

第十三届“恩智浦”杯全国大学生 智能汽车竞赛 技 术 报 告



学 校：皖西学院

队伍名称：皖西学院电协胖虎一队

参赛队员：周邦震

唐家成

叶 茂

带队教师：张晓东

林 华

关于技术报告和研究论文使用授权的说明

本人完全了解第十三届“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中。

参赛队员签字: 周翔 唐宝成 叶英
指导老师签字: 张昭东 林峰
日期: 18.8.20

目录

引言	1
第一章 方案设计.....	5
1.1 系统总体方案的设定	5
1.2 总体系统设计框图	6
1.3 小结	3
第二章 机械系统部分调整与优化.....	4
2.1 整车布局	4
2.2 智能车前置万向轮处理	5
2.3 智能车后轮轮胎处理方式	5
2.4 电感的安装方式及高度	6
2.5 电池的及主板安装	7
第三章 智能汽车硬件电路设计.....	8
3.1 单片机最小系统	8
3.2 主板电路设计	8
3.2.1 电源模块设计	8
3.2.2 电磁传感器信号处理电路	9
3.2.3 其他传感器电路	12
3.2.4 系统主板的 PCB 设计	12
3.3 电机驱动模块	13
第四章 智能汽车控制软件设计说明.....	14
4.1 软件开发平台.....	14
4.2 软件系统总体设计.....	14
4.3 方向控制算法设计.....	15
4.3.1 磁场信息获取与处理	15
4.3.2 巡线原理	15
4.3.3 圆环算法.....	15
4.4 控制算法	16
4.4.1 差速控制	16
4.4.2 电机速度控制.....	16
4.5 PID 控制算法	17
4.5.1 位置式 PID	18
4.5.2 增量式 PID	18
4.5.3 PID 参数整定.....	19
第五章 系统调试工具.....	20
5.1 在线调试	20
5.2 现场调试	20
5.3 调试过程	20
第六章 车模主要参数及说明.....	22
结论	23
参考文献.....	24
附录	34

引言

随着电子科技的进步，越来越多的自动化设备代替人工投入到现代生产中。而关于智能汽车的自动驾驶成为社会的焦点。人工智能发挥着越来越重要的地位，因此，智能控制策略显得尤为重要。

“恩智浦”杯全国大学生智能汽车竞赛是国家教学质量与教学改革工程资助项目，以恩智浦半导体公司生产的 16、32 位单片机为核心控制模块，通过利用道路传感器、以及自行制作的硬件电路、编写相应的程序，制造一个能自动识别路径的汽车模型。因此该竞赛是涵盖了智能控制、模式识别、传感技术、汽车电子、电气、计算机、机械等多个学科的综合性比赛，从第一届清华大学的邀请赛开始，受到全国各大高校的高度重视，对于当代的大学生知识融合和实践能力的提升，具有良好的推动作用。

本文采用第十三届“恩智浦”杯全国大学生智能汽车竞赛的 f 汽车模型作为研究平台，以 32 位单片机 S9KEAZ128AMLK 作为主控制单元，使用 IAR 软件作为开发软件进行智能控制策略研究。

硬件电路中采用 TPS7350 稳压芯片与 TPS7333 稳压芯片 5V 输出与 3.3V 输出，5V 给单片机、外设、运算放大器等模块供电，3.3V 给 OLED 供电。

单片机软件算法为本文主要论述的主要内容，运用 PID 控制算法理论对车模进行速度、方向的控制。本篇技术报告将从 F 车机械结构、硬件电路、控制算法等方面详细介绍整个准备过程。

在准备此次比赛的过程中，我们除去每天上课的时间外，其余时间都呆在实验室。这篇技术报告凝聚着我们团队的智慧，凝聚着一年的成果。

在此次比赛过程中，我认识了很多车友，在此感谢他们，齐鲁工业大学赵正阳同学，广东工业大学方泽彬同学，在竞赛期间相互讨论帮助。还要感谢大连东软信息学院某同学给的 3D 打印文件，非常实用好看。

最后感谢阜阳师范学院承办安徽赛区的竞赛，感谢厦门大学嘉庚学院作为总决赛的承办方。给予我们一个良好的竞赛环境。

第一章 方案设计

本章主要介绍团队制作智能车之初进行的方案的选定与总体的设计思路。

1.1 系统总体方案的设定

本届智能汽车竞赛，我队为电磁三轮组。三轮车模为今年新出车模，调试起来完全靠自我摸索。遵照本届竞赛规则规定，智能汽车系统采用飞思卡尔的 32 位微控制器 S9KEAZ128AMLK 单片机作为核心控制单元用于智能汽车系统的控制。在循迹传感器方面，用 10mh 电感和 6.8nf 电容谐振检测 20kHz 电流产生的交变磁场。针对本次比赛规则，设计传感器排布方式、方向控制、速度控制等。此外，我们采用蓝牙和上位机进行调试。以及板载蜂鸣器，OLED 显示，独立按键等调试工具。

硬件设计，前期查阅资料了解相关数电模电知识，自行利用手头材料采用洞洞板焊接主板、运放板，继而方便修改，在确保电路设计无任何问题基础下，再采用 altium designer15 软件设计电路板。设计中主要解决的问题列表见表 1.1。

