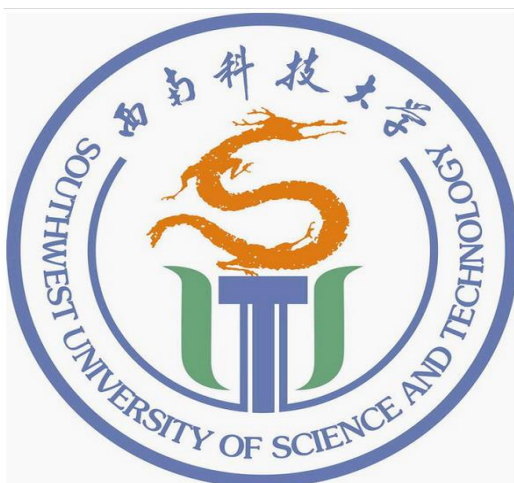


第十届“恩智浦”杯全国大学生 智能汽车竞赛

技 术 报 告



学 校：西南科技大学

队伍名称：西科一队

参赛队员：李林飞

陈宇堃

何秋森

带队教师：武丽、朱玉玉

关于技术报告和学术论文使用授权的说明

本人完全了解第十一届“恩智浦”杯全国大学生智能汽车竞赛关于保留、使用技术报告和学术论文的规定，即：参赛作品著作权归参赛者本人，比赛组委会和恩智浦半导体公司可以在相关主页上收录并公开参赛作品的设计方案、技术报告以及参赛模型车的视频、图像资料，并将相关内容编纂收录在组委会出版论文集中。

参赛队员签名：李林飞、

陈宇堃

何秋森

带队教师签名：武丽 朱玉玉

日期：2016.08.15

摘要

本文设计的智能车系统以 MK60DN512ZVLL10 微控制器为核心控制单元，通过 CMOS 摄像头检测赛道信息，提取赛道边界，用于赛道识别；通过增量式光电编码器检测模型车的实时速度，使用 PID 控制算法调节驱动电机的转速和转向舵机的角度。整个系统涉及车模机械结构调整、系统电路设计、图像处理、控制算法和策略优化等多个方面。对比了不同方案的优缺点，通过大量的测试，确定了现有的系统结构和各项控制参数。实验结果表明，该系统设计方案稳定可靠。

关键词：智能车系统，MK60DN512ZVLL10，CMOS 摄像头，PID 控制。

Abstract

In this paper we will design a smart car system based on MK60DN512ZVLL10 as the micro-controller unit. We use a CMOS image sensor to obtain lane image information. Then extracting black leading lines for track identification. An inferred sensor is used to measure the car's moving speed. We use PID control method to adjust the rotate speed of driving electromotor and direction of steering electromotor. The whole system involves mechanical models' structure adjustment , the system circuit designment, image processing, controlling algorithm and strategy optimization, and many other aspects. Comparing the advantages and disadvantages of different schemes, Through a lot of testing, we determine the current system structure and controlling parameters. The experimental results show that the parameters of the system design scheme is feasible.

Keywords: smart car system, MK60DN512ZVLL10, CMOS camera, PID control.

目录

第一章 引言	6
1.1 智能汽车系统设计情况概述	6
1.2 文献综述	6
1.3 本文主要结构	6
第二章 系统总体结构设计	8
2.1 系统总体结构	8
2.2 智能车硬件电路总体设计	8
2.3 智能车软件系统总体设计	10
2.4 本章小结	10
第三章 车体机械结构设计	11
3.1 汽车行驶的数学模型	11
3.2 车模的机械模型	13
3.3 舵机的安装	14
3.3.1 舵机的安装方式	14
3.3.2 舵机臂的调整	15
3.3.3 舵机连杆的调整	16
3.4 前轮调节	16
3.4.1 主销内倾角调整	16
3.4.2 前轮前束角调整	17
3.5 编码器的安装	18
3.6 图像传感器安装	19
3.6.1 摄像头底座安装	19
3.6.2 摄像头支撑杆的选择	19
3.6.3 摄像头固定	20
3.7 整车效果图	21
3.8 本章小结	21
第四章 硬件系统设计	23
4.1 硬件系统总体设计	23
4.2 单片机最小系统	24
4.3 图像采集模块	24
4.3.1 摄像头的选择	24
4.3.2 视频信号处理	26
4.4 速度反馈模块	27
4.5 舵机供电电源设计	28
4.6 驱动模块设计	29
4.7 电源供电模块设计	30
4.8 本章小结	30
第五章 软件系统设计	32
5.1 软件系统整体设计方案	32
5.2 图像采集算法	33
5.3 提取黑线	33
5.3.1 提取算法总述	33
5.3.2 近端黑线提取算法	34
5.3.3 远端黑线提取算法	34

5.3.4 起跑线的识别	35
5.4 黑线中心求解算法	37
5.4.1 逆透视变换计算黑线位置	37
5.5 控制算法	39
5.5.1 电机控制策略	39
5.5.2 电机的 PID 控制	42
5.6 本章小结	44
第六章 调试方法	45
6.1 本系统中所使用调试工具简介	45
6.2 IAR Embedded Workbench for ARM 在线调试工具使用	47
6.3 串口示波器+无线组合工具使用	48
6.4 无线+MATLAB 组合工具使用	48
6.5 本章小结	49
第七章 结论	50
7.1 模型车主要技术参数说明	50
7.2 主要内容总结	50
7.3 存在问题及解决方案	51
7.4 心得体会	51
参考文献	53
附录 A: 系统主控板原理图	55
附录 B: 源程序	56
1、搜底三线	56
2、左搜线再补线程序	60
3、起跑线以及障碍检测	67
4、图像逆透视还原世界坐标	70

第一章 引言

1.1 智能汽车系统设计情况概述

根据竞赛规则及功能要求，设计以恩智浦半导体公司的 32 位单片机 K60 和数字 CMOS 摄像头为核心的智能车控制系统。在组委会提供统一车模平台上，构建完整智能小车系统。小车通过摄像头采集道路图像信息送入单片机，单片机对输入的原始图像信息进行处理。提取出赛道的特征信息，据此选择最优行进路线并进行速度控制。整个系统设计包括车体机械结构设计和系统的软/硬件系统设计。车体机械结构设计主要包括编码器和图像传感器的安装、舵机和后轮调节等；硬件系统设计完成了电源管理模块、电机驱动模块、速度测量模块、辅助调试模块、图像采集处理模块、舵机控制模块和单片机模块等的电路设计和调试；软件系统设计完成了各功能模块的算法及程序设计，包括图像采集算法设计、提取黑线及黑线中心求解算法设计以及舵机和电机的 PID 算法设计。采用 IAR Embedded Workbench IDE 调试工具进行在线调试所设计的系统经过测试，赛车能够快速安全的行驶。

1.2 文献综述

在智能车制作过程中，参考了大量的相关资料文献。从恩智浦公司官方网站，下载得到 K60 单片机开发技术手册。了解所需要的寄存器的功能。配合《单片机嵌入式应用的在线开发方法》。熟悉了 K60 单片机的内部结构，具备了开发所需要的程序基础。

关于控制算法的文章很多，对于此次比赛，重点在于速度的提高，不求很精确入弯，因此在阅读这些文献时，重点参考模糊控制和算法容易实现方面得内容。很多算法都是通过实验测试，最后得到了一个比较优越的速度控制方案。其他参考文献在文后参考文献中列出。

1.3 本文主要结构

本文针对第十一届恩智浦杯智能车准备阶段的各个方面做了一个详细的总结。采用先总后分的结构，先对系统总体设计进行介绍，然后分别对各部分进行介绍，突出强调了系统机械设计、硬件电路和软件编程。

本文有 7 个章节，第一章为引言，简单介绍了本校智能车制作的一些情况；第二到六章为主体部分，对机械结构系统、硬件和软件设计系统进行了详细介绍，并对调试方法和调试过程进行了说明；第七章为总结，主要阐述了在设计过程中遇到的问题和解决办法，以及还存在的问题。

第二章 系统总体结构设计

根据大赛组委会的统一要求，须在组委会提供的统一车模平台上，自主选择所需要的传感器类型，设计系统硬件电路，开发软件算法。所以本系统主要包括三个大部分，分别为车模平台、硬件电路系统、软件算法。每一个部分又由各个小的模块构成。所以要构建一个完整的小车系统，必须先对各个模块进行论证和设计，在将整个系统组合成一个完整系统，进行系统联调。

2.1 系统总体结构

从智能汽车设计规则及其自主寻迹运行的要求进行分析，可将智能汽车系统设计在结构上分为两个部分，即智能车车载系统设计和上位机系统设计。从智能车功能需求上分析，智能车车载系统设计需完成赛道信息采集、赛道信息综合判断、直流电机测速及转速大小控制、舵机转向控制、无线通信等基本功能。系统总体结构如图 2.1 所示。

2.2 智能车硬件电路总体设计

系统的硬件系统是整个系统的基础，是软件平台得以稳定运行的基础，所以硬件电路的设计是非常重要的。在硬件系统的设计过程中，本着系统安全、稳定的原则，电路设计尽量模块化，主要包括系统微控制器、图像采集模块、智能汽车供电系统、电机驱动模块、速度反馈检测模块、辅助调试模块等。

在本系统中，以恩智浦半导体公司的微控制器作为主控制器。系统工作过程为：首先，微控制器按照一定时序接收来自图像采集模块的摄像头图像信息，采集完后存储。其次，微控制器对存储的赛道信息进行处理，提取出赛道中线的信息。并按照处理后的赛道中线信息以 PD 控制方式控制舵机转角。最后，微控制器根据此时测速装置反馈而回的电机实际转速大小和当前赛道状况下电机期望转速大小，采用 PID 控制器以调节直流电机转速来控制智能车自主寻迹运行速度。此外，通过智能车车载辅助调试模块，将智能车运行参数及采集的轨道引线信息，以无线通信方式发送于上位机通信系统，由上位机监控界面对智能车自主寻迹运行情况进行实时跟踪。

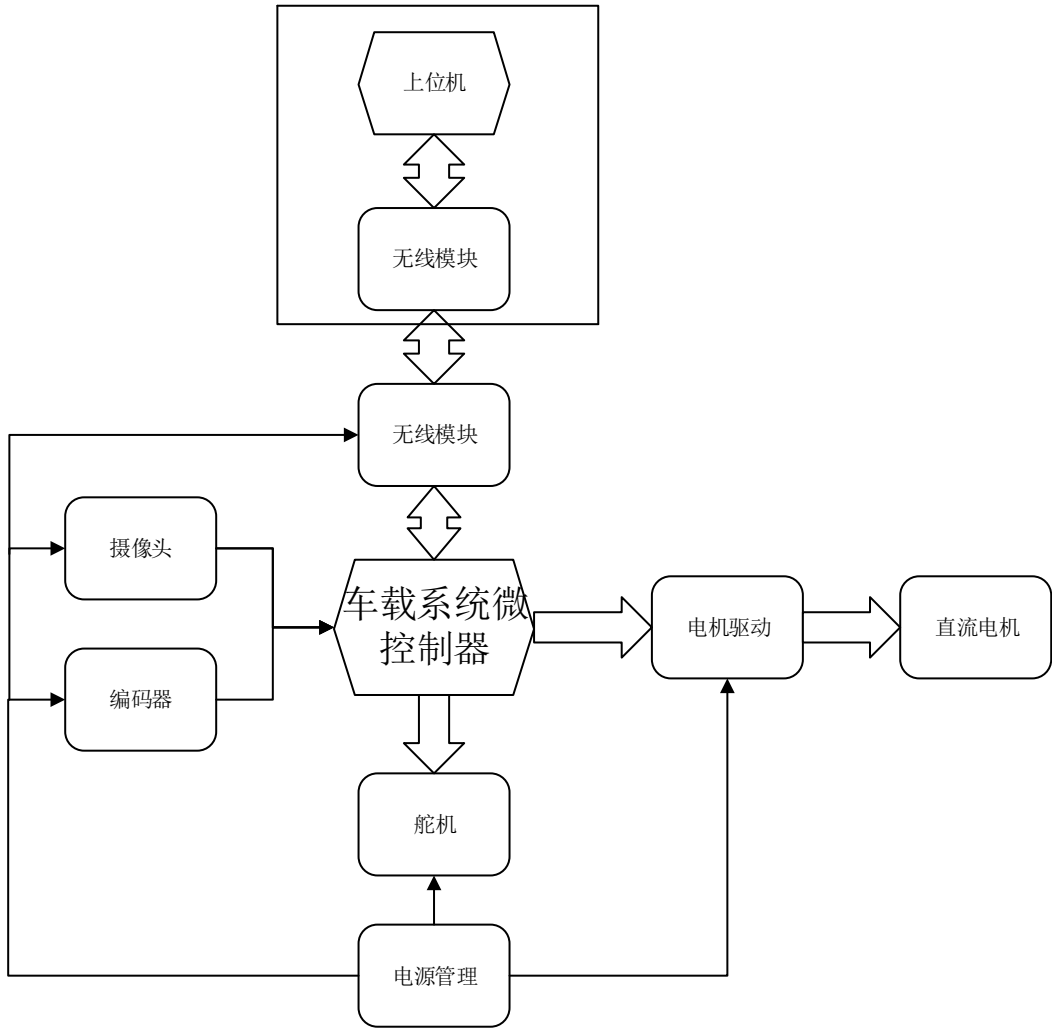


图 2.1 系统总体结构

根据比赛的统一规则，全部硬件电路的电源由 7.2V、2A/h 的可充电镍镉电池提供。由于电路中的不同电路模块所需要的工作电压和电流容量各不相同，因此电源模块应该包括多个稳压电路，将充电电池电压转换成各个模块所需要的电压。考虑到整个系统的功耗情况，经过测试，列出整个系统的功耗情况明细表，如表 2.1 所示，根据系统的总体功耗来对硬件电路进行设计。

模块名称	小系统	摄像头	测速模块	舵机模块	其他电路损耗
功耗	250mw	120mw	60mw	600mw	200mv

表 2.1 系统功耗统计表

单片机的硬件资源是有限的，合理的利用单片机的硬件资源是保证系统稳定的前提。在本系统中，既涉及到信号的采集，同时涉及到控制。在采集部分，使用的是中断触发 DMA 采集方式进行，而在控制部分，主要使用的是单片机自带的 PWM 方波。

2.3 智能车软件系统总体设计

系统硬件对于赛车来说是最基础的部分，软件算法则是赛车的核心部分。如果把一辆车和一个人做个类比的话，我们可以说，赛车的硬件结构相当于人的身体；赛车的软件算法相当于人的思想。只有“身体健康，思想进步”，才会取得好成绩。所以软件系统对于赛车来说至关重要。首先，赛车系统通过图像采样处理模块获取前方赛道的图像数据，同时通过速度传感器模块实时获取赛车的速度。然后 K60 利用边缘检测方法从图像数据中提取赛道黑线，求取赛车行驶路径点并通过图像逆透视变换得到世界坐标，从而获取准确的行驶半径，接着采用 PID 方法对舵机进行反馈控制，并在 PID 算法的基础上，整合加入模糊控制算法，有利于对小车系统的非线性特性因素的控制。最终赛车根据检测到的速度，结合我们的速度控制策略，对赛车速度不断进行恰当的控制调整，使赛车在符合比赛规则情况下沿赛道快速前进。

2.4 本章小结

本章主要介绍了大赛的相关背景和小车整体结构设计的概述。硬件，软件和机械部分的有效融合是小车能否跑出好成绩的关键因素。小车采用组委会统一提供的车模，由控制处理芯片 MK60DN512ZVLL10 图像采样模块，速度传感模块，舵机驱动模块，电机驱动模块和辅助调试模块组成，通过图像采集、黑线提取、速度控制等环节使小车在规则下沿赛道快速前进。

