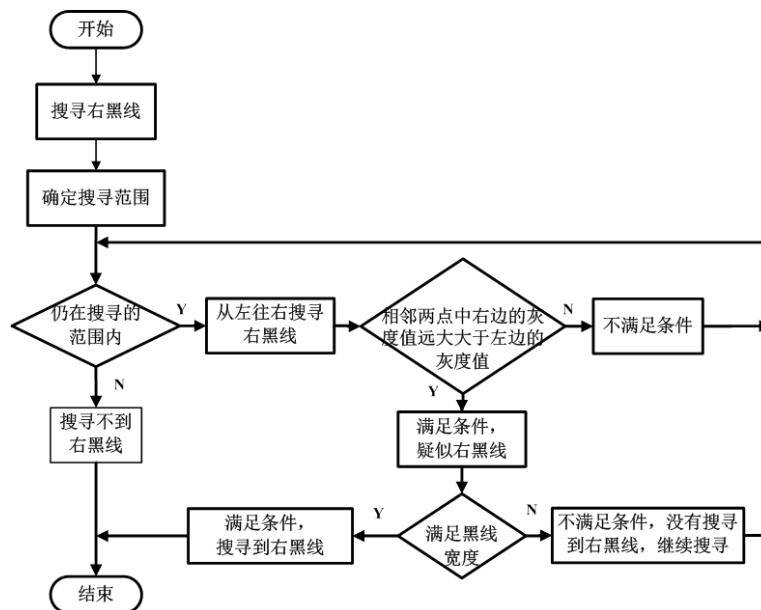


图4.3 右黑线辨别程序流程图

4.3.2 提取左右黑线

根据往年经验，搜寻黑线的方案一般有 2 种，介绍如下：

1、全行扫描算法。采取了整行扫描搜寻黑线的方法，但是在实际运用中发现其具有以下两大缺点：(1)效率低下。在之前的主程序^[10]流程分析中得知：我们要在间隔行中进行图像的处理。一旦采取了整行搜寻黑线的方法，则一行图

像处理所花费的时间极有可能超过行与行的间隔时间，从而达不到我们所设想的要求。(2)噪点多。

利用 CCD 摄像头获取到的图像如下所示。



图4.4 CCD摄像头获取的道路图像

通过观察发现：图像最近端是 45cm，而最远端大概为 5*45cm 的大小，有 4/5 的范围是地板，所以对于我们提取黑线来说有 4/5 的数据是完全无用的。假设我们在最远端也采取整行搜寻黑线的方法，那意味着至少有 4/5 的数据是噪点，在很大程度上会影响我们黑线的提取。

2、跟踪算法

所以我们有必要寻找一种更为有效的黑线提取方案，所以在调试后期我们重新设计了黑线提取方案，即跟踪算法，它在很大程度上弥补全行扫描算法的两大弊端。

(1)一场图像的跟踪算法

根据赛道信息，可以很明显发现一个特点：黑线是连续的。基于该思路我们可以进行如下的设想：假设第一行右黑线的坐标位置为 20，那么第二行的右黑线必定会在 20 附近出现。如果我们在 10 到 30 的范围内从左往右搜寻右黑线，则必定会搜寻到第二行右黑线。一旦第二行右黑线的坐标位置确定了，采取同样的黑线提取策略，就能搜寻到第三行右黑线。以此类推，一场图像的右黑线也便提取完毕了。同理，一场图像的左黑线也能采取同样的提取机理。这种提取黑线算法^[7]具有诸多优点：(1)效率高。可以发现，其搜寻黑线的范围从全行扫描算法的 100 降到了跟踪算法的 20，效率整整提高了 5 倍，使得一行图像处理所花费的时间小于行与行的间隔时间成为可能。(2)噪点少，具有一定的滤波

