

第五届飞思卡尔杯全国大学生 智能汽车竞赛

技术报告

学 校： 华中科技大学
队伍名称： 华中科技大学五队
参赛队员： 方华启
张江汉
诸金良
带队教师： 何顶新
罗惠

关于技术报告和学术论文使用授权的说明

本人完全了解第五届全国大学生“飞思卡尔”杯智能汽车竞赛关保留、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

第 1 章 引言	1
1.1 概述	1
1.2 全文安排	2
第 2 章 电路设计	3
2.1 电路系统框图	3
2.2 电源部分	4
2.3 电机驱动部分	5
2.4 电磁传感器	6
第 3 章 机械设计	8
3.1 车体结构和主要参数及其调整	8
3.2 舵机的固定	10
3.3 传感器的固定	11
3.4 编码器的固定	11
第 4 章 软件设计	12
4.1 程序整体框架	12
4.2 前台系统	13
4.3 后台系统	13
4.4 软件详细设计	14
第 5 章 调试	15
第 6 章 全文总结	16
6.1 智能车主要技术参数	16
6.2 不足与改进	16
6.3 致谢与总结	17

参考文献.....	18
附录 A 源代码	18

第1章 引言

教育部为了加强大学生实践、创新能力和团队精神的培养，在已举办全国大学生数学建模、电子设计、机械设计、结构设计等4大竞赛的基础上，委托教育部高等学校自动化专业教学指导分委员会主办每年一度的全国大学生智能汽车竞赛（教高司函[2005]201号文）^[1]。

为响应教育部的号召，本校积极组队参加第五届“飞思卡尔”杯全国大学生智能汽车竞赛。从2009年12月开始着手进行准备，历时近8个月，经过设计理念的不断进步，制作精度的不断提高，经历2代智能车硬件平台及相关算法的改进，最终设计出一套完整的智能车开发、调试平台。作为电磁组的华中科技大学五队采用轻质量机械设计、大前瞻传感器和连续化算法处理的基本技术路线，在前瞻距离、噪声抑制、驱动优化、整车布局等方面加强研究创新，在有限计算能力下获得了较高的赛道信息准确率。使智能车能够满足高速运行下的动力性和稳定性需求，获得了良好的综合性能和赛场表现。

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。

1.1 概述

1.1.1 电路设计

飞思卡尔电磁组智能汽车硬件主要分为主控板，传感器板。本车在主控板上主要特色为电机使用H桥驱动，从性能和扩展性上优于集成驱动器方案。传感器板设计着重考虑提高传感器的前瞻量和信号的抗干扰能力。

1.1.2 机械设计

机械方面，主要是对舵机的安装进行了研究，加长了舵机的连杆，以增加反应速度。另外，主要研究车差速性能的研究以及传感器支架的固定。

1.1.3 控制程序设计

一方面使用免费的 μ COS操作系统，这给智能车的整体调试提供了很多方便；另一方面，在大前瞻传感器的基础上设计出合理的舵机、电机控制算法，在满足稳定性要求的基础上提高速度。

1.1.4 调试平台

为了提高调试效率，并在制作的过程中减少控制决策的工作量，在 MATLAB 平台的基础上，设计了相应的数据调试工具，方便对采集的数据进行处理，并对小车的运行参数进行实时监控，达到了事半功倍的效果。

1.1.5 预赛成绩

由于华南赛区比赛赛道摩擦力与学校调试环境相差较大，但从整体稳定性和速度上在华南赛区靠前，并以华南赛区第 3 名的成绩获得决赛资格。以下是参数对比：

环境	本校调试赛道	华南赛区赛道
速度	约2.2米/秒	约2.0米/秒
稳定性	很稳定	很稳定
抖动	不抖动	基本不抖动

1.2 全文安排

本文在第一章中简要介绍智能车竞赛的情况。

本文在第二章中详细说明电路部分的设计。

本文在第三章中详细说明机械部分的改装。

本文在第四章中详细说明软件部分的实现。

本文在第五章中详细说明调试平台部分的设计。

本文在第六章中做全文总结。

由于作者水平有限，报告中的错误和不足之处，希望各位读者能及时批评和指证，并提出建议和意见。

第2章 电路设计

智能车的电路设计在整个智能车的设计过程中占有相当重要的分量, 稳定合理的电路是智能车高速运动的必要条件。本文智能车硬件的设计遵循电路简单、功能强大、方便调试等原则, 设计了集成度很高的主控制电路板, 整合了除传感器以外的所有电路。

2.1 电路系统框图

2.1.1 电源部分

电源部分主要包括: 电源开关保护指示、电源管理、主控制板和数字传感器板电源、模拟传感器板电源、舵机电源、电机驱动。电源部分功能结构如下:

