

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：河北联合大学

队伍名称：联大飞龙一队

参赛队员： 张学东

 田 明

 陈家程

带队教师：陈 波、刘振东

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有关保留、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

本文设计的智能车系统以 MK60DX256ZVLQ10 微控制器为核心控制单元,通过 CCD 摄像头检测赛道信息,使用外部高速 AD 进行模数信号转换,提取黑色引导线,用于赛道识别;通过光电编码器检测模型车的实时速度,使用 PID 控制算法调节驱动电机的转速和转向舵机的角度,实现了对模型车运动速度和运动方向的闭环控制。为了提高模型车的速度和稳定性,使用 LABVIEW、MATLAB 仿真平台、SD 卡模块、键盘模块等调试工具,进行了大量硬件与软件测试。实验结果表明,该系统设计方案确实可行。

关键字: 智能车, MK60DX256ZVLQ10 微控制器, CCD 摄像头, PID 闭环控制

目录

第一章 引言	1
1.1 概述	1
1.2 论文结构	1
第二章 系统总体设计	3
2.1 智能车系统概述	3
2.2 机械结构总体设计	3
2.3 硬件电路总体设计	4
2.4 控制算法（软件）总体设计	5
2.5 小结	5
第三章 机械系统设计及实现	7
3.1 车体模型改造	7
3.1.1 前轮调整	7
3.1.2 后轮轮距的调整	9
3.1.3 差速器的调整	9
3.2 底盘连接	9
3.3 舵机安装	10
3.4 摄像头安装	10
3.5 小结	11
第四章 硬件电路设计及实现	13
4.1 硬件系统设计总论	13
4.1.1 可靠性	13
4.1.2 高效性	13
4.1.3 简洁性	13
4.2 传感器及图像采集电路	14
4.3 电源系统	17
4.4 电机驱动电路（动力系统）	18
4.5 舵机转向控制电路	19
4.6 速度检测电路	20
4.7 其他接口电路	21
4.8 小结	21
第五章 控制算法设计及实现	23
5.1 总体控制思路及实现方案	23
5.1.1 总体设计	23
5.1.2 算法实现方案	23
5.2 图像采集算法	24
5.3 黑线提取算法	25
5.4 图像滤波算法	26
5.5 控制算法	27
5.5.1 转向控制算法	27
5.5.2 速度控制	28
5.6 小结	28
第六章 智能车系统调试	30

6.1 CodeWarrior IDE 在线调试.....	30
6.2 matlab 仿真和数据分析.....	30
6.3 串口调试工具	31
第七章 总结	32
7.1 智能车主要技术参数	32
7.2 体会	32

第一章 引言

1.1 概述

“飞思卡尔”杯全国大学生智能汽车竞赛是由教育部高等学校自动化专业教学指导分委员会主办全国大学生智能汽车竞赛。该竞赛以“立足培养，重在参与，鼓励探索，追求卓越”为指导思想，旨在促进高等学校素质教育，培养大学生的综合知识运用能力、基本工程实践能力和创新意识。智能车竞赛涉及自动控制、模糊识别、传感技术、电子、电气、计算机、机械与汽车等多个学科，为大学生提供了一个充分展示想象力和创造力的舞台，吸引着越来越多来自不同专业的大学生参与其中，激发了大学生的创新思维，对于其实践、创新能力和团队精神的培养具有十分最重要的价值。

参赛队伍在组委会提供统一车模平台上，根据竞赛规则及功能要求，利用 Freescale 半导体公司的 32 位单片机和数字摄像头为核心部件，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，据此进行行进路线和速度控制。整个系统设计包括车体机械结构设计、硬件电路设计和控制算法（软件）设计。

智能车的制作过程包含着我们的辛勤努力，这份报告凝聚了我们智慧，是我们团队共同努力的成果。在准备比赛的过程中，我们小组成员涉猎控制、模式识别、传感器技术、汽车电子、电气、计算机、机械等多个学科，一年来的经历，培养了我们电路设计、软件编程、系统调试等方面的能力，以系统的角度看待问题，锻炼了我们知识融合、实践动手并付诸创新的能力，对今后的学习工作都有着重大的实际意义。

1.2 论文结构

本技术报告采用先总后分的结构，先对系统总体设计进行介绍，然后依次对车体机械结构、硬件电路和控制算法（软件）三部分进行了详尽的阐述，其中包含了我们在硬件电路和控制算法方面的创新设计和独特想法。本报告的具体内容和组织方式如下所述。

全文分为七章，第一章为引言，简单介绍了智能车竞赛的基本情况、基本组成和总体设计；第二章为系统总体设计，主要介绍了智能车的总体设计思路 and 实现方法；第三章为车体机械结构设计，主要包括编码器和图像传感器的安装、舵机和前轮调节等；第四章为硬件电路设计，主要包括电源管理模块、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；第五章为控制算法（软件）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；第六章为总结，总结了本设计的主要特点以及在设计过程中遇到的问题和解决办法。

第二章 系统总体设计

本系统主要包括三个大部分，分别为车模总体机械结构、硬件电路系统、软件算法，每一个部分又由各个小模块构成，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。同时，为了使系统达到最优，必须从总体的角度出发进行对各部分进行综合设计。

2.1 智能车系统概述

根据智能车系统的基本要求，我们设计了系统结构图，如图 2.1 所示。智能车系统的总体工作模式为：CCD 摄像头传感器拍摄赛道图像，输出 PAL 制式模拟信号；将该模拟信号转换为数字信号，利用 LM1881M 进行视频同步分离，将获得的场同步信号和、行同步信号输入到 MK60DX256ZVLQ10 微控制器中，进行处理获得赛道信息；通过光电编码器检测车速，采用 MK60DX256ZVLQ10 输入捕捉功能进行脉冲计算获得速度和路程；转向舵机采用经典的 PID 算法控制；驱动电机采用 PID 控制，通过 PWM 控制驱动电路调整电机的功率；通过即插型的接口用来整车的调试工作；而车速的目标值由默认值、运行安全方案和基于图像处理的优化策略进行综合控制。

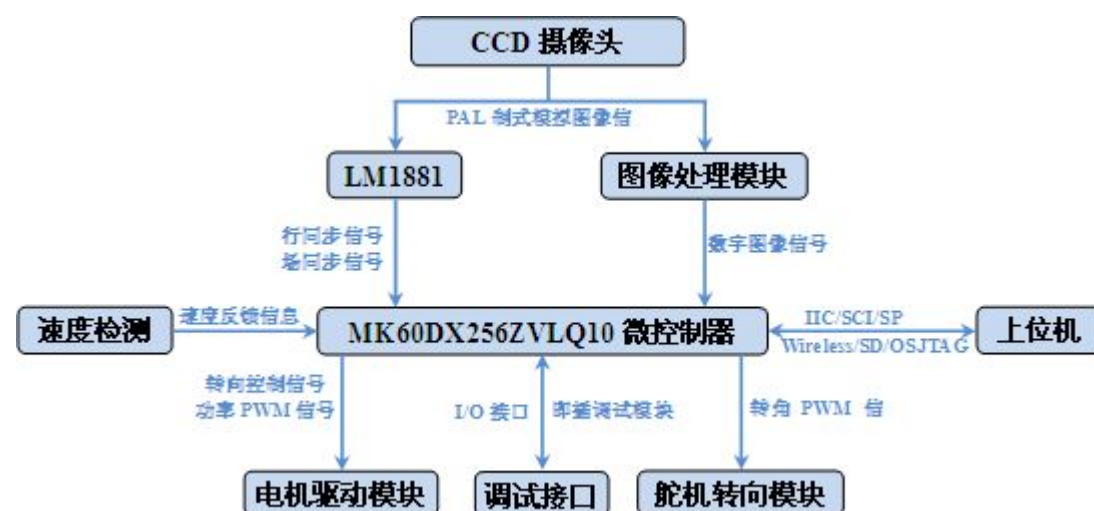


图 2.1 系统结构图

2.2 机械结构总体设计

通过对组委会规定的 A 车模进行机械建模分析，本设计本着轻量化设计，整车布局如图 2.2 所示，具有以下特点。

- (1) 架高舵机并直立安装，以提高舵机响应速度；
- (2) 主板采用 SMT 技术以提高主板的集成度，减小主板的空间体积；
- (3) 主板和电池低位放置，以降低整车的重心；

- (4) 采用强度高、质量轻的材料作为摄像头的支架，以保证摄像头的稳定；
- (5) 摄像头后置于模型车的后方，减少赛车前方盲区。
- (6) 改进原有的避震系统，使车在运行过程中稳定性更好。

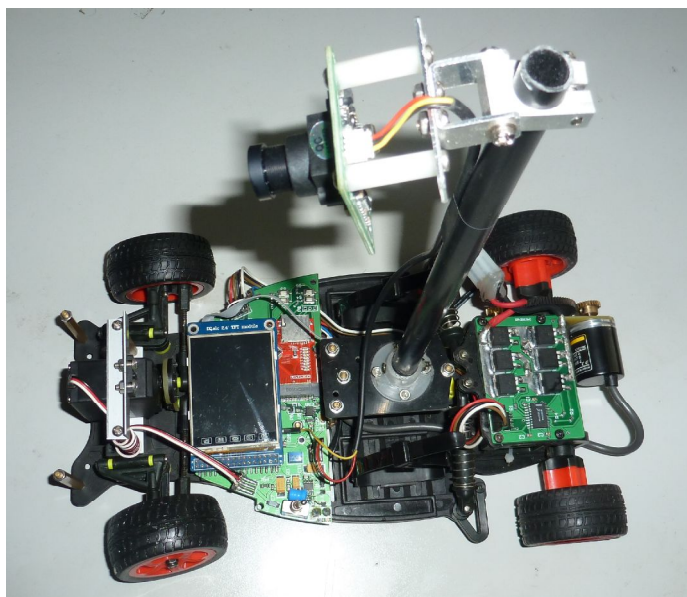


图 2.2 整车布局

2.3 硬件电路总体设计

在满足比赛要求的情况下，本设计力求系统简单高效，因而在设计过程中尽量简化硬件结构，避免了因硬件而出现的问题。赛车的硬件电路主要有七个部分组成：MC9S12XS80 最小系统板，摄像头图像采集与处理电路，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) MC9S12XS80 最小系统板是系统的核心部分，通过对赛道图像数据和赛车速度等信息进行处理，产生舵机与驱动电机的控制量。