第三章 车体机械结构设计

在小车的制作和调试过程中，随着车速的进一步提高，由机械引发的问题日益变得严重，制约着速度的提高。经过理论分析和实践调试效果，得到了一套比较适合本小车的机械结构。本章主要介绍赛车车模的机械特点和调整方案。从理论方面进行论证，在结合具体实践成果，通过不同的方案效果对比，为读者提供一个全面的认识。

3.1 汽车行驶的数学模型

汽车是现代社会的主要交通工具，在对汽车研究过程中，形成了一大批研究成果。我们在查阅相关资料后，掌握了汽车运动的原理，并建立了汽车行驶的数学模型。

模型一：ACKman 运动模型（见图 3.1），当智能车运动时，可以认为四个轮胎绕着同一个圆心(驱动轮轴心)做纯滚动。

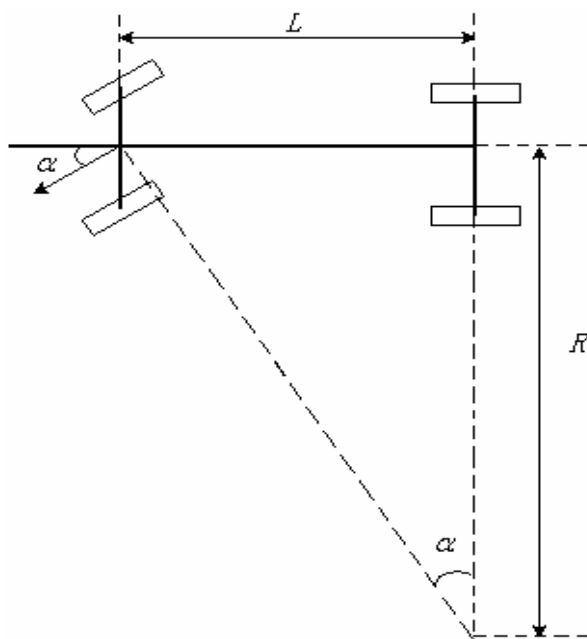


图 3.1 Ackman 汽车运动模型一

当小车转角为 α 时候可以根据车长来求出小车当前的转弯半径

$$R = \frac{L}{\tan \alpha} \quad \text{公式 1}$$

式中 L 是小车的轴长， α 为小车的转角， R 为前轮转弯半径。

本模型来源于后轮驱动，前轮转向的汽车运动系统。

我们知道，由于传感器图像采集和处理带来的延时较长，造成整个运动控制系统并不能对采集到的当前状况进行实时控制，下面将具体分析小车前轮轴心的运动模型，在此模型上可以进行对运动系统的延时进行有效的补偿。

考虑图 3.2 所示的以小车驱动轴中点为参考的运动模型， M_k 和 M_{k+1} 为参考点运动的两个连续位置， X_k 、 X_{k+1} 为前轴中点的速度方向，即小车的纵向， θ_k 为小车初始的方向角， O 为小车的转向中心，那么可以得到：

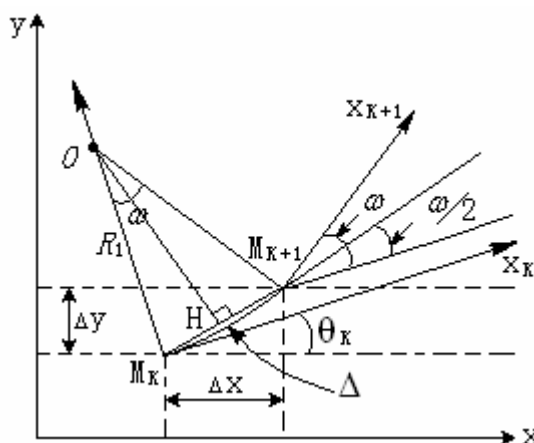


图 3.2 智能车前轴中点运动模型

$$\Delta = R_1 \cdot \omega \quad \text{公式 2}$$

其中 Δ 是 t_k 到 t_{k+1} 时间内小车所走过的距离， ω 是小车的横向角速度， R_1 是小车的前轮的转弯半径。

由图中的几何关系可以得到：

$$\begin{cases} x_{k+1} = x_k + \Delta \cdot \cos(\theta_k + \frac{\omega}{2}) \\ y_{k+1} = y_k + \Delta \cdot \sin(\theta_k + \frac{\omega}{2}) \\ \theta_{k+1} = \theta_k + \omega \end{cases} \quad \text{公式 3}$$

上式的结果即为小车前轴中点在运动过程中的轨迹方程，在 Δt 足够小的情况下，随着 θ_k 和 x_k ， y_k 的不断变化可以画出其运动轨迹。当给出小车的初始方向角和初始坐标时，就可以推导出任何时刻小车的坐标值，从而用经过一定延迟时间后的控制量来进行当前的控制以补偿系统实时控制。

3.2 车模的机械模型

本次比赛采用北京科宇通博科技有限公司 1:10 B 型仿真车模，车模长：200mm，宽：172mm。车模正方向运行，采用转向轮在前，动力轮在后，示意如图 3.3 和图 3.4 所示。

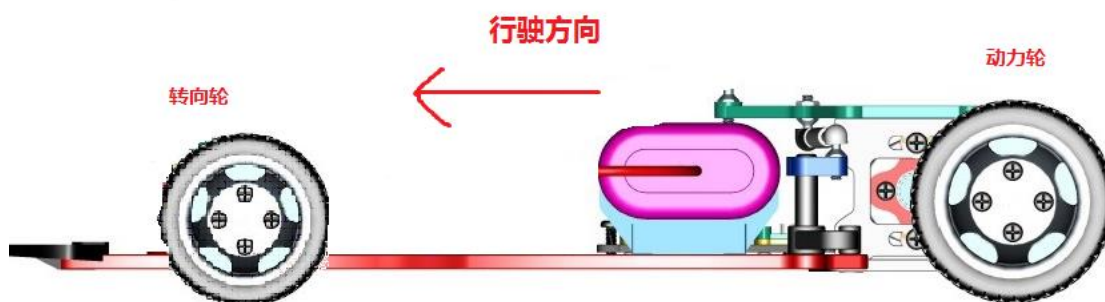


图 3.3 运动示意图

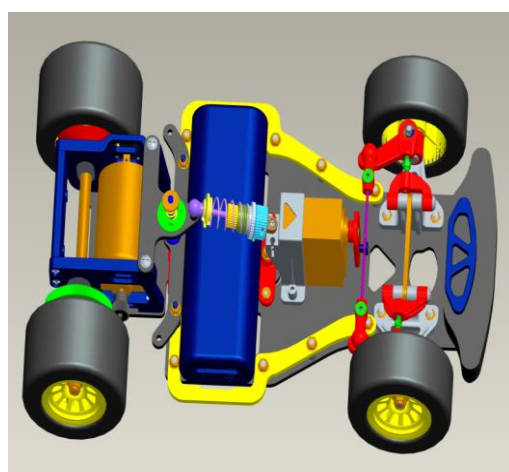


图 3.4 B 型车整车模型

首先，整车机械结构稍显复杂。涉及到主销前倾或后倾的调整，转向轮束角以及内倾外倾的调整；其次，动态传动系统性能较差。特别体现在从舵机输出到舵机臂，从舵机臂到转向车轮，这些传动机构之间存在较大的虚位，因此会造成智能车运动控制中转向的灵敏度和准确度降低；再者，重心位置偏高。虽然车模配件中提供了一定的摆臂垫片，但可留给我们自由发挥的依然不多；最后，轮胎摩擦力很小。本届轮胎采用硬胎，较之以前的软胎来说，一是摩擦力严重不足，二是轮廓呈外凸状，易导致车模过弯时翻车。

当然，本届摄像头组车模依然配备了各型车模中性能最为优越的 540 驱动电机，带载能力强，电机本身在带载情况下的加速性能比较好，这使得在速度提升上有更大的空间。同时，车模的本身结构也有利于参赛选手根据自身情况进行适当的改装，满足自身控制的需求。

3.3 舵机的安装

舵机最早出现航模运动中，控制航模上的发动机，翼舵转向。在“恩智浦”智能汽车大赛中，主办方提供的舵机内部具有位置反馈电路，使它的舵盘输出转角正比于给其的脉冲信号的宽度。

舵机是具有较大延迟特性的器件，其延迟与其转角大小成正比，但如果能使舵机转过一个越小的角度而使车轮转过一个越大的角度，则会大大提高舵机过弯的响应速度。这不仅与舵机的安装方式有关，而且也与舵机输出臂的长度有关。

3.3.1 舵机的安装方式

对比各种舵机安装方式，我们决定采用最经典的输出臂朝下的输出方式，对于此种安装方式，同样以主销为 Z 轴，前轴为 Y 轴建立空间直角坐标系如图 3.5 所示。为了使小车左右转向一致，在舵机的安装中，一般将舵机安装在后轮轴的中垂线上，并使舵机的转轴与 OA 杆同处于 XOY 平面上，且当模型车直线行驶时，输出臂 AB 垂直与 OA 杆，并使之保持水平。

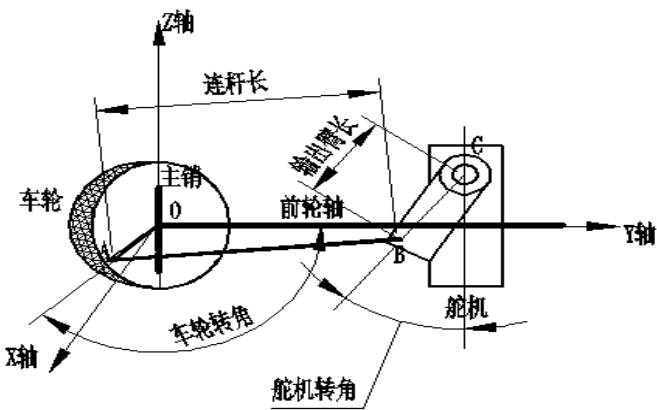


图 3.5 舵机安装方式关系图

根据理论计算和实际效果对比，得到最终舵机安装图，如图 3.6 所示。

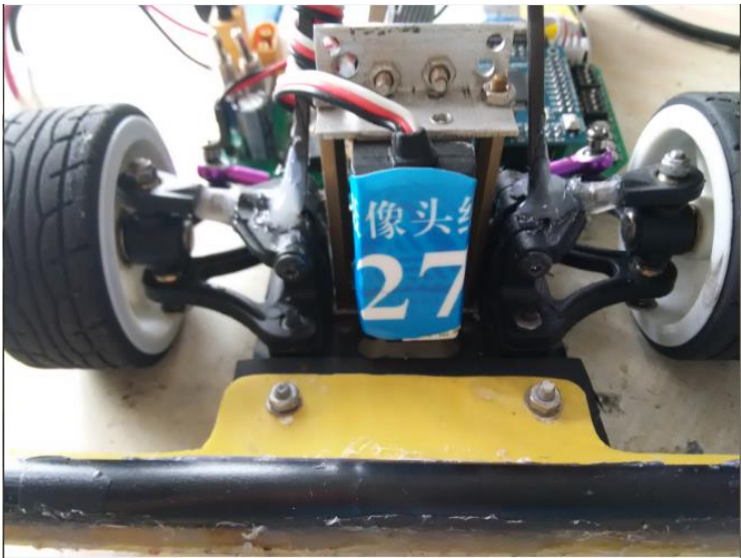


图 3.6 舵机安装

3.3.2 舵机臂的调整

由于舵机本身的滞后特性，对于诸如小车这样实时系统，必须尽量提高舵机的响应速度（及舵机的灵敏度）。舵机的灵敏度和舵机的安装方式有关，而且还与舵机臂的长短有关系，在安装方式一定的前提下。舵机输出臂越长，则舵机的灵活性越大，但是舵机的转矩会随着输出臂的增加而减小。在实际调整过程中，既要寻找最优的舵机长度，又要兼顾舵机的输出转矩。最后舵机臂的长短还应考虑到后轮的调整。

3.3.3 舵机连杆的调整

车模原装舵机提供的只有一个支点由舵机臂输出到连杆，如下图 3.7 所示，这也是我们最初采用的安装方式。这种方式有一个比较大的缺点即是由于两个连杆必须前后叠在一起，这就造成左右连杆互成一个角度而非平行，在调试中我们也发现这种方式能导致左右打角不对称，并且这种连杆所带的球头帽及各结合处缝隙太大。最终确定用两个支点，连杆平行安装，球头帽缝隙较小。改装后的连杆安装如图 3.8。

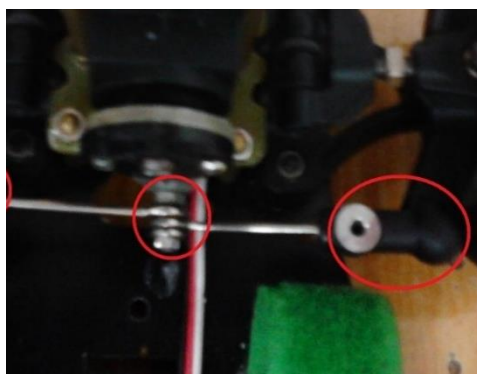


图 3.7 原装舵机连杆

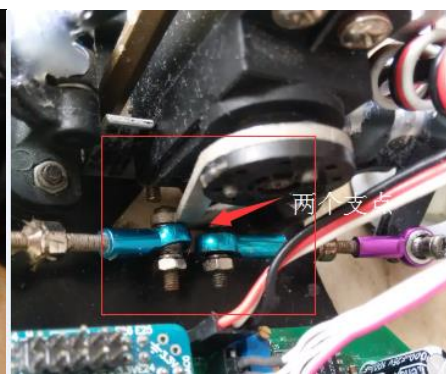


图 3.8 改装后的舵机连杆

3.4 前轮调节

现代汽车在正常行驶的过程种，为了使汽车直线行驶稳定，转向稳定、轻便，转向后能及时回正，并减少轮胎和转向系零件的磨损等，在转向轮，转向节和前轴之间须形成一定的相对安装位置，叫车轮定位，其主要参数包括：主销内倾，主车轮前束。智能汽车大赛提供的车模的四项后轮定位参数均可调，并给规格不同配件。

3.4.1 主销内倾角调整

主销后倾是指在汽车的纵向平面内（汽车的侧面）有一个向后的倾角，即主销轴线与地面垂直线在汽车纵向平面内的夹角。采用主销后倾角的原因是由于汽车在车轮偏转后会产生一回正力矩，纠正车轮的偏转。



图 3.9 主销后倾角调整（前摆臂拉杆）

3.4.2 前轮前束角调整

前轮外倾角是指通过车轮中心的汽车横向平面与车轮平面的交线与地面垂线之间的夹角。当轮胎呈“八”字形张开时称为“负外倾”，而呈现“V”字形张开时称为“正外倾”。前轮外倾角可以减少转向阻力，使汽车转向轻便。模型车提供了序号为 EX-19 的配件用来调节前轮外倾的角度。

当车轮有了外倾角后，在滚动时就类似与圆锥滚动，从而导致两侧车轮向外滚开。由于转向横拉杆和车桥的约束使车轮不可能向外滚开，车轮将在地面上出现边滚边向内滑移的现象，从而增加了轮胎的磨损。在安装车轮时候，为了消除车轮外倾带来的这种不良后果，可以使汽车两前轮的中心面不平行，并使两轮前边缘距离小于后面的边缘距离，两者之差称为“前轮前束”。如图 3.10 所示，模型车是由舵机带动左右横拉杆实现转向的。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改变前束的大小。



