

第十届“飞思卡尔”杯全国大学生

智能汽车竞赛

# 技 术 报 告

学    校：    武汉工商学院

队伍名称：    风火队

参赛队员：    文小林

李钰文

涂划

带队教师：    曾鹤琼

邓明华

## 关于技术报告和研究论文使用授权的说明

本人完全了解第十届“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：\_\_\_\_\_

## 目录

引言.....	1
第一章 智能车的总体设计方案.....	2
1.1 总体设计框架简介.....	2
1.1.1 系统结构框图.....	2
1.1.2 整车布局.....	2
第二章 机械结构的安装和调试.....	3
2.1 转向轮的参数调整.....	3
2.2 底盘及重心的调整.....	5
2.3 轮胎表面的打磨.....	5
2.4 小结.....	6
第三章 硬件电路的设计.....	6
3.1 硬件电路框架.....	6
3.2 电源电路的设计.....	7
3.3 驱动电路的设计.....	8
3.4 PCB 制版.....	9
3.5 小结.....	9
第四章 控制程序的设计与调试.....	10
4.1 系统软件流程图.....	10
4.2 跟踪边沿检测法简介.....	11
4.3 赛道类型识别.....	12
4.3.1 直角弯识别.....	12
4.3.2 中心引导线识别.....	13
4.3.1 红外信号识别.....	14
4.3.4 障碍识别.....	15
4.3.5 不对称坡道识别.....	15
4.4 舵机控制.....	16
4.4 电机控制.....	17
4.5 小结.....	18
第五章 参数的整定与调试.....	18
5.1 开发环境.....	18
5.2 调试工具.....	19
5.3 小结.....	22
第六章 全文总结.....	23
6.1 智能小车的主要参数.....	23
6.2 总结.....	24
参考文献.....	25
附录：.....	26

## 引言

随着信息技术的发展，汽车的电子化模块越来越多，智能汽车领域受到了大量的关注。高等学校自动化专业教学指导分委员会主办“飞思卡尔”杯全国大学生智能汽车竞赛，以迅猛发展的汽车电子为背景，涵盖了控制、模式识别、传感技术、电子、电气、计算机、机械等多个学科交叉的科技创意性比赛。飞思卡尔智能车竞赛是以智能汽车为研究对象的创意性科技竞赛，是面向全国大学生的一种具有探索性工程实践活动，是教育部倡导的大学生科技竞赛之一。比赛要求在组委会提供统一智能车竞赛车模下选用飞思卡尔公司的指定控制芯片，以较快的速度在符合组委会规定下完成比赛。

本文介绍的智能车系统选用32位微处理器MK60FX512VLQ15为控制核心，采用具有较好高速动态性能的OV7725数字摄像头，利用蓝牙模块、液晶模块、上位机等工具辅助分析。本文主要通过四个部分来阐述智能车的设计过程，即机械结构的环境适应、硬件电路的设计、赛道环境的获取、执行行程的决策和算法的。通过这四个部分的协调工作使智能车能够满足高速运行下的高速性和稳定性需求，本文将对智能车的总体设计和各部分的详细设计进行一一介绍

# 第一章 智能车的总体设计方案

## 1.1 总体设计框架简介

### 1.1.1 系统结构框图



图 1-1 智能车系统结构框图

摄像头车系统整体结构如图 1-1 所示。MK60FX256ZVLL10 微处理器通过采集 OV7727 数字摄像头的硬件二值化信号，得到赛道边沿信息；通过编码器获取左右车轮的速度，得到车行进的速度数据，进行闭环及差速控制；通过红外接收管获取发、停车的信号，做减速、停车、发车的控制；例外，本文介绍的智能车系统还采用了陀螺仪和红外光电对管辅助识别赛道信息。微处理器对角度、速度进行 PID 控制实现转向和速控。在调试过程中，通过上位机观察摄像头采集的图像、赛道数据等实时数据，便于发现问题、进行调试。

### 1.1.2 整车布局

在保证车子灵活性和稳定性的前提下，尽量降低重心和减轻质量。为获取最佳最佳的机械性能，还需注重各个小细节。比如编码器的选型与固定方式，以及有利于车模进行巡线、提速条件下的电池、电路板和传感器的固定的研究和设计。在电路方面，稳定高效的电路设计是赛车能够稳定较好运行的保障，在经过多次改进之后，我们形成了以 K60 为核心的一整套稳定的电路系统。系统中子模块可

分为电源部分、摄像头部分、电机驱动、编码器、陀螺仪和加速度计等几个部分。

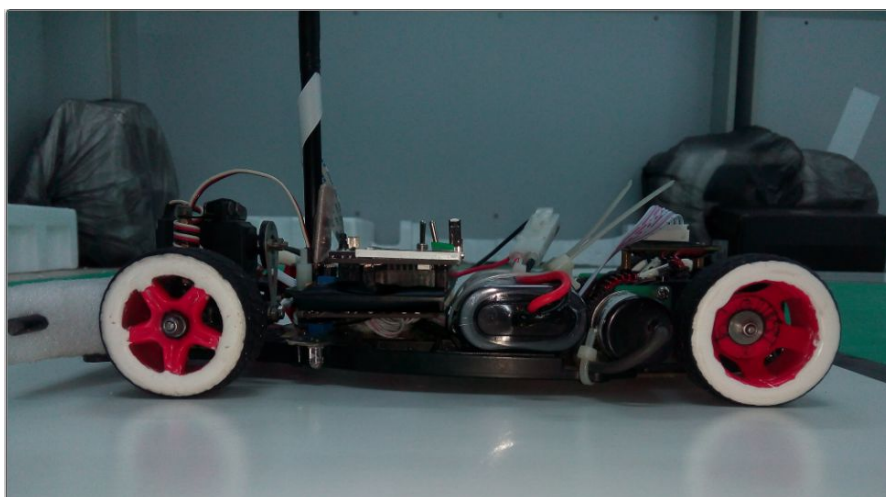


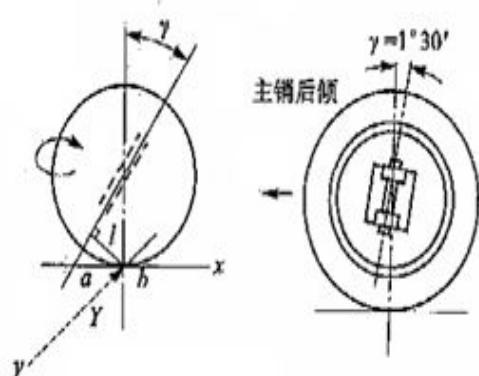
图 1-2 整车布局图

## 第二章 机械结构的安装和调试

### 2.1 转向轮的参数调整

此套车模在车轮定位方面可以改动的地方有：主销前后倾角、主销内外倾角、前轮前束。

主销前后倾角，如图 2-1 所示，可以通过改变悬挂上前后黄色小垫片的数量（原车前后各有两个黄色小垫片，主销处于垂直状态），当模型车需过坡道，或在比较颠簸的路面行驶时，主销宜有一定后倾角，但后倾角不能过大，否则会产生过大的回正力矩，导致模型车在出弯进入直道时发生左右摆动。



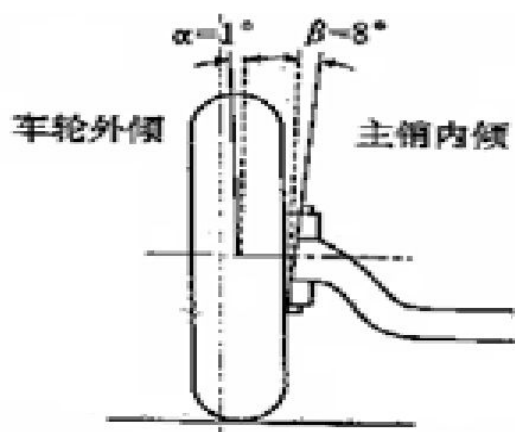
(a)示意图



(b) 实物图

图 2-1 主销后倾角

主销内外倾角，如图 2-2 所示，可通过旋转悬挂上导向臂螺杆，改变上导向臂螺杆长度，在不同的赛道调试时，主销内外倾角会有不同的设置，在急弯较多的赛道上行驶时，宜使主销有一定的内倾角，提高过弯时的稳定性，但倾角不宜过大。主销的倾角设置应在调试赛车时不断调整，不同的赛道和不同的程序会有不一样的倾角设置，具体的角度设置应视调车时车辆的行驶情况而定。