主要解决问题	解决方法简述
赛道检测	电感电容构成谐振回路电路
车速获取	编码器
坡道	通过陀螺仪检测角度特殊性
圆环	通过传感器测回的特征值以及陀螺仪检测角度
终点线	九轴陀螺仪磁力计检测磁铁
方向控制	PD 控制
速度控制	增量式 PI 控制
调试方式	OLED 参数显示、软件开发、上位机调试、按键调试

表 1.1 设计解决的问题及方法

1.2 总体系统设计框图

智能车作为一个整体系统，包括机械、电路硬件、软件三个主要部分，三个部分相互联系，相互影响。具体实现系统框图见图 1.1。

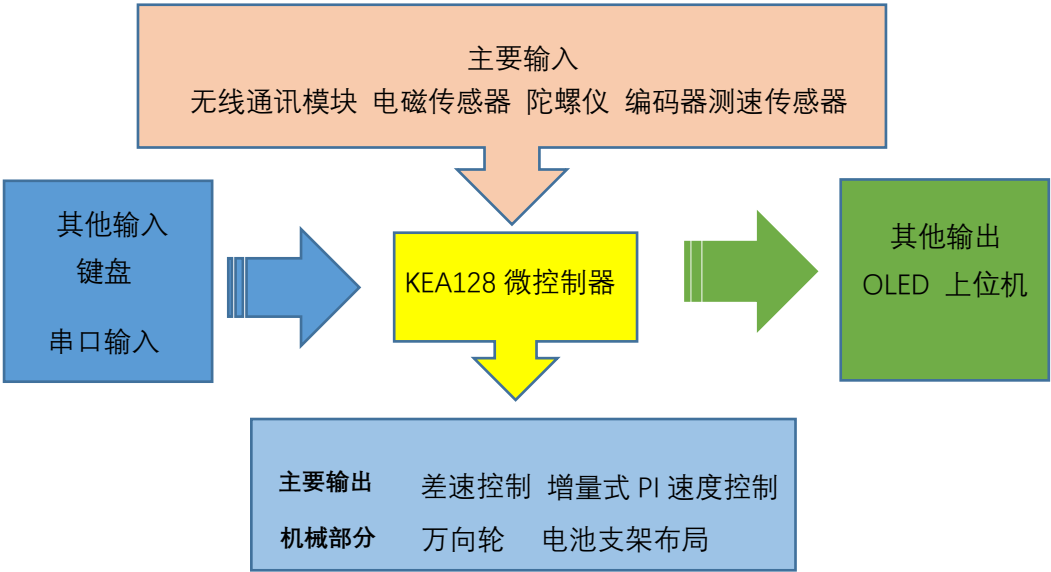


图 1.1 整体框图

根据以上系统方案设计，智能车包括七大模块：KEA128 主控模块、传感器模块、电源模块、电机驱动模块、速度检测模块、辅助调试模块。各模块的作用如下：

KEA128 主控模块，作为整个智能车的“大脑”，将处理采集的信息进行处理，根据相应的控制算法做出控制决策，驱动 380 直流电机完成对智能车的控制与运行。

传感器模块，作为智能车的“眼睛”，具有一定的前瞻性，提前感知前方的赛道信息，为智能车的“大脑”做出决策提供足够的信息和充足的反应时间。

电源模块，为整个系统提高合适稳定的电源。

电机驱动模块，驱动直流电机和伺服电机完成智能车的加减速速度控制和转向控制。

速度检测模块，检测速度反馈，用于速度的闭环控制。

辅助调试模块，主要用于智能车在整车进行调试，以及赛车状态测试等方面。

1.3 小结

本章重点分析了我们的制作智能汽车系统总体方案的选择，并介绍了系统的总体设计和要解决的问题，简要地分析了系统各模块的作用。在接下来的章节中，将对整个系统的各个模块进行详细介绍。

由于团队经验有限，且时间仓促，尽管团队努力追求完美，但是技术报告以及在智能车制作中还存在很多问题，报告中存在不恰当的或者错误的地方，欢迎大家批评指正，希望与各高校相互交流，共同提高。

第二章 机械系统部分调整与优化

决定智能车速度有多快的是软件算法，决定速度的上限是智能车的机械结构。因此在设计整个软件和算法之前一定要对整个车模的机械结构有一个充分的认识与了解，F 车模是今年新出的车模，它是由两个强劲的 380 电机，前置万向轮。由转差速形成转向，有着直立车的灵活，强劲的动力系统，小巧灵活，因此可以跑出很快的速度。

在此章节，将主要介绍智能车的机械结构和调整方案，实际调试过程中遇到的问题与解决的方式。

2.1 整车布局

此次竞赛选用博思威龙有限公司生产的智能车竞赛专用模型车（F 车模），匹配电机为双 380 电机。智能车的转向控制采用的是双电机差速转弯的方式。智能车的整体外形如图 2.1.1。