功能。因为扫描范围只有 20，所以绝大部分滤除了无关的地板和一些图像噪点，使图像处理更为稳定健全。

其流程图如下：

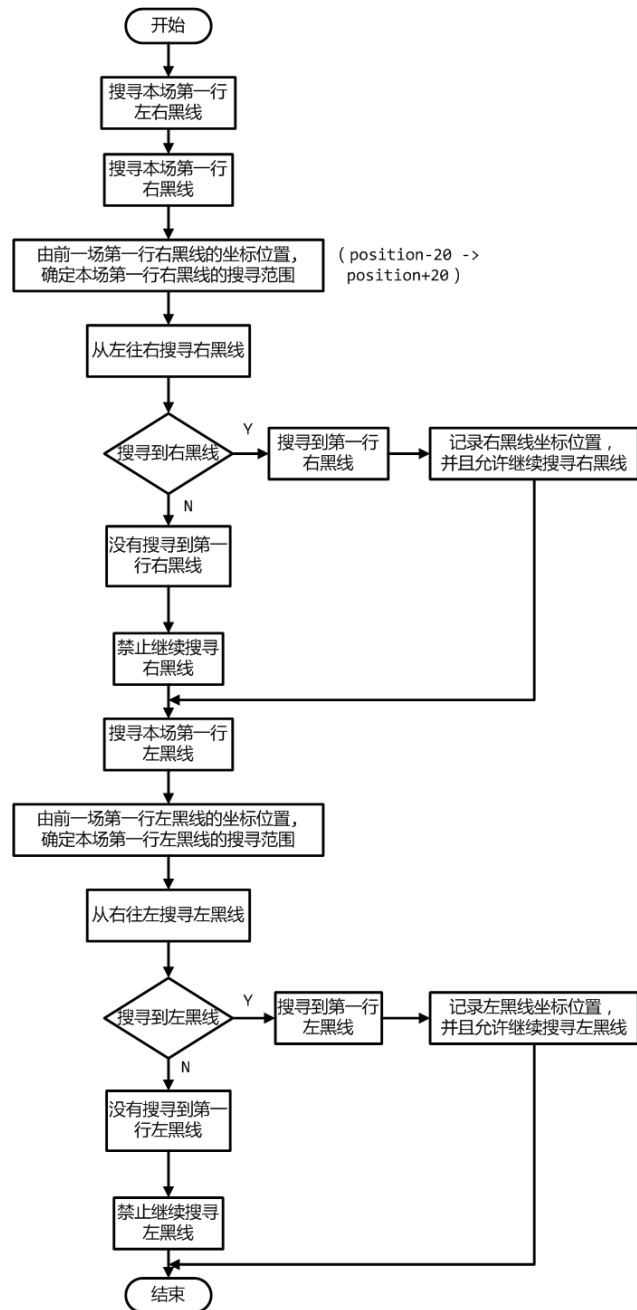


图4.5 一场图像的跟踪算法流程

(2) 各场图像的跟踪算法

虽然一场图像的跟踪算法有诸多优点，但还是有一点的瑕疵：搜寻第一行黑线时并没有参照点，只能进行全行扫描搜寻黑线，所以有必要对上述算法中加以改进。根据车模的速度以及赛道的特征，提出了以下改进方案。因为智能车在 20ms 内不可能跑很长的路程，即使速度为 4m/s，那 20ms 经过的路程也只有 $20\text{ms} \times 4\text{m/s} = 8\text{cm}$ 。所以根据该特征值提出了以下的假设：第一场图像第一行右黑线的坐标位置为 20，因为 20ms 内经过的路程较短，所以可以得知第二场图像的第一行右黑线必定是在 20 附近。如果我们在 0 到 40 的范围内从左往右搜寻右黑线，则必定会搜寻到第二场图像的第一行右黑线。一旦第二场图像的第一行右黑线坐标位置确定了，采取同样的黑线提取策略，就能搜寻到第三场图像的第一行右黑线。以此类推，各场图像第一行的右黑线都得以确定了。

其流程图如下所示：

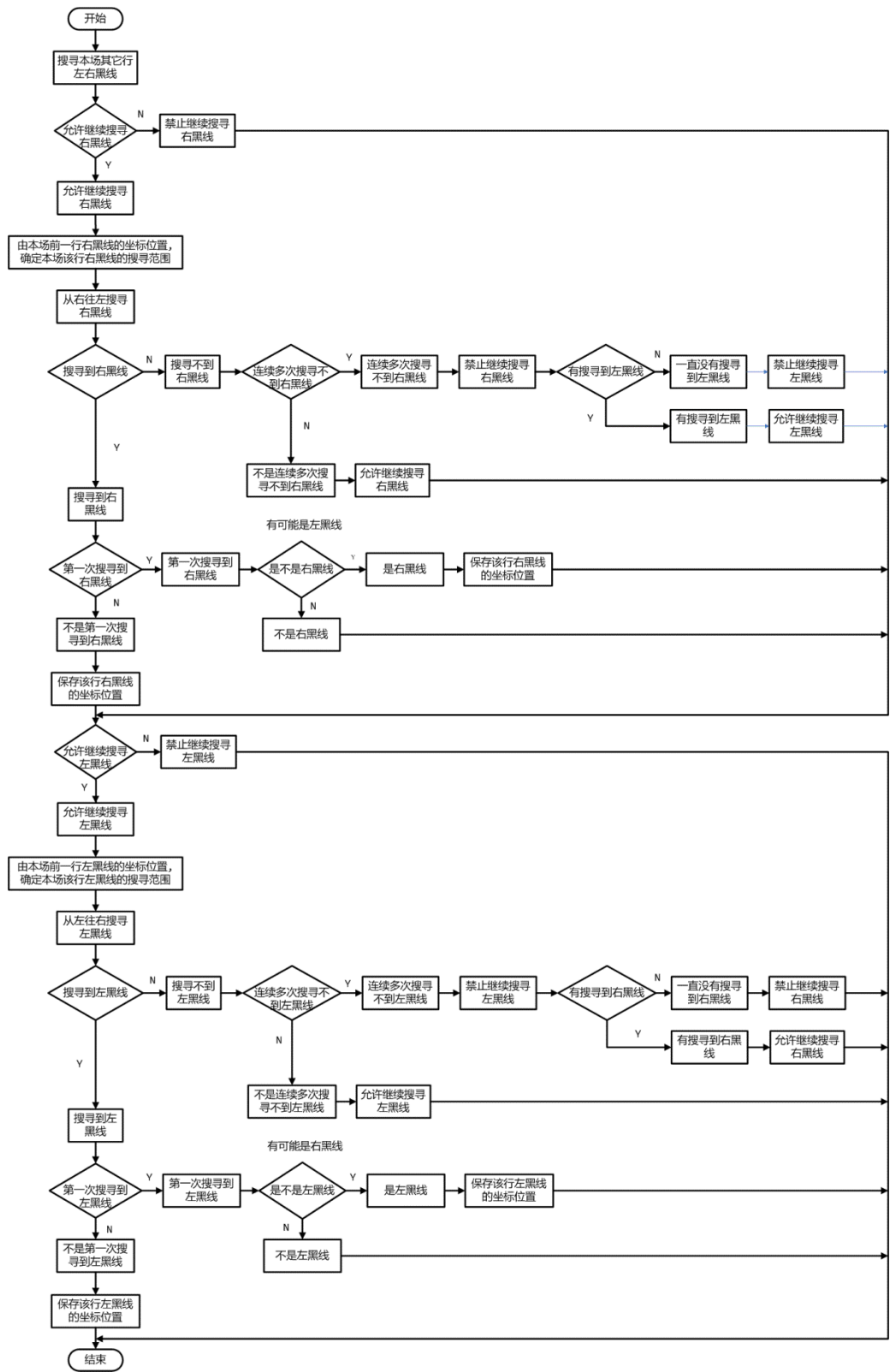


图4.6 各场图像的跟踪算法流程

所以一场图像的跟踪算法结合上各场图像的跟踪算法就形成了一个完整的黑线提取算法，两者是缺一不可。

4.4 赛道识别与特殊标记识别

完成黑线提取以后，通过左右黑线的位置求平均值便可确定中心线位置，将一场中的得到的每一行的中心线位置再一次求平均值以此作为区分赛道的主要标志。使用该方法实现的赛道识别对于小 S 弯非常有利，一般情况下可以完成小 S 弯直接通过。

4.4.1 起跑线识别模块

起跑线的标准如下图所示。车模到达起跑线前的状态有两种可能：1.车模直线进入起跑线，如黑箭头所示。2.车模倾斜进入起跑线，如红箭头所示。起跑线的特征明显。首先从图像的近端行开始从两边黑线路径分别向右和向左用同样的提取方法搜索由黑到白的边缘，继续搜索由白到黑的边缘，再继续搜索由黑到白的边缘，无论如何其中一边的 3 个边缘可被搜索到，最好的情况搜索到 6 个边缘的位置，用 3 个边缘的位置即可计算其中一边的白色空隙和黑块，将白色空隙和黑块的值相加，不管车模的进入状态如何，相加所得的值都非常接近，以此来判定起跑线标记。