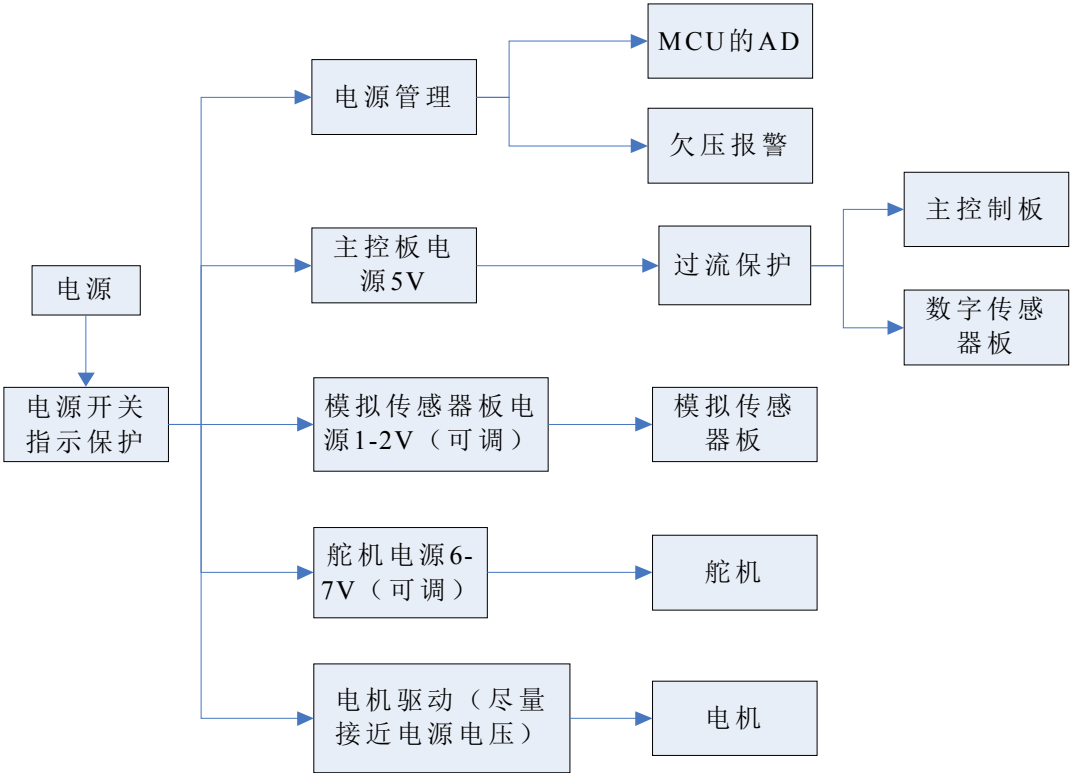


图 2.1 电源部分系统框图

2.1.2 MCU 部分

MCU 最小系统部分主要包括：MCU 复位、通讯、人机交互、与外围器件接口，结构框图如下：

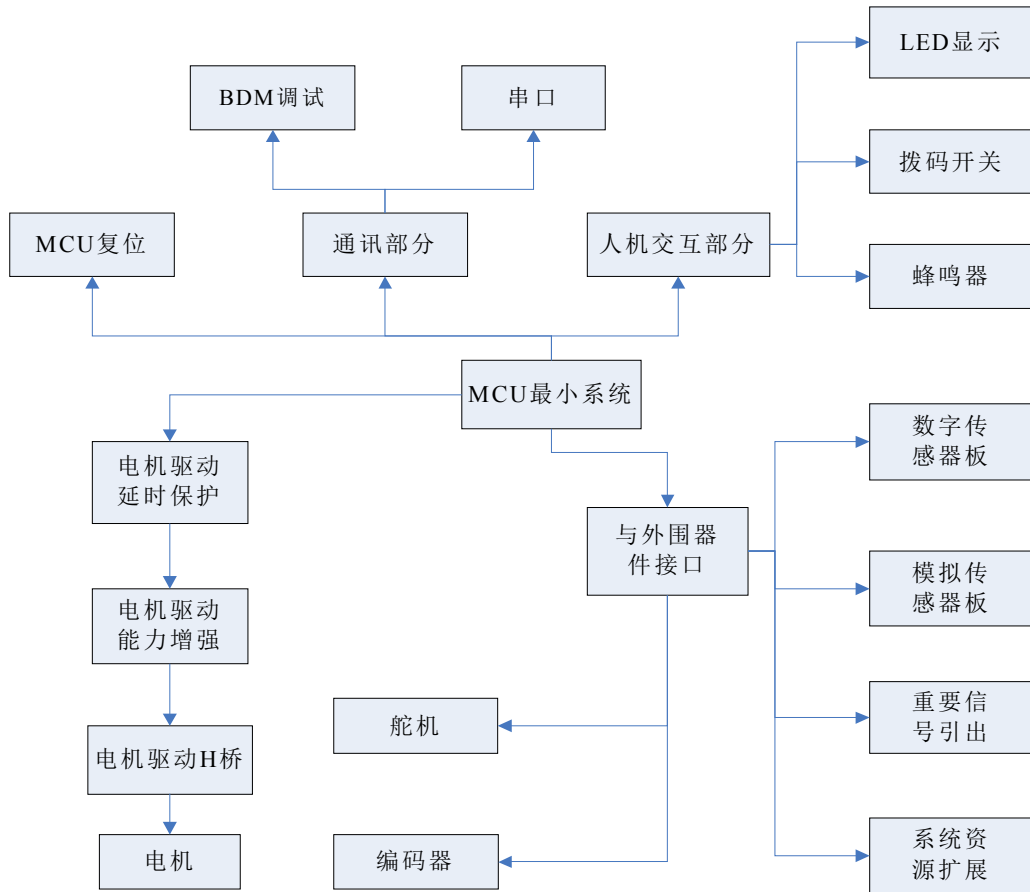


图 2.2 MCU 部分系统框图

2.2 电源部分

为了方便了解系统电源的工作情况，故需要加入电源监视电路即时提醒注意电池电压状况。而且电源管理部分体积、功耗都不能太大，不能影响电路的正常工作。

2.2.1 电源开关保护指示模块

由于智能车使用的直流电源，故一旦电池的正负极接反将造成很严重的后果，很有可能烧坏主控制板，故考虑在电源的最开始部分设计防插反的结构和

插反后的保护电路。一般来说，二极管具有良好的单向导电性，故可考虑用二极管作为反接保护器件。

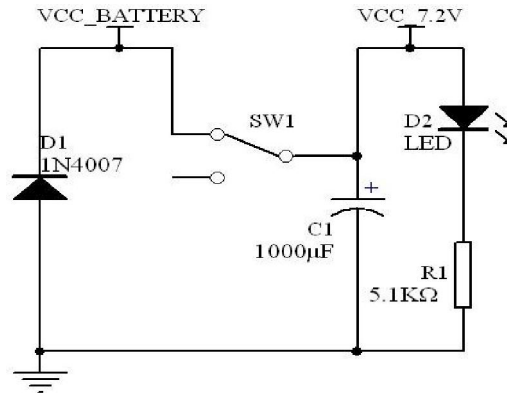


图 2.3 电源开关指示保护电路原理图

2.2.2 主控制板和数字传感器板电源模块

由于主控制板和数字传感器板两部分都需要 5V 直流稳压，为简便故两块板子共用一个稳压芯片。根据分析可得：

$$U=+5.0V\pm 5\% (4.75V-5.25V), I \approx 600mA。$$

故 LM2940-5V 可以作为合适的稳压芯片选择。

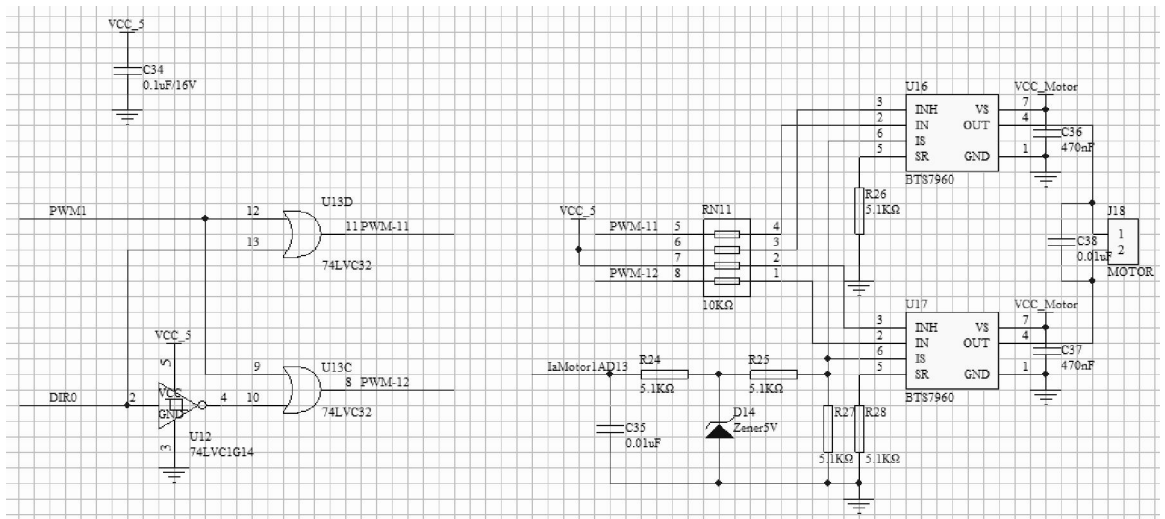