(2) 摄像头图像采集与处理电路由高速 AD 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) 速度检测电路用于实时测量小车的运行速度，由旋转编码器、比较器以及 S12 的 ECT 脉冲捕捉功能构成。

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

(5) 舵机驱动模块用于控制舵机的转向进而控制小车的运行方向。

(6) 电源管理模块给整个系统供电。

(7) 辅助调试模块有串行通信等，主要用于赛车系统的程序烧写，赛车状态监控等方面。

2.4 控制算法（软件）总体设计

控制算法是本赛车系统的核心部分，最能体现赛车的智能化水平。本设计的控制算法采用模块化实现方法，力求简洁高效。

控制算法主要包括赛道信息和偏差提取模块、舵机的 PID 控制模块、电机的转速控制算法。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 S12 利用边缘检测方法从图像数据中提取赛道黑线，求得赛车于黑线位置的偏差，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。

2.5 小结

本章首先介绍了赛车整体设计，然后对机械、硬件、软件的总体设计思路进行了概述。只有机械、硬件和软件做到有效融合，赛车才能够跑出好成绩。

第三章 机械系统设计及实现

为了提高智能车的竞技性能，必须了解其机械结构，并且对其结构上的不足之处加以改进。想要取得好的成绩，智能车底盘的优化和硬件设备的可靠性所占的比重不低于软件程序及其控制策略所占的比重。在比赛备战之初，我们就对该车模进行了详细的系统分析。今年的车模精度不是很高，因此在规则允许范围内尽量改造车模，提高车模整体精度是很必要的。本章将主要介绍智能汽车车模的机械结构及调整方案。

3.1 车体模型改造

第七届智能车竞赛的车模由来自东莞市博思电子数码科技有限公司生产的 A 型车模作为摄像头组比赛用车模，车模外形如图 3.1 所示。通过对 A 型车模进行系统的建模分析，在比赛规则的允许范围内，制定了整车机械系统的设计安装和改进方案，使其能够符合智能车的竞技要求。



图 3.1 车模外形

3.1.1 前轮调整

前轮是转向轮，决定了整车的转向性能，是整车机械系统设计的关键所在。它的安装位置由车高和重心、前轮的内束角和外张角、主销后倾、主销内倾、前轮前束等参数决定。

(1) 车高和重心

车高指的是当车子满载的时候(也就是竞速状态的时候)底盘离地面的高度，一般分前后两个高度测量。在保证顺利通过坡道的前提下，底盘尽量降低，从

整体上降低模型车的重心，可使模型车转弯时更加稳定、高速。

通过实际的测试最终在前轮两侧各安装两个 2mm 的调节块，后轮使用了最低的调节块安装方式，将整车前后底盘高度控制在 5~6mm。

（2）前轮的内束角和外张角

前轮的内束角和外张角是描述从车的正上方看，车轮的前端和车辆纵线的夹角。少量的前轮内束角可以使车在加速时保持稳定。但这样做也会减少车子在进入弯角时的转向反应，同时增加加速出弯时的转向反应。如果需要在进入弯角时获得更多的转向，可以使用外张角。但这样做会使车子在加速时、或者通过起伏路面时，变得不稳定，偏离直线。越大的内束角或外张角，越会减低车辆在直线行走的速度。通过实际的调试最终将智能车的前轮定位内束角。

（3）主销后倾

主销后倾是指主销装在前轴，上端略向后倾斜的角度。这与摩托车的前轮向后倾的道理一样，它使车辆转弯时产生的离心力所形成的力矩方向与车轮偏转方向相反，迫使车轮偏转后自动恢复到原来的中间位置上。由此，主销后倾角越大，车速越高，前轮稳定性也愈好。但后倾角大也会使转向费力，但对于“飞思卡尔”智能车竞赛用车模的机械结构而言，主销后倾角也有一个范围，一般不大于 3°。在“飞思卡尔”车模上，主销后倾角由前轮悬挂的垫片来调节。

前轮内束角和主销后倾都有使汽车转向自动回正，保持直线行驶的功能。不同之处是前轮内束角的回正与车速无关，主销后倾的回正与车速有关，因此高速时后倾的回正作用大，低速时内倾的回正作用大。“飞思卡尔”车模的速度较快，所以在进行前轮内束角和主销后倾的调整时要注意找到不同速度下的最佳的前轮内束角和主销后倾的配合关系，建立相应的车模机械模型。

（4）主销内倾

主销内倾是指主销装在前轴略向内倾斜的角度，它的作用是使前轮自动回正。角度越大前轮自动回正的作用就越强烈，但转向时也越费力，轮胎磨损增大；反之，角度越小前轮自动回正的作用就越弱，因此这个都有一个范围，约 5°~8°之间。

（5）前轮前束

所谓前轮前束是指两轮之间的后距离数值与前距离数值之差，也指前轮中心线与纵向中心线的夹角。前轮前束的作用是保证汽车的行驶性能，减少轮胎的磨损。前轮在滚动时，其惯性力会自然将轮胎向内偏斜，如果前束适当，轮胎滚动时的偏斜方向就会抵消，轮胎内外侧磨损的现象会减少。

在“飞思卡尔”车模上，前轮前束由拉杆的螺纹来调节。在实际调试中，我们发现前轮前束的值为负值，可以提高舵机的动作灵敏性，使车转向更灵活，

减小因高速而引起的转向不足的情况。

3.1.2 后轮轮距的调整

轮距是左右轮胎外侧最低点之间的距离。后轮轮距的宽度可以影响车子的稳定性，转向特性，重力转移特性等。智能车的 A 型车模只能调整后轮的轮距。

“飞思卡尔”智能车模型的轮距调整可通过更换圈中红色垫圈来调整后轮轮距。在实际调试中，我们发现在组委会允许的范围内适当增大轮距的确可以增大转弯时后轮的稳定性，从而减少车模发生侧滑的情况，使车转向更灵活，减小因高速而引起的侧滑危险，提高了车模运行的稳定性。

3.1.3 差速器的调整

汽车在拐弯时，左右两轮的转速不同，差速器的作用就是利用“能量最低原理”，自动分配两轮的转速，使汽车平稳过弯。智能车的差速器如图 3.2。

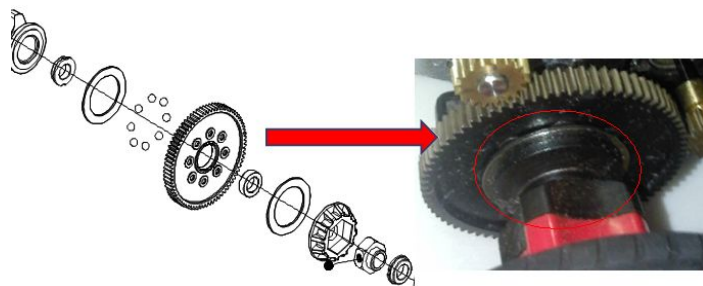


图 3.2 智能车的差速器实物图

“飞思卡尔”智能车使用的是滚珠差速器，虽然差速效果不及行星轮差速器，但如果调整得好的话，可以满足车模的需要。车模的差速器的松紧程度可以通过后轮的紧固螺母来调节。在差速器的调整的过程中要注意，差速器越紧后轮的驱动能力越强，但差速效果变差；差速器越松，差速效果越好，但后轮的驱动能力变差，所以在调整时要兼顾两方面的要求综合考虑。

3.2 底盘连接

在准备华北赛时我们将原来的避震系统去掉，采用了自制的 PCB 基板进行车体两部分的连接，如图 3.3 所示。但是发现这样的连接存在很多的问题，车体的前后连接之后四轮承重不一致，在调试的时候发现车体震动严重，为改善这些缺点，在保持原有的机构的情况下，采用减震弹簧改进了悬挂系统，如图 3.4 所示，这样的悬挂系统很灵活，减震弹簧可以拆卸，这样调节整车的重心变得比较容易。

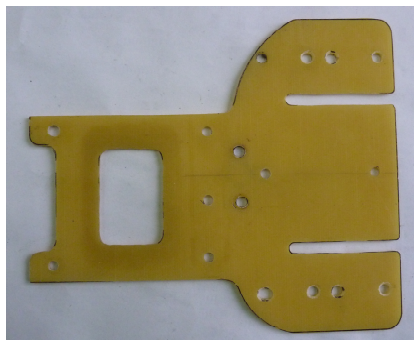


图 3.3 PCB 基板进行车体两部分的连接



图 3.4 采用减震弹簧改进的悬挂系统

改变减震弹簧的预紧力可以改变底盘的刚度，从而改变行驶的效果。“飞思卡尔”车模的弹簧预紧力可以通过垫圈的个数来调节。

3.3 舵机安装

原装车模的舵机为卧式安装，考虑到主板的安装方便以及车模转向性能，我们对舵机安装结构进行了较大的调整。比赛车模的转向是通过舵机带动左右横拉杆实现。舵机的转动速度和功率是一定，要想加快转向机构的响应速度，唯一的办法就是优化舵机的安装位置及其力矩延长杆的长度。由于功率是速度与力矩乘积的函数，过分追求速度，必然要损失力矩，力矩太小也会造成转向迟钝，因此设计时就要综合考虑转向机构响应速度与舵机力矩之间的关系，通过优化得到一个最佳的转向效果。我们可以利用实际参数进行计算，得出一套可以稳定高效工作的参数及结构。

最终，我们设计了一套舵机支架对其固定以及舵机连片(转向拉杆)，综合考虑了速度与力矩的关系，并根据模型车底盘的具体结构，简化了安装方式，最终安装效果如图 3.5 所示。