起跑线标记黑块的纵向宽度为 25mm，由于采集的行数足够多，图像的纵向分辨率足以满足起跑线占 1 行以上的宽度，无论在图像近端或远端都不会因为分辨率不足导致丢失起跑线的情况出现。因此程序只要在同一行中搜索起跑线即可，不需要在不同的几行中搜索。

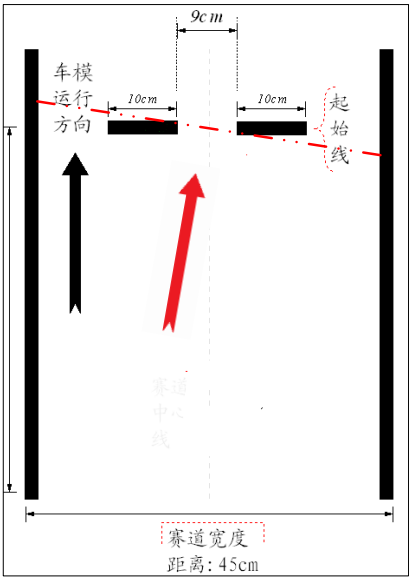


图 4.7 起跑线标记图

4.4.2 十字弯识别模块

与起跑线类似，车模进入十字弯的状态也有直入而斜入两种情况。直入十字弯时，可以采用对十字弯补黑线的方式使车模顺利通过。车模斜入十字弯时(示意图如下所示)，容易造成误判而不同顺利通过。此时，记录黑线的位置可以发现，其中一边黑线位置的远端部分贯穿了视场中心位置，如果连续几场都出现这种情况，则判定出现了十字弯，并将贯穿视场中心位置的远端黑线位置舍弃掉，以此修正路径，通过十字弯。

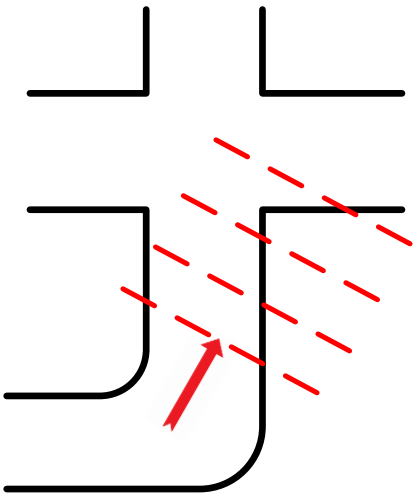


图 4.8 十字弯

4.5 PID 控制原理

PID 控制^[8]为比例、积分、微分控制。PID 控制器是一种线性控制器，它将给定值与实际输出值构成控制偏差的比例(P)、积分(I)和微分(D)通过线性组合构成控制量，来控制被控对象，故称 PID 控制器。实际中也有 PI 和 PD 控制。原理框图如图 4.9 所示。

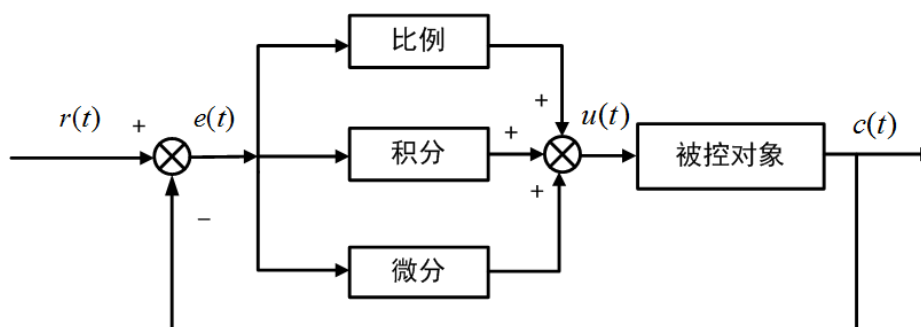


图4.9 PID控制器原理框图

在计算机控制系统中，使用的是数字 PID 控制器，控制规律为：

$$e(k) = r(k) - c(k) \quad (4.1)$$

$$u(k) = K_p \{e(k) + \frac{T}{T_I} \sum_{j=0}^k e(j) + \frac{T_D}{T} [e(k) - e(k-1)]\} \quad (4.2)$$

式中 K ：采样序号， $k = 0, 1, 2 \dots$ ； $r(k)$ ：第 k 次给定值； $c(k)$ ：第 k 次实际输出值； $u(k)$ ：第 k 次输出控制量； $e(k)$ ：第 k 次偏差； $e(k-1)$ ：第 $k-1$ 次偏差； T ：采样周期； K_p ：比例系数； T_I ：积分时间常数； T_D ：微分时间常数。

简单说，PID 控制器^[8]各校正环节的作用如下：

比例环节：成比例反映控制系统的偏差信号，一旦偏差产生，控制器立即产生减少偏差的控制。

积分环节：消除静差，提高系统的无差度。积分时间常数越小，积分作用越强，反之则越弱。

微分环节：反映偏差信号的变化速率，并在该偏差信号变得太大之前，在系统中引入一个有效的早期修正信号，以加快系统的动作速度。

数字 PID 控制算法通常分为位置式 PID 控制算法和增量式 PID 控制算法。

4.5.1 位置式 PID

通常称公式 (4.2) 为位置式 PID 控制算法。位置式 PID 控制算法的缺点：因每次的全量输出均与过去的状态有关，计算时累加过去 $e(k)$ 的工作量大；如计算机出现故障，因计算机输出的 $u(k)$ 对应执行机构的实际位置， $u(k)$ 的大幅度变化将引起执行机构位置的大幅度变化，这在生产实践中是不允许的。

4.5.2 增量式 PID

增量式 PID 是指数字控制器的输出只是控制量的增量 $\Delta u(k)$ 。当需要 $\Delta u(k)$ 时，可由式 (4.2) 推导出提供增量的 PID 控制算式。由式 (4.2) 推出式 (4.3)，式 (4.2) 减去式 (4.3) 得式 (4.4)。

$$u(k-1) = K_p \{e(k-1) + \frac{T}{T_I} \sum_{j=0}^{k-1} e(j) + \frac{T_D}{T} [e(k-1) - e(k-2)]\} \quad (4.3)$$

$$\Delta u(k) = K_p \{[e(k) - e(k-1)] + \frac{T}{T_I} e(k) + \frac{T_D}{T} [e(k) - 2e(k-1) + e(k-2)]\} \quad (4.4)$$

公式 (4.4) 即为增量式 PID 控制算法，可见因控制系统采用恒定的采样周期 T ，一旦 K_p 、 T_I 、 T_D 确定，只需使用前后三次测量值的偏差，即可由式 (4.4) 求出 $\Delta u(k)$ 。