2.2.3 舵机和模拟传感器板电源模块

由于舵机和模拟传感器需要的电源电压都要求尽量能接近于电池所提供的电压，但是电压太高又会对电路稳定性带来一些影响，故需要选择稳压值可调的稳压芯片

根据需求，结合实际使用经验，本文认为 LM1117-ADJ 为比较合适的选型。

2.3 电机驱动部分

本文研究的重点是如何给电机提供尽量大的驱动电压（尽量接近电源电压）和电流，以及尽量减少驱动环节的能量损耗。



2.4 电磁传感器

2.4.1 双水平线圈方案

单独一个线圈感应电动势输出电压 V_{ox} 是位置 x 的偶函数，只能反映到水平位置的绝对值 x 的大小，无法分辨左右。为此，可以使用相距长度为 L 的两个线圈电感，计算两个电感感应电动势的差值：

$$E_d = E_1 - E_2 = \frac{h}{h^2 + x^2} - \frac{h}{h^2 + (x-L)^2} \quad (3.23)$$

令 $L = 30cm$ ，计算得到两个线圈电动势差值 E_d 如下图所示：

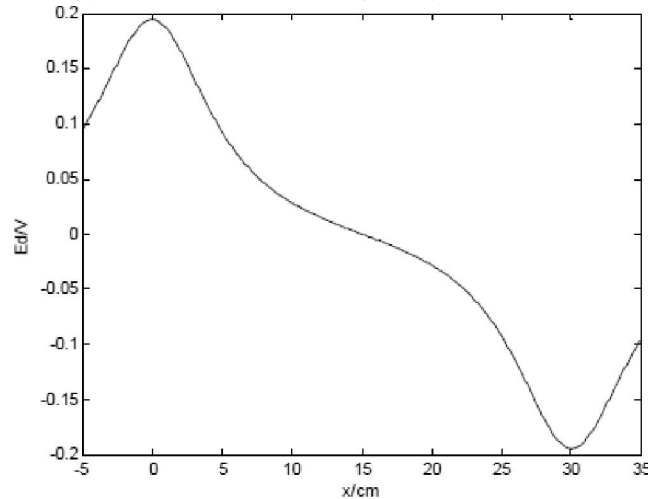


图 3.22 双水平线圈电压差值与横向位移的关系

由图可知，在位移 $0\sim 30\text{cm}$ 内，电动势差值 E_d 与位移 x 关系是一个单调函数。可以使用这个量对于小车转向进行负反馈控制，从而保证两个线圈的中心位置跟踪赛道的中心线。通过改变线圈高度 h 、线圈之间距离 L 可以调整位置检测范围以及感应电动势的大小。