图3.10 前束角调整

3.5 编码器的安装

编码器是测速用的核心模块，必须将其和电机牢牢咬合才能准确测量实际速度值。开始我们选取的 300 线的编码器，但是调试过程中发现其精度稍低，后来换成 500 线的大编码器。考虑到本届 B 型车模要求驱动轮在后，所以在安装位置上，尝试过两种安装方式，第一种是将安装在差速盘和电机齿轮确定的直线上，使三者位于一条直线。这就将编码器暴露在车体最后方。第二种是安装在差速盘上方，利于保护编码器不至于受撞击受损。同时也采用一个线切割的薄铝片和车身固定在一起这样便使得编码器能非常牢的和电机齿轮咬合。我们西部赛一直采用第一种安装方式，效果较好，具体安装图如图 3.11 所示。

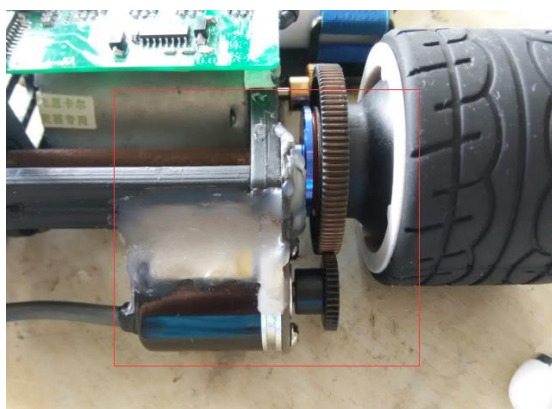


图 3.11 编码器的安装

3.6 图像传感器安装

信号采集传感器我们采用的是摄像头。摄像头的安装也光电管和电磁器件安装稍微不同，摄像头安装主要考虑四个方面，一是图像的失真要小，而是整车重心要低，三是前瞻要比较大，四是智能车运动时传感器足够稳定不至于晃动。

3.6.1 摄像头底座安装

在摄像头和车体的安装位置方面，应该兼顾车模重心位置，稳定性，采集信息的准确性及左右对称性。在调试中我们知道，如果摄像头安装在车体太后面，将会导致图形畸变严重，以致给处理图像带来太多的复杂因素，但要是安装在车体太前面就会在保证一定的前瞻的情况下，导致小车车前盲区的过大，这样在弯道的识别控制时，是相当不利的。而当架设位置远离重心时，摄像头左右前后晃动厉害，当整个车模重心正好在中心处时所取得的控制效果较好。而根据物理特性，车模结构中最为稳定的位置即在重心处。另外，我们研究过将摄像头在重心处架设能够将盲区控制在 15cm 以内并能保证 190cm 的前瞻量，足以满足我们的控制要求。综合之下，我们最终确定将摄像头安装在重心位置处，这样使得采集的图像不至于畸变过于厉害，也使得小车的盲区控制在 14cm 左右，并保证了摄像头的稳定性。

在摄像头底座的选择上做了两种安装方式，第一种直接用车模轮配件来将摄像头支架直接用 AB 胶固定在底盘上，第二种是采用线切割底座支架将摄像头安装在做的底座上，显然第二种安装方式更加稳定，两种安装方式的具体情况如图 3.12 所示。

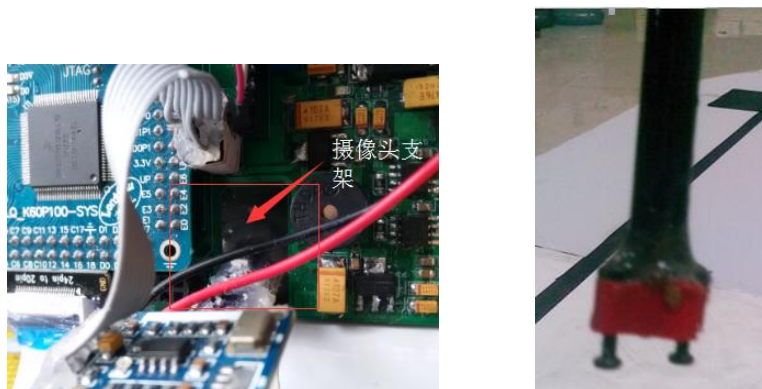


图 3.12 摄像头两种底座安装方式对比

3.6.2 摄像头支撑杆的选择

在摄像头支架杆上的选择也有两种，前一种是碳素杆非常轻，但很容易破碎，后一种采用轻质铝材，牢固耐用，但比起碳素杆还是重了许多。对比两种材质的支撑杆，结合本届车模的具体情况，整车重心应尽量低，避免翻车。这就要求架设摄像头的模块在满足硬度要求的前提下应尽量轻。最后决定采用碳素杆材料，从而减轻车模重量。

3.6.3 摄像头固定

要获得稳定可靠的信息，摄像头的固定必须相当牢固。在摄像头固定方面，其可以上下左右进行调节，在调节好视角之后，为了使其更加牢固采用了热熔胶进行二次定型。在实际调试中，先后采用了很多种固定方式，最后采用亚克力板进行线切割制成的固定块，这样的模块既质轻又可以很方便的左右上下移动，确定好前瞻和角度后，用热熔胶粘牢，可得到比较好的固定效果。如图 3.13 所示。



图 3.13 摄像头安装方式

3.7 整车效果图

整车的机械结构设计参数见表3.1



图3.14 整车效果图

项目	参数
路径检测方法（赛题组）	摄像头组
车模几何尺寸（长、宽、高）（毫	285mm/185mm/300mm
车模轴距/轮距（毫米）	200mm/140mm
赛道信息检测空间精度（毫米）	150mm（近端）、1900mm(远
车模重量（带有电池）（千克）	1.2

表 3.1 整车机械结构参数表

3.8 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问题。由于前期过于注重控制算法的研究，而没太注重整车的机械结构。所以随着调试的进一步

进行，机械结构问题已经严重影响到了车速的进一步提高。特别是小车的过弯速度，即使速度很低，过弯也存在一些问题。后来在多次调整小车机械结构后，小车在过弯速度在同样的控制算法基础上都有了很到的改观。对了这样一个车载系统，要实现机械与控制的完美结合，才能达到最佳的速度效果。

第四章 硬件系统设计

4.1 硬件系统总体设计

系统的硬件电路是整个系统的基础，是软件平台得以稳定运行的基础，所以硬件电路的设计是非常重要的。在硬件系统的设计过程中，本着系统安全、稳定的原则，电路设计尽量模块化，主要包括系统微控制器、图像采集模块、智能汽车供电系统、电机驱动模块、速度反馈检测模块、辅助调试模块等。（具体硬件系统框图见图 4.1）

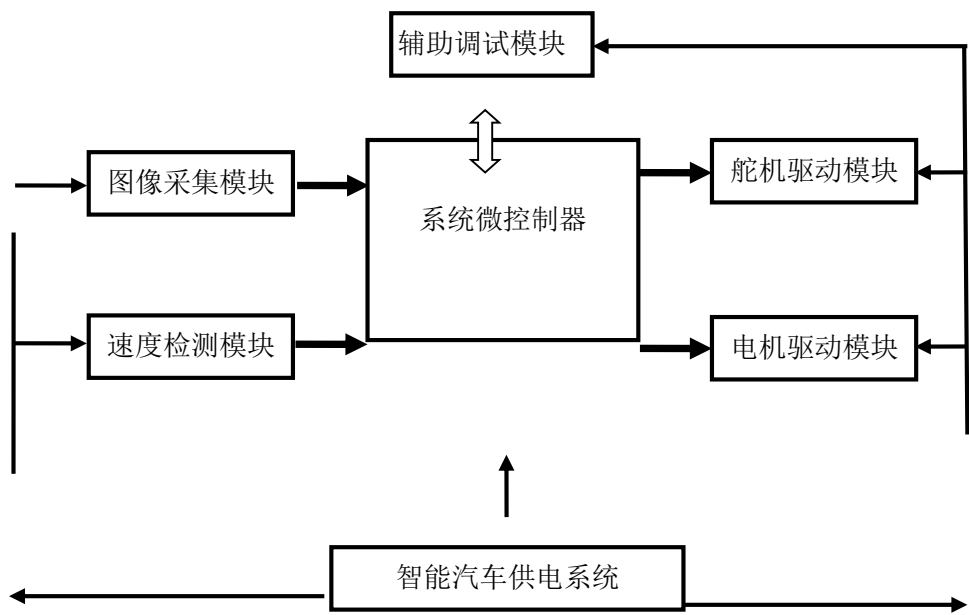


图 4.1 硬件系统框图

采用摄像头进行道路识别，赛车的硬件电路主要有七个部分组成：

MK60DN512ZVLL10 最小系统板，图像采样处理模块，速度检测电路，电机驱动电路，舵机驱动模块，电源管理模块，辅助调试模块。

(1) MK60DN512ZVLL10 最小系统板是系统的核心部分，负责接收赛道图像数据，赛车速度等反馈信息，并对这些信息进行恰当的处理，形成合适的控制量来对舵机与驱动电机进行控制。

(2) 图像采样处理模块由数字电路及摄像头组成，是智能小车的“视觉系统”，用于获得前方道路情况以供单片机处理。

(3) 速度检测电路由旋转编码器，高速比较器以及 K60 的 FTM 模块正交解码模式构成，实现速度的反馈。

(4) 电机驱动电路使用专用 MOS 管搭建的全桥驱动，可以实现电机的正反转。

(5) 舵机驱动模块用于给舵机提供稳定的电源。

(6) 电源管理模块给整个系统供电，保障系统安全稳定运行。

(7) 辅助调试模块有串行通信、无线 24L01，OLED 液晶显示等，主要用于小车系统的程序烧写，功能调试和测试，小车状态监控，小车系统参数和运行策略设置等方面。

4.2 单片机最小系统

单片机最小系统采用大赛组委会统一指定的 32 位微处理器——Kinetis 系列 MK60DN512ZVLL10 单片机。K60 最小系统核心板是系统的核心部分，负责接收并处理赛道图像数据，赛车速度反馈等信息，并对这些信息进行恰当的处理，在控制算法的控制下，形成合适的控制量来对舵机与驱动电机进行控制。从而控制整个小车的安全、快速前进，达到对小车的控制。单片机小系统原理图附录 A 所示。

4.3 图像采集模块

4.3.1 摄像头的选择

智能车制作中，摄像头起着举足轻重的作用，所以摄像头的选择也是至关重要的。摄像头又分为数字摄像头和模拟摄像头，常用的数字摄像头有 CMOS 数字摄像头，常用的模拟摄像头包括 CCD 和 CMOS 两种。下面从摄像头的重量、噪声、解析度、感光度、电能和功耗等方面作一个对比，以选择适合我们小车的摄像头。

(1) 重量

重量上,因为体积和远见数量的区别,CMOS 摄像头也优于 CCD,大约有 30%-50%的重量差别。这个重量的差异,随摄像头安装的位置的不同而对整个赛车的稳定性有不同的影响。摄像头安装的位置越高,摄像头重量对赛车重心的影响越高。另外 CMOS 摄像头有一种数字式的摄像头,它在重量和体积上和 CCD 摄像头相比更小。

(2) 噪声

在图像噪声方面,CCD 好于 CMOS 摄像头。这一点是这两种技术本身先天决定的,CMOS 的工艺无法保证几万个感光点的一致性,而 CCD 因为所有感光点均通过一个电荷-电压转换器,一致性可以得到很好的保障。图像的噪声,在智能车比赛中,影响不是特别大。因为拍摄图像只有全白全黑两种,来自传感器的电平轻微跳动,还不至于影响到赛道的识别。同时在一个设计不良的采集电路中,来自 A/D 模块的误差,也足以掩盖来自传感器的误差噪声,这一点却没有引起很多人的重视。

因此,我认为图像传感器的噪声不足以影响智能车比赛中对黑线的识别,因此可以在摄像头选型时,作为优先级较低的一个参考标准。而在其他需要高分辨率图像采集和识别的应用中,两种传感器的图像质量差别,还是需要引起一定的重视的。

(3) 解析度

针对智能车使用单片机对图像型号进行采集的应用,解析度的高低直接关系到采集的难度。

我们知道,对于模拟摄像头标准的视频信号每秒有 50 或 60 场图像,这是固定不变的。越高的分辨率,包含了越多的行数。即在 20ms 的时间内,有更多的行数,以至于每行图像所占用的时间越少。

一个 30 万像素的摄像头,每一场输出 312.5 行图像,行周期为 64us。我们希望采集的分辨率越高,就需要在 64us 的时间内采样更多的点数。这就需要更快的 A/D 转换器,例如片外高速视频 A/D。

既然单片机的速度远远跟不上视频信号的速度,那么选用越低分辨率的摄像头,图像采集的效果反而越好,因为每行图像有更多的时间留给 MCU 进行采

集。

目前能买到的摄像头中，30 万像素的是最佳选择，而 130 万像素就不适合了。

对于数字摄像头视频信号的速度可以在 50 帧，75 帧，112 帧，150 帧中选择，是可修改的，并且采用 DMA 的采集方式大大的提高了 MCU 的运行速度，将 MCU 从繁重的图像采集中脱离出来，减轻 MCU 的负担。从这一角度数字摄像头更优。

(4) 电能和功耗

影响 CCD 在嵌入式中应用的一个重要原因，就是供电和功耗。首先，CCD 需要 12V 供电，对智能车竞赛而言，这就意味着需要额外给 CCD 设置一个升压电路。这意味着成本和体积的增加。还好，越来越多的队伍在智能车电机驱动部分开始采用 MOS 管驱动的方式，而 MOS 管栅极电压很多都需要 12V，因此两路电源合二为一，提高了电源的利用率。

不能否认，CCD 确实是耗电大户，工作时散发的热量足以烫手。对于电池供电的系统来说，每一点电能的节约都至关重要。从节能角度来说，CCD 真的有点不适合，随着技术的发展，目前 CCD 单板机需要的电流已经从开始的 150mA 降到了现在的 50mA，但是别忘记，CCD 的供电是 12V，这么看来，功耗依然很大。

而 CMOS 在这方面大出风头，3.3V 供电和微功耗，足以让 CCD 汗颜。

通过对比了两种摄像头的各自优缺点，我们组内决定尝试将两种摄像头进行一个实际的对比，就是将摄像头固定在小车上，然后将两种摄像头所采集到的图像在上位机上进行对比、分析。经过组内的积极讨论，我们决定选择 CMOS 摄像头。最终，我们选择了一款 OV7725 摄像头。

4.3.2 视频信号处理

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;3.3V 电源供电,工作时功耗<120mW,待机时功耗<10 μ W。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。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 4.1 所示。



图 4.1

4.4 速度反馈模块

由于对车模采用的是闭环控制系统,所以必然涉及到反馈量的检测。在车模行驶过程中,最重要的一个反馈量就是车模的行驶速度。通过采用光电编码器直接测量车轮的速度。

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,是目前应用最多的传感器。一般的光电编码器主要由光栅盘和光电检测装置组成。在伺服系统中,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,其原理如图所示。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。此外,为判断旋转方向,码盘还可提供相位相差 90° 的 2 个通道的光码输出,根据双通道光码的状态变化确定电机的转向。

编码器返回的信号输入到单片机,利用单片机的正交解码模块获取当前小车的实时速度,从而实现小车的闭环控制。

4.5 舵机供电电源设计

在模型飞机和模型车中，舵机即伺服电机是必备的部分，舵机的基本工作原理为内部包括了一个小型直流马达；一组变速齿轮组；一个反馈可调电位器；及一块电子控制板。其中，高速转动的直流马达提供了原始动力，带动变速（减速）齿轮组，使之产生高扭力的输出，齿轮组的变速比率越大，伺服马达的输出扭力也越大，也就是说越能承受更大的重量，但转动的速度也越低，减速齿轮组由马达驱动，其终端（输出端）带动一个线性的比例电位器作位置检测，该电位器把转角坐标转换为一比例电压反馈给控制线路板，控制线路板将其与输入的控制脉冲信号比较，产生纠正脉冲，并驱动马达正向或反向地转动，使齿轮组的输出位置与期望值相符，令纠正脉冲趋于为 0，从而达到使伺服马达精确定位的目的。