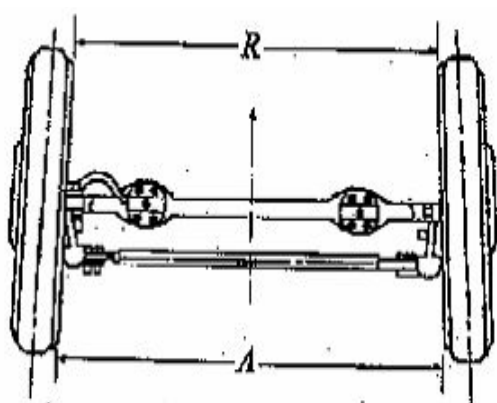
(a) 示意图



(b) 实物图

图 2-2 主销内外倾角示意图

前轮前束，如图 2-3 所示，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。



(a) 示意图



(b) 实物图

图 2-3 前轮前束示意图

如图 2-4 所示，模型车悬架上原装的弹簧属于刚度较大的弹簧，适用于较高车速，在高速过弯时可以减小车身的侧倾，提高稳定性。但模型车的悬挂上没有阻尼，在高速通过急转弯时，前轮会出现较大的跳动，大大降低车速和行驶稳

定性，当出现这种情况时，应选用刚度较小的弹簧。



图 2-4 前轮悬挂示意图

## 2.2 底盘及重心的调整

模型车悬挂的下方可以加装垫片，用来降低底盘高度，在没有坡道或比较平顺的路面行驶时，应降低底盘高度，以此降低整车重心高度，这样有利于模型车在高速行驶时的稳定性。当加装垫片后，会使底盘与悬挂连接的螺钉容易松动，应改用较长的螺钉，同时改用螺栓连接或用胶粘来防止螺钉松动。当赛道上有坡道时，底盘就不能过于降低，过低的底盘会在模型车通过坡道时，底盘与赛道发生碰撞，所以底盘高度应调整到通过坡道时底盘不与赛道发生碰撞为宜。



图 2-5 电池及主控板安放示意图

## 2.3 轮胎表面的打磨



车轮是模型车与地面接触的唯一部件，所以车轮的改装也十分必要，由于规则的限定，只能对轮胎性能做部分改动。可以将轮胎与轮毂用胶粘住，增加轮胎的侧偏刚度，提高车轮高速过弯的能力，粘胎时要十分谨慎，不能让胶撒在轮胎表面上。粘胎用的胶可选用 704，由于 704 硅橡胶，可以长时间使用。为提高轮胎与地面的附着力，可将轮胎中间的凸起线剪掉，保持轮胎与地面接触面平整。在跑车时要时刻保持轮胎清洁，必要时要用湿布擦洗，保持轮胎潮湿，但不能有水，可以提高轮胎与地面的附着力。



图 2-6 轮胎处理示意图

## 2.4 小结

机械结构对赛车在运动过程中的稳定起到了至关重要的最用，通过改变赛车的悬挂系统让小车的高速转弯过程中再受到较大向心力的情况下有更好的稳定性，通过降低和将压力平均的分配到每个轮胎上让小车有更好的控制效果。通过对轮胎的打磨让小车有更强的抓地力，让赛车不会发生侧滑和漂移。

# 第三章 硬件电路的设计

## 3.1 硬件电路框架

智能车系统主要由 CMOS 摄像头、视频采集辅助电路、测速装置、电机驱动电路、舵机、直流电机和电源几部分组成。系统硬件电路图如图 4.2 所示。

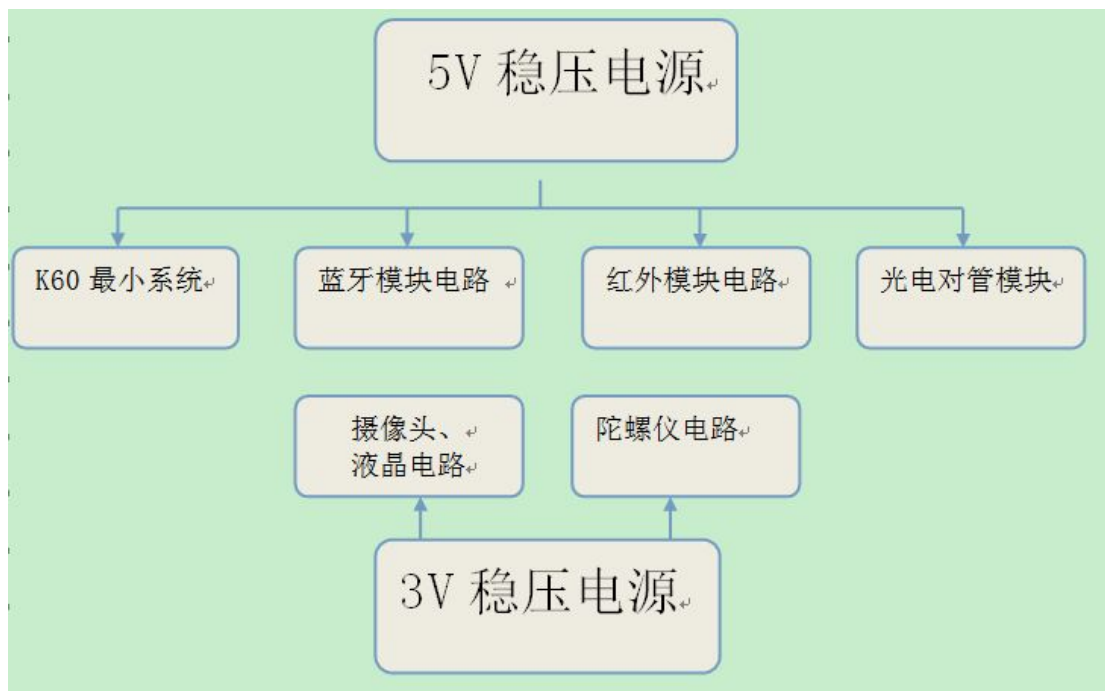


图 3-1 系统硬件电路框图

智能车系统控制由 MK60FX512VLQ15 作为核心。由 CMOS 摄像头对路况信息进行采集获取图像数据。同时，测速装置测量电机的速度，MK60FX512VLQ15 对反馈进行计数，间接反映车速。MK60FX512VLQ15 根据采集来的图像特征，通过一定的算法，来对舵机进行适当的控制。结合反馈通过 PID 速度调节输出 PWM 波形到驱动芯片 BTS7971 驱动直流电机。

系统由 7.2V 电池供电，舵机和 BTS7971 芯片电源由电池直接供给。经过电源稳压芯片 LM2940 输出 +5V 电压给 MK60FX512VLQ15 和其他芯片供电，通过 LM2937 变换器控制电路输出 3.3V 为摄像头供电。

### 3.2 电源电路的设计

供电电路主要分为 3.3V 为摄像头液晶供电，5V 为单片机传感器和驱动供电，6.5V 为舵机供电。在芯片的选择上，我们做过很多实验，对吼确定为用市场上比较常见的电源芯片，LM2940, LM2937 其主要是货源充足，芯片稳定，输出电压波动很小，芯片的外部电路简单。在舵机的驱动芯片上面考虑到舵机为感性耗电单元其驱动要求较高，所以我们选用了数字电源芯片 AS1015，其驱动能力强，电压值可变。如图 4.3 LM2937 电路 图 4.4 LM2940 电路 图 4.5 AS1015 电路。