2.4.2 传感器布局

双水平线圈方案是最简单的路径检测方案，要想达到较好的巡线竞速效果，仅仅两个传感器的分辨率显然是不够的。一种改进方案是增加传感器数量，采用一字型布局。

布局之前应分析每个传感器的检测范围。车体中心线完全平行于赛道中心导线时，归一化后每个传感器检测范围的测量结果见下图：

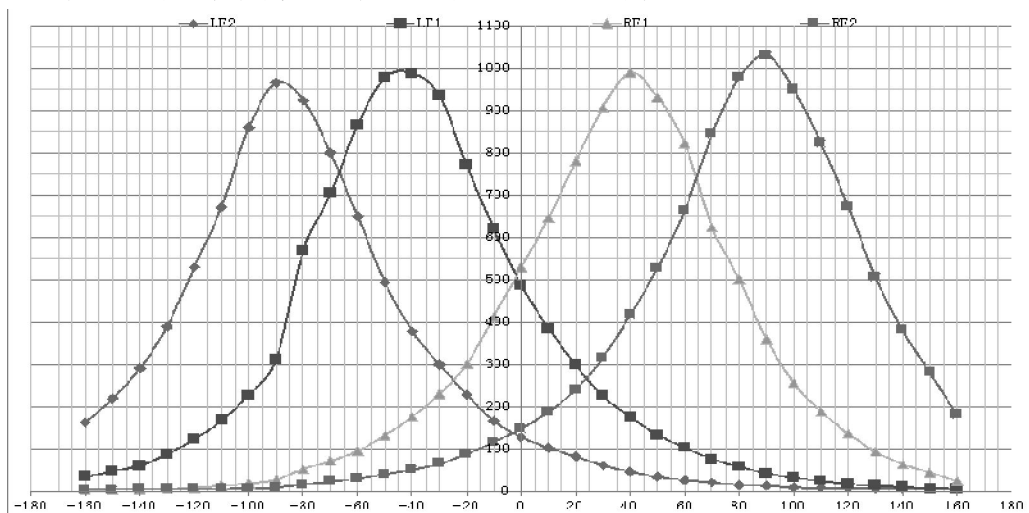


图 3.23 微控器采集的传感器信号分布

可见每个传感器检测范围约为 $\pm 14\text{cm}$ 。但取 14cm 间距是远远不够的，因为水平线圈方案检测的是漆包线同心圆磁场的水平分量，具有方向选择性，如果智能车转向过程中传感器序列与漆包线成的角度较大，则每个传感器检测范围会减小很多，因此传感器间距需要尽量小。不过，传感器间距受到同轴向线圈互感的限制，不能太小。实际测得 3cm 时传感器之间影响较小。

第3章 机械设计

3.1 车体结构和主要参数及其调整

智能车竞赛所使用的车模是一款带有差速器的后轮驱动模型赛车，它由大赛组委会统一提供。

表 3.1 智能车模主要参数

基本参数	尺寸
轴距	200mm
前轮距	136mm
后轮距	138mm/146mm
车轮直径	52mm
传动比	18/76

组委会要求智能车最终尺寸不得大于 400mm*250mm。

3.1.1 前轮定位

1) 主销后倾角

主销向后倾斜，主销轴线与地面垂直线在赛车纵向平面内的夹角称为主销后倾角。

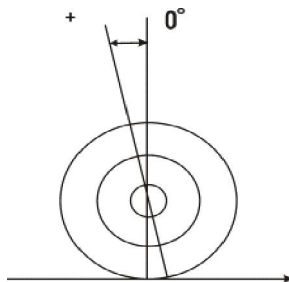


图 3.1 主销后倾角示意图

主销后倾角定义：上球头或支柱顶端与下球头的连线（转向时，车轮围绕其进行转向运动的转向轴）向前或后倾斜的角度，向前倾称为负主销后倾角，向后倾斜称为正主销后倾角。其功能影响转向稳定性及转向后车轮的自动回正力。对于智能汽车模型，通过改变前后黄色垫片的数量可以改变它的主销后倾角。

2) 主销内倾角

主销在横向平面内向内倾斜，主销轴线与地面垂直线在赛车横向断面内的夹角称为主销内倾角。

主销内倾角也有使轮胎自动回正的作用，但主销内倾角不宜过大，否则在转弯时轮胎将与赛道间产生较大的滑动，从而会增加轮胎与路面间的摩擦阻力，使转向变得沉重，同时会加速轮胎的磨损。

3) 前轮外倾角

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，称为前轮外倾角。

前轮外倾角是前轮的上端向外倾斜的角度，如果前面两个轮子呈现“V”字形则称正倾角，呈现“八”字则称负倾角。由于前轮外倾可以抵消由于车的重力使车轮向内倾斜的趋势，减少赛车机件的磨损与负重，所以赛车安装了组委会配备的配件。

3.1.2 车体重心

重心调整主要包括重心高度和前后位置的调整。

理论上，车体重心越低稳定性越好，重心低有利于赛车在高速转弯的贴地性，可以有效防止侧翻，因此在车体底盘高度、舵机安装、电路板的安装等上尽量使重心放低。

车体重心前后方向的调整对赛车的行驶性能也有很大的影响。本赛车后轮驱动，驱动力的大小直接跟轮胎与地面的附着力有关，而附着力与路面附着系数和驱动轴的轴荷有关，驱动轴的轴荷又取决于重心的水平位置，如果仅从这方面考虑，重心应靠近后轴。