B 车配置的转向舵机采用的是型号为 SD-5 的数字舵机，参见了 SD-5 舵机的各项参数（如图 4.2）。舵机的供电电压变化范围为 4.5~5.5V，为了使舵机有更快的响应速度和更大的扭矩，我们决定采用 5.5V 的供电电压。舵机的电源稳定性对于舵机的正常打角起着决定性的作用，当舵机的打角很大时，就需要电源提供足够的电压和电流。我们选择了 LT1764-EQ 这款可调稳压芯片，该电源模块外围电路不算复杂，并且能够很好的满足我们小车的需要，经过长时间的试验，该模块稳定性很好。舵机供电电源见图 4.3 所示。

S-D5 数码舵机参数													
	数字舵机，塑胶齿，双含油轴承							4.5V			5.5V		
	尺寸 (mm)					重量		速度	扭力		速度	扭力	
	A	B	C	D	E	g	oz	sec/60°	kg-cm	oz-in	sec/60°	kg-cm	oz-in
	40.5	20.5	38	49.0	9.2	44	1.55	0.16	4.2	.	0.14	5	

图 4.2

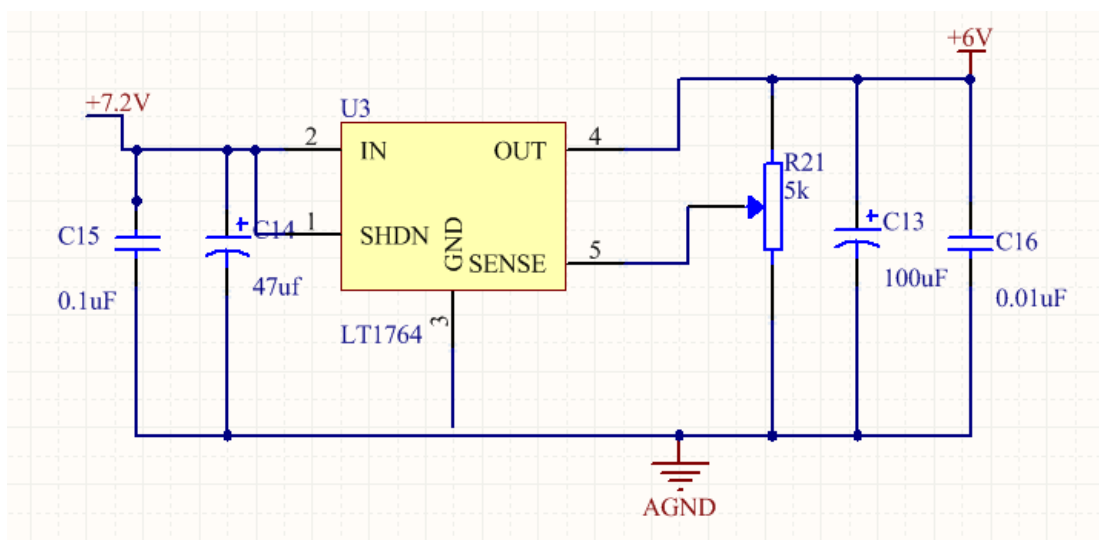


图 4.3

4.6 驱动模块设计

小车的驱动模块是小车的动力之源，所以驱动模块也是很重要的。开始的时候我们组采用了智能功率芯片 BTS7960，它是应用于电机驱动的大电流半桥高集成芯片，它带有一个 P 沟道的高边 MOSFET、一个 N 沟道的低边 MOSFET 和一个驱动 IC，集成的驱动 IC 具有逻辑电平输入、电流诊断、斜率调节、死区时间产生和过温、过压、欠压、过流及短路保护的功能。BTS7960 通态电阻典型值为 $16\text{m}\Omega$ ，驱动电流可达 43A。我们用两片 BTS7860 构成了全桥驱动。但是在后面的调试过程中发现，驱动芯片会发烫，特别是在急停、急起的时候，有时候还会出现电机停转的现象，这对于小车的正常行驶很不利，在仔细阅读了技术手册之后发现芯片内部的过温保护起了作用。

因此，我们小组开始尝试研究新的驱动电路方案，通过多方的查阅资料以及多次的制作驱动电路尝试，最后在小车上试验成功。我们采用的是一片 HIP4082 全桥驱动芯片和四片 NMOS 管组合搭建的全桥驱动。

HIP4082 是全桥驱动芯片可以驱动 4 个 N 沟道 MOSFET，能提供较大的栅极驱动电流，并具有硬件死区、硬件防同臂导通等功能。使用一片 HIP4082 芯片可以组成完整的直流电 H 桥式驱动电路。电路如图 4.5 所示。

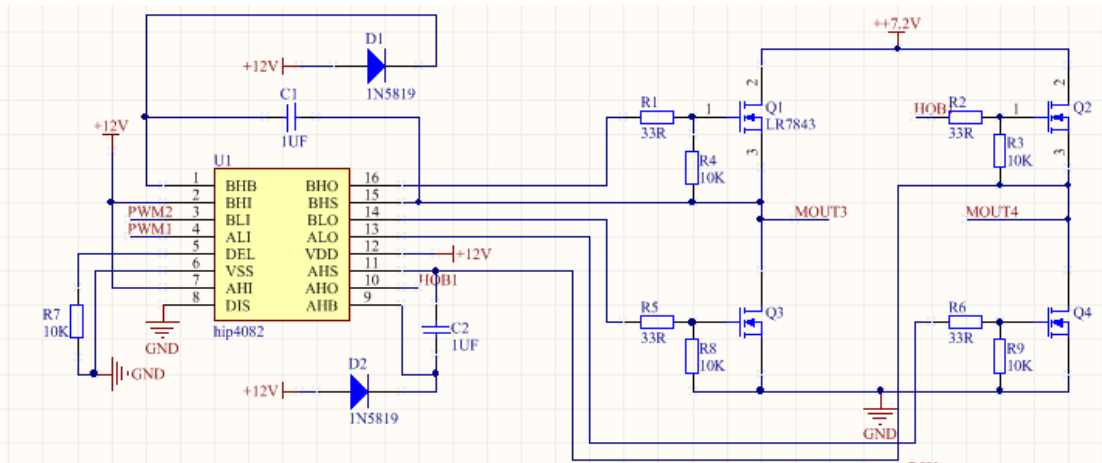


图 4.5 驱动电路

4.7 电源供电模块设计

电源模块是整个系统的能源动力机构，所以良好的电源供给是系统稳定运行的前提。常用的电源有串联型线性稳压电源和开关型稳压电源两大类。前者具有纹波小、电路结构简单的优点，但是效率较低，功耗大；后者功耗小，效率高，电流大，但电路却比较复杂，电路的纹波大。对于整个系统所需要的5V供电，我们采用了TPS7350，它是一款线性低压差稳压芯片，具有输出稳定且纹波小的优点，对于系统的稳定性很有利。对于K60单片机小系统板，需要提供稳定的3.3V电源，由于AMS1117-3.3稳压的线性度非常好，所以选用AMS1117-3.3单独对其进行供电。长时间的试验过程验证了该电源系统的稳定性与可靠性。该模块如下图4.6所示：

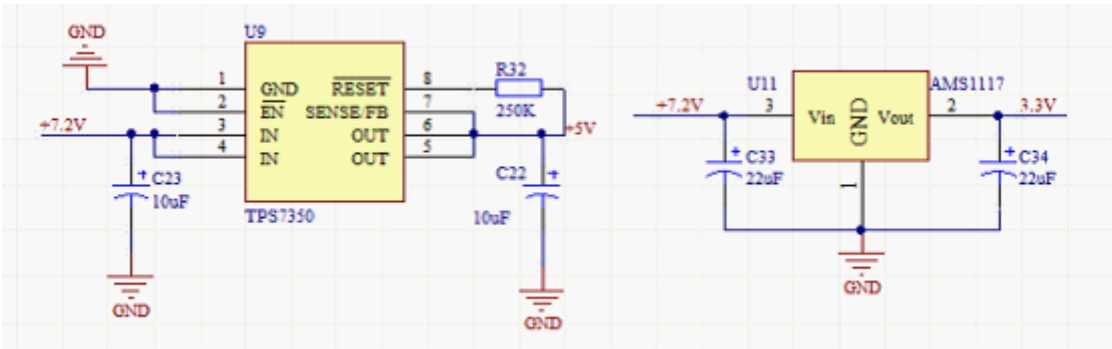


图 4.6 电源模块

4.8 本章小结

本章详细介绍了车载系统的硬件电路结构和工作过程，采用数字 CMOS 摄像头搭建系统，系统硬件电路稳定、可靠。在机械和硬件电路搭建完毕后，就需要配合软件平台。经过测试发现，该硬件平台可支持软件平台的稳定运行。

第五章 软件系统设计

5.1 软件系统整体设计方案

高效稳定的程序是智能小车平稳快速行驶的保证。本智能车采用 CMOS 作为图像采集传感器，图像采集、处理成了整个软件的核心内容之一。在小车转向和速度控制方面，智能车仍然采用的经典的 PID 控制算法，不断把理论和实际结果比较，再进行适当的补偿，使智能车在赛道上达到平稳快速的效果。具体过程为：根据赛道信息给定舵机输出量，使用编码器监测智能车的实际速度，再根据赛道信息给定智能车运行速度，运用位置式 PID 算法调节驱动电机转速，实现了电机的快速响应。软件系统整体设计框图如图 5.1 所示。

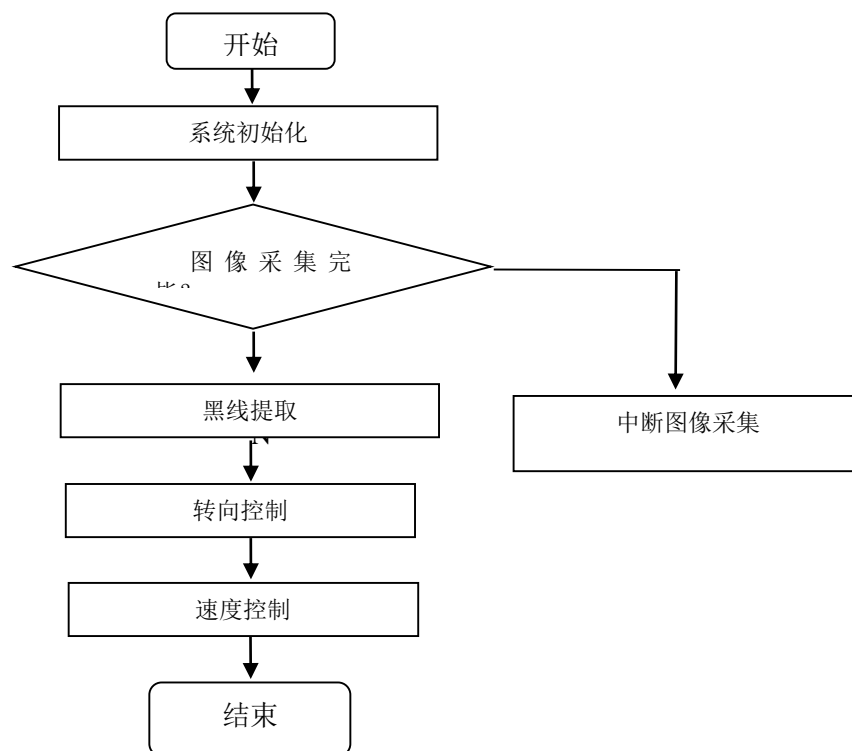


图 5.1 软件设计整体框图

图像采集模块在中断中执行，本系统的反馈测试模块亦放在图像采集程序

中，接下来分模块介绍具体算法。

5.2 图像采集算法

图像采集模块硬件使用的是 CMOS 数字摄像头，在摄像头场中断到来时触发 DMA 采集当前摄像头的图像，在采集完成时，触发 DMA 中断完成一次采集，并在主函数中，将采集到的压缩图像加以解压。

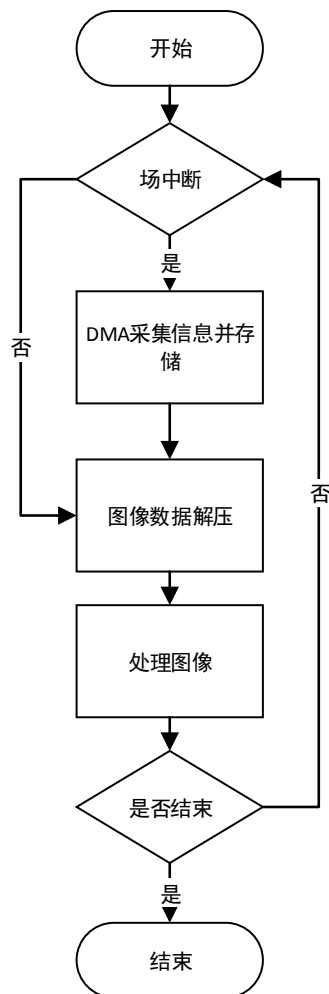


图 5.2 图像采集软件流程

5.3 提取黑线

5.3.1 提取算法总述

在黑线提取时采用了经典的边沿跟踪提取和隔行处理的方法。采集效果相对较好。

经过分析，发现摄像头返回的数据，在近端比较清楚，即使在不同光线下，变化也不是太大，于是对前三行进行可靠提取，即对近端 3 行进行全行扫描，直到找到黑线行为止，并且提取左右的黑线边缘。

在前 3 行确定的情况下，以此 3 行为基准确定是否向上提取，如果底部三行找到则继续向上搜索，反之放弃搜索。对后所有行进行边沿搜索时，依照前两行的黑线的斜率，若斜率为负，黑线的变化趋势就是向右的，此时便采取从左向右搜索，若斜率为正，黑线的变化趋势就是向左的，此时便采取从右向左搜索。但为了排除十字叉的干扰，如果前一行列值过大，则采取从右向左搜索，如过小则是从左向右搜，搜索范围会根据黑线的远近端以及前一行是否丢失黑线来确定。

若前 3 行没有搜索到黑线，则认为黑线丢失，黑线丢失标志置 1，即退出搜索，黑线提取完毕，有效行在退出前要再次进行更新。

5.3.2 近端黑线提取算法

由于近端黑线的数据很可靠，所以在近端处进行整行搜索，以获取基准值，如果近端黑线没有提取出，则认为图像丢失，会保持上一场的值。

为了防止近端黑线提取出错。在此需要将提取的基准和上一场图像的值进行比较，如果差距过大，就认为黑线提取出错，这场的采集的图像便失效，保持上一场的控制量。

5.3.3 远端黑线提取算法

根据近端 3 行提取的黑线的基准，并且在图像没有出错的情况下，进行后 117 行的黑线提取。由于摄像头采集的图像是呈现梯形畸变的，在远端会采集到赛道外面，这时的采集回的数据情况是相当复杂的，所以对于远端黑线的提取，需要依靠近端确定的基准值，以及图像远近和上一行是否提取到黑线等情况，来一定量得缩小或扩大扫描范围，并且依据图像的远近进行丢失图像的判断，在本程序中将图像从第 33 行到第 0 行分为 3 段及上两行提取的黑线位置差值来给定扫描范围。在上一行黑线丢失之后，在这不同的三段中分别扩大一定的范围进行继续搜索，直到又提取到黑线后，再将丢失的黑线进行拟合处理。在这三段内如果黑线丢失的行数超过一定的范围，就认为黑线扫描终止。远端黑线的提取算法