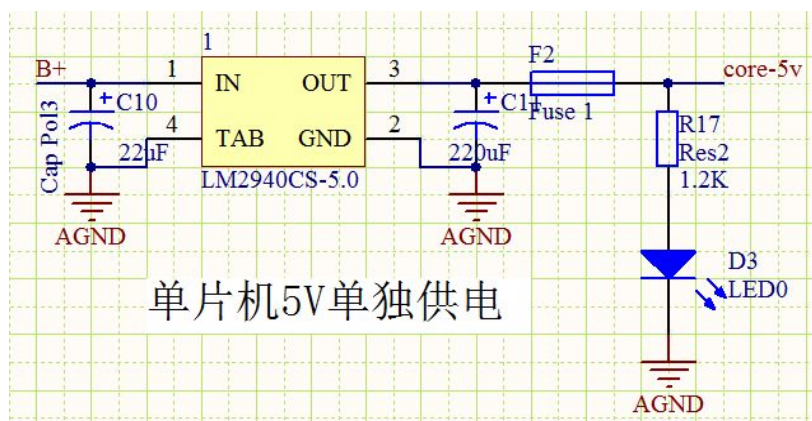


图 3-2 LM2940 5V 供电电路

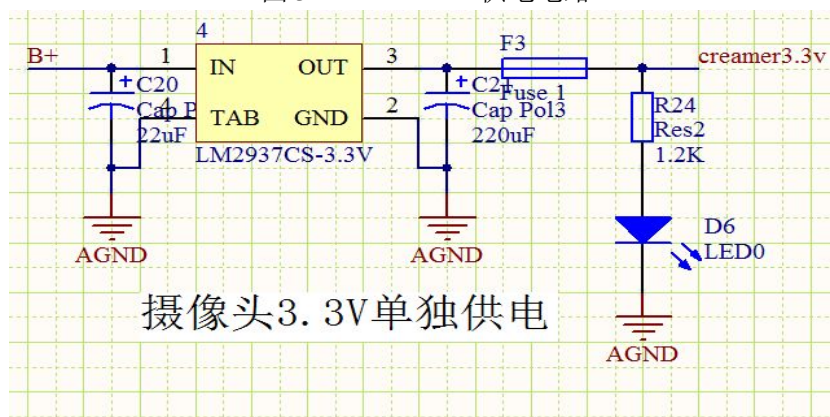


图 3-3 LM2937 3.3V 供电电路

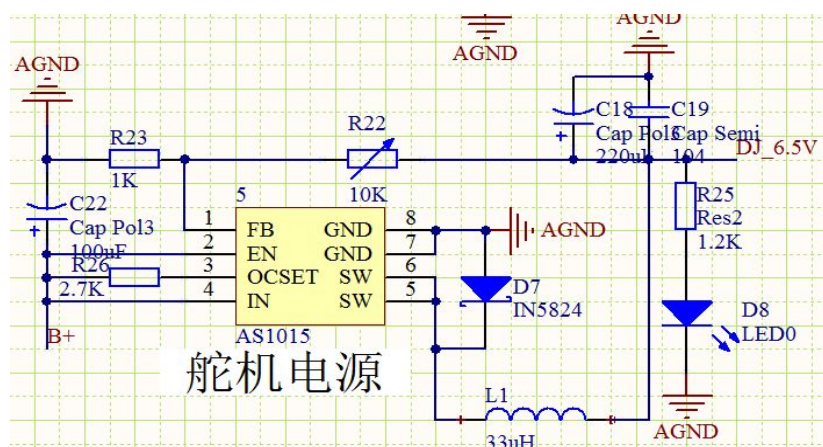


图 3-4 AS1015 舵机供电电路

### 3.3 驱动电路的设计

驱动电路芯片采用的是 BTN7971 内阻只有 3 欧姆左右, 而其驱动电流可以达到 70A, 完全可以满足对 260 电机的驱动要求并且驱动电路简单, 且输出稳定。



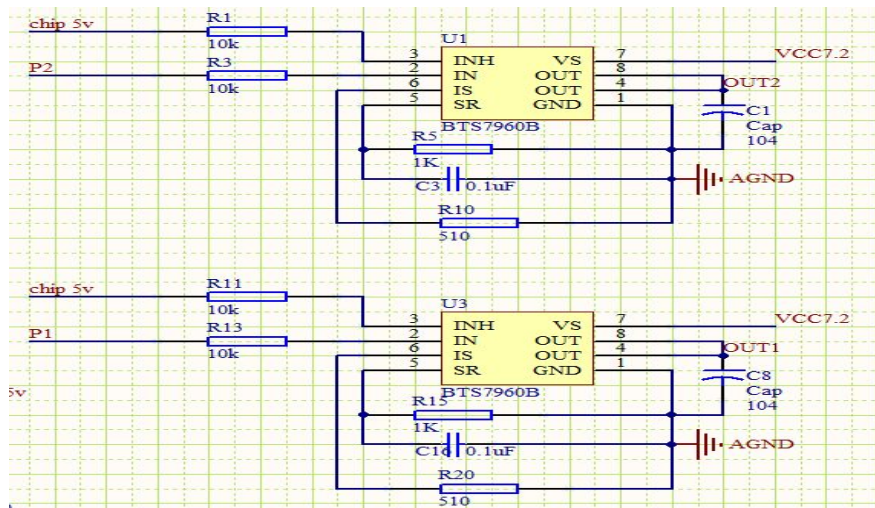


图 3-5 驱动电路

### 3.4 PCB 制版

整个电路设计采用 Altium Designer 进行设计，在整个 PCB 的制图过程中遵循这线与线扩大间距减小干扰，避免走折角，较少过孔，铺铜屏蔽的原则进行设计，通过测试，所制作的电路板电气性能良好，稳定。为了减轻小车的重量和减少连接线的数量，将所有电路全部都集中在了一块板子上从而提高板子的运行效率和稳定性。PCB 板如图 3-6。

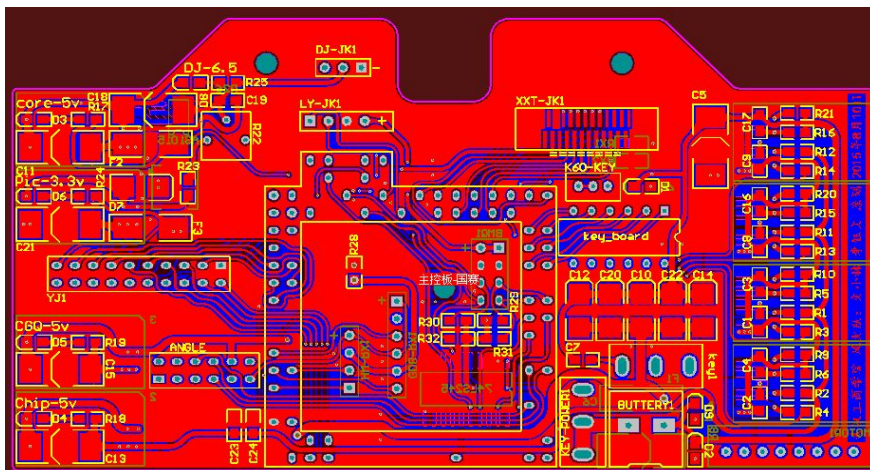


图 3-6 PCB 板

### 3.5 小结

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。

硬件电路部分在发送出去制作之前一定要再三检查错误。避免出现因为一个小错误导致整个板子都不能用的情况。从原理图到 PCB 的封装，都要仔细检查，做到无纰漏。另外制作板子也要跟机械方面相结合，使得板子大小能够放进底盘空缺处，降低重心。还有板子上打孔要跟车模上原装的孔对应，方便安装。制作出来的板子虽然经过再三的检查，但是还是会出现各种问题，不可能尽善尽美的。因此可能会出现多次制版，我们需要不断改进前面的问题。我从第一次洞洞板到最后第 9 块 PCB 版，虽然每块都可以使用，但还是在不断改进前面的问题。所以硬件上的每一个小细节对我们都至关重要。