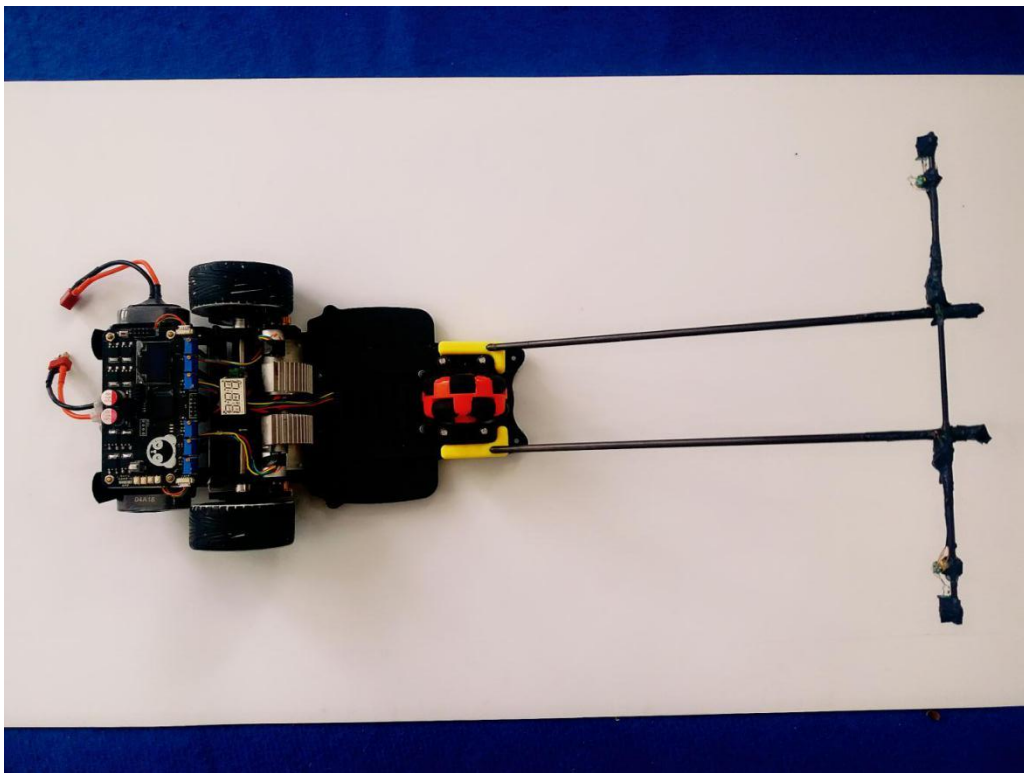


图 2.1.1 智能车整车外形

(1) 为了是车模底盘尽量放低，主板尽量放低，前置固定万向轮固定件加入垫片尽量降低重心。

- (2) 使用 3D 打印支架固定前置碳素杆，稳定，轻巧，转向灵活。
- (3) 主驱运放三合一，一体板，便于结构搭建，降低重心。
- (4) 整车重心应该在后轮轴心，增大后轮的摩擦力，防止高速入弯时甩尾现象。
- (5) 漆包线连接传感器从碳钎维杆中走线，防止比赛时损坏，同时也很美观。

2.2 智能车前置万向轮处理

由于今年加入颠簸路面，我们在调试中发现，没改装前轮时，过颠簸路面容易弹飞，因此我在固定万向轮时在其中加入垫片，降低前面地盘高度，发现过颠簸路面有了很大的改进。如图 2.2.1 所示：



图 2.2.1 前轮改装示意图

2.3 智能车后轮轮胎处理方式

如果说硬件决定速度，我觉得轮胎的影响在其中占了有 80%。三轮车模有着直立的转向灵活性，缺少了直立最难的部分直立环。所以，它的速度可以跑的很快，那么一副好的轮胎就显得尤其重要了。

F 车的胎皮与新 C 胎皮一样，橡胶材料不易磨损，不吃软化剂，软化效果不明显很是头疼。

在此期间我发现 f 车里面仅有一块海绵，但却撑不起来胎皮，当车模在行驶过程中胎皮接触路面面积不够达不到一定的摩擦力因此会甩尾。因此我在其原本的胎皮下加入一定量的海绵，使得胎皮被撑起来（不超过原本直径），此时跑起来摩擦力就大了。当速度过快时，胎皮容易外翻，因此使用 704 硅橡胶进行密封，密封时要从里面向外部密封，这样密封效果好。接下来不用软化，一直跑赛道就可以将轮胎磨得很薄，摩擦力自然上去了。合肥师范学院处理的轮胎我看了一下，它那个比我处理好的地方是将胎皮外翻在轮毂外表面再用 704 进行粘连。像这种轮子胎皮越薄摩擦力越大，但如果成了光头胎就不满足比赛要求了，所以根本不需要磨轮胎，

就一直跑就好了。如下图 2.3.1 和 2.3.2:

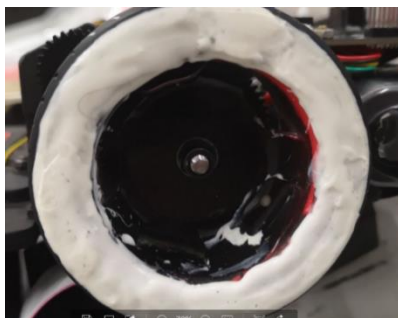


图 2.3.1 后轮胎皮处理示意图



图 2.3.2 后轮胎皮处理示意图

2.4 电感的安装方式及高度

我们在准备比赛期间换了很多结构，如双杆、单杆、架设方式等等，最后选用了双杆作为最后的结构，因为我自己处理不好单杆抖动的问题，所以选择双杆更加稳定。初期，我们用买的一些支架使用螺丝进行连接，发现这样会使前面支架变重，因此舍弃了。接下来我们又买了 8mm 单杆发现这个重量并不比双杆轻，故又换成 5mm 单杆，但 5mm 单杆很抖，我加了支撑杆都无法解决，故舍弃单杆。接着来，利用 3mm 的实心单杆搭了双杆，因为 3mm 的杆子柔韧性好，没有刚性还是会抖一点，加了很多加强杆，结构还是不如意。