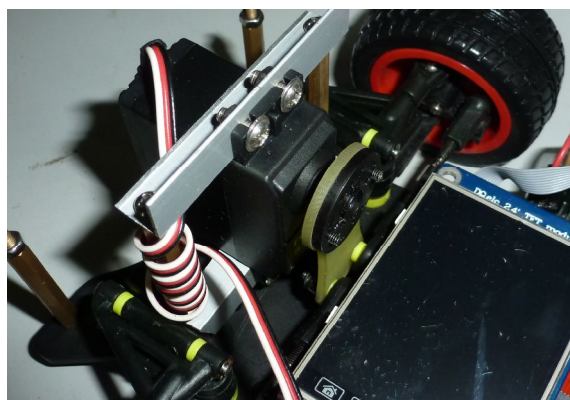


图 3.5 舵机安装方式

3.4 摄像头安装

为了获得较好的图像信息我们选择了 Sony 公司的黑白 CCD 摄像头，但是这

样的摄像头重量较大为兼顾整车重心，需要严格控制 CCD 摄像头的安装位置和重量，我们选择了适当的轻巧的铝合金夹持组件并采用了碳纤维管作为安装 CCD 的主桅，这样可以获得较大的刚度质量比，整套装置具有较高的定位精度和刚度，使摄像头便于拆卸和维修，具有赛场快速保障能力。摄像头的安装如图 3.6 所示。

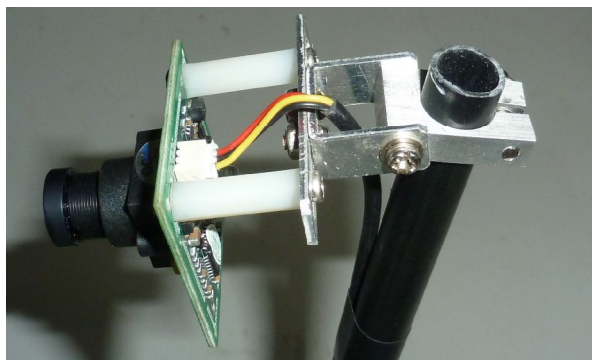


图 3.6 摄像头安装

3.5 小结

本章通过对 A 型车模进行系统的建模分析，在比赛规则的允许范围内，制定了整车机械系统的设计安装和改进方案，使其能够符合智能车的竞技要求，主要包括模型车机械部分安装及改造、传感器的设计安装、系统电路板的固定及连接等。

第四章 硬件电路设计及实现

4.1 硬件系统设计总论

可靠、高效、简洁是我们对硬件电路的设计方案既定的系统设计目标。

4.1.1 可靠性

可靠性是系统设计的第一要求，我们对电路设计的所有环节都进行了如下的改进和设计：

- (1) 电源系统的稳压性能、纹波噪声的去处、功率的充足提供的可靠性等；
- (2) 主控系统的电路优化，减少电路元件的体积和用量，减少不稳定因素；
- (3) 动力系统的动力性能可靠性，如加速、制动等性能会有大幅度的改善，为整车的性能提升提供充足的保证；
- (4) 做好各部分的接地、屏蔽、滤波等工作，将高速数字电路与模拟电路分开，使本系统工作的可靠性达到了设计要求；
- (5) 传感器的性能稳定，保证整车数据流的准确可靠；

4.1.2 高效性

高效性是指本系统的性能要足够强劲，主要从以下五个方面实现的：

- (1) 整车的机械构件的创新设计，提高整车机械传动的性能，改善机械机构的滞后性，提高整车机械结构的响应速度；
- (2) 电源系统的稳定性，提高电源的利用率；
- (3) 动力系统电路的创新设计，减少动力系统控制元件的内阻，提高动力系统的驱动能力、响应速度、电源的利用率，期望可以大大提高电动机的工作转矩和转速；
- (4) 硬件的运算电路的论证方案中，增加了图像的动态阈值可调电路；

4.1.3 简洁性

简洁是指在满足了可靠、高效的要求后，为了尽量减轻整车重量，降低模型车的重心位置，应使电路设计尽量简洁，尽量减少元器件使用数量，缩小电路板面积，使电路部分重量轻，易于安装。我们在对智能车系统进行分析后可以通过以下几点来进行对系统简洁处理：

(1) 智能车机械结构的简化，这需要我们设计出更加精巧、灵活的机械构件，提高智能车的精度，提高智能车的制作工艺，简化智能车的机械复杂程度。

(2) 在对智能车硬件电路系统进行了详细分析后，通过认真筛选所用器件、简化电路、合理设计元件排列、电路走线，最终的硬件电路板采用贴片封装形

式，使本系统硬件电路部分轻量化和机械强度部分的指标达到设计要求。

4.2 传感器及图像采集电路

(1) 传感器

固态图像传感器主要有 5 种类型：电荷耦合器件 CCD、电荷注入器件 CID、互补性氧化金属半导体 CMOS、电荷引发器件 CPD 和叠层型摄像器件。在智能车系统的传感器模块设计中，通常采用的图像传感器中主要有 CMOS 和 CCD 两种。我们在智能车传感器模块的选取这一块基本有 4 种备选方案，线阵型的 CCD、面阵型 CCD、CMOS 型传感器、数字型摄像头。线阵 CCD 可以直接接收一维光信息，而不能直接将二维图像转换为一维的电信号输出，为了得到整个二维图像的输出生，就必须用行扫描的方法来实现。面阵 CCD 传感器的感光单元呈二维矩阵排列，能检测二维平面图像。由于传输与读出方式不同，面阵图像传感器有许多类型，常见的传输方式有行传输、帧传输和行间传输三种。

综合考虑各种图像传感器的光电转换特性、灵敏度、灵敏度不均匀性、分辨率、各种噪声等各种因素，我们对 4 种传感器硬件资料进行了详细分析和测评，以数据和实际效果为依据，最终选定了 Sony 公司生产的黑白色 CCD 摄像头，具体参数如下：本着科学性的原则，我们决定对 4 种传感器硬件资料进行详细地分析之后选取代表性的产品进行实际的测评，以数据和实际效果为依据，确定切合整车系统的传感器类型。最终我们选定了 Sony 公司生产的黑白色 CCD 摄像头，具体参数如下：

图像传感器: 1/3" SONY CCD (SONY 405AL+D2463+1310 芯片方案)

有效像素: PAL:500(H)×582(V)

感光面积: 4.8mm×3.6mm

信号系统: PAL (模拟信号)

水平清晰度: 480 电视线

最低照度: 0.05Lux/F1.2

增益控制: 自动

背光补偿: 自动

电子快门: 自动

快门速度: 1/50 (1/60) — 1/100,000 (秒)

白平衡: 自动跟踪白平衡

信噪比: 大于 48db

视频输出: 1.0Vp-p, 75 Ω

工作电压: DC12V±10%

工作电流：100mA

尺寸（MM）：38×38/32×32

摄像头以隔行扫描的方式采样图像，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对应的电压值，然后将此电压值通过视频信号端输出。如图 4.1 所示，摄像头连续地扫描图像上的一行，就输出一段连续的视频信号，该电压信号的高低起伏正反映了该行图像的灰度变化情况。当扫描完一行，视频信号端就输出一个低于最低视频信号电压的电平（如 0.3 V），并保持一段时间。这样相当于紧接着每行图像对应的电压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。然后扫描新的一行，如此下去，直到扫描完该场的信号，接着会出现一段场消隐信号。其中有若干个复合消隐脉冲（简称消隐脉冲），在这些消隐脉冲中，有一个消隐脉冲远宽于其他的消隐脉冲（即该消隐脉冲的持续时间远长于其他的消隐脉冲的持续时间），该消隐脉冲又称为场同步脉冲，标志着新的一场的到来。摄像头每秒扫描 25 帧图像，每帧又分奇、偶两场，故每秒扫描 50 场图像。

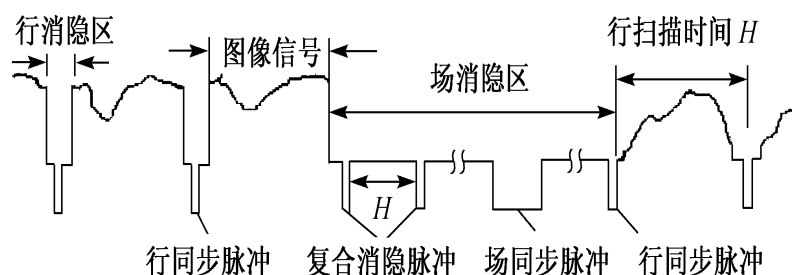


图 4.1 摄像头工作方式

由于智能车对摄像头的采集速率有一定的时序要求，所以在进行数据提取时要有一定的算法以弥补硬件滞后性的不足，需要有创新性的软硬件设计来应对摄像头的数据特殊性。

（2）图像采集电路

路径识别单元是智能车控制系统的输入采集单元，其优劣直接影响智能车的快速性和稳定性。在摄像头方案中，其前瞻距离及检测到的赛道信息是红外线光电管方案远不能比拟的，但其软、硬件设计也较红外线光电管方案难。

要能有效地采样摄像头视频信号，首先要处理好的技术问题就是能提取出摄像头信号中的行同步脉冲、消隐脉冲和场同步脉冲。否则，单片机将无法识别所接收到的视频信号处在哪一场，也无法识别是在该场中的场消隐区还是视频信号区，更无法识别是在视频信号区的第几行。要处理好行同步脉冲和场同步脉冲提取的问题，有以下两种可供参考的方法。方法一：直接采用 A/D 转换进行提取。当摄像头信号为行同步脉冲、消隐脉冲或场同步脉冲时，摄像头信号电平就会低于这些脉冲模式之外时的摄像头信号电平。据此，可设一个信号

电平阈值来判断 A/D 转换采样到的摄像头信号是否为行同步脉冲、消隐脉冲或场同步脉冲。方法二：就是给单片机配以合适的外围芯片，此芯片要能够自己提取出摄像头信号的行同步脉冲、消隐脉冲和场同步脉冲，以供单片机控制之用。