和近端黑线的提取算法一样，都是通过判断二值化结果来进行黑线的提取。前 3 行最稳定的一行算法流程图如图 5.3 所示。

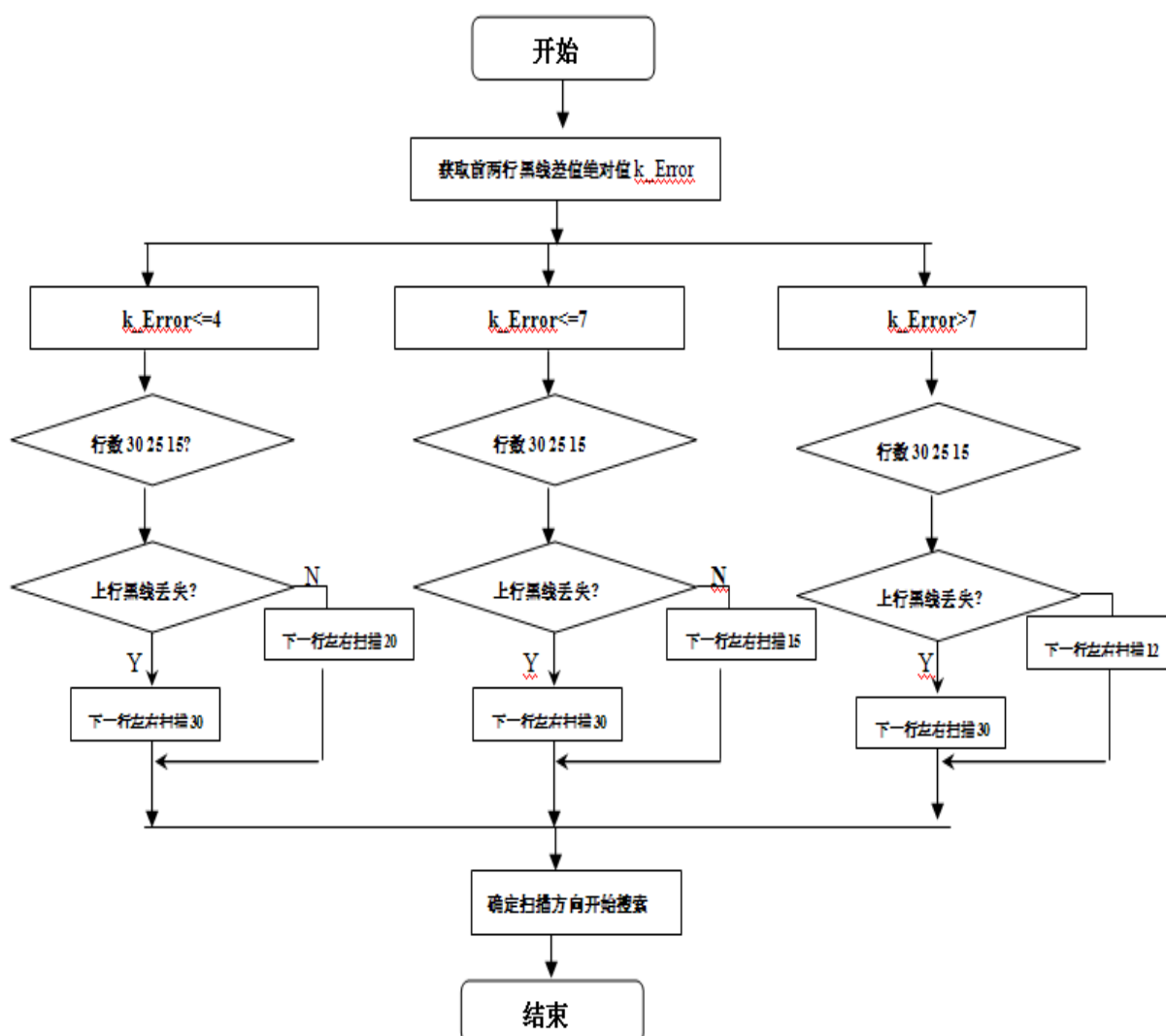


图 5.3 远端黑线提取算法流程图

5.3.4 起跑线的识别

要能够准确的识别起跑线那么我们首先要能够分析出是跑线的特征，并且抓住最主要的特征，也就是他不同于赛道其它的地方。

图 5.4 是标准的起跑线示意图。赛道有一个长为 1 米的出发区，图 5.5 为串口上位机发回的起跑线效果图，计时起始点。两边分别有一个长度 10 厘米黑色计时起始线，赛车前端通过起始线作为比赛计时开始或者结束时刻。由上可知，起跑线前后

最少有 1m 的直道，有两条 10cm 的黑带，并且他与黑色引导线的距离为 4.5cm，最后是起跑线不会出现在窄道区域。有了上面对起跑线的特征的分析就可以对起跑线进行识别。

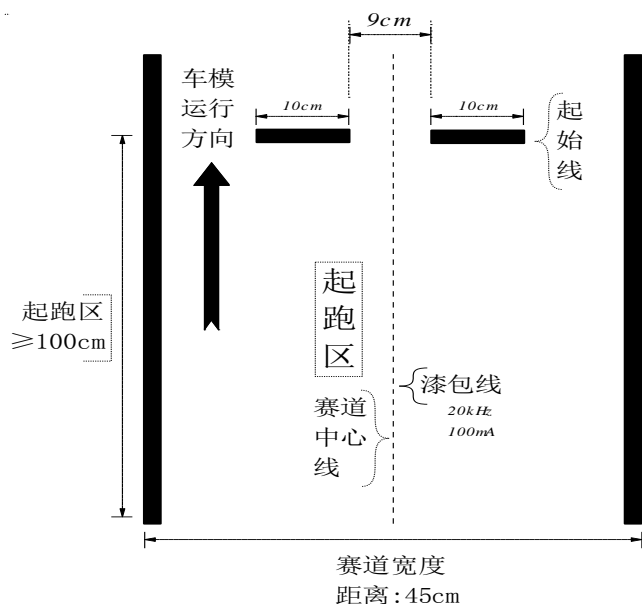


图 5.4 起跑线特征图

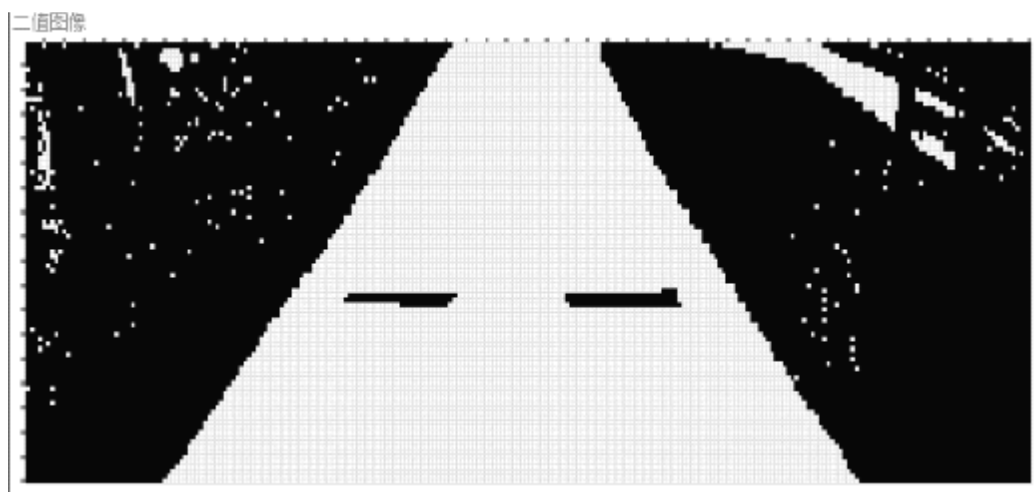


图 5.5 串口返回的起跑线数据

起跑线区别于十字交叉和别的干扰的最主要特征应该是由左边沿黑色引导线到起跑线左边部分的黑线段的跳变，和起跑线右边部分的黑线段到左边沿黑色引导线的跳变，因为在正常的情况下，黑线的跳变次数为左右各一次。那么我们就在检测到跳变次数大于 1 次的时候检测起跑线，而且在车出发大概 5s（通

过统计图像采集的场次来估算时间，5s 大概 150 场）后才开始检测起跑线。

步骤如下：

- (1) 系统时间是否达到起跑线检测时间。
- (2) 检测是否为直道。
- (3) 判断跳变的次数是否达到检测要求。
- (4) 从中心向两侧确定中间那段白色赛道。
- (5) 对白色赛道的宽度做进一步的要求。

因为前面已经说过起跑线的在直道上，弯道不可能出现起跑线，所以主要检测赛道是否为直道，在弯道上不检测。在车开始 5s 内不可能出现起跑线，因此在 5s 内也不检测起跑线，然后当它的跳变次数超过 1 的时候，就从黑线的中心位置向两侧搜索另外一条黑线，并且记录这两次搜索的距离，这个距离应该左右大致相等，并且距离有个上限和下限。当达到这些条件的时候就认定是起跑线。本方案在实际调试和应用中，发现其算法简单，特征明确，并且准确率较高。

5.4 黑线中心求解算法

5.4.1 逆透视变换计算黑线位置

由于摄像头与地面成一定角度安装,造成采集到的图像存在桶形失真，为了对小车进行速度和转向控制，求出黑线的具体位置，需要将图像校正成正投影的形式，故采用逆透视变换。透视变换是中心投影的射影变换，在用非齐次射影坐标表达时是平面的分式线性变换，具有如下的形式：

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \frac{\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}}{1 + \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}}$$

确定两个平面之间的投影变换只需要在两个平面上找到四对“物点和像点”，有这四对点的对应关系就可以确定 a, b, c, d, e, f, u, v。具体而言，用一个规范化坐标系（平面）作为重投影平面，首先建立从规范化坐标（Normal Coordinate）

$$\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \frac{\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}}{1 + \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}}$$

到原始图像坐标（Original Coordinate）的变换，其形式为：

在地面上取定一个矩形（不一定是正方形），要求它的边平行或垂直于车体轴线，这样的矩形在原始图像中显示为一个等腰梯形。

设 A 点的坐标为 (x_1, y_1) ，C 点坐标为 (x_2, y_2) ，则 B、D 的坐标分别是 $(\Phi - x_1, y_1)$ 和 $(\Phi - x_2, y_2)$ 。经过投影，ABCD 四个点依次与规范化坐标系中的 $(0, 0)$ ， $(1, 0)$ ， $(0, 1)$ ， $(1, 1)$ 四个点相对应。把此对应关系代入上式，得：

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \frac{\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}}{1 + \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix}}$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \frac{\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}}{1 + \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}}$$

$$\begin{bmatrix} \Phi - x_1 \\ y_1 \end{bmatrix} = \frac{\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}}{1 + \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}}$$

$$\begin{bmatrix} \Phi - x_2 \\ y_2 \end{bmatrix} = \frac{\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}}{1 + \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}}$$

可以求得如下的各待定常数：

$$\begin{cases} c = u = 0 \\ a = \Phi - 2x_1 \\ v = \frac{a}{\Phi - 2x_2} - 1 \\ b = x_2v + x_2 - x_1 \\ d = y_2v + y_2 - y_1 \\ e = x_1 \\ f = y_1 \end{cases}$$

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \frac{\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}}{1 + \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}}$$

以上变换的逆变换 OrgToNormal 的形式是

把四对点的对应关系代入上式，得到待定系数的计算方法为：

$$\begin{cases} a = (y_1 - y_2) / \Delta \\ v = 2(x_2 - x_1) / \Delta \\ b = \frac{x_2 - x_1}{y_1 - y_2} a \\ e = -ax_1 - by_1 \\ d = \frac{1 + y_2 v}{y_2 - y_1} \\ f = -y_1 d \\ c = u = 0 \end{cases}$$

其中 $\Delta = y_1(\Phi - 2x_2) - y_2(\Phi - 2x_1)$,

公式中只需确定 (x_1, y_1) 和 (x_2, y_2) ，确定这几个参数的方法如图 5.6 所示：

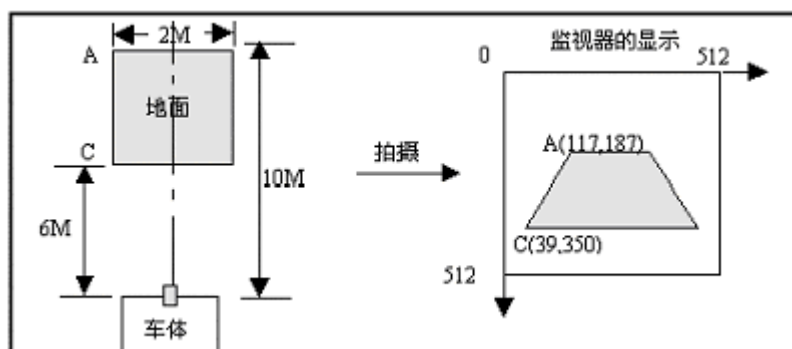


图 5.6

这样就可求解出对应的世界坐标，再通过一系列的转化变为离车前后左右的距离坐标，从而用于精确控制。

5.5 控制算法

5.5.1 电机控制策略

整体思想：用 PWM 波的等面积效应来控制电流大小，即控制电机的转速，用全桥来控制电流的方向，从而可控制电机的正反转。根据赛道的情况，通过比

例或者建立的模型进行给定速度，再通过光电编码器，对轮子实际速度的测量，通过反馈偏差进行 PID 校正控制。

小车行驶途中，在不同的路况时，摄像头所采集到的图像的偏离中心和需要的转弯半径是会变化的。在过弯时，采集到的图像偏差会增大，半径会减小，这样便可利用偏差和半径的大小来控制速度。所以我们详细分析了这两种控制量作为速度给定的变量，而共同点是人为地给定一个最高速度和一个最低速度，而所计算出的其它速度值介于两者之间，不同点在于构造的数学模型的特性不同，最终实现的效果也有较大差异。具体分析如下：

速度给定：

模型一： $\text{TestSpeed} = f \cdot \sqrt{\text{Absolute}(\text{true_r})}$ ， f 为人为给定速度系数， f 越大，整体速度越高。如图 5.12。我们理想的速度给定应该满足这样一个原则，在长直道入弯时能够及时降到需要的速度，而在一定半径的弯道时能够保持相对较高的弯道速度。而这些性能主要体现在速度模型的曲线斜率上，长直道入弯时需要斜率略有降低而不能太高，而在连续弯道时斜率不能太低，否则过弯速度久低不上。分析此模型曲线可知，这种模型在长直道入弯（即 R 由大减小）的情况时，利于减速，所以在长直道较多的情况下，小车能够在直道上保持较高的速度，而在弯道较多时也能够保持比较合理的速度。