增量式 PID 优点：

- (1) 因计算机输出增量，导致误动作时影响小，可用逻辑判断的方法关掉。
- (2) 手动/自动切换时冲击小，便于实现无扰动切换。发生故障时，执行装置可锁存信号。
- (3) 算式中不需累加。 $\Delta u(k)$ 的确定仅与最近 k 次的采样值有关，可通过加权处理获得较好的

4.6 舵机控制

4.6.1 舵机的工作原理

PWM 信号由接收通道进入信号解调电路进行解调，获得一个直流偏置电压。该直流偏置电压与电位器的电压比较，输出电压差，送入电机驱动集成电路以驱动电机正反转。当电机转速一定时，通过级联减速齿轮带动电位器旋转，直到电压差为 0，电机停止转动。舵机的控制信号是 PWM 信号，利用占空比的变化改变舵机的位置^[9]。

标准的舵机有 3 条导线，分别是：电源线、地线、控制线。电源线和地线用于提供舵机内部的直流电机和控制线路所需的能源。电压通常介于 4~6V，一般取 5V。注意，给舵机供电电源应能提供足够的功率。我们这里取的是 6.9V。控制线的输入是一个宽度可调的周期性方波脉冲信号，方波脉冲信号的周期为 20 ms(即频率为 50 Hz)。当方波的脉冲宽度改变时，舵机转轴的角度发生改变，角度变化与脉冲宽度的变化成正比。脉冲宽度为 0.5ms~2.5ms，对应舵盘的位置为 0~180 度，呈线性变化^[10]。

4.6.2 舵机的 PID 控制

舵机控制采用的 PID 闭环控制，以每场图像黑线提取后求得平均值偏差与舵机 PID 参考角度值构成一次线性关系。若偏离中心线很小，则将 K_p 值置小些，接近零，之所以没有直接给零，希望即使在直道上也有一定的修正能力，如此可以更加稳定。而只要有偏离， K_p 值接近为偏离值的二次函数。车子转弯较流畅，在直道上基本保持直线加速，车身左右抖动较小。由于保证视场范围足够大，前瞻亦足够大，可以覆盖赛道中的各种赛道类型，所以在程序中，并没有对 S 弯、大弯及长直道进行特殊的识别算法，只是根据计算的平均值(中心线的位置)动态改变 PID 参数，便可以得到较好的控制效果。

经不断调试，最终选择一组 PID 参数，得到了较为理想的转向控制效果。

其舵机控制流程图如下图所示：

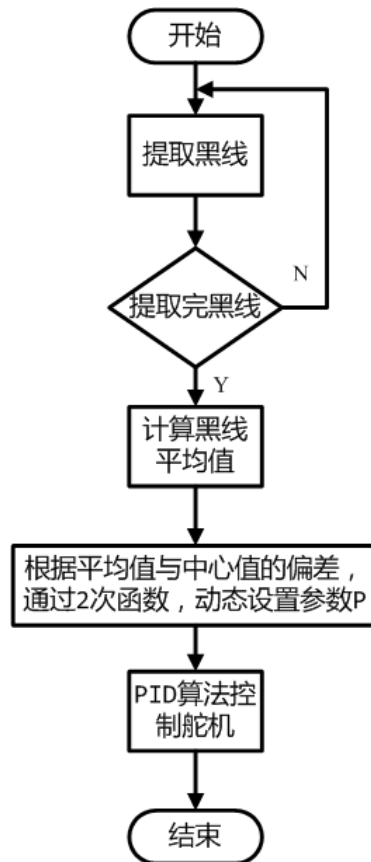


图4.10 舵机控制流程图

4.7 电机控制

电机控制包括电机的 PID 控制策略与速度分配策略。

1、速度的 PID 控制策略：

对于速度控制，最终采用 PID 控制。

2、速度分配策略：

速度分配的基本思想为：直道给全速，弯道中给最低速度。有效前瞻量越远，给定速度越高。所以，要做的是利用好 1.4m 的前瞻量，让车模在入弯前及时减速。测试发现，在入弯和出弯时无法分辨的情况下，入弯减速提前也意味着出弯加速延后。因此，要将车速始终保在最合理的状态需要有一条合理的速度分配曲线。入弯时的目标速度的给定根据车模所处状态不同而不同，若车模位置与中线偏差较大，则由前瞻量来决定需要给定的速度，两者成线性关系；若与中心线偏差较小，则只由偏差来决定目标速度。