采用第一种方法，无需配以外围芯片在硬件上就可以较为简便。然而，此方法在智能车控制系统的设计中存在两大局限性。首先，S12 的 A/D 转换时间还不够短，在不超频的情况下，该单片机的 A/D 转换时间最短为 7 μ s，而行同步脉冲一般只有 4.7 μ s 左右的持续时间，大多数消隐脉冲更只有 3.5 μ s 的持续时间，持续时间都小于 7 μ s，所以 A/D 转换很有可能漏检行同步脉冲或消隐脉冲。一旦漏检一两个脉冲，就会使摄像头视频采样的效果大打折扣。其二，在智能车控制系统中，S12 除负责摄像头视频采样方面的处理之外，还要负责黑色引导线的提取、方向速度控制等方面的处理，但毕竟 MC9S12DG128 的处理能力还是有限的，若采用此方法，会使得在视频采样上花费较多的 S12 处理资源，这样摄像头视频采样本身的效率较低。此外，黑色引导线的提取、方向速度控制等方面的设计也会局限于所剩余的单片机处理资源。

采用第二种方法，则需要配以专门的外围芯片。虽然硬件上相对要繁琐一些，但在资源的合理配置上则大大提高。目前，LM1881 视频同步信号分离芯片就是一款合适的芯片，它提取摄像头信号的行同步脉冲、消隐脉冲和场同步脉冲，并将它们转换成数字式电平直接输给单片机的 I/O 口作为控制信号，其硬件连接如图 4.2 所示。摄像头视频信号端接 LM1881 的视频信号输入端，同时也接入 S12 的一个 A/D 转换器口（选用 PAD1）。LM1881 的行同步信号端（引脚 1）接入 S12 的一个外部中断 IRQ 口。LM1881 的奇-偶场同步信号输出端接 S12 的普通 I/O 口即可（选用 PORTM0）。

本设计中采用使用第二种方案，有以下优点：

- a. 减少单片机的资源占用，这样单片机就可以利用有限的资源进行复杂的数据处理和整车控制，从而整车的精确控制提供保证。
- b. 数据提取更加简便，通过增加硬件资源可以尽可能减少软件的设计复杂程度，更加关注于软件算法和控制策略的设计，加快开发周期。

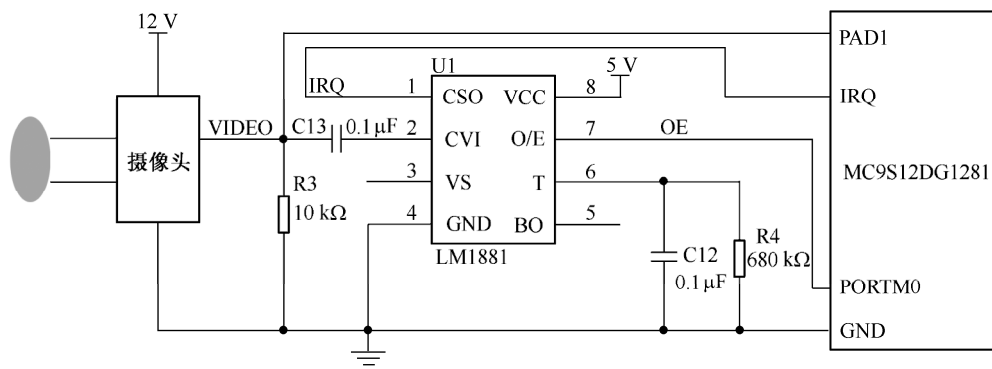


图 4.2 图像采集电路

4.3 电源系统

电源系统的设计宗旨：稳定，高效，简洁，驱动负载能力强。电源设计方案依据模拟电子技术、电力电子技术关于直流稳压电源的设计进行理论论证。电源系统是智能车的动力来源，电源的稳定与否直接关心智能车运行的稳定性，因此要求电源有非常好的稳压性能，极小的纹波噪声。

（1）电气参数

智能车的瞬间启动、频繁加速减速，动力电机驱动的瞬间浪涌电流会很大，会使车载电源的变化很大，由于大赛规定的整车总电容量有限制，所以需要整车的电源系统所采用的稳压芯片要有很好的稳压性能，此模块的电容、电阻、电感集体参数要求较为精确，需要加强理论的可靠指导；考虑到功率的要求，在绘制电路板时走线及线宽设置有一定的技术要求，需要时刻注意；直流电机在转动时会产生很大的纹波噪声，为保证主控芯片的稳定运行，需要电源系统有很好的降噪能力；车模的整车功耗很大，要求电源系统要有很好的带负载能力和较高的效率。

（2）测试环节

在电源模块的测试时需要有一个比较科学完整的数据记录表格，测试的情况要考虑全面，完整记录在不同情况下的数据以便进行数据比较，在进行详细的比较之后最终确定电源系统的芯片型号和具体的供电系统细节设计。对每种电源方案要有详细的数据测试手册，形成书面格式的技术报告，有真正的理论指导，在测试的过程中遇到问题可以有针对性的解决问题，减少因盲目测试造成时间上的浪费。

（3）主控系统

第七届智能车竞赛组委会要求采用美国飞思卡尔公司生产的 Freescale 16 位 HCS12 系列单片机也称 MC9S12 系列，简称 S12 系列芯片作为智能车系统的主控芯片，MC9S12X 系列是 HCS12 系列的增强型产品，基于 S12 CPU 内核，可达到 25MHz 的 HCS12 的 2~5 倍的性能。S12X 系列增加了 172 条额外指令，

可以执行 32 位计算(共 280 条指令)，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。S12X 系列的 CPU 以复杂指令集 CISC 架构，集成了中断控制器，有丰富的寻址方式。中断有 7 个优先级并且内核支持优先级的调度，最多可有 117 个中断源，S12X 可访问最多 8M 的全部存储空间(包括片内和片外资源)。而我们采用的是龙丘科技生产的 112 脚的最小系统板作为整车的控制器。我们可以省去对最小系统电路的设计，只须提供稳定的工作电源和相关的接口电路即可，大大提高了系统的稳定性和电路的复杂程度。

4.4 电机驱动电路（动力系统）

第七届智能车竞赛规则规定摄像头组别使用东莞市博思电子数码科技有限公司生产的 A 型车模，其中随车模的驱动电机为：RS380-ST/3545，实物如图 4.3 所示，基本参数如下：

- 直径：Φ27.5 mm，
- 机身长度：32.5 mm，
- 输出轴直径：Φ2.3 mm，
- 安装螺孔：2-M2.5 mm。
- 工作电压：DC 1.2~36 V



图 4.3 车模的驱动电机

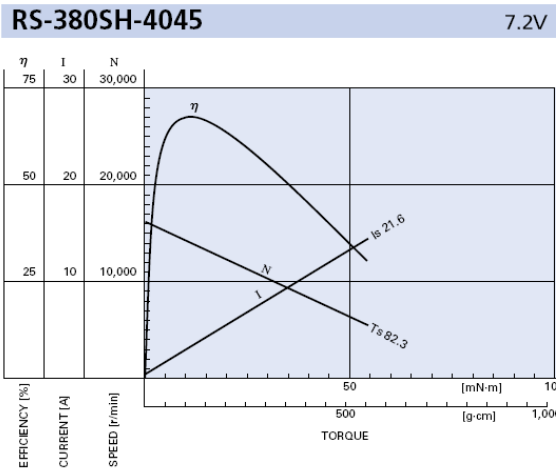


图 4.4 驱动电机特性

驱动电路是摄像头组别硬件系统中关键的环节之一，是速度的重要保障，驱动电路又分为集成芯片系列和分立元件系列，其特色各不相同。依据电机特

性（图 4.4），设计驱动电路时需要注意以下几个方面：

（1）马达在瞬间启动时产生的很大浪涌电流，这一点在驱动设计时很重要；摄像头组别竞速水平较其他组别的速度快，在车模的行驶中对驱动性能要求更为严格，包括制动、加速、牵引方面需要有突出的性能。

（2）驱动系统的耗电较大，效率问题也是考虑因素之一；马达的散热问题，功率管的散热问题等，在选择器件方面要谨慎，在电路板的绘制时也要十分的注意结构问题。

（3）在达到基本要求的以后，采用将 PWM 信号的幅值进行升压处理后在加到 MOS 管上，可以减小 MOS 管的导通电阻，从而改善 H 桥的驱动性能。同时需要考虑 MOS 管的延时、功耗、响应速度等问题。

综合考虑以上因素，设计出的驱动电路如图 4.5 所示。



图 4.5 电机驱动电路

4.5 舵机转向控制电路

在智能车上，舵机的输出转角通过连杆传动控制前轮转向。舵机本身是一个位置随动系统是系统中一个具有较大时间常数的惯性环节。其时间延迟正比于转过的角度，反比于舵机的响应速度。对于快速性要求极高的智能小车来说，舵机的响应速度是影响其过弯最高速度的一个重要因素，特别是对于前瞻不够远的智能小车更是如此。

Futaba S3010 舵机是由舵盘、减速齿轮组、位置反馈电位计、直流电机和控制电机电路组成，实物如图 4.6 所示。通过内部的位置反馈，使它的舵盘输出转角正比于给定的控制信号，因此对于它的控制可以采用开环控制方式。在负载力矩小于其最大输出力矩的情况下，它的输出转角正比于给定的脉冲宽度。但实际上，由于舵机反应的延迟性，智能车的舵机转角通常不能在一个控制周期内达到指定的设定角度，因此，可以在舵机外部在安装一个位置反馈装置，构成闭环系统，以实时监测和控制舵机的转动角度。