## 第四章 控制程序的设计与调试

### 4.1 系统软件流程图

主程序流程图：

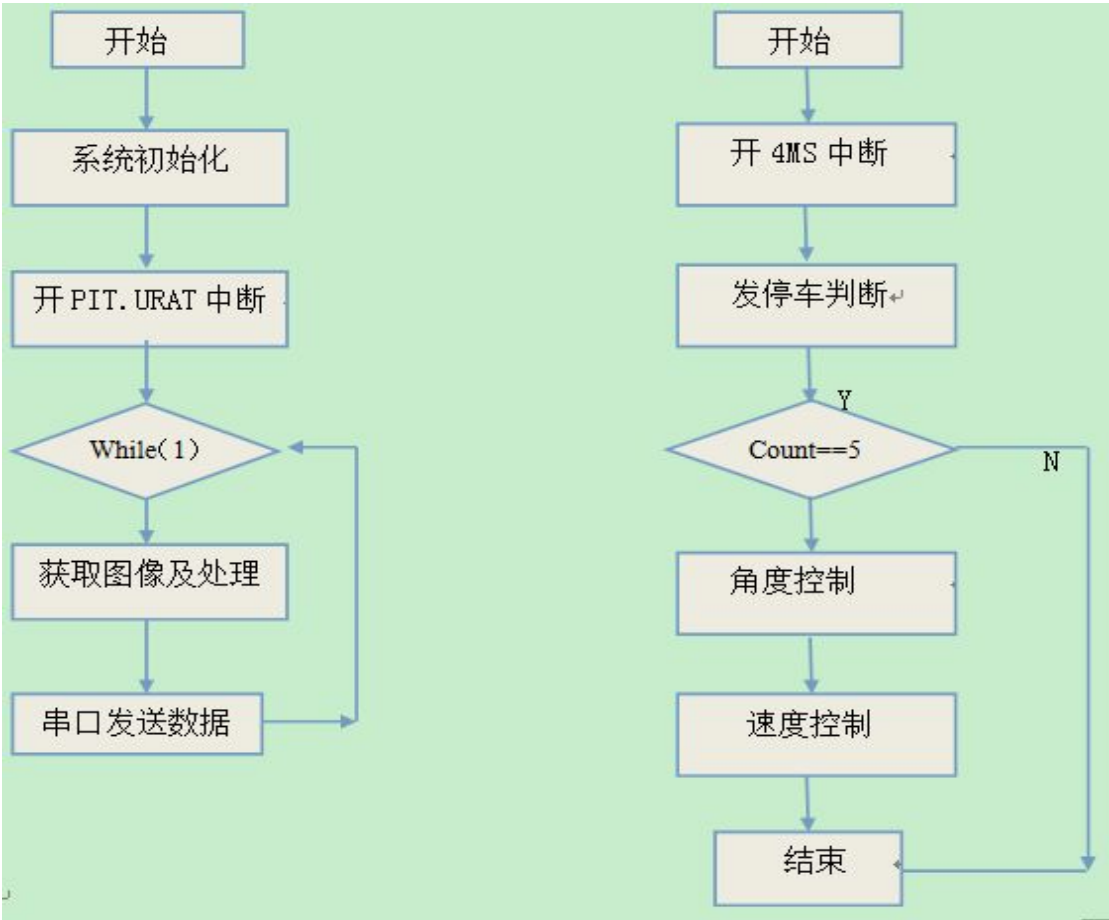


图 4-1 程序框图

## 4.2 跟踪边沿检测法简介

(1) 跟踪边沿检测法算法流程图：

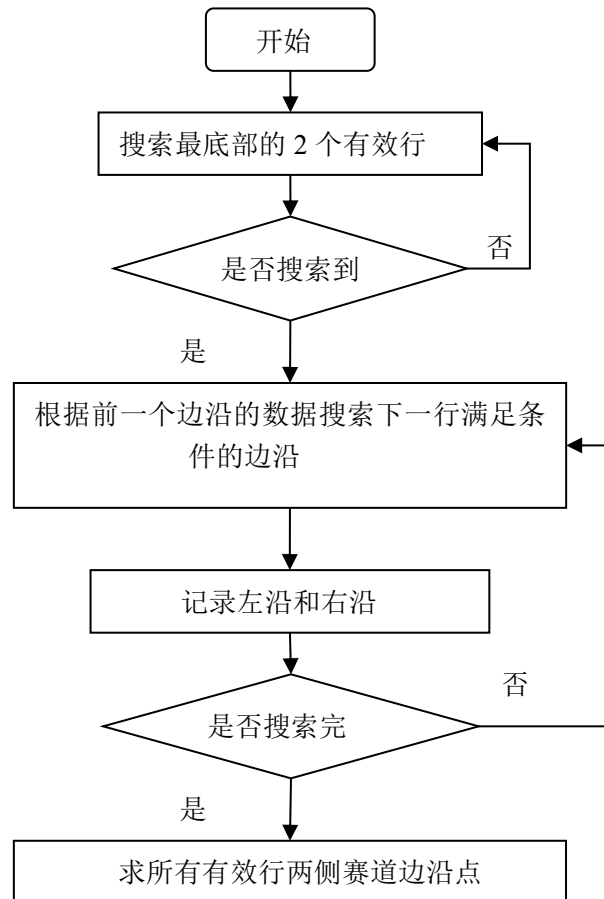


图 4-2 算法流程图

(2) 具体步骤如下

1) 首先准备提取黑线

2) 用检测跳变的方法提取出前 2 行中每行的两个跳变点，然后求平均值就可以得到前 2 行的黑线位置。当搜索到多个跳变的时候，我们根据上一行跳变的位置确定出最优的那个跳变的位置作为本行的黑线跳变位置。当前 10 行都没有找到黑线的时候，我们就认为这幅图像的黑线丢失了，然后依据前一幅图像黑线的位置，给这幅图像的整体赋极值。当只有 10 行中的几行丢失时，我们就继续搜索黑线直到找完前 10 行位置。

3) 当前 2 行黑线存在时，我们利用前 10 行黑线的位置确定第 3 行黑线的位置，

然后在这个区间搜索黑线，依次类推用前一行黑线的位置确定后一行黑线的位置，当本行黑线没有找到时，此行黑线位置保持上行的值，下行搜索的位置相应的扩大。有连续 3 行黑线搜索不到时我们就认为黑线丢失，退出搜索。这样既可以去除干扰，还可以大大的提高算法的效率。

值得注意的是：第 2 行和前 3 行这个接口位置边缘确定非常重要，要考虑的非常全面，不然有可能就只能搜索到前 2 行的黑线，后面的黑线因为边缘的问题所有不到。

4) 搜索完成后推出搜索。对搜索到的黑线进行中值滤波和限幅滤波。只要搜索范围合理，这种算法有很强的抗干扰能力，并且可以滤除十字交叉和三角黑区的干扰。

### 4.3 赛道类型识别

今年竞赛的特殊赛道类型相比往届相差较大。特殊赛道的类型包括：中心引导线、直角弯、楔形障碍等。常见的特殊赛道有十字路、小 S、大 S、不对称坡道等。一般情况下，我们的程序在着手写时分为两步走，即先在不误判的情况下识别。再以合适的处理方式寻线，特殊赛道元素的处理直接影响小车速度，应予以足够的重视，而处理过程中难点也在这些特殊的赛道类型处理，因其往届并未出现这些元素，所以需要我们独立创新的去寻找合适的处理方法，然后在不断的摸索过程中，笔者认为没有所谓的最好、绝对有效的处理方法，只有适合自己的小车的处理方式，因此，我在下面讲述的方法是我在长期摸索过程研究出来的一套适合本文智能车系统的一套方法，望予读者启发。