根据车辆运动学理论，车身重心前移，大部分重量压在前轮，转向负荷增大，会增加转向，对模型车的制动性能和操纵稳定性有益，但降低转向的灵敏度，同时降低后轮的抓地力；重心后移，会减少转向，但增大转向灵敏度，后轮抓地力也会增加。但综合起来看，重心应靠近后轴一点。

3.1.3 离地高度及底盘刚度调整

1) 底盘高度的调整

车模底盘的高度主要由赛道中的坡决定，在顺利过坡的前提下底盘越低越好。

2) 后悬挂纵向减震刚度

适当增大底盘的刚度有利于车体走直线的稳定性，可通过增加垫片来增大弹簧的预紧力，另外还可以通过调整弹簧的另一个支点的位置来改变预紧力，从而提高底盘的刚度。

3.1.4 差速机构的调整

差速机构的作用是在车模转弯的时候，降低后轮与地面之间的滑动，还可以保证在轮胎抱死的情况下不会损害到电机。当车辆正常的过弯时（假设：无转向不足亦无转向过度），4个轮子的转速(轮速)皆不相同。

差速器的特性是：阻力越大的一侧，驱动齿轮的转速越低；而阻力越小的一侧，则相反。

差速器的调整中要注意滚珠轮盘间的间隙，过松过紧都会使差速器性能降低，转弯时阻力小的车轮会打滑，从而影响车模的过弯性能。好的差速机构，在电机不转的情况下，右轮向前转过的角度与左轮向后转过的角度之间误差很小，不会有迟滞或者过转动情况发生。

3.2 舵机的固定

舵机有竖直和水平放置两种方案，竖直放置可以适当的降低重心，但安装较为麻烦，水平放置固定方式简单，但重心升高，故先采用竖直放置方式。

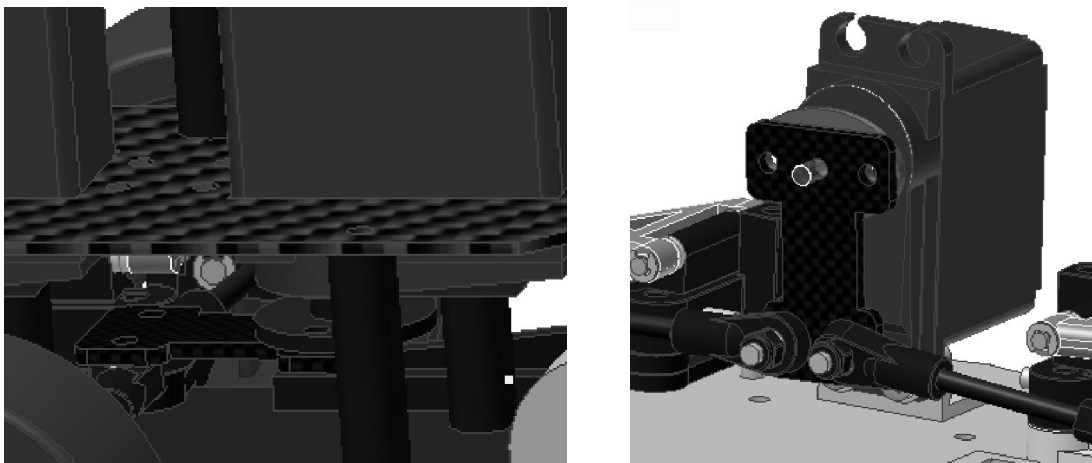


图 3.2 舵机安装 3D 图

3.3 传感器的固定

传感器应尽量伸出车体以获得足够的前瞻量。

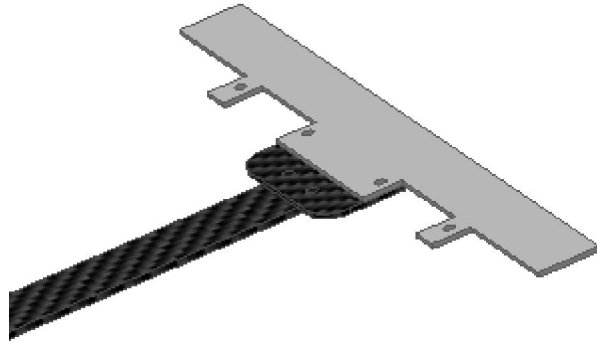


图 3.3 传感器板固定 3D 图

3.4 编码器的固定

电磁编码器的安装精度较高，要求编码器轴与赛车后轴同轴，通过齿轮与赛车差速器相联。本文用组委会提供的电机上自带的齿轮进行改装，因为该齿轮不仅模数与差速器齿轮同，而且反馈回来的是电机的实际转速。编码器直接固定于赛车尾部。

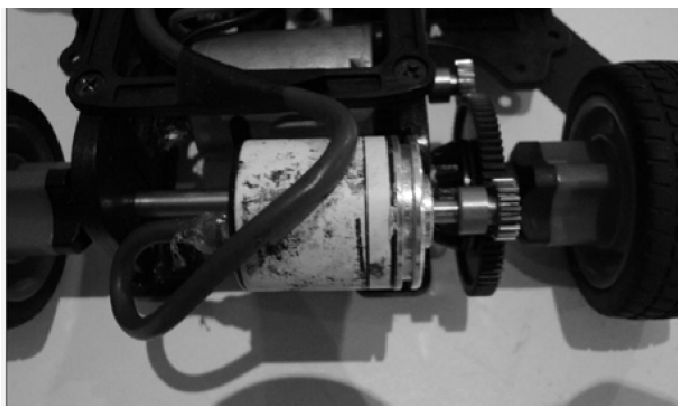


图 3.4 编码器固定图

第4章 软件设计

4.1 程序整体框架

整个系统由飞思卡尔 H12 系列 16 位单片机为主控 MCU，对模拟和数字两种传感器信号进行采集和处理，并对输出的两路 PWM 分别控制，进行舵机和电机的控制，一路 PWM 控制舵机来决定小车的角度，另一路 PWM 通过硬件电路驱动来控制电机，决定小车的速度。另外使用编码器采集速度值，作为辅助速度控制。