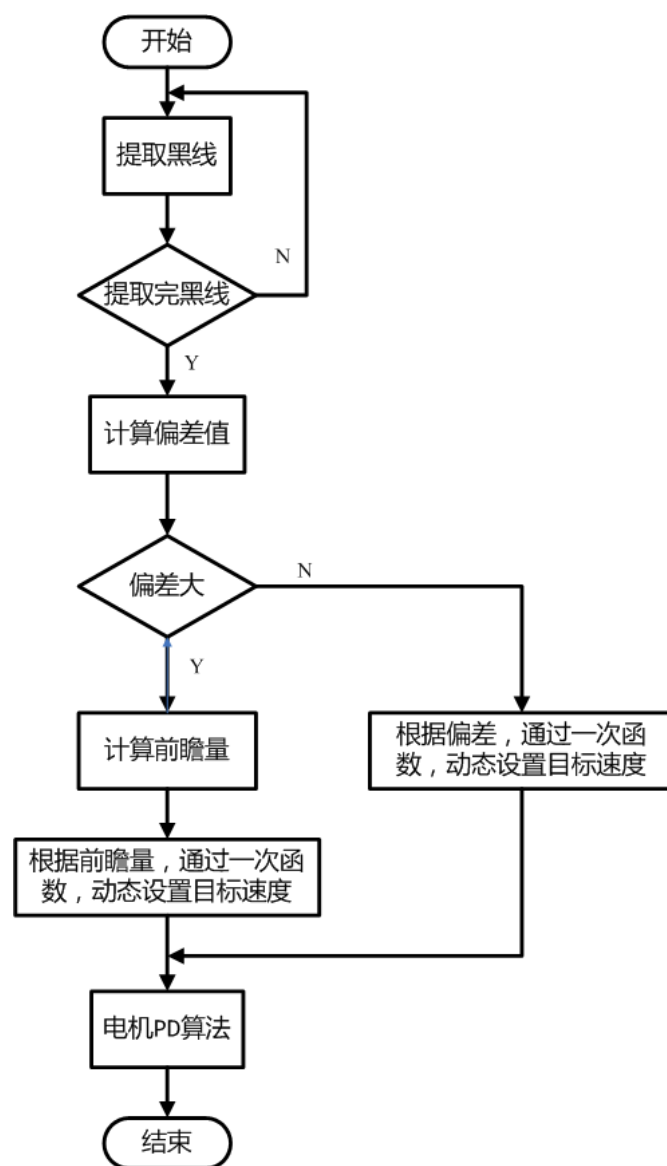


图 4.11 电机控制流程图

速度 PID 控制中必不可少的一个量便是当前的车模的速度值。这就需要通过测速模块来获得。我们选择的是 500 线的光电编码器，通过脉冲计数的方法来实现对小车速度的检测：在靠近小车右轮的轴上装光电编码器，这样当小车前进，车轮转动时，光电编码器跟随车轮同步转动，当一个黑色脉冲被摄像头检测到时，速度传感器的输出就变为高电平，产生脉冲，送给单片机的 PA 模块，PA 模块捕捉脉冲信号并对其进行计数，同样的，当白色被检测到时，也产生一脉冲，送以单片机计数，在一特定时间内(5ms)读出脉冲总数，将该总数除以车

轮转动一圈移过的脉冲数目，便可以计算出车轮的转动圈数，再乘以车轮周长，得到行驶路程，再除以计数时间，最后得到小车的速度。假设 N 为一个采样周期 T 内 ECT 模块记录的脉冲个数， T 为采样周期（单位为 s）， l 为小车后轮周长， s 为小车前进距离， v 为小车的速度， n 为小车车轮转过的圈数，此光电编码器上有 24 个齿轮，后轮上的装有的黑色减速器的齿轮为 76。则：

$$n = \frac{N}{500} \times \frac{24}{76} \quad (4.5)$$

$$s = l \times n \quad (4.6)$$

经过测量，小车后轮直径 $D=50\text{mm}$ ，而采样周期 $T=5\text{ms}$ ，从而速度为：

$$v = \frac{s}{T} = \frac{N}{500} \times \frac{24}{76} \times \frac{\pi D}{T} = 0.0198N \text{ (m/s)} \quad (4.7)$$

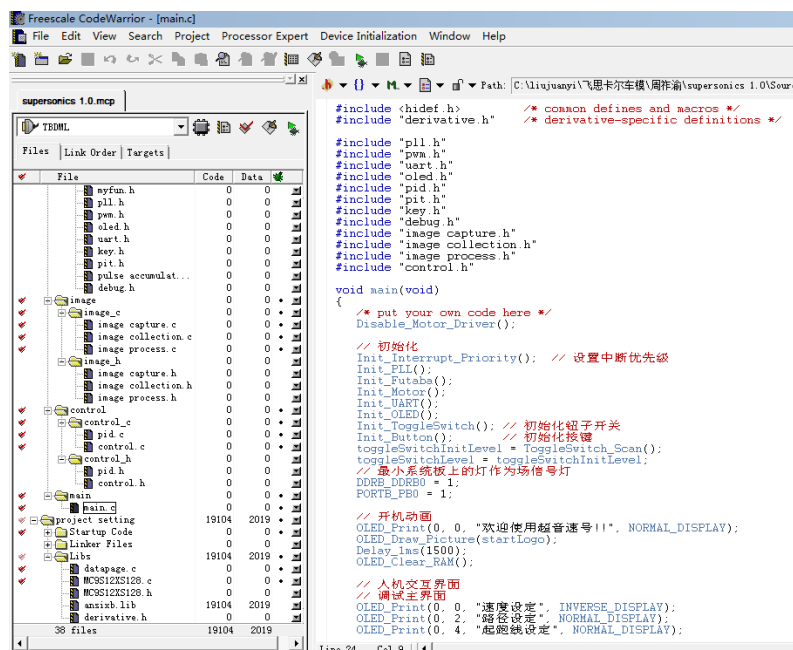
4.8 小结

软件部分的设计的原则为高效稳定。智能车系统的优劣很大一部分取决于软件设计中所选用的算法的合理性、高效性、可靠性。从图象的采样、图像的处理、赛道信息的提取到电机和舵机的控制，每一步都非常关键。不仅如此，PID 参数整定时不仅需要理论知识，更需要对理论知识的深刻理解，结合小车的实际情况，总结经验，耐心调试。

第五章 系统开发及调试工具

5.1 开发工具

开发工具为组委会提供的专门面向 Freescale 所有 MCU 与 DSP 嵌入式应用的 Metrowerks 公司的开发软件工具 CodeWarrior，CodeWarrior 开发环境提供 TBDML 在线调试环境。只要用 BDM 连接 PC 和 MCU，就可以看到单片机内部变量，可进行单步执行等调试操作^[1]。这种调试方法可以全面了解单片机内部变量和其内部寄存器的值，但其缺点是数据有时效性，对时效性很强的变量无法捕获、储存其变化，而且单片机和 PC 的距离被 BDM 的长度限制。其使用界面如下图所示。



5.2 串口调试

利用单片机的 SCI 模块，以及 S0、S1 口可以方便地利用串口通信将数据传给 PC 进行在线调试调试。在前期调试中，我们使用了串口调试助手来进行串口数据的收发。

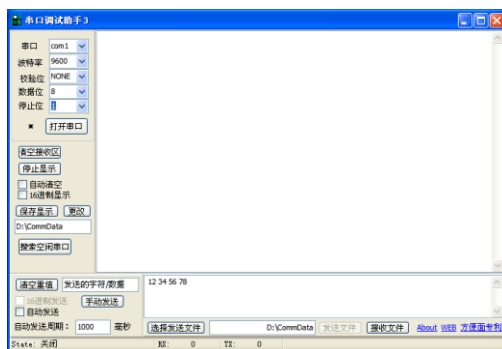


图5.1 串口调试助手

它支持多种串口通讯格式的设置，可以通过相应的下拉菜单进行详细的更改，以配合不同的应用场合。它还支持 16 进制^[12]显示和 ASCII 码显示，应用广泛。所以在简单的串口发送和接收场合中，使用这款软件将大大加快开发进度，减少开发周期。

5.3 无线调试模块

在后期的车模调试过程中，车模是处于一个运动的过程，在传输数据时单片机和电脑必须要用串口线连接起来，也就意味着当我们要采集小车运动时的数据，必须要带着电脑和车模一起跑，这十分不方便。因此采用无线的蓝牙串口[9]对正在跑道上跑的小车进行在线接收数据并离线分析。