#### 4.3.1 直角弯识别

直角弯是第十届新增加的元素，赛道特征为在直角弯道两边距离 1 米处路面上有宽度为 10 厘米、长度等于路宽的黑色标识，用于光电平衡组和摄像头组检测提前减速。

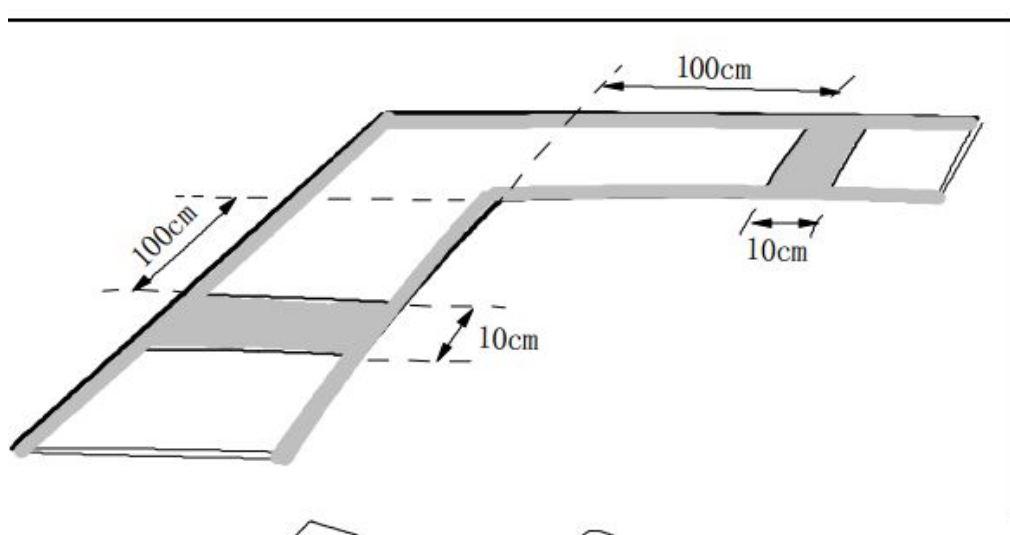


图 4-3 直角弯示意图

可以看出处理好直角弯关键在于能准确识别，我们注意到直角前的黑色标志线，只需要我们识别好黑线即可识别直角，这里有两种处理方式，一种是通过摄像头识别，另一种是通过其他传感器辅（红外光电对管）助识别，经过反复测试我们采用以摄像头和光电对管同时识别黑线的方法。这里要考虑两方面，一是保证小车在黑线前能以正确的轨迹前行，另一方面需要记录黑线的标志位保证准确过直角弯。如果通过摄像头识别黑线在直道上比较容易识别，但如果黑线紧接弯道则会因为摄像头视角失真看到一个角形弯而难以识别，而在这里我们通过比较左右线的变化趋势，可以判断小车的转向方向，给定一个打角值保证小车不出去，黑线在直道上，实际情况中依据上述寻线方式不需要处理也可较稳定的通过和识别，但保险起见，我们通过识别到黑线行后，将这一行上下 10 行的像素点全部置为白色，这样将赛道变换成十字情况，直接调用十字处理程序即可。这样车如果通过黑线光电管由于检测到黑线必然发生电平变化，通过识别光电管电平为后面通过直角做保障，在过直角弯时在黑线标志位置 1 时，如果丢线再做补线处理（或者给定舵机最大或最小值，测试中两种情况区别不大，可以达到在不减速或者加速的情况下稳定的通过直角弯。

#### 4.3.2 中心引导线识别

中心引导线是第十届新增加的元素，赛道特征为宽度为  $25 \pm 5\text{mm}$  的黑线。每段中心线的长度不超过 2 米。中心引导线只会出现在直道或者弯道上，距离赛道中的十字路口、坡道、中心引导线、直角弯道、路障有一米以上的距离。



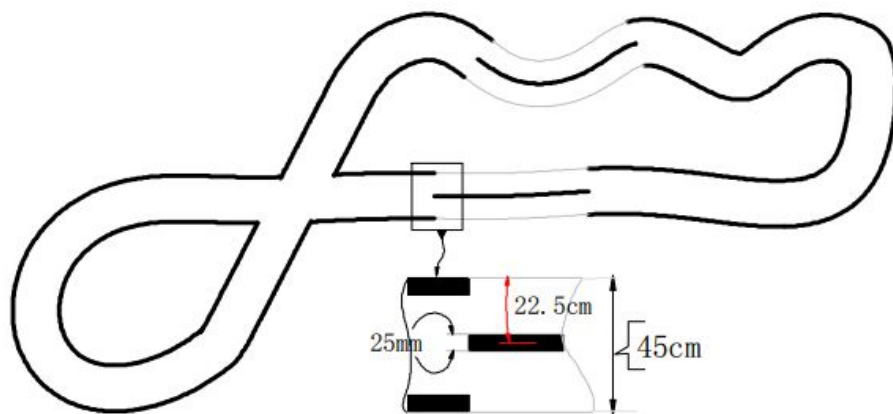


图 4-4 中心引导线示意图

由于采用跟踪边沿检测法，在遇到单双线切换的时候会出现问题，因此我们对中心引导线的情况加以区分为双线情况、单线情况、单线切双线情况、双线切单线情况。分别对以上四种情况进行处理，对于单线切双线、单线切双线我们仍然是采用中线法，寻找出完整的中线。

#### 4.3.1 红外信号识别

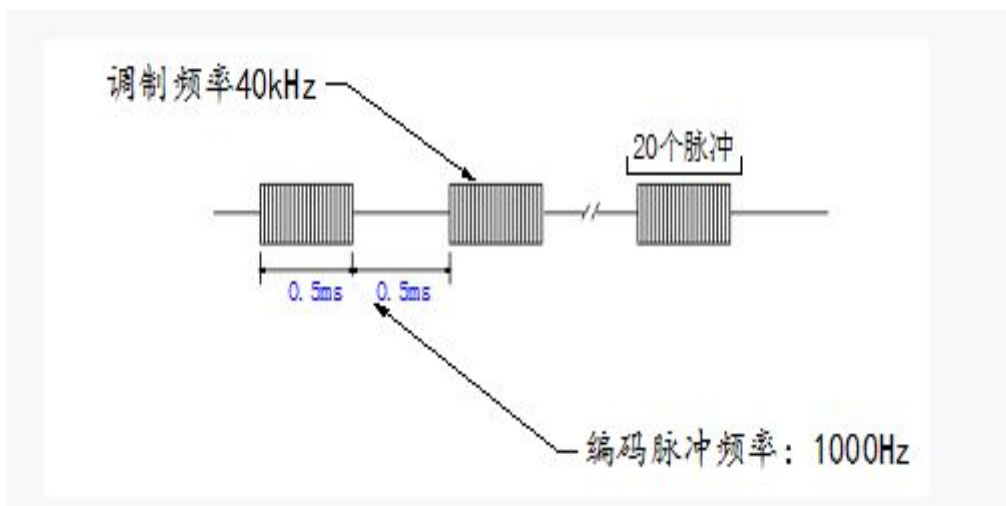
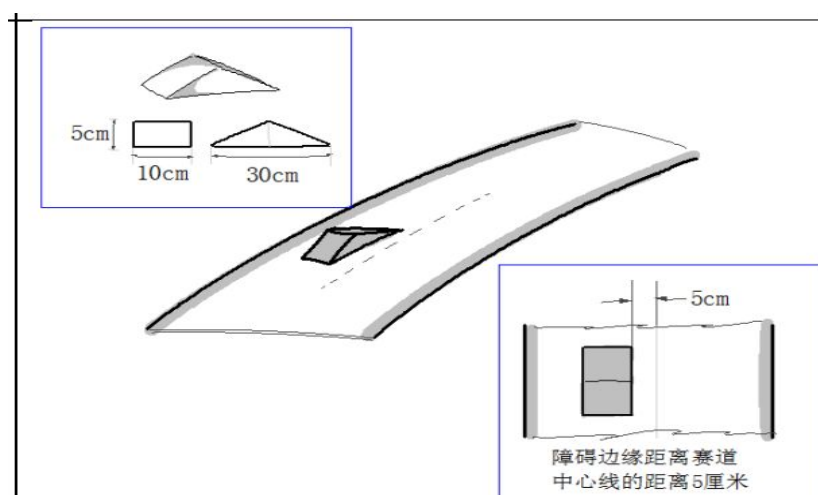


图 4-5 红外发射信号波形示意图

对于红外信号的识别主要用于发停车的判断，比赛要求比赛中灯塔信号中如上图，在原有的 40kHz 的调制信号的基础上，增加 1kHz，占空比为 50% 的编码信号，我们采用 HS0038 红外接受管，设计一个电压比较器，设置一个适合的阈值输出电平，省去繁琐的输入捕捉、中断程序，也节省了资源。