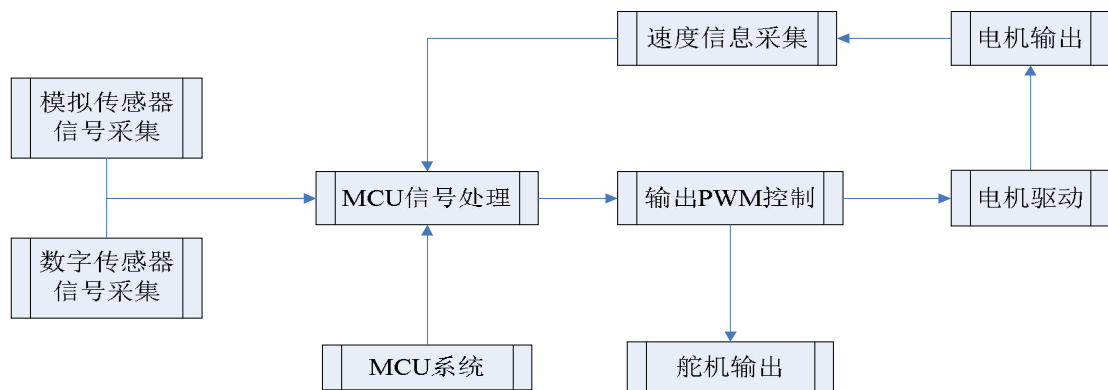


图 4.1 程序整体框架图

系统输入信号包括传感器，编码器。输出由舵机和电机组成。信号的输入输出都使用后台的实时控制来实现，前台只是将得到的输入信号进行处理，决定电机和舵机的输出。

4.2 前台系统

4.2.1 主函数流程

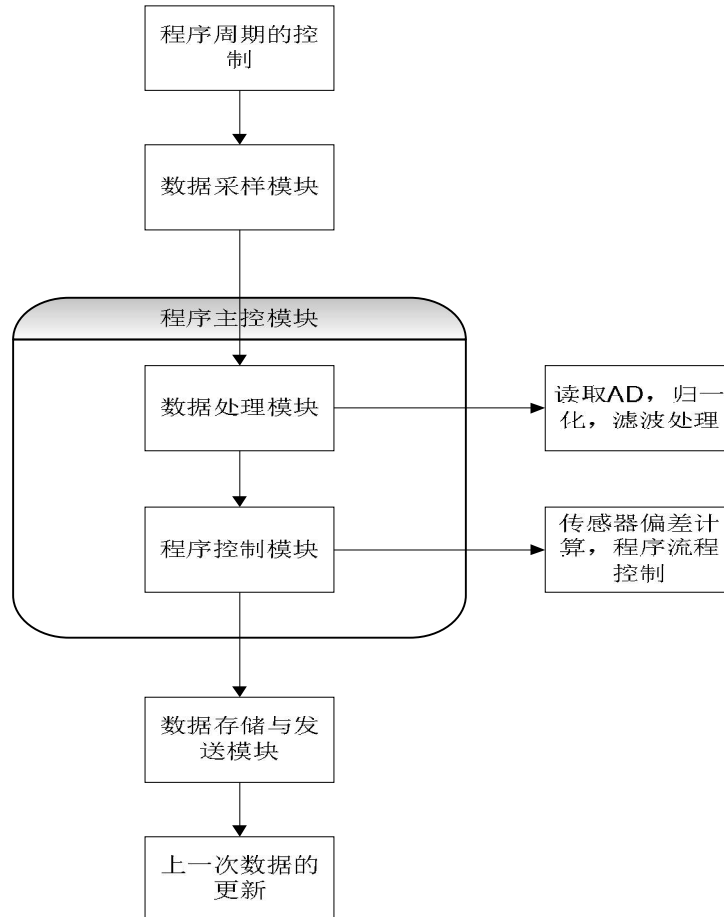


图 4.2 主程序流程图

4.2.2 流程说明

始化系统后打开中断，其实是把可屏蔽中断的中断标志位清零。

开机初始化成功，并且电池还有电，就开始标定传感器，注意标定时一定要传感器正交过漆包线。标定完毕后用户可选择手动调试模式和串口调试模式，以及运行模式。一旦进入了运行模式，就不能进入调试模式了。在运行模式中，如果检测到保护，就可以直接不需要标定直接通过拨码开关来控制实现方法。

4.3 后台系统

后台系统为 1 毫秒中断，做实时控制，设定每隔一定时间进行信号采样和硬件的输出。

假设每隔 10 毫秒操作一次舵机，设定宏 SERVO_TIME 为 10，定义静态局部变量 servorCounter，没次中断 servorCounter 加 1，计数到 SERVO_TIME 时，操作一次寄存器，然后清零 servorCounter，即可做到 5 毫秒的实时控制，其他实时操作类似，这样要求所有的实时控制时间间隔为整数毫秒，如果不是，需要再开一个中断。