无线蓝牙串口模块：无线蓝牙串口模块分为主机模块和从机模块。实物图如下所示。核心模块使用 HC-06 模块。底板设置防反接二极管，带 3.3V LDO，输入电压 3.6~6V，未配对时电流约 30mA，配对后约 10mA，输入电压不得超过 7V；接口电平 3.3V，可以直接连接各种单片机（51，AVR，PIC，ARM，MSP430 等），5V 单片机也可直接连接，无需 MAX232 也不能经过 MAX232；空旷地有效距离 10 米，超过 10 米也是可能的，但不对此距离的连接质量做保证；配对以后当全双工串口使用，无需了解任何蓝牙协议，但仅支持 8 位数据位、1 位停止位、无奇偶校验的通信格式，这也是最常用的通信格式，不支持其他格式；在未建立蓝牙连接时支持通过 AT 指令设置波特率、名称、配对密码，设置的参数掉电保存。蓝牙连接以后自动切换到透传模式；体积小（3.57cm*1.52cm），工厂贴片生产，保证贴片质量。并套透明热缩管，防尘美观，且有一定的防静电能力。该模块与一般的蓝牙设备一样，第一次使用时必须要进行配对操作。

但是一旦配对成功后，就完全可以当做全双工串口使用，无需了解任何蓝牙协议，非常方便。



图5.2 蓝牙主机模块实物图

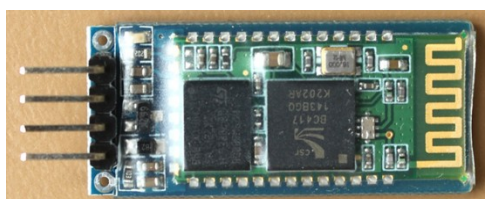


图5.3 蓝牙从机模块实物图

5.4 智能车上位机设计

串口调试助手虽然用途广泛，但是它只能显示一些比较晦涩的文字信息，不能进行效果非常直观的图形显示，这就是它的缺点所在。所以在后期调试中，我们选用了使用简单，上手较快，功能及其强大的 Labview 进行图像数据的接收和显示^[13]。效果图如下所示。

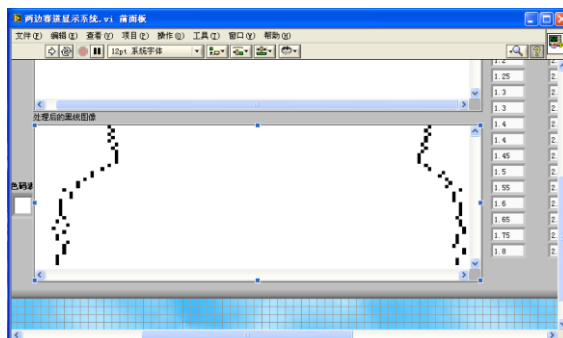


图5.4 上位机界面

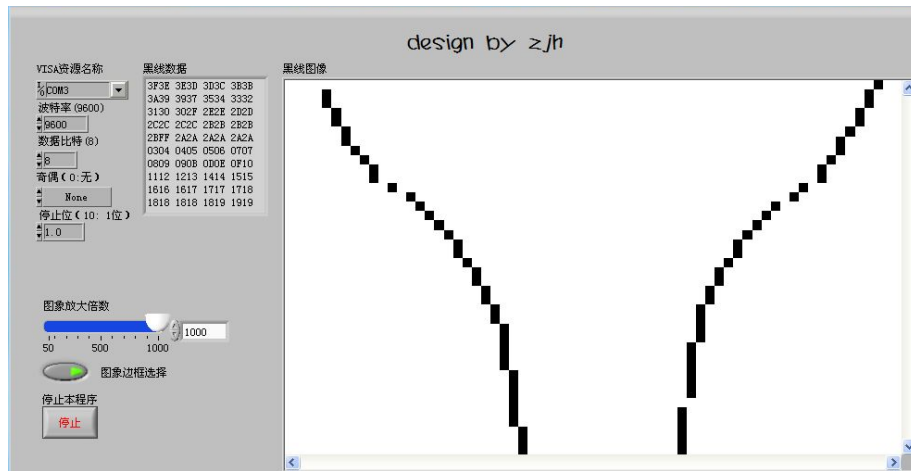


图5.5 双边线

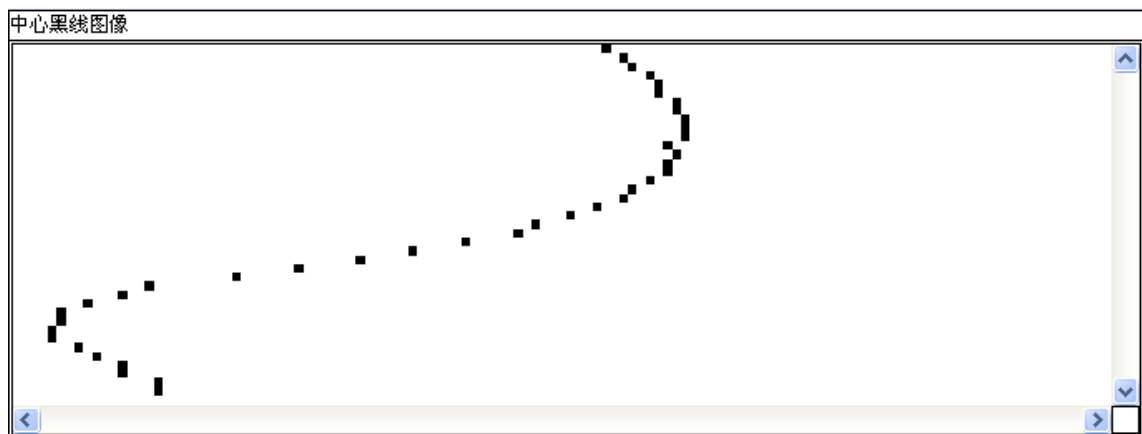


图5.6 S弯

5.5 小结

调试工具的选择直接影响到智能车调试的速度，优秀的智能车离不开优秀的调试工具，特别是在智能车动态运行时，此时如果能够将智能车采集的信息实时传送至 PC 机，并且能够在 PC 机上直观显示赛道信息，对发现问题和解决问题帮助甚大，由此智能车的性能便可得到大幅改善。由于蓝牙的传输带宽较小，无法实时传输大量的图片信息，存在不足。需采用 SD 卡等更为有利的调试工具。

第六章 智能车的主要参数统计

赛车基本参数	长	300mm
	宽	162mm
	高	300mm
车重	1kg	
电容总容量	1320 μF	
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 个	
赛道信息检测	视野范围（近/远）	15/120cm
	精度(近/远)	1.5/6cm
	频率	50HZ