图 4.6 Futaba S3010 舵机实物图

依据舵机的物理性能特点和大赛组委会的相关规定，舵机的型号必须为 **Futaba S3010**，基于此供电电路方案如下所述。舵机的响应速度与工作电压有关系，电压越大速度越快，所以应在舵机允许的工作范围内，尽量选择最大的工作电压，可以提高舵机的响应速度，鉴于大赛组委会规定不允许通过升压电路对舵机提供工作电源。实际使用过程中，蓄电池 **7.2V** 电压虽然略高于舵机允许电压范围，可以提高舵机的响应速度，此种方案值得注意的是要密切注视舵机的状态，舵机工作在额定电压之外，蓄电池的最高电压可以达到 **8.5V**，舵机很容易烧毁。经过一个二极管降至 **6.5 V** 左右后供给转向伺服电机，此种电路简洁，但在选择二极管时要注意二极管的最大的耐流值，确保二极管不被击穿。对转角的控制设计，通过对智能车的机械结构进行计算并和实际值对比，在大体上逼近准确值的基础上来设计对不同赛道类型的参数确定，实现控制的最佳效果。舵机的控制信号线与 **HCS12** 单片机的 **PWM1** 口相连，为提高舵机的精度，加大 **PWM** 信号控制范围，软件上将 **2** 个 **8** 位 **PWM** 信号寄存器合并作为一个 **16** 位的寄存器进行输出。同时要保证舵机在整车的重心中心线上，保证整个小车行驶时更加稳定。

4.6 速度检测电路

智能车的速度达到一定值时就需要对速度部分进行相应的检测构成更加精确地闭环控制系统，使智能车的运行更加稳定。测速模块有两部分组成，即车载部分和外部激光对管组成，外部激光对管主要是用来对车模的速度进行测量。车载部分的设备主要是光电编码器。

光电编码器码盘图案不均匀，码盘与码道位数相等，在相应位置可输出相应的数字码。其优点是坐标固定，与测量以前的状态无关，抗干扰能力强，光电编码器的精度很高，无累积误差，基本不会受到外界光线的干扰，具有断电位置保持，不读数时移动速度可超越极限响应速度，不需要方向判别和可逆计数，信号并行传送等；其缺点是结构复杂、价格较高，外部电路比较复杂，而且会消耗动力系统的输出转矩。要想提高光电编码器的分辨率，需要提高码道数目或者使用减速齿轮机构组成双码盘机构，将任意位置取做零位时需要进行

一定的运算。

4.7 其他接口电路

接口电路就是要把传感器后续的有关电路联系起来的电路。接口电路的设计需要考虑两个问题：一个是传感器的输出与计算机的匹配问题；另一个是选择器件的问题。

一般传感器的输出信号有三种形式：数字开关量、数字脉冲和模拟信号。数字开关量信号分为电压输出型和触点型。如果传感器的输出电压信号为 0V 或 2V，可以直接和控制装置相连，但是由于这种信号有抖动，所以在于计算机连接时要采用消除抖动电路。数字脉冲电路用计数器计数后送入计算机，通过脉冲的计数来达到对信息的采集一般计算机要对计数器拥有清零的功能，以便计数器从新计数。模拟信号比较复杂，由于模拟信号不能直接与控制器相连接，所以首先要通过适当的 A/D 转换器变为合适的输入电压，必要时还要在 A/D 前面对信号进行放大、分压等，一般称为信号的预处理。处理后的信号经过 A/D 转换器后于控制器直接相连，完成接口电路的设计。器件的选择，要兼顾成本和技术指标进行综合考虑。传感器信号要有比较好的降噪电路，一般减少噪声的方法是：传感器与预处理放大电路的接线要尽量短；传感器信号接线要采用屏蔽线，外皮接地；放大电路的输入和输出之间尽量远；放大电路的增益不要太大，以免产生自激振荡；放大电路要远传感器，以免产生电场和磁场的干扰等。

整车的测试系统全部接口化，即除一些必要的检测设备和控制开关车载之外，其他的一些检测修改参数的设备进行接口的连接，比如外接的数码管，液晶显示屏，外接的键盘，可以更加深入的了解智能车的系统特性，同时可以加快整车的调试工作，能迅速的适应多变的比赛环境，还有更少的车载设备可以减少整车的重量和行驶阻力，时整车的动力性能更加的灵活，是提速的一个开发方面。

4.8 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 CCD 型数字摄像头搭建系统，系统硬件电路简单，工作稳定。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。

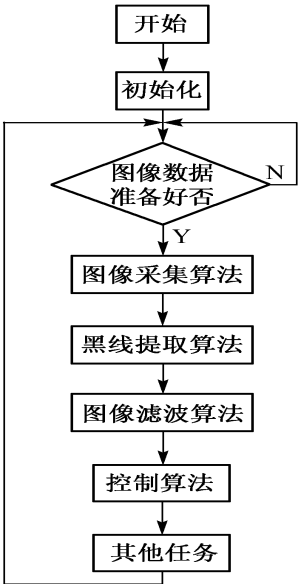
第五章 控制算法设计及实现

高效的软件程序是智能车高速平稳自动寻线的基础。我们设计的智能车系统采用摄像头进行赛道识别，图像采集处理就成了整个软件的核心内容。在智能车的转向和速度控制方面，我们预想使用鲁棒性很好的经典 PID 控制算法结合其他控制算法，配合使用理论计算和实际参数补偿的办法，期望智能车能够稳定快速寻线。

5.1 总体控制思路及实现方案

5.1.1 总体设计

对于智能车系统来说，软件的核心是算法，软件部分是智能车系统的灵魂，真正体现各参赛队智能车的优势和最后成绩好坏的往往是软件部分，尤其是核心算法的设计。本设计在小车转向和速度控制方面，智能车采用的经典的 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：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，运用增量式 PID 算法调节驱动电机转速，实现了电机的快速响应。整体流程如图 5.1 所示，通过外部中断采集程序对摄像头的视频信号进行采集，主程序在两次外部中断的间隙中完成对数据进行处理及计算并给出控制量，其中，主程序主要完成的任务是：单片机初始化和黑线提取算法；图像滤波算法；舵机控制算法及驱动电机控制算法。



5.1 控制算法总体设计

5.1.2 算法实现方案

编程语言有 C 和汇编，而针对本组的特点，已经对 C 语言的编程思想有些

了解，还未接触过汇编。考虑到 C 语言可以使程序员尽量少的对硬件进行操作，具有很强的功能性、结构性、可移植性强、易书写的特点，本组准备在开始的算法开发设计阶段用 C 语言进行算法的开发，此过程中贯穿汇编语言的学习，在算法开发完毕且调试成功后，再用汇编或者 C 和汇编进行混编的程序。因为汇编语言具有直接对硬件进行操作。执行代码的效率高等特点，可以做到 C 语言所不能做到的一些事情。摄像头组别智能车的主控芯片 HCS12 的 RAM 区内存较小，数据处理量大，所以时序要求较高，对提高数据的处理能力有很大的帮助。而且通过对汇编语言的应用可以更加深入的了解单片机底层结构，充分利用单片机的内部资源，同时能提高编程水平，在应用单片机的能力上提高一个新的层次。

5.2 图像采集算法

在完成图像采集时，有两种设计方案。

方案一：考虑到实际赛道只是在白色 KT 板上布置黑色引导线，路径识别只需大致提取出黑色引导线即可，不必每行采集。因此，我们采用隔行采集思想来压缩图像的数据。通过相关的资料显示，智能车控制系统的图像传感系统在单一方向上只要有 40 像素的分辨能力就足够用了。故我们只需对这 288 行视频信号中的某些行进行采样就可以了。假设每场采样 40 行图像数据，为了方便软件程序的编写，可以均匀地采样 288 行视频信号中的 40 行，即每隔 7 个有效行采集一行。具体实现流程图（以采集一幅 16 行×40 列的图像为例）如图 5.2 所示。

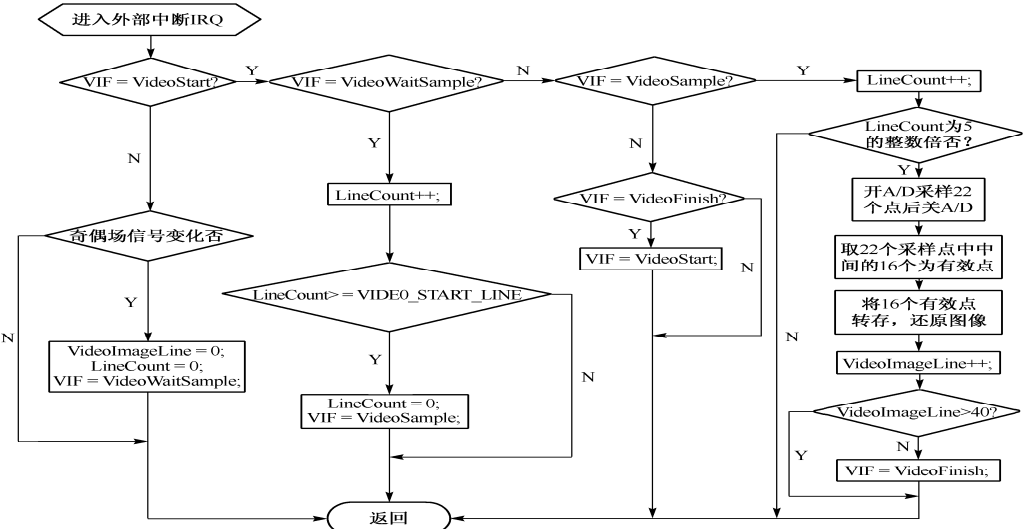


图 5.2 图像采集方案一流程图

在图 5.2 中，数据转存是由于摄像头安装时旋转了 90°后，故在图像采集后应将其还原。这样可以减轻单片机的数据处理量，同时可以减轻程序的编写难度，但这种方案的实用性有待实验验证。因为从第六届开始赛道中增加了虚线，

这种方案很可能会丢失数据或者造成数据的误判。

方案二：在不放弃任何图像数据的情况下，增加硬件二值化的运算电路，直接通过单片机的 I/O 口进行数据的快速提取和处理，将数据存放在单片机的数据寄存器中从而将数据进行处理。此种方案的优点在于，图像的分辨率很高，不会丢失虚线情况，不过相应的优点带来了相当大的数据量，如何提取关键信息，除去噪声，需要一套灵活高效的数据处理算法。

在本设计中，采用了第二种设计方案，可以通过利用汇编语言实现此种设计方案，可以通过精心设计的算法利用汇编语言简洁而强大的信息处理能力，在占用很少程序存储空间的情况下，将单片机的资源专注于数据处理和整车控制上。