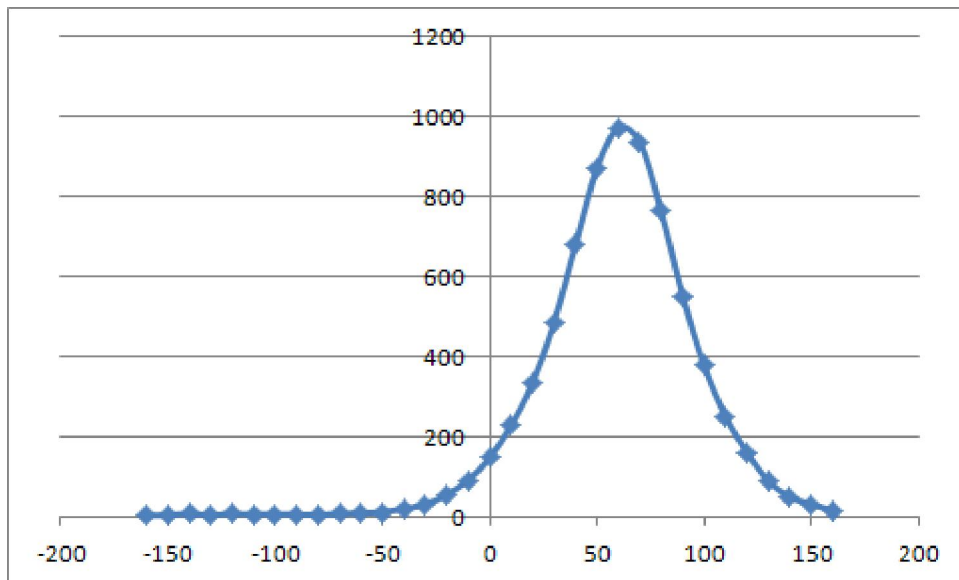
1ms 中断中主要实现以下功能：

舵机 PWM 输出，电机 PWM 输出，编码器采样(计算速度与距离，为了便于计算将采样时间设置为了 9ms)，蜂鸣器按指定周期鸣叫(周期可调，1ms 为基准)。遥控器命令检测，主要用于急停车，起到保护车的作用。

4.4 软件详细设计

4.4.1 信号处理

信号处理部分主要采用了组委会推荐的偏差算法，可以测出单个传感器曲线见下：



4.4.2 速度控制

通过阅读前几届的技术报告发现采用增量式 PID 进行控速是一个比较好的方案，增量型 PID 其与位置型 PID 算法相比，具有以下优点：增量型算法不需要做累加，增量的确定仅与最近几次偏差采样值有关，计算精度对控制量的计算影响较小，而位置型算法要用到过去偏差的累加值，容易产生大的累加误差；

增量型算法得出的是控制量的增量，而位置型算法的输出是控制量的全量输出，误动作影响大；采用增量型算法，易于实现手动到自动的无冲击切换。为了更快的响应，采用了 BangBang 控制与增量式 PID 结合的算法。

4.4.3 舵机控制

利用单排传感器进行传感器的定位，实现该函数的思路是根据传感器的偏差，进行分段线性给出舵机角度。首先计算出理论的角度，方法是测量舵机轴到传感器板的长度 L ，假设计算出传感器的距离赛道漆包线的偏差是 D ，即对应的角度为 $\alpha = \arctan(D/L)$ ，由于机械上限制了舵机的转角不能过大，现在最大就是左右各 40 度的转角，因计算出来的角度公式以及给角只能作为参考。但是线性分界点可以用来借鉴一下。

第5章 调试

程序的基本框架建立起来以后，主要要做的就是进行 PID 参数的整定，具体的整定方法可以参见《过程控制》与《微机控制技术》，在整个调试过程中可能会遇到很多问题，下面列举一下我们遇到的问题

- 1) 传感器放在同一位置，AD 采集回来的值在一定范围内抖动非常厉害这就需要程序进行相应的滤波处理，这个是不容忽视的。
- 2) 每个传感器可能都不一样，从程序上我们可以采用标定来实现，当然如果相差太远就最好是从电路上解决
- 3) 传感器的布局应该考虑到传感器之间存在互感，因此传感器的间距不能过小，我们测试发现大概间距大于 3cm，互感影响会很小。
- 4) 在实际调试过程中我们不能仅仅只参考车的姿态来进行调试，我们要使用调试工具将数据发送上来进行分析处理。

第6章 全文总结

6.1 智能车主要技术参数

表 6.1 智能车主要技术参数

项目	参数
路径检测方法（赛题组）	电磁传感器
车模几何尺寸（长、宽、高）（毫米）	392*245*225
车模轴距/轮距（毫米）	200/前轮距 136/后轮距 138
车模平均电流（匀速行驶）（毫安）	3000
电路电容总量（微法）	1150
传感器种类及个数	15
新增加伺服电机个数	0
赛道信息检测空间精度（毫米）	10
赛道信息检测频率（次/秒）	200
主要集成电路种类/数量	电机驱动/2，舵机驱动/1
车模重量（带有电池）（千克）	1.20

6.2 不足与改进

- 1) 机械改装和支架设计中，其强度和稳定性需要更加注意，以免造成支架容易撞断和运行不稳定的情况；
- 2) 硬件设计的兼容性和易扩展性考虑，在电路设计的过程中，端口分配是一个很重要的步骤，设计时需要预留一些可能用到的资源，以备后面增加是方便扩展；
- 3) 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；
- 4) 在备赛过程中，对于学校内部应加强不同组别和不同传感器方案队间的交流，同时应加强与其他兄弟院校的交流，使智能车往更好的方向上发展。

6.3 致谢与总结

经过半年多的设计、制作和调试，在队员们的分工和配合下，最终顺利完成了智能车电路、机械、程序等方面的工作。在这个过程中，大家学到了很多，积累了很多，能够一步一步走下来，靠的是整个团队的协作和团结。总体而言，大家学到了很多，提高了很多，这将对以后的学习和工作都会起到很大的作用。