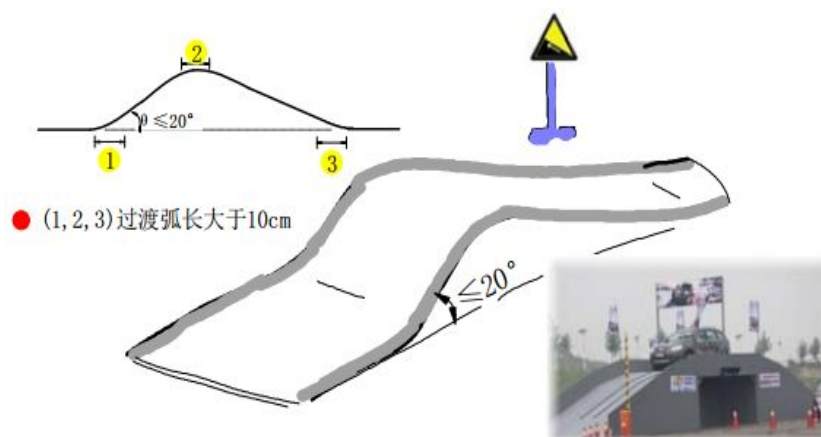
#### 4.3.4 障碍识别



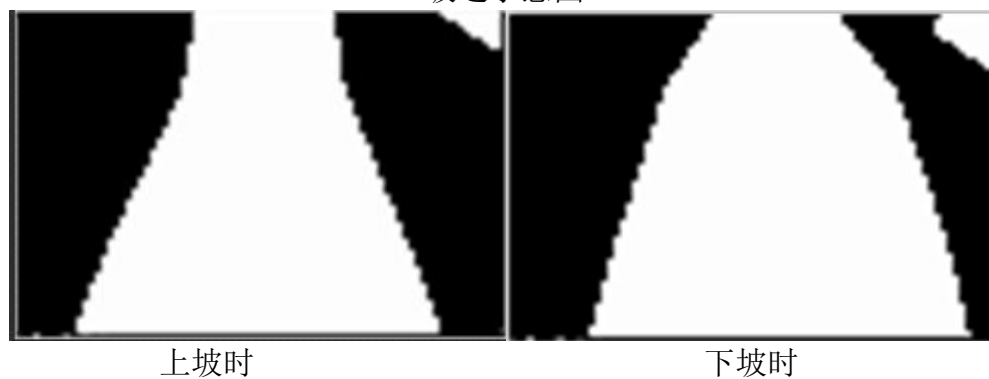
4-6 障碍物示意图

障碍是第十届改变的元素，比较去年障碍物变为楔形，位置有靠近边线移到靠近中间，由于我们采用跟踪边沿检测法，会直接出现看不到中间的障碍物的情况，我们采取才去老方法，每幅图抽取几行判断赛道宽度。宽度较小且左右线都存在，则为障碍的特征。

#### 4.3.5 不对称坡道识别



4-7 坡道示意图

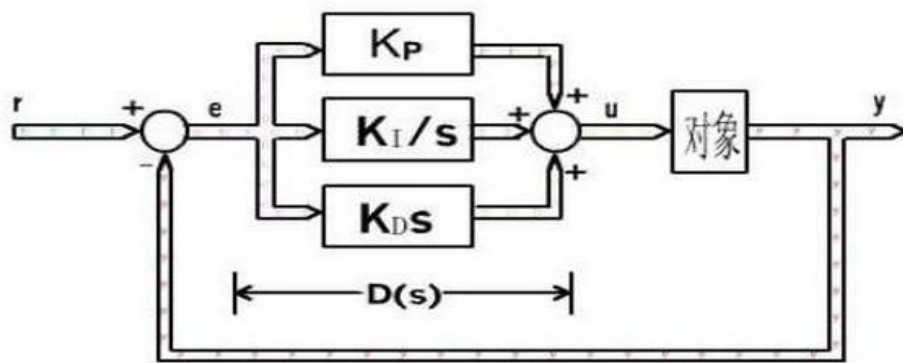


4-8 摄像头采集的坡道图

不对称坡道是第十届改变的元素，同样介绍两种方法，一种通过摄像头识别，主要由图像的宽度是识别，上坡时远处较大，下坡时远处较小。如下图所示，但由于我们的摄像头安装方式不同图像却别并不是很大，所以我们采用例外一种方式，即通过陀螺仪来辅助判断，利用在 ENC03 输出的角速度可以很好的判断坡道。

#### 4.4 舵机控制

舵机是控制赛车运动姿态的重要器件，在赛车运动过程中通过控制前轮转角来精确控制赛车的运动方向，因此让舵机的转角迅速达到指定的大小，尤为重要，同时为了让小车在入弯和出弯时能选择最优路径也需要舵机的直接控制。为了达到既让小车有良好的方向控制性又能在路况复杂的情况下选择最有路径，在舵机的控制算法上，运用了具有优良特性的 PID 算法。



4-8 PID 流程图

比例（P）：反应系统的基本（当前）偏差  $e(t)$ ，系数大，也以加快调节，减小误差，但是过大的比例会让系统的稳定性下降，甚至是造成系统的不稳定；

积分（I）：反应系统的累计偏差，使系统消除稳态误差，提高误差度，因为误差，积分调节就进行，直至没有误差；

微分（D）：反应系统偏差信号的变化率  $e(t) - e(t-1)$ ，具有预见性，能预见偏差变化的趋势，产生超前的控制动作，在偏差没有形成之前，已被微分调节作用消除，因此可以改善系统的动态性能，但是微分对噪音干扰又放大作用，加强为分对系统的抗干扰能力不利。

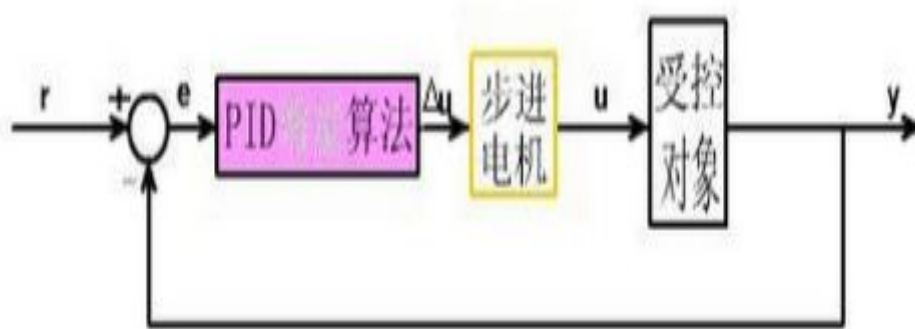
位置式 PID 公式：

$$\text{公式: } u(k) = k_p e(k) + k_i \sum_{j=0}^k e(j) + k_d (e(k) - e(k-1))$$

鉴于以上的 PID 控制算法的特性，在舵机的控制算法中选择了 PD 控制，即超前调整的控制输出，当赛车快要进入弯道时通过微分运算得出小车运动趋势，通过微分的控制加大舵机的打角值，从而实现提前打角的效果，同理在小车出弯回直时提前减小舵机的打角直减少舵机打角。防止小车出现角打多或者时打慢了的情况，从而让小车有更好的行驶路径。

#### 4.4 电机控制

当赛车在入弯和出弯时，为了达到入弯前减速，出弯时加速的效果，需要对小车的电机进行实时的精确控制，当小车再入弯减速和出弯加速时通常由于受到小车在运动过程中的受力不断变化、机械结构滞后以及电机本身性能的影响，总是不能达到理想的运动状态，所以为了让小车达到预定的运动状态，在电机的控制代码中加入了增量式 PID 算法，已实现小车快速达到预定动作。如图增量式 PID 框图。



4-8 电机控制流程图

上一次控制后采集回来的值与上一次输出的值之间的差值加上上一次控制的输出值为此次的输出值。通过增量式 PID 的算法在速度没有达到预定值时控制电机的输出使赛车尽快达到设定的运动状态，使小车既能提高速度又能提高稳定性。计算公式如下：。

$$\text{公式: } \Delta u(k) = k_p(e(k) - e(k-1)) + k_i e(k) + k_d(e(k) - 2e(k-1) + e(k-2))$$

相关代码如下：

```
float speed_pid_2(int32 speed1,int32 speed2 ) //speed1=设定的理想速度,
speed2=采集回来的反馈速度
```

```

{
    ee2=ee1;    //存储当前误差，用于下次计算
    ee1=ee0;
    ee0=speed1-speed2; //增量计算，即速度偏差
    incpid=(int32) (KKi*ee0+KKp*(ee0-ee1)+KKd*(ee0-2*ee1+ee2));
    SP=SP+incpid;
    if (SP>=9500)
        SP=9500;
    if (SP<0)
        SP=0;
    return SP ;
}

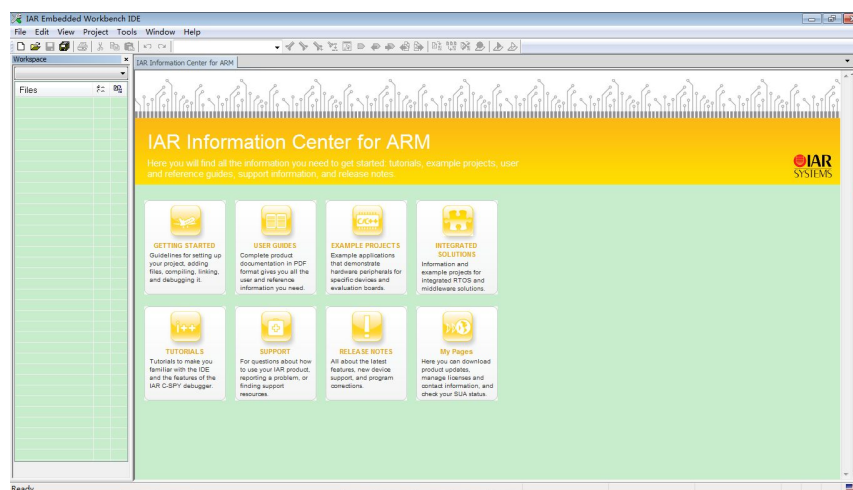
```