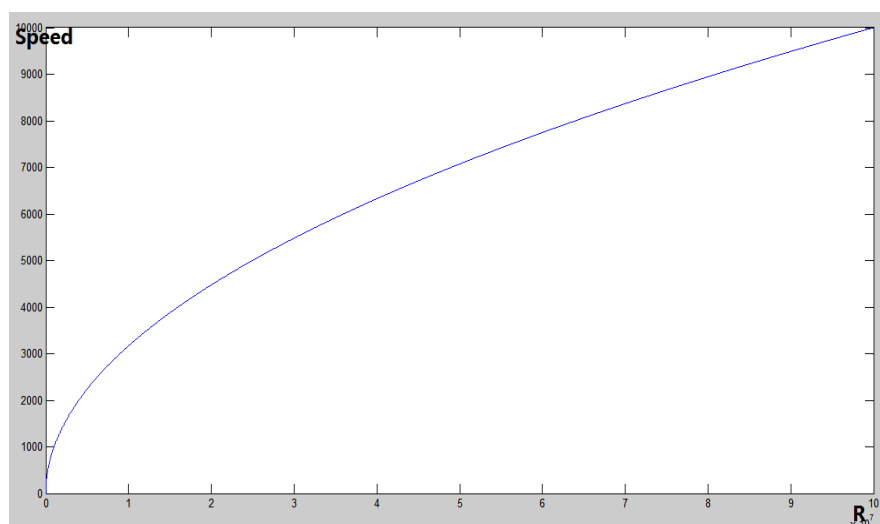


图 5.7 半径——速度曲线模型一

模型二：二次函数控制曲线

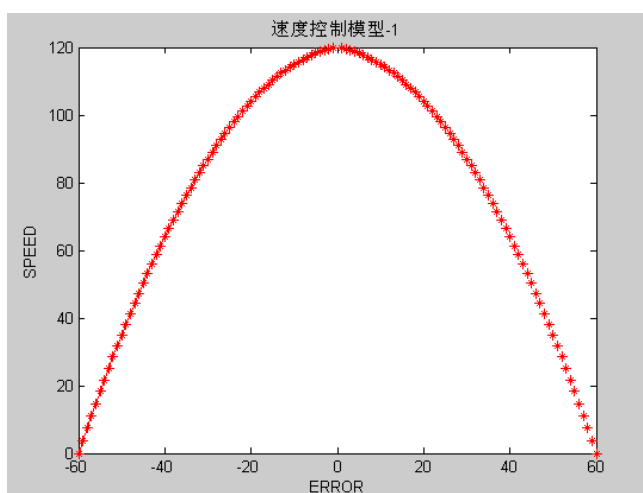


图 5.8 偏差—速度模型二

此模型相应的控制函数为：

$$\text{Error} > 0 \quad \text{speed} = (12000 - 2 * \text{Error} * \text{Error} - 20 * \text{Error}) / 100;$$

$$\text{Error} < 0 \quad \text{speed} = (12000 - 2 * \text{Error} * \text{Error} + 20 * \text{Error}) / 100;$$

这种模型在长直道入弯的情况时，不利于减速，所以在长直道较多的情况下，并要求小车在直道上保持较高的速度，这种模型小车是很难减速成功的。精度上可能无法满足这中函数的控制要求，这就无形的加大了调试的难度。

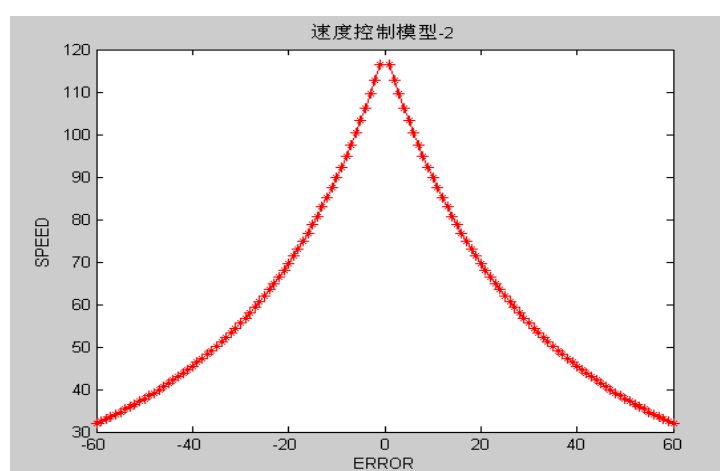


图 5.9 偏差---速度模型三

此模型相应的控制函数为：

$\text{Error} > 0$ $\text{speed} = 120 / ((\text{Error} / 20 + 1) * (\text{Error} / 15 + 1));$

$\text{Error} < 0$ $\text{speed} = 120 / ((\text{Error} / 20 - 1) * (\text{Error} / 15 - 1));$

这种模型的减速相当明显，所以在一些小弯道较多的情况下，小车减速非常迅速，导致弯道速度缓慢。还有个难题同模型一是一样的，单片机的计算精度问题，会影响这种模型的实际控制效果，不利于调试。

最终确定使用模型一作为速度给定方式，通过测试，这样的速度给定方式能够发挥出小车在各种半径赛道时保持最高速度。

(1) 刹车处理

当反馈速度值 $\text{Feedback_rate} > \text{TestSpeed}$ 时，表示当前需要减速，而减速的程度则需要按照反馈值与期望速度值的差值大小决定，相差越大的情况下为保证及时减到安全速度，则需要将电机反转进行减速。当然，这里还应考虑到编码器反馈速度的准确度，需要适当进行补偿。另外，可以结合有效行数进行减速判别。

(2) 低速状态下的突加速处理

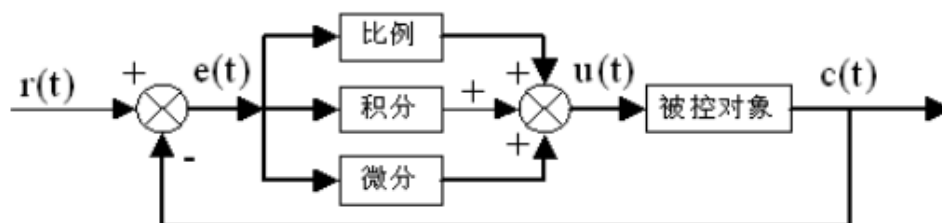
在转弯半径很大的情况下，如果出现速度很低的情况，这时小车的舵机已经完全有时间改变转向，此时给定的速度过低便会影响整个小车的行驶速度，这时便需要给定全速，让小车突加速。

(3) 启动方式处理

采用按期望速度开环启动，开环启动后迅速转换为闭环启动。整个智能车控制系统速度控制采取闭环控制，但在调试过程中，我们发现如果启动瞬间采用闭环，由于初始速度为零，根据 PID 控制理论将使得电机 PID 系统有很大的输入，从而导致启动速度产生严重的超调，导致舵机等模块的不稳定工作，严重的时候小车会直接冲出跑道。解决此种情况的方案有两种，一是采用缓启动，即初始时给定较小的 KP 比例系值，待数场后，变换为正常的 KP 系数，这种方案的缺点是启动速度过慢，丢失时间多。二是开环进行启动，随后变换为闭环，这种方案既提升了启动速度，又不至于使系统不稳定。综上，我们采用后者，并取得了不错的控制效果。

5.5.2 电机的 PID 控制

图 5.10 PID 控制器原理框图



$$u(k) = K_p(e(k) + \frac{T}{T_I} \sum_{i=0}^k e(i) + \frac{T_D}{T} [e(k) - e(k-1)]) \quad \text{公式 3}$$

$$u(k-1) = K_p(e(k-1) + \frac{T}{T_I} \sum_{i=0}^{k-1} e(i) + \frac{T_D}{T} [e(k-1) - e(k-2)]) \quad \text{公式 4}$$

$$\Delta u = K_p([e(k) - e(k-1)] + \frac{T}{T_I} e(k) + \frac{T_D}{T} [e(k) - 2e(k-1) + e(k-2)]) \quad \text{公式 5}$$

$$= K_p \Delta e(k) + K_I e(k) + K_D [\Delta e(k) - \Delta e(k-1)]$$

k: 采样序列号;

r(k): 第 k 次实际输出值;

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;

u(k): 第 k 次输出控制量;

u(k-1): 第 k-1 次输出控制量;

T: 采样周期;

K_p : 比例系数;

$K_I = \frac{T}{T_I}$ 积分系数;

: 微分系数;

利用光电编码器获得第 k 次实际输出的实际速度值，通过电机本身的参数和给定的 PWM 占空比获得给定值，然后按照上述公式和时间形成相应关系进行

PID 控制。

PID 参数的调节：进行分步调节法，先让积分系数 T_I 和微分系数 T_D 为 0，设定好比例系数 K_P ，然后调节 T_I 系数，再调节 T_D 系数。

5.6 本章小结

本章详细介绍了系统软件算法，包括图像采集、处理和电机、舵机控制算法。详细描述了舵机控制量的求解以及智能车在各种赛道类型的速度模型建立过程，明确了智能小车系统控制的数学理论基础，取得了较为突出的成效。

第六章 调试方法

通过调试工具才能知道单片机正在做的事，从而通过分析能知道单片机是否在按照人的意图去执行，如果没有调试工具，想要做好智能车的控制，将是不可能的。

6.1 本系统中所使用调试工具简介

在本系统的完成中，所使用到了一下三种调试组合工具：

(1) IAR Embedded Workbench for ARM 在线调试

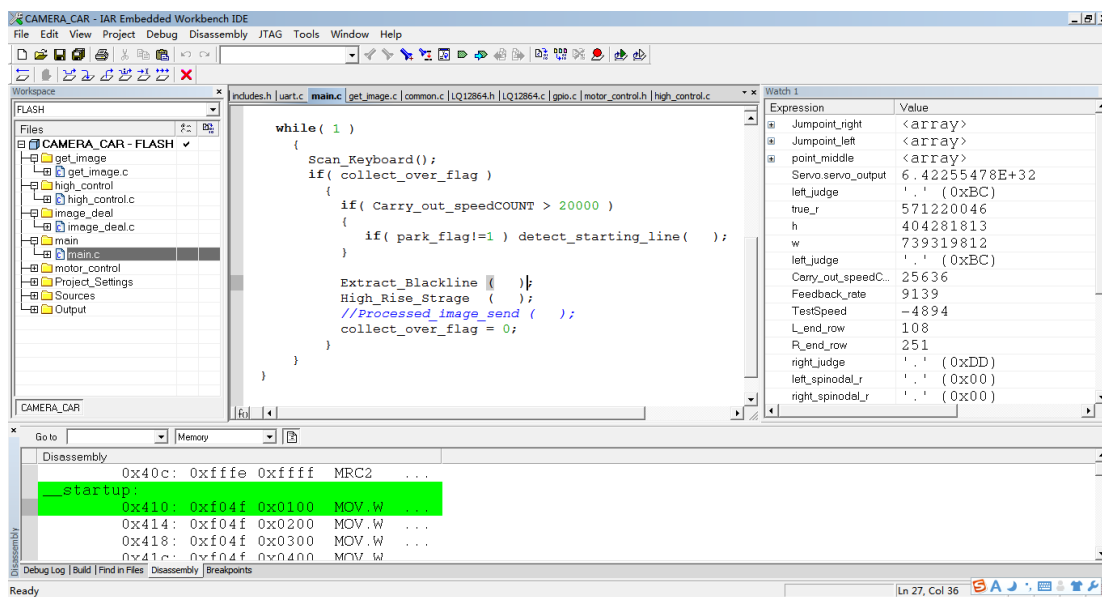


图 6.1 IAR Embedded Workbench for ARM 调试界面

IAR 是最基本的调试工具，在没有其他调试工具的时候这个已经能完成很多事，在后面会单独讲解利用这套调试工具可以做的事，并且熟练使用这个最基本的调试工具是做智能车最基本的要求。