作为一个融合多学科交叉的复杂系统，完成整智能车的设计、制作和调试是一个非常庞大的工程，仅靠几名队员是很难完成的，它需要一个高效运作、规范管理的团队的紧密合作才能完成。本文所论述的内容不仅仅是作者个人工作的结晶，同时也蕴含了他人的劳动、汗水和智慧。在本文即将结束之际，特向整个智能车团队的成员以及其他提供过帮助、建议和意见的人表示衷心的感谢。

感谢指导老师的指导、支持和帮助。没有他的指导和帮助，整个智能车队就很难高效稳定运行下去，设计的方向也很难把握得这么准确。

感谢华中科技大学启明学院和控制科学与工程系在调试场地，物资上的支持和帮助，正是有了学校的大力支持，才使我们能专心做事，仔细研究。

还要感谢教育部，自动化分教指委给大家提供的这个锻炼团队合作和创新能力的竞赛平台，以及清华大学和本届竞赛中给大家提供赛场的湖南大学、杭州电子科技大学以及其他赛区承办学校，感谢各个院校为大家提供优良的竞赛环境，使大家能够在竞赛中发挥出高水平。

参考文献

- [1] 卓晴 黄开胜 邵贝贝 等.学做智能车. 北京:北京航空航天大学出版社,2007年3月第1版
- [2] 第四届全国大学生“飞思卡尔”杯智能汽车竞赛秘书处, 竞速比赛规则与赛场纪律, 2008/12
- [3] National Semiconductor, LM2940 Datasheet, 1999/06
- [4] National Semiconductor, LM1117 Datasheet, 2004/06
- [5] 杨杰 吴凡.粗糙表面可见光散射特性的实验研究. 中国测试, 2009, 02期
- [6] HAMAMATSU, Light modulation photo IC S7136, 2007/07
- [7] 邵贝贝.单片机嵌入式应用的在线开发方法. 北京:清华大学出版社, 2004年10月第1版.
- [8] 邵贝贝.UCOS-II 嵌入式实时操作系统. 北京:清华大学出版社, 2002年10月第1版
- [9] MC9S12DG128B Device User Guide ,Version 1.2, Motorola,2000/05/12—2000/08/17

附录 A 源代码

因篇幅限制, 只附上舵机和电机控制部分源代码

- 1) 电机驱动


```

void InitialMotorPWM(void)
{
    DISABLE_MOTOR_PWM();           // Moto PWM Output PP'n' Immediately Disabled.

    MOTOR_PWM_HIGH_LEVEL_MODE();  // Moto PWM Channel is Set High at the Beginning of the Cycle.

    MOTOR_PWM_LEFT_ALIGNED();     // Set Moto PWM Port as Output Left Aligned.

    MOTOR_DIR_DDRROUT();          // Motor PWM direction init *****

    MOTOR_PWM_PER_REG = MOTOR_PWM_PER_1MS; // Motor PWM Period is Set at 1ms.
    MOTOR_PWM_DTY_REG = 0;         // Motor PWM Duty is Set at 0.

    ENABLE_MOTOR_PWM();           // Motor PWM Enabled.
}

void OutputMotorPWM(void)
{
    int motorTemp = 0;             // Moto Output Temp, to Set PWM Duty.

    /* PWM Value Should not be Bigger than 100 */
    if (gl_motorTemp > 100)
    {
        gl_motorTemp = 100;
    }
    else if (gl_motorTemp < -100)
    {
        gl_motorTemp = -100;
    }

    /* for "H" Bridge Drive,there is an "IO" Port to Control Motor Direction */
    if (gl_motorTemp >= 0)
    {
        MOTOR_DIR_FORWARD_TURN(); // If PWM is Positive Number, Motor Running Forward.
    }
    else
    {
        MOTOR_DIR_BACKWARD_TURN(); // Else, Motor Running Backward.
        gl_motorTemp = 0 - gl_motorTemp; // PWM Value Should always be Positive.
    }

    motorTemp = ( (100 - gl_motorTemp) * MOTOR_PWM_PRESCALE ) / 10;
    // Motor Speed is Set from 0 to 100
    // Since the MOTOR_PWM_DTY_REG(PWMDTY1) is a byte so here must use forced type_convert (unsigned char)
    MOTOR_PWM_DTY_REG = (unsigned char)motorTemp; // PWM Duty is Set here
}

```

2) 舵机驱动

第五届全国大学生智能汽车竞赛技术报告

```
void InitialServoPWM(void)
{
    DISABLE_SERVO_PWM();          // disable servo pwm

    SERVO_PWM_HIGH_LEVEL_MODE(); // All PWM Channels are High at the beginning of the Cycle.

    SERVO_PWM_LEFT_ALIGNED();    // PWM Port as Output Left Aligned.

    SERVO_PWM_PER_REG = SERVO_PWM_PER_10MS;
    SERVO_PWM_DTY_REG = gl_servoCenter;

    ENABLE_SERVO_PWM();          // Servo PWM enabled.
}
void InitialLRServoPWM(void)
{
    DISABLE_LRSERVO_PWM();        // disable servo pwm

    LRSERVO_PWM_HIGH_LEVEL_MODE(); // All PWM Channels are High at the beginning of the Cycle.

    LRSERVO_PWM_LEFT_ALIGNED();  // PWM Port as Output Left Aligned.

    LRSERVO_PWM_PER_REG = LRSERVO_PWM_PER_5MS;
    LRSERVO_PWM_DTY_REG = gl_LRservoCenter;

    ENABLE_LRSERVO_PWM();        // Servo PWM enabled.
}
```