## 4.5 小结

在程序设计和调试的过程中，不能太模糊化，需要进行实际的建模和仿真；对智能车的运行一定要进行定量的分析和计算，对智能车运行中出现的各种情况都能做到理解—解决，站在理论支持的角度进行分析和调试；在调试过程中，注重上位机和 IAR 在线调试使用，通过车的状态和返回的数据，来提供调节参数的依据，可以为提高调测效率。

# 第五章 参数的整定与调试

## 5.1 开发环境



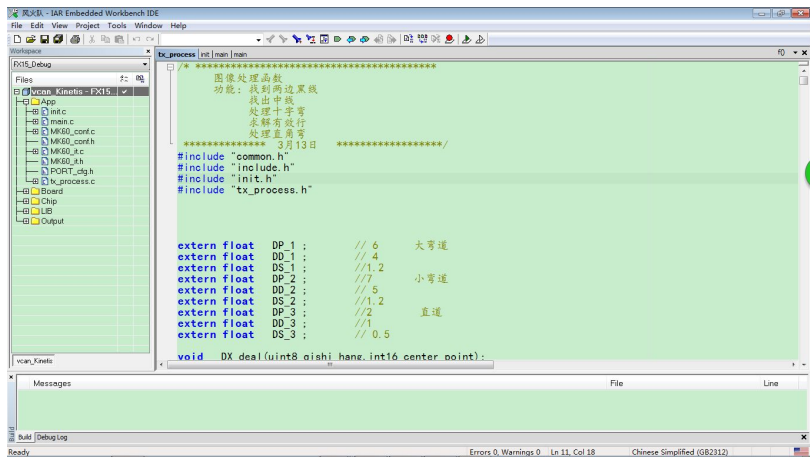
4-9 IAR 界面

如上图本文叙述的智能车系统采用 IAR Embedded Workbench for ARM 软件编译，J-LINK 下载调试。

## 5.2 调试工具

在本系统的完成中，所使用到了以下四种调试组合工具：

### (1) IAR Embedded Workbench for ARM 在线调试



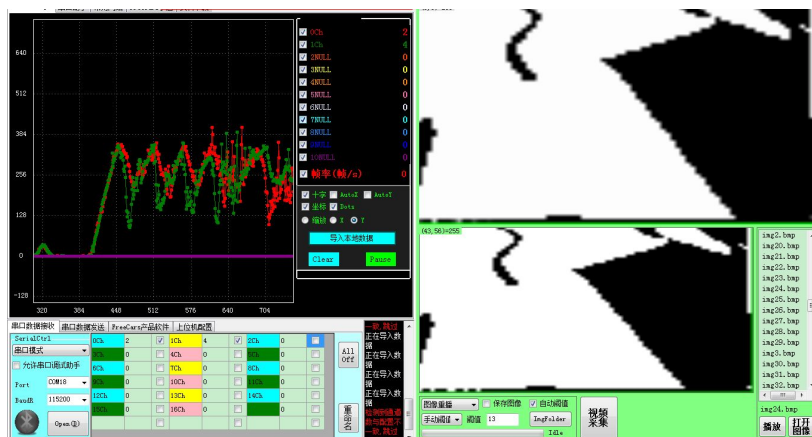
4-10 IAR Embedded Workbench for ARM 调试界面

IAR 是最基本的调试工具，前期可以通过 IAR 的在线调试窗口观察各变量的值，可以发现程序中的各种问题，利用好 IAR 的在线调试可以帮组我们在程序编写过程中事半功倍。熟练使用这个最基础的调试工具是做智能车最基本的要求。

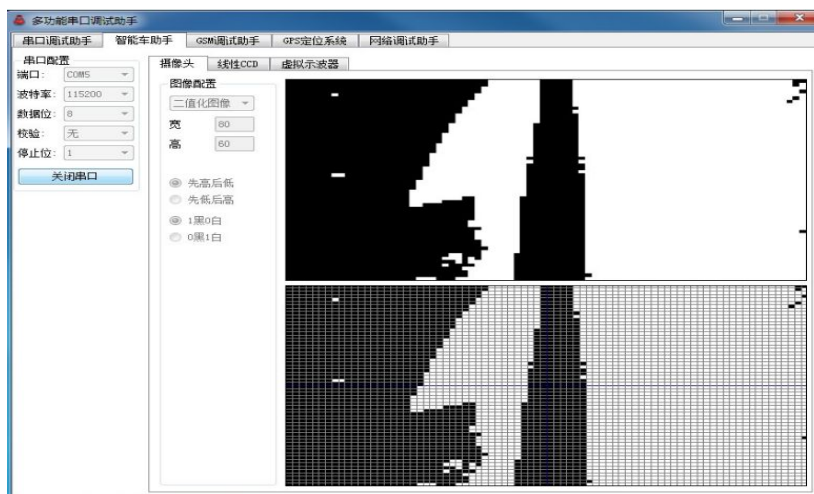
### (2) 上位机的使用

这套调试工具可以使我们在调试的过程中，直接将信息通过蓝牙发送到电脑的上位机中，这样我们调试的时候就可以实时查看我们所需要的信息，但需要注意的是有蓝牙发送数据有的数据的速度有限，且会因为数据的长度、发送模块和接受模块的距离不同时间也不相同，而对于摄像头组的数据量比较大，我们一般只在前期调试用来观察图像，后期主要利用串口观察数据和在线调参而不看图像，且在一次观察数据时我们尽可能一次只观察少量的数据以免影响摄像头数据的采集和处理。





4-11 上位机 1



4-12 上位机 2

### (3) 液晶调试

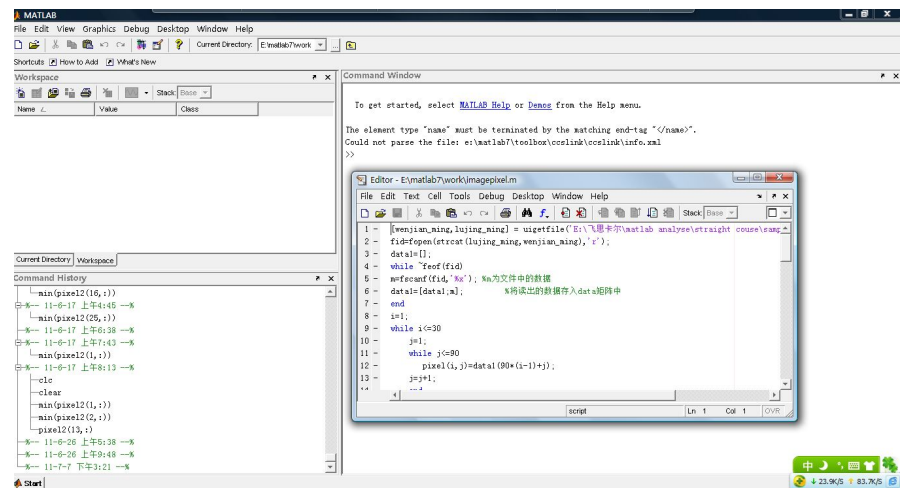


4-13 液晶调试窗口

液晶的实时性好，用液晶加上按键来调试既不会影响摄像头程序的执行，又可以较为方便的改变参数的值，我们采用液晶模块如上图。利用液晶可以随时显示摄像头采集的图像，用实时监视各个变量的变化，帮助我们查找问题