5.3 黑线提取算法

在单片机采集图像信号后需要对其进行处理以提取主要的赛道信息，同时，由于交叉线、起点线的存在，光线、杂点、赛道连接处以及赛道外杂物的干扰，图像效果会大打折扣。因此，在软件上必须排除干扰因素，对赛道进行有效识别，并提供尽可能多的赛道信息供决策使用。在图像信号处理中我们提取的赛道信息主要包括：赛道中心位置，赛道宽度，赛道曲率，赛道变化幅度，赛道类型判别。由于摄像头自身的特性，图像会产生梯形式变形，这使得摄像头看到的图像信息不真实。为此我们将利用赛道进行测量，创建函数还原出了真实赛道信息。原始图像是一个将模拟图像经模拟电路转换得到的二维数据矩阵，矩阵的每一个元素对应一个像素点，图像的第一行对应最远处。

三种黑线提取算法：二值化算法、直接边缘检测法、跟踪边缘检测算法。

(1) 二值化算法。设定一个阈值 `valve`，对于视频信号矩阵中的每一行，从左至右比较各像素值和阈值的大小，若像素值大于或等于阈值，则判定该像素对应的是白色赛道；反之，则判定对应的是黑色的目标引导线。记下第一次和最后一次出现像素值小于阈值时的像素点的列号，算出两者的平均值，以此作为该行上目标引导线的位置。该算法的思想简单，但是这种提取算法的鲁棒性较差，当拍摄图像中只有目标引导线一条黑线时，尚能准确提取出该目标引导线，但当光强有大幅度变化或图像中出现其他黑色图像的干扰时，该算法提取的位置就有可能与目标引导线的实际位置偏离较大。

(2) 直接边缘检测法。采用逐行搜索的算法，首先找到从白色像素到黑色像素的下降沿和从黑色像素到白色像素的上升沿，然后计算上升沿和下降沿的位置差，如果大于一定的标准值，即认为找到了黑线，并可求平均值算出黑线的中心点。至于上升沿、下降沿的检测，可以通过上上次采样数与这次采样数

的差值的绝对值是否大于一个阈值来判断，如果“是”且差值为负，则为上升沿；如果“是”且差值为正，则为下降沿。这里，阈值可以根据经验设定，基本上介于 30~46 之间（当 A/D 模块的参考电压为 2.5 V 时），也可以采用全局自适应法设定，每次采样后首先都遍历一次图像，得到图像灰度值的平均值，然后用这个平均值乘以一个调试系数即可得到所要的阈值。该算法较二值化方法而言，抗环境光强变化干扰的能力更强，同时还能削弱或消除垂直交叉黑色引导线的干扰。

（3）跟踪边缘检测算法。由于黑色的目标引导线是连续曲线，所以相邻两行的左边缘点比较靠近。跟踪边缘检测正是利用了这一特性，对直接边缘检测进行了简化。其思路是：若已寻找到某行的左边缘，则下一次就在上一个左边缘附近进行搜寻。这种方法的特点是始终跟踪每行左边缘的附近，去寻找下一行的左边缘，所以称为“跟踪”边缘检测算法。该算法的优点：在首行边缘检测正确的前提下，该算法具有较强的抗干扰性，能更有效地消除垂直交叉黑色引导线的干扰，以及引导线外黑色图像的影响，始终跟踪目标引导线。另外，较之前两种算法，跟踪边缘检测算法的时间复杂度更低，因此效率更高。但该算法的问题在于：由于是在连续邻域上跟踪引导线边缘，若第一行左边缘位置的检测位置 and 实际导引线偏差较大，就会产生一连串的错误，甚至造成智能车失稳。

综上所述，从算法的简洁性和实用性综合考虑，直接边缘检测算法相对于其他两种算法是一个较好的选择，因此，在本设计中，我们采用直接边缘检测算法提取黑线。

5.4 图像滤波算法

虽然采用了边缘检测的方法进行黑线提取，但由于赛道对小车有十字交叉线的干扰、光线对小车有反光的影响、小车本身也存在视野狭窄的缺陷，故用黑线提取算法得到的路径信息有可能错误，不能如实地反映赛道情况。为了避免决策失误，必须对图像加以滤波，并且对错误数据适当地进行校正。

在图像滤波算法中，主要应考虑以下几个方面：

首先，根据图像模型去噪，例如，由于赛道的黑色引导线是绝对连续的，故两个中间有黑线的行之间不能有全白行（注意中间二字：如果黑线在边缘，则可能是由于摄像头的视野太窄或智能车身不正导致在过弯道时只能看到部分黑色引导线），这主要是解决光线对摄像头的反光问题；其次，在理想的情况下，根据赛道的黑色引导线的连续性，如果某一行求取的中心线位置与相邻的两行都相差很大，则可以认为该行数值错误，抛弃该行的数据或使用其前后两行数

据的平均值来替代该错误数值用以校正。在调试过程中会出现两段黑线情况，这种图像信息比较复杂，这时可能存在三种路况：赛道比较密集时，检测到多条赛道；大“S”弯道；“十”字交叉道。

处理策略为：保留最底下的一段，这主要是从安全角度考虑的，防止赛道周围环境对小车的干扰。另外，由于智能车上安装的摄像头相对于赛道存在一定的倾斜角度，因此会造成采集到的赛道图像具有一定的梯形失真，即图像中的赛道远端窄、近端宽，因而也会对路径的正确识别产生影响。对于这种失真，可以通过对每行提取的赛道位置添加一个线性修正值来消除，一般通过实验的方法确定线性补偿的系数。

5.5 控制算法

PID 控制是比例、积分、微分控制的简称。在自动控制领域中。PID 控制是历史最久，生命力最强的基本控制方式。PID 控制器的原理是根据系统的被测量实测值与设定值之间的偏差，利用偏差的比例、积分、微分三个环节的不同组合计算对广义被控对象的控制量。基于摄像头采集数据的滞后性，算法开发这一块要进行长期的论证，调试，修改，优化的过程。

5.5.1 转向控制算法

影响赛车速度成绩的一个重要因素就是对弯道和直道的提前识别判断，从而实现安全过弯，快速过直道、S 弯道，以提高比赛成绩。而摄像头方案在这方面有天然的优势：相对于光电管传感器，可以获得较远的路径信息；不仅可以得到单行的黑线信息，还可以同时获得多行的黑线信息。经过图像处理算法后，得到的信息是关于前瞻范围内的黑线的具体位置，它一般是一个二维数组。现在就是要从这个二维数组中提取出智能车前方的路径信息，以便于后面转向和速度的控制。为了使小车运行得快而稳，可以从中提炼出一个黑线距离车身中心轴的偏差量 **Offset**，要求出这个偏差量必须至少考虑三个因素：最远行黑线位置、黑线平均位置、全白行行数。

从理论上讲，相对于偏差量 **Offset**，曲率是智能车更好的一个控制变量。但由于路径检测单元的局限性，很难计算出非常精确的曲率。在此我们设计了一套的曲率算法方案，并采用了分段控制算法，提高了车模精准定位性能。

(1) 分段比例控制。因为小车处于弯道和直道的转向模型不同，若采用统一的比例系数设置，那么该系数过大会导致小车振荡，过小会导致最大控制量偏小，小车转向不足，过弯时易冲出赛道。使用分段比例控制既方便又可以解决以上两种问题。

(2) 前馈补偿控制。由于摄像头采集的赛道图像存在失真的情况，越远地

方的图像失真越厉害，而对于计算 `offset` 偏差量来说，越远的路径信息越有参考价值，故其比重也相对较大。这样，就导致小车转向微调时的效果往往不太如人意。鉴于此，在小车前述的分段比例控制基础上又加入了前馈补偿控制进行微调。

5.5.2 速度控制

速度调整要求对智能车的速度的调整既要快速，又要准确，而且不能频繁波动。故采用 PID 控制算法不失为一种简单而有效的策略。考虑到摄像头方案的控制周期较长，假设按 2.5 m/s 的平均速度计算，则一个控制周期小车大概可以跑过 5 cm ，如果按这种周期用上述 PID 调节速度，则会导致加速、减速均过长的后果，严重地影响小车的快速性和稳定性。为了解决这个问题，可以在 PID 调速控制中加入 BANG-BANG 控制思想。

5.6 小结

本章详细介绍了系统算法及软件实现，包括图像处理算法和控制算法。

第六章 智能车系统调试

完成智能车软、硬件设计和制作后，需要进行系统调试和优化，包括舵机调试、电机调试以及稳定性调试等，最终获得理想的运行状态。通过各种调试工具获得控制器的运行数据，进而通过分析数据判断单片机是否按照设计的算法执行。

6.1 CodeWarrior IDE 在线调试

程序开发及调试在组委会提供的 CodeWarrior IDE 下进行，Code Warrior 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。其中在本设计中主要使用集成开发环境和调试器，即 IDE。CodeWarrior 是最基本的调试工具，可以实现在线程序下载和在线背景调试功能，其开发和调试界面如图 6.1 所示。