(2) 串口示波器和无线联合调试工具

这套调试工具是我们学校自己独立开发的串口调试工具，其可以直接使用串口实现单片机和PC机的通讯，通过这个调试工具可以查看静态和动态的数据。

其上位机的人机界面如下图，操作简单，功能强大。

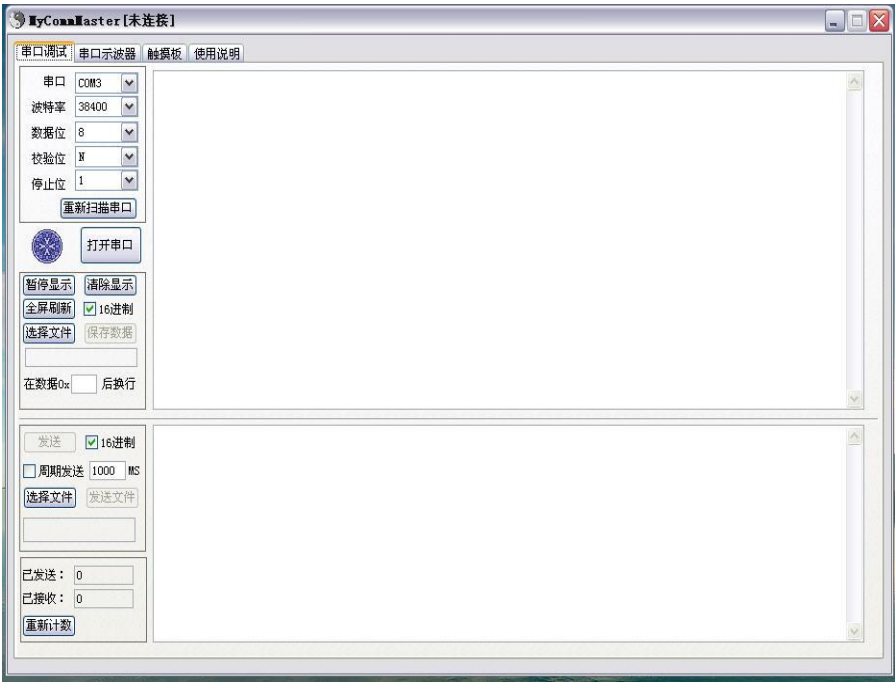


图 6.2 串口示波器调试界面

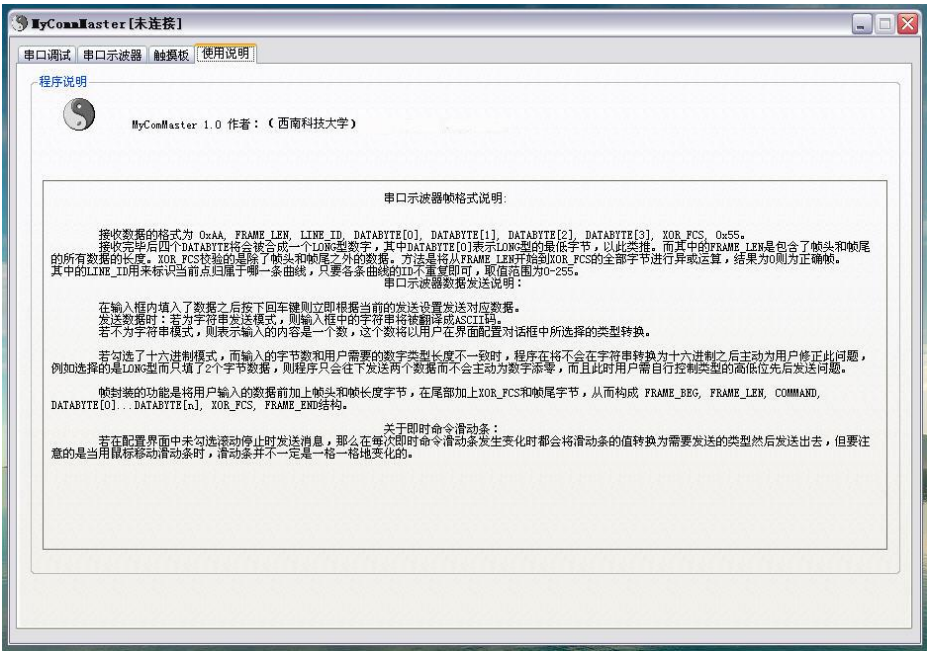


图 6.3 串口示波器使用说明

(3) 无线和 MATLAB 组合调试工具

这套调试工具是我们组内人员自主设计开发，其主要用来分析查看动态数

据，和监控小车速度等信息，加上 MATLAB 的强大功能，使其具有更多的数据分析功能。

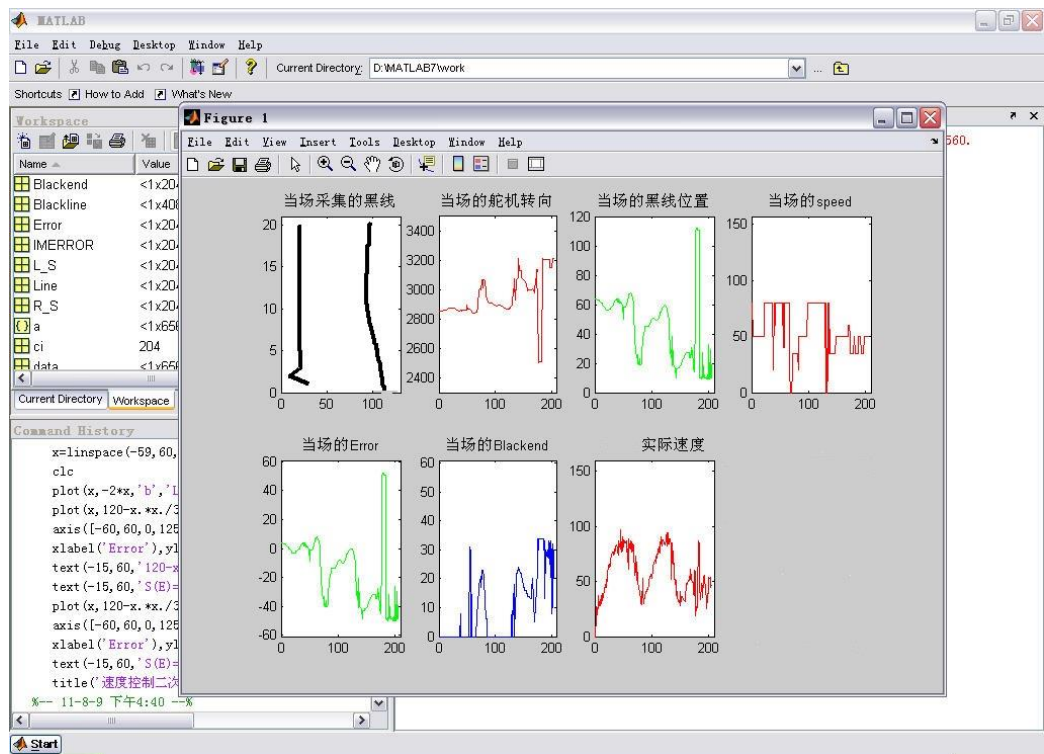


图 6.4 matlab 调试界面

6.2 IAR Embedded Workbench for ARM 在线调试工具使用

通过在线调试，可以静态的查看定义的所有变量值，可以看到摄像头在某一时刻的采集到的数据值，可以用这种调试方法做到：

- (1) 摄像头的安装固定，摄像头前瞻的确定，每两采集行见距离的测量；
- (2) 摄像头采集数据正误的检查；
- (3) 在静态时，观测黑线提取的是否正确；
- (4) 可以检查各个硬件模块是否工作在正常情况下，并对程序的执行正误做出分析；
- (5) 可进行测速模块的反馈值检测；

这套调试工具是在前期软件编写过程中最常使用的一套工具，也是和小车机械结构安装的辅助工具。对于这套调试工具的使用之处还有许多地方，在此就

不一一说明。

6.3 串口示波器+无线组合工具使用

关于串口示波器的使用，本系统主要用于返回黑线数据的检测，查看摄像头采集的数据情况以及黑线提取情况，以便在做黑线提取算法时分析用。其是写好黑线提取算法的关键性调试工具，如果没有这个调试工具，那就根本不知道黑线提取为何出错，从何修改。在后期的数据处理的算法上，这套工具的作用相当大，在网上也有许多类似的串口示波器，在调试也可以使用，但最终采用了我们学校自己开发的软件，使用起来也一样的方便。上位机显示的处理过后的图像信息如图 6.5、图 6.6 所示。

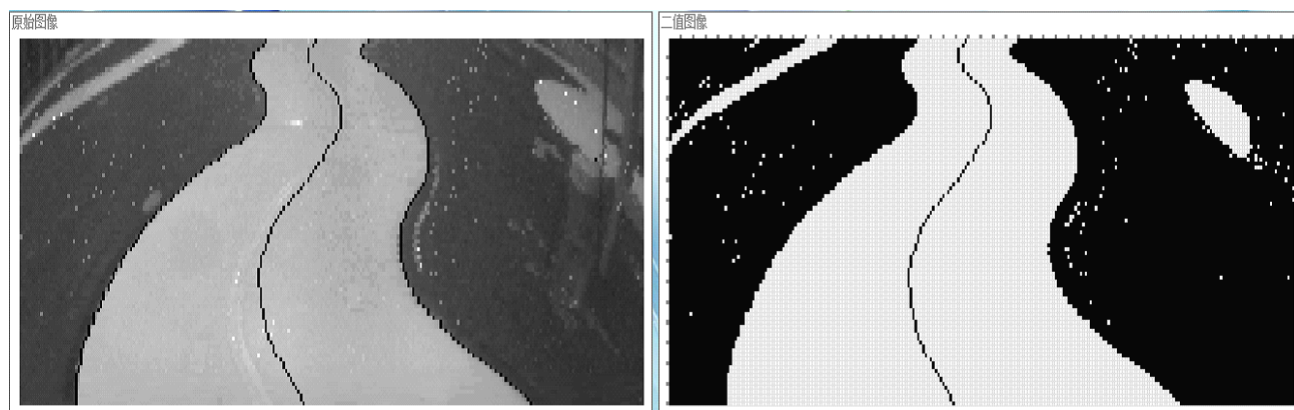


图 6.5 小 S 弯图像处理

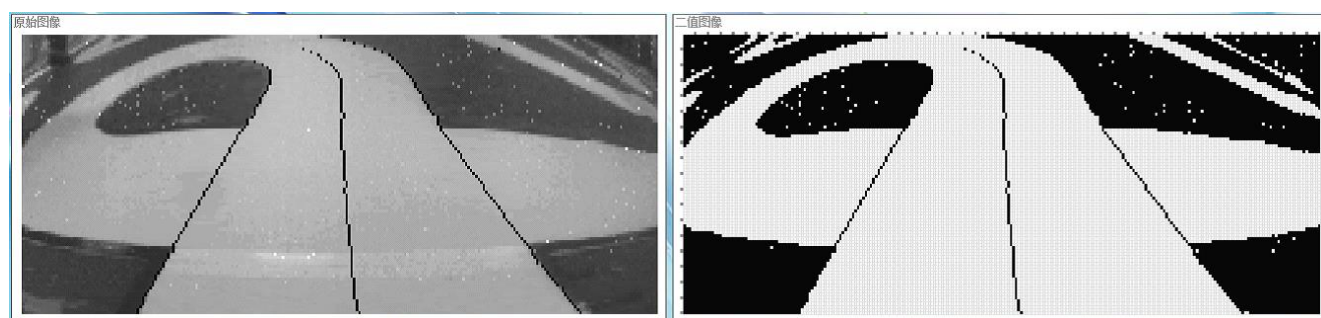


图 6.6 十字弯图像处理

6.4 无线+MATLAB 组合工具使用

对于这套工具的使用，主要用于小车在跑动过程中，测试其的动态数据，先通过无线发回数据，在经过 matlab 读取数据，再使用 matlab 强大的数据分析功

能，便可以制作多种多样的数据监控程序，在本系统的设计中主要使用这套调试工具实现的功能有：

- （1）对于动态提取黑线正误的监控；
- （2）对于舵机速度控制量的监控和检查；
- （3）通过 matlab 分析速度的变化情况，拟合出合适的控制曲线；
- （4）通过 matlab 分析数据优化速度舵机等控制策略；
- （5）调节 PID 参数；

6.5 本章小结

本章介绍了调试方案的运用，良好的辅助调试手段更有利于小车的调试，但是由于水平有限，所用调试方案仍然不是很好。希望以后发现更优的调试方案。

第七章 结论

7.1 模型车主要技术参数说明

表7.1 系统整体参数

项目	参数
路径检测方法（赛题组）	摄像头组
车模几何尺寸（长、宽、高）（毫米）	285mm/185mm/300mm
车模轴距/轮距（毫米）	200mm/140mm
车模平均电流（匀速行驶）(毫安)	3000mA
电路电容总量（微法）	1900 μ F
传感器种类及个数	摄像头/1 速度传感器 /1
赛道信息检测空间精度（毫米）	150mm（近端）、1900mm(远端)
赛道信息检测频率（次/秒）	112
主要集成电路芯片种类	AMS1117 OV7725 LT1764 TPS7350
车模重量（带有电池）（千克）	1.2

7.2 主要内容总结

首先根据竞赛规则及功能要求，设计以恩智浦公司的 32 位单片机 K60 和数字摄像头为核心的智能车系统。通过摄像头采集的送入单片机，在单片机内计算出赛道的轨迹状况，选择最优行进路线。整个系统包括车体机械结构设计和系统的软件、硬件系统设计。车体机械结构设计主要包括编码器和图像传感器的安装、

舵机和后轮调节等；硬件系统设计完成了电源管理模块、电机驱动模块、速度测量模块、辅助调试模块、图像采集处理模块、舵机控制模块和单片机模块等的电路设计和调试；软件系统设计完成了各功能模块的算法及程序设计，包括图像采集算法设计、提取黑线及黑线中心求解算法设计以及舵机和电机的 PID 算法设计。所设计的系统经过测试，赛车能够快速安全行驶。

7.3 存在问题及解决方案

(1) 在小车的制作过程中，小车出现了死机后重新启动的现象。在更换大功率稳压芯片和更换了系统的供电系统后，这个问题得到了很好的解决。

(2) 模型车速度和转弯的协调问题。速度提高了，转弯就不稳，要顺利的转弯，速度又上不去。为了协调这个问题，在算法上多次改进，但是效果不好。后来在机械上加以改进，从而最后取得比较好的效果。从而认识到了机械对于小车的重要性，同时也认识到系统各个部分都是不可分离的。

(3) BTS7960所存在的驱动问题。在调试过程中，芯片发热严重，有时候还会出现过热保护。后来我们尝试自己用MOS管搭建全桥驱动，最后经过试验，电路搭建成功，发热的问题得到解决。

一个良好的系统仍然需要不断完善和更新，为了保证车子能够稳定达到最高平均速度，智能车系统的设计还是在不断的改进的过程。但是因水平有限，我们的小车仍然有很多需要改进的地方。

7.4 心得体会

小车的制作涉及许多方面，包括机械、电子、控制等方面。在小车的制作过程中，前期主要把精力放在了控制方面，但是随着速度的提高，机械方面的问题便暴露出来了，但由于在机械方面缺乏经验，在借鉴了别人的经验和自己的多次尝试后，小车的机械问题也得到了比较好的解决，但是小车还是有很大的调整空间。

在智能车制作过程中，遇到过很多问题，在老师和师兄们的帮助下，再加上我们的努力调试，终于在不断的改进中，完成了智能小车的制作。在智能小车的制作过程中，不仅丰富了我们的知识，并且锻炼了动手能力，同时培养了团队合作精神。在此感谢智能汽车竞赛组委会对本次比赛的大力支持，感谢学校以及学

院对我们的大力支持，同时也感谢帮助我们的老师和实验室师兄师弟们。

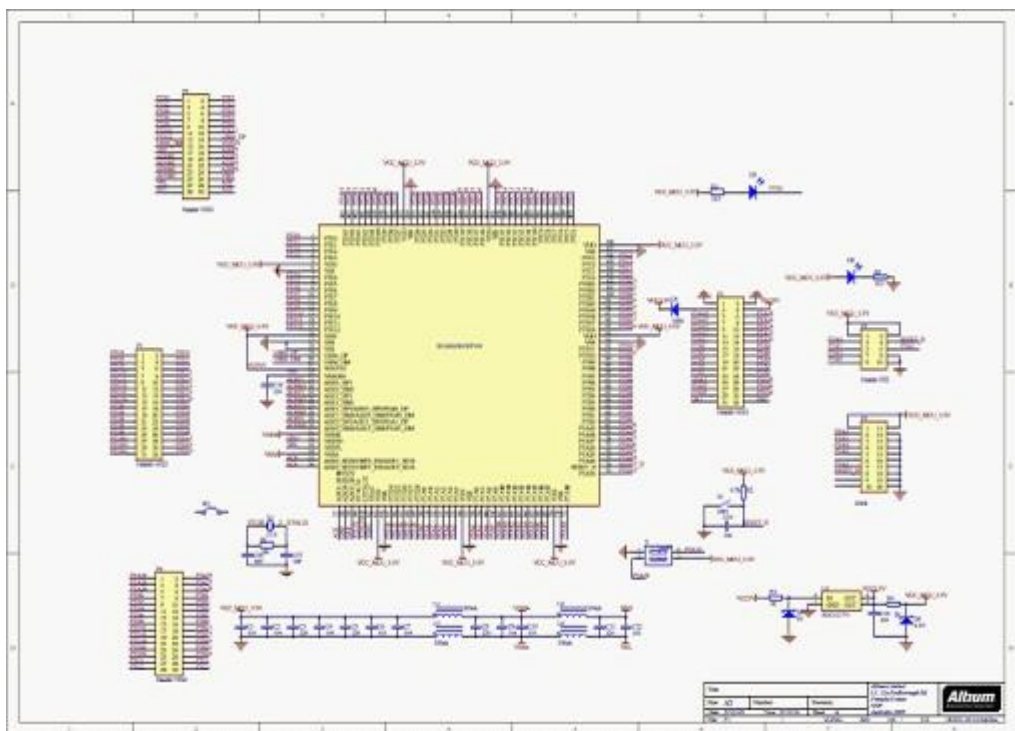
参考文献

- [1]卓晴、黄开胜、邵贝贝等编. 学做智能车—挑战“飞思卡尔”杯. 北京: 北京航空航天大学出版社, 2007. 366~369
- [2]黄开胜、金华民、蒋狄南著. 韩国智能模型车技术方案分析. 北京: 北京航空航天大学出版社, 2007. 45~51
- [3]卓晴著. 智能汽车自动控制器方案设计. 北京: 北京航空航天大学出版社, 2007. 5
- [4]邹朗豪、李文昌、苏启健著. Rapid prototype and smartcar design. 北京: 北京航空航天大学出版社, 2007. 71~86
- [5]卓晴、王璿、王磊著. 基于面阵 CCD 的赛道参数检测方法. 北京: 北京航空航天大学出版社, 2007. 35~40
- [6]Freescale, Inc. MC9S12DT128B Device User Guide V01.09. Freescale Semiconductor, Inc, 2002
- [7]Freescale, Inc. MC9S12DT128B CRG Block User Guide V03.08. Freescale Semiconductor, Inc, 2002
- [8]Freescale, Inc. MC33886 Datasheet. U.S.A:Freescale Semiconductor, Inc, 2005
- [9]李立国、刘旺、郝杰等著. 基于大前瞻光电识别和道路记忆方法的智能车. 北京: 北京航空航天大学出版社, 2007. 338~350
- [10] ON Semiconductor. NCP630 3.0A Fast Linear Voltage Regulators
- [11] 余灿键、程东成、李伟强著. PID 算法在智能汽车设计上的应用. 基于大前瞻光电识别和道路记忆方法的智能车. 北京: 北京航空航天大学出版社, 2007. 128~137
- [12] 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. 23
- [13]广州周立功单片机发展有限公司著. I²C 总线规范. 广州: 广州周立功单片机发展有限公司, 2000
- [14]林辛凡,李红志,黄颖. 第五届“飞思卡尔”杯全国大学生智能汽车邀请赛技术报告[R], 2007 清华三角洲队(原清华一队)