最后，我采用了某同学的 3D 打印件，最终 H:15cm，发现这样的结构很好，转弯前瞻重量小不容易形成很大的转动惯量。下面图 2.4.1 展示我所搭过的结构。结论：前瞻支架的搭建对小车运行的转向影响很大。



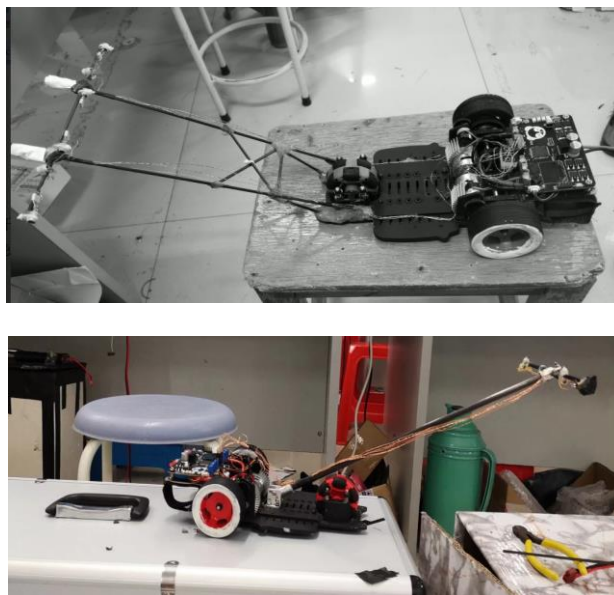


图 2.4.1 前瞻搭建过程示意图

2.5 电池及主板的安装

电池安装影响整车的重心，同时也影响车子的转向。因为三轮类似直立，需要把整车的重心压在后轮轴心上，这样才可以使得后轮转向时有足够的摩擦力。下图为电池安装示意图 2.5.1

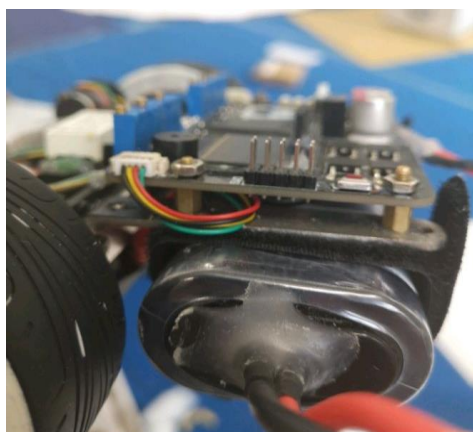


图 2.5.1 电池安装示意图

第三章 智能汽车硬件电路设计

智能车在赛道上可靠运行的最基础的条件是硬件电路的稳定与可靠。经过查阅相关芯片的资料和长时间的调车实验，确定了整个系统的硬件设计，本章将重点介绍各个模块（包括主板电路、驱动电路、运放电路）的电路原理和设计方案。我自己电路板也是开了 3 次才整版调试成功，建议大家画板时要仔细认真检查。