结 论

本技术报告详细阐述了智能车系统的设计过程，包括机械系统、硬件系统和软件系统的设计。机械部分主要包括车模的整体布局，重心的调整，摄像头的安装，编码器的安装，轮胎和底盘位置的调整等，目的是让小车强健美观，既稳定又灵活。硬件系统的设计本着可靠的原则，包括传感器的选取，电源电路的设计、电机驱动电路、视频分离电路的设计等，重点在电源部分设计以及防止电机部分的干扰上。软件部分的设计的重要程度不言而喻，主要由信息的采集和提取，赛道的识别，速度和方向的控制。除了上述三大部分外，调试模块虽作为辅助模块，但不可或缺，优秀的调试方式可加快智能车的测试速度，有利于及早发现问题和解决问题，所谓工欲善其事，必先利其器。

智能车的制作和比赛过程中遇到过不计其数的问题，以下列出一些相对比较重要的问题。

1、十字弯的检测。

当遇到半径较小的十字弯时，智能车容易冲出赛道，分析原因是由于出弯时小车来不及调整路线，到达十字弯时，车身是倾斜的，造成对十字弯的误判，通过不断的分析和尝试，最终标识了十字弯的特征以区分普通弯道，并对搜索到的黑线进行修正，即修正行使路径，使车模顺利通过。这种方式过十字弯的效果较好，比较流畅。

2、起跑线的检测。

直道中的起跑线检测问题不大，此时车身已经调整完毕，容易检测起跑线。但是遇到急转弯后检测起跑线，此时车身来不及调整，同时速度较快，起跑线检测困难，参考十字弯检测的方式，加入标志位来加强起跑线的检测，并且抓住起跑线特有的标记，每一边的空白间隙和黑块的相加所得的值为常值以此来判定起跑线，这种方式可以不用顾及车模进入起跑线之前的状态便可识别起跑线。

3、速度的控制时机。

由于没有实际的赛车经验，并不清楚入弯时速度的控制时机，过早控制，影响车模的整体速度，而过晚控制则容易冲出赛道，不利于提速。经过大量反复的测试，最终决定以直道给最高速度，处在弯道中时给最低速度为原则，入急弯时使用前瞻来给定速度，当弯比较缓和时使用偏差控制目标速度的方式。通过大量的试验，这种处理办法的整体效果不错。

参考文献

- [1] 党宏社, 韩崇昭, 段战胜. 智能车辆系统发展及其关键技术概述[J]. 公路交通科技. 2002, 19(6):127-130
- [2] 孙同景, 陈桂友. Freescale 9S12 十六位单片机原理及嵌入式开发技术[M]. 北京: 机械工业出版社, 2008
- [3] 臧杰, 阎岩. 汽车构造[M]. 北京: 机械工业出版社. 2005
- [4] 黄开胜、陈宋著. 汽车理论与智能模型车机械结构调整方法. 北京: 北京航空航天大学出版社, 2007
- [5] 黄萍莉, 岳军. 图像传感器 CCD 技术[J]. 信息记录材料, 2005, 6(1):50-54
- [6] 卓晴、黄开胜、邵贝贝等编. 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
- [7] 卓晴, 王璘, 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.
- [8] 尹怡欣, 陶永华. 新型PID控制及其应用. 北京: 机械工业出版社, 1998
- [9] 黄洪卫, 黄永卫. 介绍一种瑞萨单片机软件实现的直流电机 PWM 调速方法[J]. 大众科学, 2009, 11:135-137
- [10] 李旭, 谢运祥. PWM技术实现方法综述[J]. 电源技术应用, 2005, 0240-43
- [11] 安鹏, 马伟. S12 单片机模块应用及程序调试[J]. 电子产品世界. 2006, 211:162-163
- [12] 马忠梅. 单片机的 C 语言应用程序设计[M]. 北京: 北京航空航天大学出版社, 2006
- [13] 王磊, 陶梅. 精通 LabVIEW8.X[M]. 北京: 电子工业出版社, 2008:318-320.

致 谢

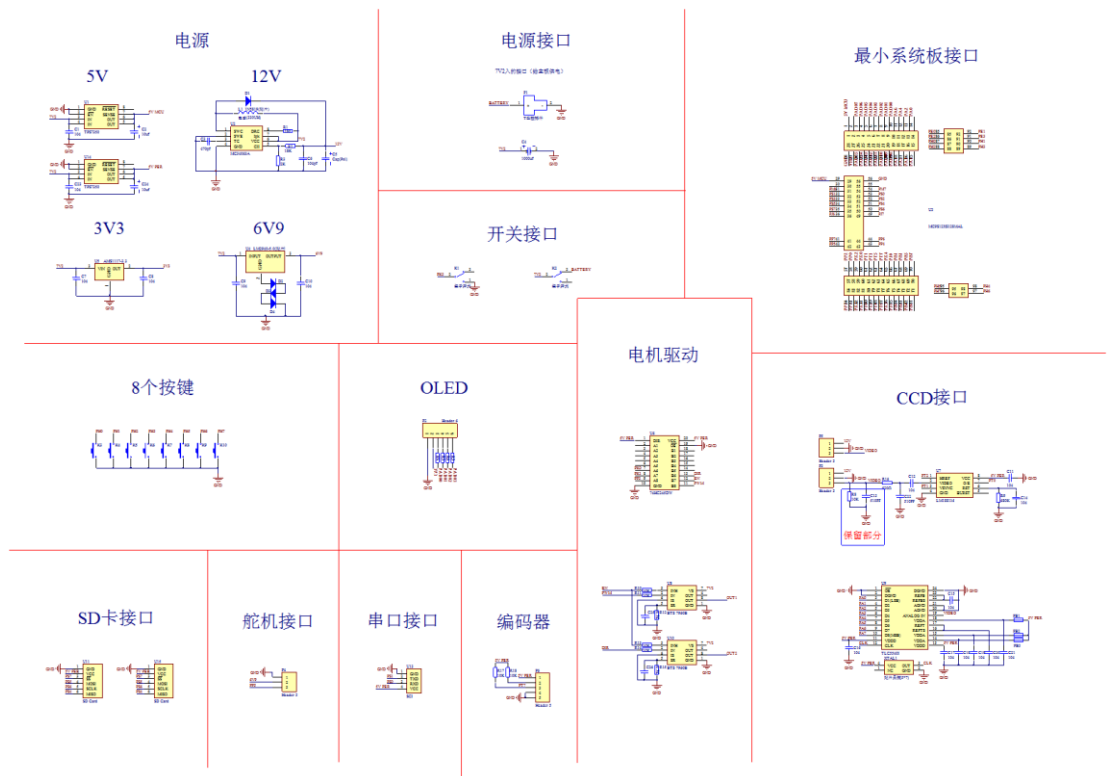
在为本次大赛制作智能车期间，我们遇到过很多问题，从最初的传感器选型与方案确定，到后来的软硬件联合调试。在解决一个问题之后，另外一个问题随之出现，再次解决，一次次的积累，我们发现，我们的技术在不断的成长，思想在不断的成熟。而在这过程中，离不开学校，老师和同学的支持。