[15]吴小平、刘万春、朱玉文著. 基于 TMS320C32 的视觉图像处理系统[J]. 探测与控制学报. 2000 年第 22 卷第 1 期: 18~22

[16] 徐济仁、牛纪海、陈家松等著. FIFO 存储器 IDT7205 及其在接收机采样处理板中的应用. 有线电视技术, 2004 年第 24 期: 62~64

附录 A：系统主控板原理图



附录 B：源程序

1、搜底三线

```
if(position_old>0)

{

    /*****搜索左右黑线*****/

    extract_left_blackline(0,COLUMN_MID+15,0);

    extract_left_blackline(1,COLUMN_MID+15,0);

    extract_left_blackline(2,COLUMN_MID+15,0);

    if(Flag[1].left_flag==1&&Flag[2].left_flag==1&&Absolute(Jumppoint_Left[0]-
Jumppoint_Left[2])<10)

    {

        extract_right_blackline(0,Jumppoint_Left[0],CAMERA_W -1);

        extract_right_blackline(1,Jumppoint_Left[1],CAMERA_W -1);

        extract_right_blackline(2,Jumppoint_Left[2],CAMERA_W -1);

    }

    else

    {

        Flag[0].left_flag=0;

        Flag[1].left_flag=0;

        Flag[2].left_flag=0;

        Jumppoint_Left[0]=0;

        Jumppoint_Left[1]=0;

        Jumppoint_Left[2]=0;

        extract_left_blackline(0,COLUMN_MID+60,0);

        extract_left_blackline(1,COLUMN_MID+60,0);

        extract_left_blackline(2,COLUMN_MID+60,0);
```

```

        if(Flag[1].left_flag==1&&Flag[2].left_flag==1&&Absolute(Jumppoint_Left[0]-
Jumppoint_Left[2])<10)

        {

            extract_right_blackline(0,Jumppoint_Left[0],CAMERA_W -1);

            extract_right_blackline(1,Jumppoint_Left[1],CAMERA_W -1);

            extract_right_blackline(2,Jumppoint_Left[2],CAMERA_W -1);

        }

    else

    {

        extract_right_blackline(0,COLUMN_MID,CAMERA_W -1);

        extract_right_blackline(1,COLUMN_MID,CAMERA_W -1);

        extract_right_blackline(2,COLUMN_MID,CAMERA_W -1);

        if(Flag[1].right_flag==0&&Flag[2].right_flag==0)

        {

            extract_right_blackline(0,COLUMN_MID-60,CAMERA_W -1);

            extract_right_blackline(1,COLUMN_MID-60,CAMERA_W -1);

            extract_right_blackline(2,COLUMN_MID-60,CAMERA_W -1);

        }

    }

    if(Jumppoint_Right[0]-Jumppoint_Left[0]<CAMERA_W *0.25)

    {

        Image_error_flag=1;

    }

}

```

```

    }

    else

    {

        extract_right_blackline(0,COLUMN_MID-15,CAMERA_W -1);

        extract_right_blackline(1,COLUMN_MID-15,CAMERA_W -1);

        extract_right_blackline(2,COLUMN_MID-15,CAMERA_W -1);

if(Flag[1].right_flag==1&&Flag[2].right_flag==1&&Absolute(Jumppoint_Right[0]-
Jumppoint_Right[2])<10)

    {

        extract_left_blackline(0,Jumppoint_Right[0],0);

        extract_left_blackline(1,Jumppoint_Right[1],0);

        extract_left_blackline(2,Jumppoint_Right[2],0);

    }

    else

    {

        Flag[0].right_flag=0;

        Flag[1].right_flag=0;

        Flag[2].right_flag=0;

        Jumppoint_Right[0]=CAMERA_W -1;

        Jumppoint_Right[1]=CAMERA_W -1;

        Jumppoint_Right[2]=CAMERA_W -1;

        extract_right_blackline(0,COLUMN_MID-60,CAMERA_W -1);

        extract_right_blackline(1,COLUMN_MID-60,CAMERA_W -1);

        extract_right_blackline(2,COLUMN_MID-60,CAMERA_W -1);

```

```

if(Flag[1].right_flag==1&&Flag[2].right_flag==1&&Absolute(Jumppoint_Right[0]-
Jumppoint_Right[2])<10)

    {

        extract_left_blackline(0,Jumppoint_Right[0],0);

        extract_left_blackline(1,Jumppoint_Right[1],0);

        extract_left_blackline(2,Jumppoint_Right[2],0);

    }

else

    {

        extract_left_blackline(0,COLUMN_MID,0);

        extract_left_blackline(1,COLUMN_MID,0);

        extract_left_blackline(2,COLUMN_MID,0);

        if(Flag[1].left_flag==0&&Flag[2].left_flag==0)

            {

                extract_left_blackline(0,COLUMN_MID+60,0);

                extract_left_blackline(1,COLUMN_MID+60,0);

                extract_left_blackline(2,COLUMN_MID+60,0);

            }

    }

}

if(Jumppoint_Right[0]-Jumppoint_Left[0]<CAMERA_W *0.25)

    {

        Image_error_flag=1;

    }

}

```

2、左搜线再补线程序

```
void extract_left_crossroad()
{
    uint8 row=0;

    uint16 lose_count_L=1;

    uint16 Left_flag_count=0;

    uint8 Left_row=0;

    // int16 forecast_left=0;

    uint8 j=0;


    Left[0]=0;

    Left[1]=0;

    Left[2]=0;


    Left_flag=0;

    if(L_end_row<=2)    //前提 L_end_row<=37
    {
        L_end_row=0;
    }


    for(int i=0;i<37;i++)
    {
        if(Image_error_flag==1)
        {
            //Image_error_flag=0;
```

```

        break;
    }

for(row=L_end_row+3*lose_count_L+2;row<L_end_row+3*lose_count_L+5;row++)// 搜 三
行空三行
{
    if(row>CAMERA_H-1) break;

    if(Jumppoint_Right[R_end_row]<COLUMN_MIN+20)

        break;

    if(Jumppoint_Left[L_end_row]>CAMERA_W -20)

        break;

    if((Jumppoint_Right[R_end_row]<COLUMN_MIN+20)&&row>R_end_row)

        break;

    // if(init_R_end_row>CAMERA_H/2&&row>CAMERA_H/2+5)break;

    if((R_end_row>CAMERA_H*3/4)&&(row>=R_end_row+5))

        break;

    if(R_end_row>5)

    {

        if(Jumppoint_Right[R_end_row-3]-
Jumppoint_Right[R_end_row]>1&&row>R_end_row)

            break;

            if(Jumppoint_Right[R_end_row-6]-Jumppoint_Right[R_end_row-
3]>1&&row>R_end_row)

                break;

                //if(Jumppoint_Right[R_end_row-3]-Jumppoint_Right[R_end_row-
2]>1&&row>R_end_row)break;

```

```

        //if(Jumppoint_Right[R_end_row-4]-Jumppoint_Right[R_end_row-
3]>1&&row>R_end_row)break;

    }

    if(row<R_end_row)

    {

        extract_left_blackline(row,Jumppoint_Right[row],Jumppoint_Left[L_end_row]);

    }

    else

if(Jumppoint_Left[L_end_row]<=COLUMN_MIN+3&&Jumppoint_Right[0]!=CAMERA_
W -1)

    {

extract_left_blackline(row,Jumppoint_Right[R_end_row],Jumppoint_Left[L_end_row]);

    }

    else

if(L_end_row==0&&Jumppoint_Left[L_end_row]>COLUMN_MIN+3&&Jumppoint_Left[L
_end_row]<=COLUMN_MIN+10&&Jumppoint_Right[0]!=CAMERA_W -1)

    {

extract_left_blackline(row,Jumppoint_Left[L_end_row]+COLUMN_MID,Jumppoint_Left[L
_end_row]);

    }

    else

if(L_end_row==0/*&&Jumppoint_Left[L_end_row]>COLUMN_MIN/*&&Jumppoint_Right
[0]==CAMERA_W -1)

    {

```

```
extract_left_blackline(row,Jumppoint_Left[L_end_row]+COLUMN_MID+60,Jumppoint_Left[L_end_row]);
```

```
}
```

```
else
```

```
{
```

```
extract_left_blackline(row,Jumppoint_Left[L_end_row]+COLUMN_MID-15,Jumppoint_Left[L_end_row]);
```

```
}
```

```
if((Flag[row].left_flag ==1)&&Absolute(Jumppoint_Right[L_end_row]-Jumppoint_Left[row])<15)
```

```
break;
```

```
if(Flag[row].left_flag ==1&&Jumppoint_Left[row]<Jumppoint_Left[L_end_row])
```

```
break;
```

```
if(Flag[row].left_flag ==1)
```

```
{
```

```
Left_flag_count++;
```

```
Left[j]=row;
```

```
j++;
```

```
}
```

```
else
```

```
{
```

```
// Left_flag_count=0;
```

```
break;
```



```

    }

    if(Left_flag_count>=3)
    {
        Left_flag_count=0;
        j=0;
        if(Absolute(Jumppoint_Left[Left[2]]-Jumppoint_Left[Left[1]])<=3&&
            Absolute(Jumppoint_Left[Left[1]]-Jumppoint_Left[Left[0]])<=3)
            {
                Left_row=Left[0];
                Left_flag=1;
                /*Jumppoint_Left[Left[1]]=0;
                Jumppoint_Left[Left[2]]=0;
                Flag[Left[1]].left_flag =0;
                Flag[Left[2]].left_flag =0;*/
                add_line(Jumppoint_Left,L_end_row,Left_row);// 左边补线函数
                if(Left_flag==1)
                {
                    init_add_L_row=Left_row;
                    break;
                }
            }
        }

        if(row>CAMERA_H-1)
        {
            break;

```

```

    }

}

if(Left_flag==0)
{
    j=0;
    Left_flag_count=0;
    lose_count_L++;
    Left[0]=0;
    Left[1]=0;
    Left[2]=0;
    //Left[3]=0;
    // Left[4]=0;
}
else
{
    break;
}

}

//////////补完线后继续搜线//////////

if(Left_flag==1)
{
    // init_add_L_end_row=Left_row;

```

```

if(add_L_count==0&&L_add_line_check_flag==0)
{
    for(row=Left_row+3;row<CAMERA_H;row++)
    {
        forecast_left =(int16)( Jumppoint_Left[row-1]+(Jumpppoint_Left[row-1]-
Jumpppoint_Left[row-3])/2);//预测的该行的跳变位置

        extract_left_blackline(row,forecast_left+5,forecast_left-5);//根据前两行
跟踪查找

        if(Flag[row].left_flag==0)//当前行没有找到
        {
            L_end_row=row-3;

            break;
        }
    }

    if(row==CAMERA_H)

        L_end_row=CAMERA_H-1;

    first_add_L_end_row=L_end_row;
}

else
{
    for(row=Left_row+3;row<CAMERA_H;row++)
    {
        forecast_left =(int16)( Jumppoint_Left[row-1]+(Jumpppoint_Left[row-1]-
Jumpppoint_Left[row-3])/2);//预测的该行的跳变位置

        extract_left_blackline(row,forecast_left+5,Jumppoint_Left[row-1]-3);//

```

根据前两行跟踪查找

```
        if(Flag[row].left_flag==0)//当前行没有找到
        {
            L_end_row=row-3;
            break;
        }
    }

    if(row==CAMERA_H) L_end_row=CAMERA_H-1;
}

add_L_count++;
}

else// if(Left_flag==2)
{
    //for(int i=Left_row+2; i>L_end_row; i--)

    for(int i=CAMERA_H-1; i>L_end_row; i--)
    {
        Flag[i].left_flag=0;
        Jumppoint_Left[i]=0;
    }
}

}
```

3、起跑线以及障碍检测

```
if(!Image_error_flag)
{
```

```

        if((Jumpoint_Right[0]-Jumpoint_Left[0]<CAMERA_W *0.5))
        {

if(init_L_end_row!=0&&add_R_count==0&&((add_L_count==0&&R_end_row>L_end_row)||
        (add_L_count==1&&Jumpoint_Left[init_L_end_row]>20)))
        {
            path_point_row=init_L_end_row;
            path_point_column=Jumpoint_Middle[init_L_end_row];//+20;
        }

if(init_R_end_row!=0&&add_L_count==0&&((add_R_count==0&&L_end_row>R_end_row)||
        (add_R_count==1&&Jumpoint_Right[init_add_R_row]<CAMERA_W -20)))
        {
            path_point_row=init_R_end_row;
            path_point_column=Jumpoint_Middle[init_R_end_row];//-20;
        }
    }
    else
    {
        if(R_end_row>90&&L_end_row>90)
            row_start=90;
        else
            row_start=R_end_row>L_end_row?(L_end_row):(R_end_row);
        for(row=row_start; row>=0; row--)
        {
            for(column=Jumpoint_Left[row]+3;

```

```

column<(Jumppoint_Left[row]+Jumppoint_Right[row])/2; column=column+20)

{
    if(image_buffer[row][column]==0)

        object_count_1++;

}

}

for(row=row_start; row>=0; row--)

{

    for(column=Jumppoint_Right[row]-3;
column>(Jumppoint_Left[row]+Jumppoint_Right[row])/2; column=column-20)

    {

        if(image_buffer[row][column]==0)

            object_count_2++;

    }

}

if(object_count_1>3)

    object_left=1;

if(object_count_2>3)

    object_right=1;

if(object_left==1&&object_right==1)

    stop=1;

else if(object_left==1)

{

    path_point_row=40;

    path_point_column=(2*Jumppoint_Right[40]+3*Jumppoint_Middle[40])/5;

    obstacle_Left_flag=1;

```

```

    Caculate_real( path_point_row, path_point_column, 2 );
}
else if(object_right==1)
{
    path_point_row=40;
    path_point_column=(3*Jumpoint_Middle[40]+2*Jumpoint_Left[40])/5;
    obstacle_Right_flag=1;
    Caculate_real( path_point_row, path_point_column, 2 );
}

```

4、图像逆透视还原世界坐标

```

    if(row>=duan1)
    {
        realx = 10.0*(a71*column+a72*row+a73)/(c71*column+c72*row+1) ;
        realy = 10.0*(b71*column+b72*row+b73)/(c71*column+c72*row+1) ;
    }
    else if(row>=duan2)
    {
        realx = 10.0*(a61*column+a62*row+a63)/(c61*column+c62*row+1) ;
        realy = 10.0*(b61*column+b62*row+b63)/(c61*column+c62*row+1) ;
    }
    else if(row>=duan3)
    {
        realx = 10.0*(a51*column+a52*row+a53)/(c51*column+c52*row+1) ;
        realy = 10.0*(b51*column+b52*row+b53)/(c51*column+c52*row+1) ;
    }
}

```

```

else if( row>=duan4)

{

    realx = 10.0*(a41*column+a42*row+a43)/(c41*column+c42*row+1) ;

    realy = 10.0*(b41*column+b42*row+b43)/(c41*column+c42*row+1) ;

}

else if( row>=duan5)

{

    realx = 10.0*(a31*column+a32*row+a33)/(c31*column+c32*row+1) ;

    realy = 10.0*(b31*column+b32*row+b33)/(c31*column+c32*row+1) ;

}

else if( row>=duan6)

{

    realx = 10.0*(a21*column+a22*row+a23)/(c21*column+c22*row+1) ;

    realy = 10.0*(b21*column+b22*row+b23)/(c21*column+c22*row+1) ;

}

else

{

    realx = 10.0*(a11*column+a12*row+a13)/(c11*column+c12*row+1) ;

    realy = 10.0*(b11*column+b12*row+b13)/(c11*column+c12*row+1) ;

}

```