3.1 单片机最小系统

第十三届“恩智浦”智能车竞赛，我们选择“恩智浦”公司 32 位的微处理器 S9KEAZ128AMLK 作为核心控制单元。如图 3.1.1

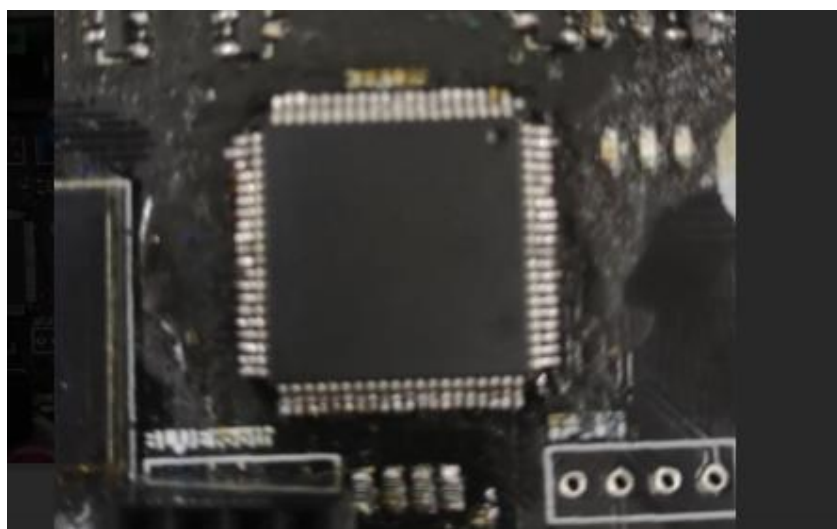


图 3.1.1 单片机最小系统

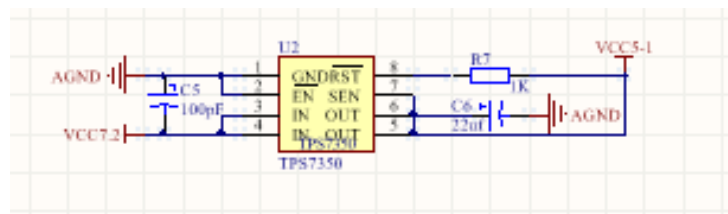
3.2 主板电路设计

3.2.1 电源模块设计

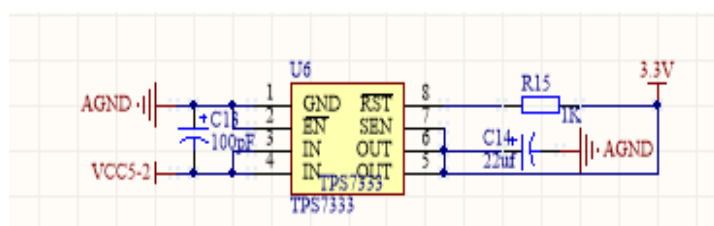
电源模块为其他各个模块提供稳定所需要的电源。设计中，除了需要考虑电源电压范围和电流容量的基本参数之外，还要在电路转换效率、降低噪声、防止干扰和电路简单等方面进行优化。可靠的电源方案是整个硬件电路稳定可靠运行的基础。

为了系统需要稳定工作，则需要多种稳定的电源。本系统中设计了多路电源。智能车使用镍镉充电电池，充满时电压在 7.8V-8.2V, 可直接用于电机驱动供电。

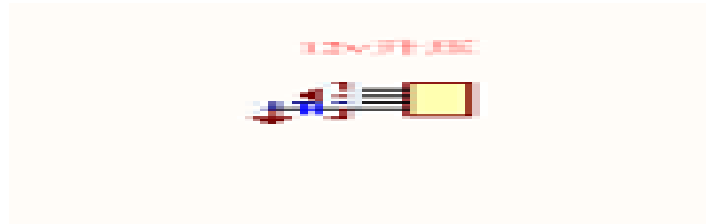
1) 使用 TPS7350 用于输出电压 5V。用于 OLED、AD 运放、编码器、超声波等模块供电。



2) 使用稳压芯片 TPS7333 输出电压 3.3V，用于 OLED 等供电。



3) 使用稳压芯片 b0512，调节电压为 12v，用于驱动芯片 4082 供电。



3.2.2 电磁传感器信号处理电路

确定电感作为检测导线的传感器，但是其感应信号较微弱，且混有杂波，所以要进行信号处理，要进行一下三个步骤才能得到较为理想的信号，信号的滤波，信号的放大，信号的检波。

(1) 信号的滤波：

由于比赛选择 20kHz 的交变磁场作为路径导航信号，在频谱上可以有效地避开周围其他磁场的干扰，因此信号放大需要进行选频放大，使得 20kHz 的信号能够有效的放大，并且去除其它干扰信号的影响。使用 LC 并联谐振电路来实现选频 电路（带通电路），如图 3.2.2 所示。

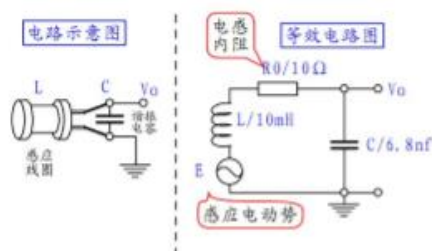


图 3.2.2 信号滤波电路

其中, E 是感应线圈中的感应电动势, L 是感应线圈的电感值, R_0 是电感的内阻, C 是并联谐振电容。电容谐振频率为:

$$f = \frac{1}{2\pi\sqrt{LC}}$$

已知感应电动势的频率 $f=20\text{kHz}$, 感应线圈电感为 10mH , 可以计算出谐振电容的容量为 $C=6.33 \times 10^{-9}\text{nF}$ 。通常在市场上可以购买到的标称电容与上述容值最为接近的电容器为 6.8nF 。故选用 6.8nF 的电容器作为谐振电容。

(2) 信号的放大

由第一步处理后的电压波形已经为较规整的 20kHz 正弦波, 但是幅值较小, 随之距离衰减的迅速, 达不到电压的采样, 所以要进行放大, 我们采用集成运放进行信号的放大处理, 集成运放较三极管优势是很准确且受温度的影响较小, 可靠性高。

测量放大后的感应电动势有很多种, 最简单的就算用二极管检波电路将交变的电压信号检波形成直流信号, 然后在通过单片机的 AD 采集获取正比于感应电压幅值的数值。

我们采用的是利用两个二极管进行倍压检波, 为了使获得更大的动态范围, 我们推荐采用肖特基二极管或锗二极管, 因为此类二极管的开启电压一般在 $0.1 \sim 0.3\text{V}$ 左右, 小于硅二极管的 0.7V 。

由以上的理论知识, 我们在选用集成运放时, 我们遴选了三种方案:

方案 1: 使用三极管放大电路。

方案 2: 使用 LM386 一级放大电路。

方案 3: 使用 opa2350ua 一级放大电路。

经过实验比较, 方案 3 比其他两种方案得到的数据更稳定, 波纹小, 输出稳定。

电路图见 3.2.3、3.2.4、3.2.5。

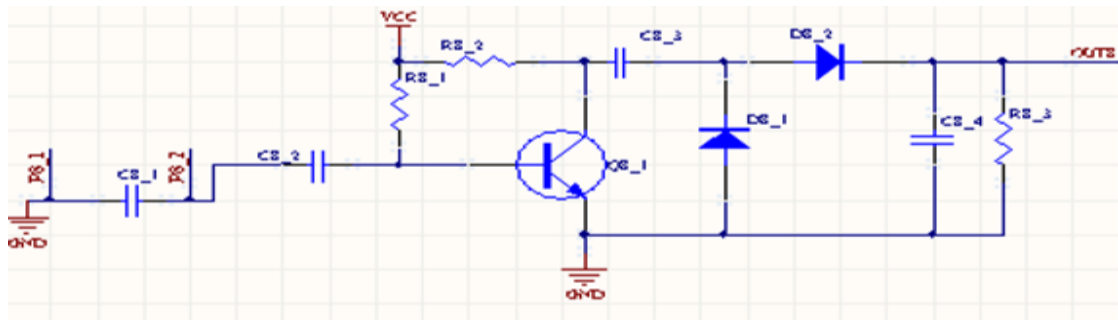


图 3.2.3 三极管放大电路

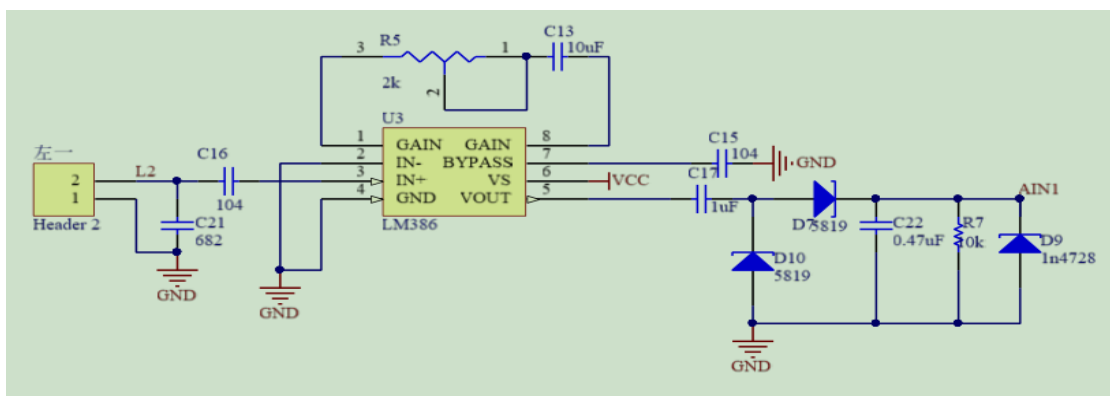


图 3.2.4 LM386 一级放大电路

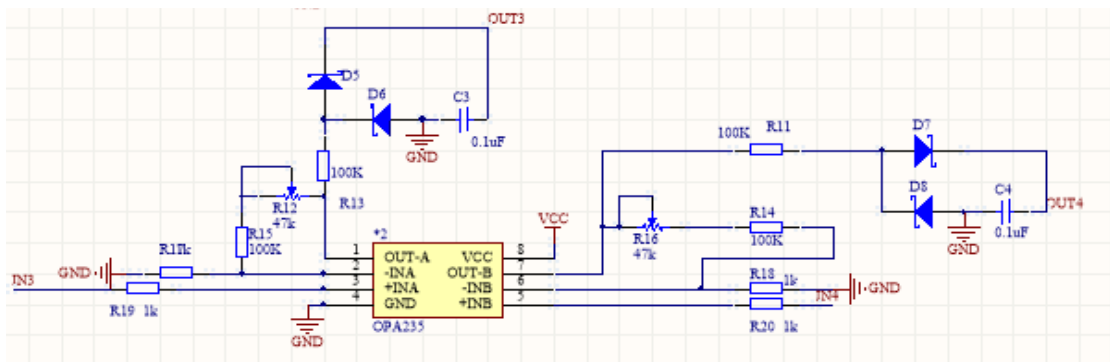


图 3.2.5 OPA2350 一级放大电路

3.2.3 其他传感器电路

在主板上我加入了 LED 灯组、蜂鸣器、OLED 显示等等外设装置。如下图 3.2.6 所示：

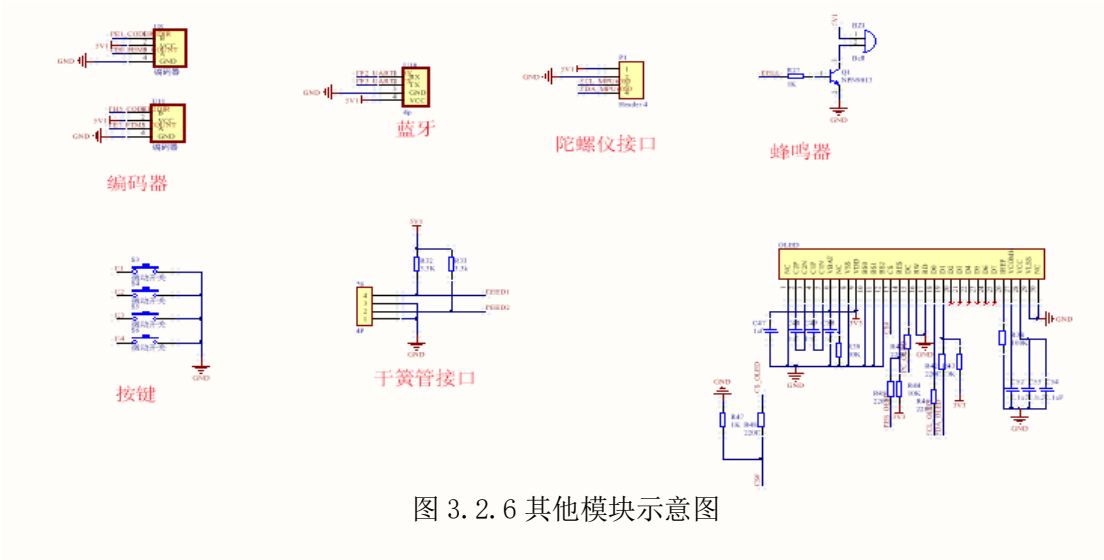


图 3.2.6 其他模块示意图

3.2.4 系统主板的 PCB 设计

我在设计 pcb 过程中，参考上届直立组学长的主驱一体板，其实在设计过程中也遇到很多复杂的问题，如 MC34063 升压电路出问题，运放电路采值乱跳受驱动板影响较大等问题。