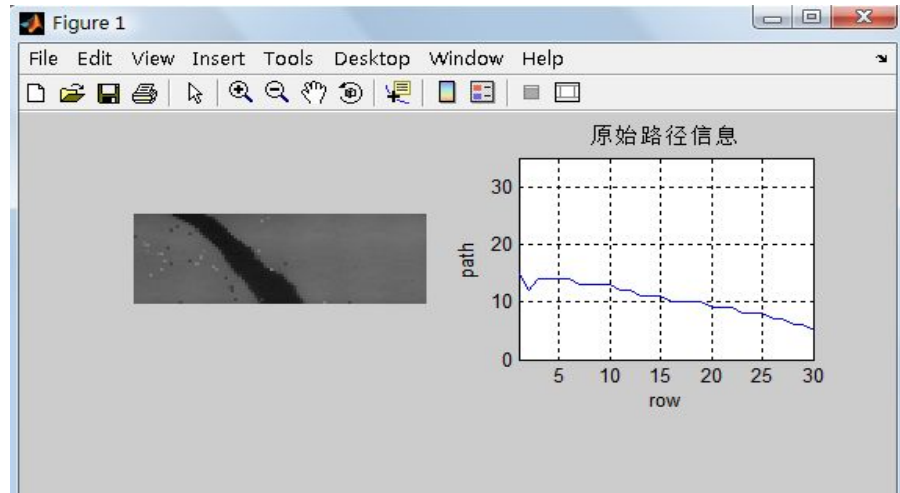
(4)MATLAB 组合调试工具

我们在程序调试过程中也运用到了MATLAB 7.0进行算法辅助分析。下面是关于一些MATLAB 7.0的界面介绍。



4-14 MATLAB 分析界面

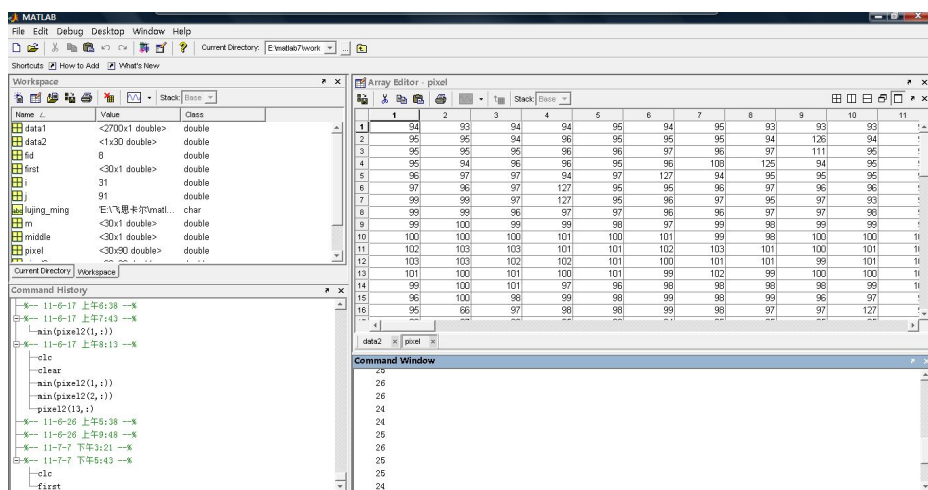
我们在使用 MATLAB 7.0 时运用了它强大的图形表达功能，下面是我们利用 MATLAB 7.0 图像处理功能还原的中心引导线图。



4-15 中心引导线

我们也利用了 MATLAB 7.0 的外部数据导入功能，和串口联合使用。微控制器执行的程序中间结果通过串口接收，再导入 MATLAB 进行观测和验证。如下图 4-16 所示为用 MATLAB 观察一些程序中的变量。





## 第六章 全文总结

### 6.1 智能小车的主要参数

项目	参数
路径检测方法（赛题组）	摄像头
车模几何尺寸（长、宽、高）（毫米）	270mm、160mm、370mm
车模轴距/轮距（毫米）	140
车模平均电流（匀速行驶）（毫安）	3000
电路电容总量（微法）	1504
传感器种类及个数	8 个（1 个面阵 CCD，2 个编码器，2 个光电对管，2 个红外接收管，1 个陀螺仪）
新增加伺服电机个数	0
赛道信息检测空间精度（毫米）	30
赛道信息检测频率（次/秒）	50
主要集成电路种类/数量	MCU/1    BTN/4    245/1 AS1015/1    2940/3    2937/1
车模重量（带有电池）（千克）	1.3

## 6.2 总结

开始做智能车到现在已有一年的时间，历经多少每位做智能车的同学应该是感同身受，在反复的摸索中，总结出几点经验和不足，给读者一些启发。

(1) 程序的编写需要一定的逻辑，我们在着手之前需要先有一个框架，然后在框架填入一个个赛道元素的处理方式，个个突破。

(2) 一辆好的智能车离不开一个合适机械结构，在这里我认为没有所谓最好的机械或者最好的算法，只有最合适的机械和算法，只有当两者能融合的最恰当时，车才能发挥出他最大潜能。

(3) 注重细节，在小车遇到问题时，我们最常用也是最有效的方法，无非排除法和控制变量法，当然听起来这些方法我耳熟能详，但实践中我们如果能很好的运用，会给我们设计带来很大效率，前提是读者要熟悉自己车的每一个细节，每一句代码。所以我们学会独立思考研究，智能车系统虽大，也要我们自己一砖一瓦盖起来。

通过全队的努力，所设计的智能车已通过调试，运行正确、稳定，其平均速度可以达到理想状态，所设计智能车系统稳定、快速地跑完赛道全程。就目前的系统而言，还是存在着需要改进的地方：

(1) 赛车的机械结构还有待调整。和有些队伍相比，我们赛车的转弯还不够灵活。在这方面需要继续研究，争取把小车的状态调整到最优。

(2) 光照对小车的的影响较大，对光照条件不好的情况要有相应的应对策略。

(3) 对于舵机的控制算法上还有待改进。

接下来，我们会昂首向前，突破再突破，向着全国总决赛大步迈进。

## 参考文献

1. 卓晴, 黄开胜, 邵贝贝. 学做智能车——挑战“飞思卡尔”杯[M]. 北京: 北京航空航天大学出版社, 2007.
2. 俞雪婷. 飞思卡尔智能车系统的设计与实现[D]. 沈阳: 东北大学, 2008.
3. 潘丰厚. 基于飞思卡尔 XS128 微控制器的智能车系统的研究与设计[D]. 沈阳: 东北大学, 2010.
4. 卓晴. 智能汽车自动控制器方案设计[D]. 清华大学自动化系, 2006.

## 附录：

### 主程序源代码

```
/*
 * @file      main.c
 * @date      2015-07-18
 * author     风火队
 */
#include "common.h"
#include "include.h"
#include "init.h"
#include "tx_process.h"

extern uint16_t val_z, val_y;
uint8_t _lms_flag=0, heixian_flag_0=1;    //lms 时间标志
int8_t tingche_flag=0;
/*
void PORTA_IRQHandler();
void DMA0_IRQHandler();
void sending(uint8_t *imgaddr, uint32_t imgsize);
void img_extract(uint8_t *dst, uint8_t *src, uint32_t srclen);
void PIT0_IRQHandler();
void PIT1_IRQHandler();
uint8_t imgbuff[CAMERA_SIZE];    //定义存储接收图像的数组
uint8_t img[CAMERA_H][CAMERA_W];
extern int32_t angle_pwm;
void main(void)
{
    NVIC_SetPriorityGrouping(3);    //设置优先级分组, 4bit 抢占优先级, 没有亚优先级
    NVIC_SetPriority(PORTA_IRQn, 0);    //配置优先级
    NVIC_SetPriority(PIT0_IRQn, 1);    //配置优先级
    NVIC_SetPriority(DMA0_IRQn, 2);    //配置优先级
    NVIC_SetPriority(PIT1_IRQn, 3);    //配置优先级
    FTM_init();
    Site_t site = {0, 0};    //显示图像左上角位置
    Size_t imgsize = {CAMERA_W, CAMERA_H};    //图像大小
    Size_t size;
    LCD_init();
    size.H = CAMERA_H;
    size.W = CAMERA_W;
    camera_init(imgbuff);    //配置中断复位函数
    set_vector_handler(PORTA_VECTORn, PORTA_IRQHandler);
    set_vector_handler(DMA0_VECTORn, DMA0_IRQHandler);
}
*/
```

```

pit_init_ms(PIT0, 4);
set_vector_handler(PIT0_VECTORn, PIT0_IRQHandler);
enable_irq(PIT0_IRQn);
/***** 按键消息 初始化 *****/
key_event_init();
pit_init_ms(PIT1, 10);
set_vector_handler(PIT1_VECTORn, PIT1_IRQHandler);
enable_irq(PIT1_IRQn);
/***** 按键消息 初始化 *****/
while(1)
{
    camera_get_img();          //9MS 摄像头获取图像
    if (PTA12_IN==0)
    {
        if( CAMERA_COLOR == 0)
            LCD_Img_Binary_Z(site, size, imgbuff, imgsize);
    }
    img_extract((uint8*)img, (uint8*)imgbuff, CAMERA_SIZE);
    PROCESS();
    deal_key_event();
}
}

```