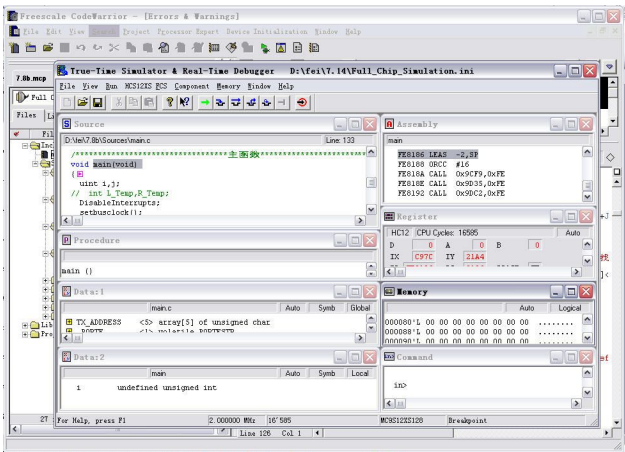


图 6.1 CodeWarrior IDE 调试界面

6.2 matlab 仿真和数据分析

利用 matlab7.8.0 软件对控制算法进行仿真分析，同时结合智能车的实际运行数据，调节 PID 参数，优化舵机等控制策略，如图 6.2 所示。

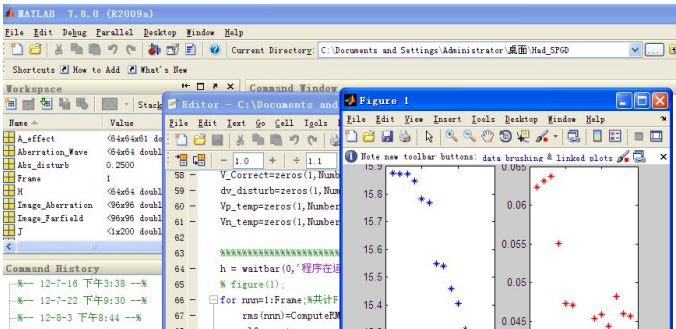


图 6.2 matlab 调试界面

6.3 串口调试工具

通过串口实现单片机与上位机的通信，实时地发送采集到的图像信息，寻线程序的寻线结果，调试过程中可以直观地看出在哪里出了错误，以及图像的阈值，中心值为调试提供了极大的便利。

第七章 总结

7.1 智能车主要技术参数

最终设计并完成的智能车主要参数如下表所示。

车模几何尺寸（长/宽/高）	285mm/174mm/280mm
车模重量（带有电池）	1.145Kg
电路功耗	7.2V-3A
电容总量	1871 μ F
传感器种类及个数	CCD1 个、光编码器 1 个
除了车模原有的驱动电机、 舵机之外伺服电机个数	无
赛道信息检测精度	2mm（近端）、12.5mm(远端)
赛道信息检测频率	50Hz

7.2 体会

在小车的制作过程中，前期把主要精力都放在了控制方面，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。在此感谢飞思卡尔组委会对本次比赛的大力支持，同时感谢帮助我们的老师和同学。

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] 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
- [3] Freescale, Inc. MC9S12DT128B Device User Guide V01.09. Freescale Semiconductor, Inc, 2002
- [4] 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
- [5] 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
- [6] 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
- [7] 程启明, 黄林. 船舶操纵的自整定 PID 参数模糊控制器设计[J]. 船舶工程, 1997 (4): 45247.
- [8] 石辛民, 郝整清. 模糊控制及其 MATLAB 仿真. [M] 北京: 清华大学出版社, 北京交通大学出版社, 2008.
- [9] 张涛, 蒋静坪, 薛鹏骞. 模糊控制在位置随动系统中的应用研究[J]. 2007 (10), 1007 - 144X.
- [10] 谭浩强. C 程序设计[M]. 北京: 清华大学出版社, 2005

附录

源程序

```
#include "Include.h"

CntStructTypedef CntStruct;

void Speed_Angle_Init(void)
{
    RunLevel          = 1;
    Speeds[1][0] = (mOTORfULLsPEED/12)*2;
    Speeds[1][1] = (mOTORfULLsPEED/12)*2;
    Speeds[1][2] = (mOTORfULLsPEED/12)*2;
    Speeds[1][3] = (mOTORfULLsPEED/12)*3;
    Speeds[1][4] = (mOTORfULLsPEED/12)*3;
    Speeds[1][5] = (mOTORfULLsPEED/12)*4;
    Speeds[1][6] = (mOTORfULLsPEED/12)*4;
    Speeds[1][7] = (mOTORfULLsPEED/12)*5;
    Speeds[1][8] = (mOTORfULLsPEED/12)*5;
    Speeds[1][9] = (mOTORfULLsPEED/12)*5;
    Speeds[1][10]  = (mOTORfULLsPEED/12)*5;
    Speeds[1][11]  = (mOTORfULLsPEED/12)*6;
    Angles[1][0] = (cARaNGLELIMIT/100)*(100-0);
    Angles[1][1] = (cARaNGLELIMIT/100)*(100-0);
    Angles[1][2] = (cARaNGLELIMIT/100)*(100-10);
    Angles[1][3] = (cARaNGLELIMIT/100)*(100-20);
    Angles[1][4] = (cARaNGLELIMIT/100)*(100-30);
    Angles[1][5] = (cARaNGLELIMIT/100)*(100-40);
    Angles[1][6] = (cARaNGLELIMIT/100)*(100-50);
    Angles[1][7] = (cARaNGLELIMIT/100)*(100-80);
    Angles[1][8] = (cARaNGLELIMIT/100)*(100-90);
    Angles[1][9] = (cARaNGLELIMIT/100)*(100-94);
    Angles[1][10]  = (cARaNGLELIMIT/100)*(100-94);
    Angles[1][11]  = (cARaNGLELIMIT/100)*(100-97);
```

```
}
```

```
void NoCrossControl(void)//近处品均面积
```

```
{//没有交点时，控制修正角度
```

```
    unsigned int RowCt;
```

```
    signed int    WindageL, WindageR;
```

```
    LPLD_GPIO_Set_b(PTD_BASE_PTR, 8, 0);//进入没有交点控制，前 2 灯（8、9）亮
```

```
    LPLD_GPIO_Set_b(PTD_BASE_PTR, 9, 0);
```

```
    LPLD_GPIO_Set_b(PTD_BASE_PTR, 10, 1);
```

```
    LPLD_GPIO_Set_b(PTD_BASE_PTR, 11, 1);
```

```
    LPLD_GPIO_Set_b(PTD_BASE_PTR, 12, 1);
```

```
    LPLD_GPIO_Set_b(PTD_BASE_PTR, 13, 1);
```

```
    LPLD_GPIO_Set_b(PTD_BASE_PTR, 14, 1);
```

```
    LPLD_GPIO_Set_b(PTD_BASE_PTR, 15, 1);
```

```
    if( ( (RoadInf.LInf==0x03)|| (RoadInf.LInf==0x07) ) &&
```

```
        ( (RoadInf.RInf==0x03)|| (RoadInf.RInf==0x07) ) )
```

```
    { //2 边都有有效行
```

```
        if( RoadInf.LUp < RoadInf.RUp )
```

```
        { //左有效行短
```

```
            if( RoadInf.LUp < (row/2) )
```

```
            {
```

```
                WindageL = 0;
```

```
                WindageR = 0;
```

```
                for(RowCt=0; RowCt<=RoadInf.LUp; RowCt++)
```

```
                {
```

```
                    WindageL    += Lline[RowCt]-RoadLline[RowCt];
```

```
                    WindageR    += Rline[RowCt]-RoadRline[RowCt];
```

```
                }
```

```
                WindageL /= (signed int)(RoadInf.LUp+1);
```

```
                WindageR /= (signed int)(RoadInf.LUp+1);
```

```
                CntStruct.LstAngle = CntStruct.UniAngle;
```

```
                CntStruct.UniAngle = (WindageL+WindageR)/2;//////////
```

```
                //ntStruct.UniAngle /= 8;
```

```

    }else
    {
        WindageL = 0;
        WindageR = 0;
        unsigned int MMMM=row/2;
        for(RowCt=0;RowCt<MMMM;RowCt++)
        {
            WindageL    += Lline[RowCt]-RoadLline[RowCt];
            WindageR    += Rline[RowCt]-RoadRline[RowCt];
        }
        WindageL /= (signed int)MMMM;
        WindageR /= (signed int)MMMM;
        CntStruct.LstAngle = CntStruct.UniAngle;
        CntStruct.UniAngle = (WindageL+WindageR)/2;//求品均
        CntStruct.UniAngle /= 8;//搞系数
    }
}
}else
{
    //右有效行短或者一样长
    if( RoadInf.RUp < (row/2) )
    {
        WindageL = 0;
        WindageR = 0;
        for(RowCt=0;RowCt<=RoadInf.RUp;RowCt++)
        {
            WindageL    += Lline[RowCt]-RoadLline[RowCt];
            WindageR    += Rline[RowCt]-RoadRline[RowCt];
        }
        WindageL /= (signed int)(RoadInf.RUp+1);
        WindageR /= (signed int)(RoadInf.RUp+1);
        CntStruct.LstAngle = CntStruct.UniAngle;
        CntStruct.UniAngle = (WindageL+WindageR)/2;////////////////////
        CntStruct.UniAngle /= 8;
    }
}
}
{
    WindageL = 0;

```

```

        WindageR = 0;
        unsigned int MMMM=row/2;
        for(RowCt=0;RowCt<MMMM;RowCt++)
        {
            WindageL    += Lline[RowCt]-RoadLline[RowCt];
            WindageR    += Rline[RowCt]-RoadRline[RowCt];
        }

        WindageL /= (signed int)MMMM;
        WindageR /= (signed int)MMMM;

        CntStruct.LstAngle = CntStruct.UniAngle;
        CntStruct.UniAngle = (WindageL+WindageR)/2;////////////////////
        CntStruct.UniAngle /= 8;
    }
}
}else
{
    //不是每条线都有有效行
    if( (RoadInf.LInf==0x03)||(RoadInf.LInf==0x07) )
    {
        //左线有效
        if( RoadInf.LUp < (row/2) )
        {
            WindageL = 0;
            for(RowCt=0;RowCt<=RoadInf.LUp;RowCt++)
            {
                WindageL    += (signed int)Lline[RowCt]-(signed int)RoadLline[RowCt];
            }
            WindageL /= (signed int)(RoadInf.LUp+1);
            CntStruct.LstAngle = CntStruct.UniAngle;
            CntStruct.UniAngle = WindageL;////////////////////
            CntStruct.UniAngle /= 8;
        }
    }
    {
        WindageL = 0;
        unsigned int MMMM=row/2;
    }
}