其中运放采值影响较大为一体板最大的原因，我们发现硬件滤波的效果还是大于软件滤波的效果，在然后就是分区敷铜，再将数字地模拟地用磁珠或 0 欧电阻隔开，效果比之前好了很多。其中 PCB 走线也有讲究，大家可自行百度走线技巧。

具体 PCB 图如图 3.2.8 所示。

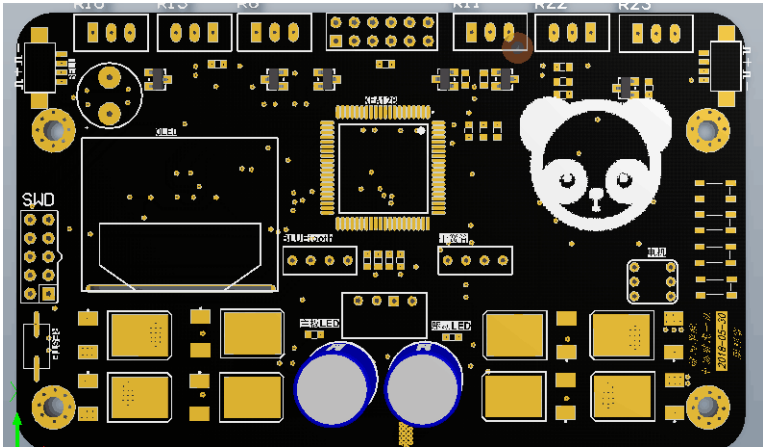


图 3.2.8 PCB 示意图

3.3 电机驱动模块

电机驱动电路为一个由分立元件制作的直流电动机可逆双极性桥式驱动器，其功率元件是由 4 只 N 沟道功率 MOSFET 管组成，额定工作电流可以轻易达到 100A 以上，大大提高了电动的工作转矩和转速。该驱动器主要由以下部分组成：PWM 信号接入口、电机接入口、电源接入口、12V 升压电路、隔离电路、5V 稳压电路等等。

电路图如下所示，首先是使 dis 拉低时，驱动桥才能有作用。PWM1 为某一占空比，PWM2 为低时使得 HIP4082 输出分别为低电平和相应的电压值。驱动电路中的左边两个 mos 导通，此时电机正转。因此，一个电机由两个 mos 管控制。PWM 输入状态与上面相反则为反转。