首先，我们要感谢李良儿老师和刘娟意老师，因为他们让我们接触到了智能车项目和竞赛。李老师作为智能车竞赛的指导老师，在智能车的制作过程中，不仅为我们奔波劳累，指导我们研究的方向，并且提供了良好的工作环境，让我们方便的钻研智能车的设计。

其次，我们要感谢学校和机电学院的王老师和陈老师。在这一年的准备中，他们在器件工具等方面给了我们很多支持。

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。还要感谢我们的队长周佐谕，没有他就没有这次国赛。总而言之，天道酬勤，感谢在一起的日日夜夜，感谢人生中这段美好的时光。

附录 1 电路原理图



附录 2 程序源代码

```
#include <hidef.h>          /* common defines and macros */
#include "derivative.h"      /* derivative-specific definitions */

#include "pll.h"
#include "pwm.h"
#include "uart.h"
#include "oled.h"
#include "pid.h"
#include "pit.h"
#include "key.h"
#include "debug.h"
#include "image capture.h"
#include "image collection.h"
#include "image process.h"
#include "control.h"

void main(void)
{
    /* put your own code here */
    Disable_Motor_Driver();

    // 初始化
    Init_Interrupt_Priority(); // 设置中断优先级
    Init_PLL();
    Init_Futaba();
    Init_Motor();
    Init_UART();
    Init_OLED();
    Init_ToggleSwitch(); // 初始化钮子开关
    Init_Button();       // 初始化按键
    toggleSwitchInitLevel = ToggleSwitch_Scan();
    toggleSwitchLevel = toggleSwitchInitLevel;
    // 最小系统板上的灯作为场信号灯
    DDRB_DDRB0 = 1;
    PORTB_PB0 = 1;
```



```

// 开机动画
OLED_Print(0, 0, "欢迎使用超音速号!!", NORMAL_DISPLAY);
OLED_Draw_Picture(startLogo);
Delay_1ms(1500);
OLED_Clear_RAM();

// 人机交互界面
// 调试主界面
OLED_Print(0, 0, "速度设定", INVERSE_DISPLAY);
OLED_Print(0, 2, "路径设定", NORMAL_DISPLAY);
OLED_Print(0, 4, "起跑线设定", NORMAL_DISPLAY);
OLED_Print(0, 6, "定时试车", NORMAL_DISPLAY);
while (toggleSwitchLevel == toggleSwitchInitLevel)
{
    Human_machine_Interface();
}
OLED_Clear_RAM();

// PID 初始化
Init_Motor_PID(&motor);
Set_PID_Parameter(&motor, 120, 2.0, 0.0, 13.0);
Init_Futaba_PID(&futaba);
Set_PID_Parameter(&futaba, KT_BOARD_CENTER, 0.0, 0.0, 10.0);

// 延迟 3S 后启动小车
Delay_1ms(3000);

// 初始化摄像头
Init_CCD();
TFLG1 = 0xff; // 清行场中断标志位!!!

EnableInterrupts;

for (;;)
{
}
#include "image capture.h"

/*****
- 功能描述：在 PT0 引脚上捕捉场信号(上升沿/下降沿)
- 参数说明：

```

- 返回说明:
- 修改记录:
- 注意事项:

*****/

```
void Init_Vsync_Capture(void)
```

```
{
    // IOC0 选择了输入捕获功能
    TIOS_IOS0 = 0;

    // IOC0 为上升沿/下降沿触发中断
    TCTL4_EDG0B = 1;
    TCTL4_EDG0A = 1;
}
```

*****/

- 功能描述: 在 PT2 引脚上捕捉行信号(上升沿)
- 参数说明:
- 返回说明:
- 修改记录:
- 注意事项:

*****/

```
void Init_Href_Capture(void)
```

```
{
    // IOC2 选择了输入捕获功能
    TIOS_IOS2 = 0;

    // IOC2 为上升沿触发
    TCTL4_EDG2B = 0;
    TCTL4_EDG2A = 1;
}
```

*****/

- 功能描述: 初始化场，行信号的捕捉和数据端口
- 参数说明:
- 返回说明:
- 修改记录:
- 注意事项:

*****/

```
void Init_CCD(void)
```

```
{
```

```

DDRA = 0x00;    // PTA 设置为输入状态

TSCR1_TEN = 0; // 禁止定时/计数器
Init_Vsync_Capture();
Init_Href_Capture();
TSCR1_TEN = 1; // 使能定时/计数器
// 使能场，禁止行
Enable_Vsync_Interrupt();
Disable_Href_Interrupt();
}

/*****
- 功能描述：使能场中断
- 参数说明：
- 返回说明：
- 修改记录：
- 注意事项：必须先清标志位，再使能才能进入下一个中断!!!
*****/
void Enable_Vsync_Interrupt(void)
{
    TFLG1 |= (1<<0); // 清中断标志位
    TIE |= (1<<0);   // 场中断使能
}

/*****
- 功能描述：禁止场中断
- 参数说明：
- 返回说明：
- 修改记录：
- 注意事项：必须先清标志位，再使能才能进入下一个中断!!!
*****/
void Disable_Vsync_Interrupt(void)
{
    TFLG1 |= (1<<0); // 清中断标志位
    TIE &= ~(1<<0);  // 场中断使能
}

/*****
- 功能描述：使能行中断
- 参数说明：

```

- 返回说明:
- 修改记录:
- 注意事项: 必须先清标志位, 再使能才能进入下一个中断!!!

*****/

void Enable_Href_Interrupt(void)

```
{
    TFLG1 |= (1<<2); // 清中断标志位
    TIE |= (1<<2);    // 行中断使能
}
```

- 功能描述: 禁止行中断
- 参数说明:
- 返回说明:
- 修改记录:
- 注意事项: 必须先清标志位, 再使能才能进入下一个中断!!!

*****/

void Disable_Href_Interrupt(void)

```
{
    TFLG1 |= (1<<2); // 清中断标志位
    TIE &= ~(1<<2);  // 行中断禁止
}
```