```



```

for(RowCt=0;RowCt<MMMM;RowCt++)
{
    WindageL    += (signed int)Lline[RowCt]-(signed int)RoadLline[RowCt];
}
WindageL /= (signed int)MMMM;
CntStruct.LstAngle = CntStruct.UniAngle;
CntStruct.UniAngle = WindageL;////////////////////
CntStruct.UniAngle /= 8;
}
}
else if( (RoadInf.RInf==0x03)||(RoadInf.RInf==0x07) )
{//右线有效
    if( RoadInf.RUp < (row/2) )
    {
        WindageR = 0;
        for(RowCt=0;RowCt<=RoadInf.RUp;RowCt++)
        {
            WindageR    += (signed int)Rline[RowCt]-(signed int)RoadRline[RowCt];
        }
        WindageR /= (signed int)(RoadInf.RUp+1);
        CntStruct.LstAngle = CntStruct.UniAngle;
        CntStruct.UniAngle = WindageR;////////////////////
        CntStruct.UniAngle /= 8;
    }
}
else
{
    WindageR = 0;
    unsigned int MMMM=row/2;
    for(RowCt=0;RowCt<MMMM;RowCt++)
    {
        WindageR    += Rline[RowCt]-RoadRline[RowCt];
    }
    WindageR /= (signed int)MMMM;
    CntStruct.LstAngle = CntStruct.UniAngle;
    CntStruct.UniAngle = WindageR;////////////////////
    CntStruct.UniAngle /= 8;
}
}

```

```

    }else
    {
        //没有有效行
        CntStruct.LstSpeed = CntStruct.UniSpeed;
        CntStruct.UniSpeed = 0;
        CntStruct.UniAngle = CntStruct.LstAngle;//////////保持上次角度
    }
}
}
}

```

```

void CacItAngle1(void)

```

```

{
    unsigned int RowCt;
    signed int    WindageL, WindageR;

    if(RoadInf.AffUp==AFF_FAIL)
    {
        //没有找到种子段
        CntStruct.LstSpeed = CntStruct.UniSpeed;
        CntStruct.UniSpeed = 0;
        CntStruct.UniAngle = CntStruct.LstAngle;
        return;
    }
    if(RoadInf.AffUp==0xff)
    {
        //没有感染到
        CntStruct.LstSpeed = CntStruct.UniSpeed;
        CntStruct.UniSpeed = 0;
        CntStruct.UniAngle    = CntStruct.LstAngle;
        return;
    }
}

```

```

if( RoadInf.LInf!=0x03 && RoadInf.LInf!=0x07 && RoadInf.RInf!=0x03 &&
RoadInf.RInf!=0x07 )

```

```

{
    //没有有效行
    CntStruct.LstSpeed = CntStruct.UniSpeed;
    CntStruct.UniSpeed = 0;
    CntStruct.UniAngle = CntStruct.LstAngle;
}

```

```

    return;
}

```

//以下是 (有有效行) 才执行的_____

```

if( (RoadInf.LCrossX==0xffff)&&(RoadInf.RCrossX==0xffff) )
{
    //没 Cross
    NoCrossControl();
    CntStruct.LstSpeed = CntStruct.UniSpeed;
    CntStruct.UniSpeed = 6000;//mOTORfULLsPEED;//全速前进!!!
    return;
}

```

//以下是 (有有效行&&有交点) 才执行_____

```

if( ( (RoadInf.LInf==0x03)|| (RoadInf.LInf==0x07) ) &&
    ( (RoadInf.RInf==0x03)|| (RoadInf.RInf==0x07) ) )
{
    //2 跳线都存在有效行
    if( RoadInf.LUp>RoadInf.RUp )
    {
        //左边有效行长
        WindageL = 0;
        for(RowCt=0;RowCt<=RoadInf.LUp;RowCt++)
        {
            WindageL += Lline[RowCt]-RoadLline[RowCt];
        }
        WindageL /= (signed int)(RoadInf.LUp+1);

        CntStruct.LstAngle = CntStruct.UniAngle;
        CntStruct.UniAngle = WindageL;/////////////////////////////////
    }
    else
    {
        //右边有效行长，或者一样长
        WindageR = 0;
        for(RowCt=0;RowCt<=RoadInf.RUp;RowCt++)
        {
            WindageR += Rline[RowCt]-RoadRline[RowCt];
        }
    }
}

```

```

    }
    WindageR /= (signed int)(RoadInf.RUp+1);

    CntStruct.LstAngle = CntStruct.UniAngle;
    CntStruct.UniAngle = WindageR;/////////////////////////////////
}

return;
}

//以下是 (有且仅有一个有效行&&有交点) 才执行


---



if( RoadInf.LInf==0x03 || RoadInf.LInf==0x07 )
{
    //左线有有效行且其上有交点
    WindageL = 0;
    for(RowCt=0;RowCt<=RoadInf.LUp;RowCt++)
    {
        WindageL    += Lline[RowCt]-RoadLline[RowCt];
    }
    WindageL /= (signed int)(RoadInf.LUp+1);

    CntStruct.LstAngle = CntStruct.UniAngle;
    CntStruct.UniAngle = WindageL;/////////////////////////////////
}
else
{
    //右线有有效行且其上有交点
    WindageR = 0;
    for(RowCt=0;RowCt<=RoadInf.RUp;RowCt++)
    {
        WindageR    += Rline[RowCt]-RoadRline[RowCt];
    }
    WindageR /= (signed int)(RoadInf.RUp+1);

    CntStruct.LstAngle = CntStruct.UniAngle;
    CntStruct.UniAngle = WindageR;/////////////////////////////////
}

```

```

    }
}

void CacItAngle(void)//计算角度
{

    unsigned int RowCt;
    signed int    WindageL, WindageR;

    if(RoadInf.AffUp==AFF_FAIL)
    {//没有找到种子段

        CntStruct.LstSpeed = CntStruct.UniSpeed;
        CntStruct.UniSpeed = 0;
        CntStruct.UniAngle = CntStruct.LstAngle;
        return;
    }
    if(RoadInf.AffUp==0xff)
    {//没有感染到

        CntStruct.LstSpeed = CntStruct.UniSpeed;
        CntStruct.UniSpeed = 0;
        CntStruct.UniAngle    = CntStruct.LstAngle;
        return;
    }

    if( RoadInf.LInf!=0x03 && RoadInf.LInf!=0x07 && RoadInf.RInf!=0x03 &&
        RoadInf.RInf!=0x07 )
    {//没有有效行

        CntStruct.LstSpeed = CntStruct.UniSpeed;
        CntStruct.UniSpeed = 0;
        CntStruct.UniAngle    = CntStruct.LstAngle;
        return;
    }

    //以下的內容是 (有有效行) 才执行的_____

```

```

if( (RoadInf.LCrossX==0xffff)&&(RoadInf.RCrossX==0xffff) )
{
    //没 Cross
    NoCrossControl();
    CntStruct.LstSpeed = CntStruct.UniSpeed;
    CntStruct.UniSpeed = 6000;//mOTORfULLsPEED;//全速前进!!!
    return;
}

```

//以下内容是 (有有效行&&有交点) 才执行_____

```

if( ( (RoadInf.LInf==0x03)||(RoadInf.LInf==0x07) ) &&
( (RoadInf.RInf==0x03)||(RoadInf.RInf==0x07)) )
{
    //2 跳线都存在有效行
    if( RoadInf.LUp>RoadInf.RUp )
    {
        //左边有效行长
        WindageL = 0;
        for(RowCt=0;RowCt<=RoadInf.LUp;RowCt++)
        {
            WindageL += Lline[RowCt]-RoadLline[RowCt];
        }
        WindageL /= (signed int)(RoadInf.LUp+1);

        CntStruct.LstAngle = CntStruct.UniAngle;
        CntStruct.UniAngle = WindageL;////////////////////////////////////
    }else
    {
        //右边有效行长，或者一样长
        WindageR = 0;
        for(RowCt=0;RowCt<=RoadInf.RUp;RowCt++)
        {
            WindageR += Rline[RowCt]-RoadRline[RowCt];
        }
        WindageR /= (signed int)(RoadInf.RUp+1);

        CntStruct.LstAngle = CntStruct.UniAngle;
    }
}

```

```

        CntStruct.UniAngle = WindageR;////////////////////
    }

    return;
}

//以下是 (有且仅有一个有效行&&有交点) 才执行


---



if( RoadInf.LInf==0x03 || RoadInf.LInf==0x07 )
{
    //左线有有效行且其上有交点

    WindageL = 0;
    for(RowCt=0;RowCt<=RoadInf.LUp;RowCt++)
    {
        WindageL    += Lline[RowCt]-RoadLline[RowCt];
    }
    WindageL /= (signed int)(RoadInf.LUp+1);

    CntStruct.LstAngle = CntStruct.UniAngle;
    CntStruct.UniAngle = WindageL;////////////////////
}
else
{
    //右线有有效行且其上有交点

    WindageR = 0;
    for(RowCt=0;RowCt<=RoadInf.RUp;RowCt++)
    {
        WindageR    += Rline[RowCt]-RoadRline[RowCt];
    }
    WindageR /= (signed int)(RoadInf.RUp+1);

    CntStruct.LstAngle = CntStruct.UniAngle;
    CntStruct.UniAngle = WindageR;////////////////////
}
}

void TillerControlPID(void)

```

```

{
    unsigned short TillerDir;
    unsigned int TillerAng;
    int          Kp, Kd;//PD 控制
    //Kp          = 1.6;
    //Kd          = 0.6;

    Kp= 10;
    Kd= 0;
    CntStruct.UniSteer = Kp*CntStruct.UniAngle +
    Kd*(CntStruct.UniAngle-CntStruct.LstAngle);// + 2.5*(err_s_e);//入弯

    if( CntStruct.UniSteer > 0 )
    {
        TillerDir = tILLERrdIR;
        if( CntStruct.UniSteer>cARaNGLEIIMIT )
            TillerAng= cARaNGLEIIMIT;
        else
            TillerAng= CntStruct.UniSteer;
    }else if( CntStruct.UniSteer ==0 )
    {
        TillerDir = tILLERcENTER;
    }else
    {
        TillerDir = tILLERldIR;
        if( CntStruct.UniSteer+cARaNGLEIIMIT<0 )
            TillerAng= cARaNGLEIIMIT;
        else
            TillerAng= CntStruct.UniSteer*(-1);
    }

    TillerAngleSet(TillerDir, TillerAng);
}

```