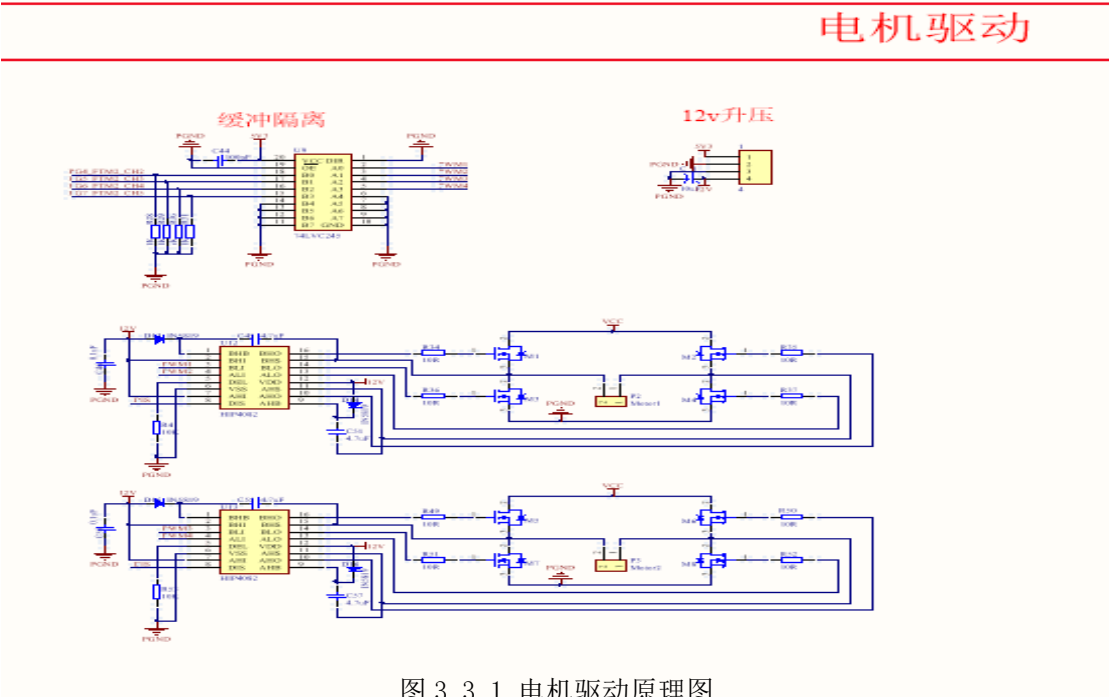


图 3.3.1 电机驱动原理图

第四章 智能汽车控制软件设计说明

4.1 软件开发平台

此次智能车软件开发环境采用了恩智浦公司的飞思卡尔 S9KEAZ128AMLK 系列单片机开发软件 IAR。该软件具有支持多种语言、开发界面统一、交叉开发平台以及支持插件工具等特点。在界面编译完成之后，使用 J-LINK 下载工具下载程序，j-link 工具使用简单，十分方便且支持在线调试。

4.2 软件系统总体设计

程序使用了 KEA128 芯片中的 AD 模块、I/O 模块、FTM 模块。其中编码器和电机分别通过 FTM1、FTM0、FTM2 控制，电机频率 $14 \times 1000\text{Hz}$ 。AD 模块用来读取电磁传感器采样回来的电压信息，I/O 主要用来作为人机交互模块使用，PIT 模块作为定时器产生 1MS 的定时中断，其中陀螺仪和转向控制程序和速度控制服务程序均在 PIT 中断中完成。

程序大概流程如下：（1）进行各个模块的初始化。（2）在按下启动键之后程序进入大循环，扫描是否有暂停键按下。（3）AD 中断服务程序进行 AD 采样，PIT 中断服务程序，进行电感数据分析。（4）转向控制、速度控制，相应 PWM 波值控制电机，使得车模正常运行。如下图 4-1 所示：

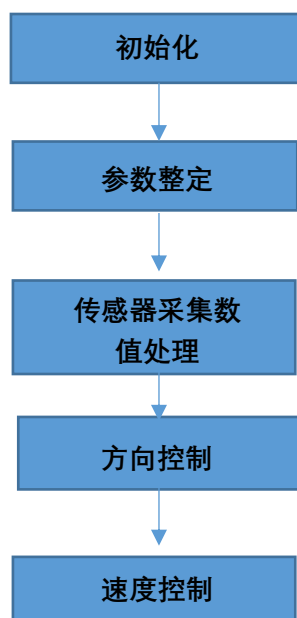


图 4.1 程序控制流程图

4.3 方向控制算法设计

本系统在位置算法设计上经过大量测试，利用四个水平电感的之间的相互关系采用差比和进行计算偏差给电机差速转弯效果能够让车模很好的围绕中线行驶。

4.3.1 磁场信息获取与处理

在小车运动过程中不可避免的遇到一些干扰，所以我们要把采集回来的信号进行滤波和标定处理。为了使单片机获得接近真是的数据我们要进行滤波防止某次电感采值过大或过小对小车识别路径造成误判。

具体实现方式每 5ms 内采集 6 次数据，将采集到 6 个值中间三个取平均。再采用滑动滤波进行 ad 数值处理，最后除去个位数得到最终的数值。

传感器的信号值标定是因为我们在调试小车过程中可能会有电流的变化，规则上是电磁信号发生器调在 80-120mA，所以防止赛道电流的变化，我们并没有采用归一化，而我是直接采用螺丝刀拧电感值的方法解决，另外一种就是可编程电阻可以了解一波。

4.3.2 巡线原理。

巡线部分是本系统软件部分的核心。本系统一共用了六个电感，四个水平电感、两个垂直电感。

我们利用四个水平电感进行差比和计算出误差 error，再利用前面的两个竖直电感加入到偏差中处理圆环的。利用电感值可以识别出直道，弯道，圆环。根据不同的赛道信息利用不同的偏差，分配不同的 PD 参数，使路径尽可能圆滑。

4.3.3 圆环算法

在环岛处电感特征非常明显，很好识别但是入环时机不好把握，我开始尝试了网友所说利用竖直电感直接循迹进去，我发现这种方式不好处理大小环岛而且路径并不是很好，出环岛会抖，因此放弃了这种方案。

如何设定一个程序满足所有的半径环岛呢？最终，我发现竖直电感总会在环岛入口处有特征点，环岛也可以用竖直电感+水平电感进行判断。

我程序总体思路为先判断到了环岛，再次判断入环口，利用竖直电感拐进去。再切为水平电感循迹。其中利用陀螺仪进行判断车身姿态进行清除标志位，因为电感的特征不足以区别环岛一些位置特征。

4.4 控制算法

智能车的控制包括电机和舵机的控制。具体算法的控制算法有 PID 控制、bang-bang 算法控制。

4.4.1 差速控制

差速控制就是赛车的方向控制，以赛车车体为参考系，控制后轮差速转向，始终使导线逼近赛车中心线，因此选择一个好的算法，对于提高电磁车的运动性能，有着重要的作用。

我们选用的差速转弯的算法其实很简单，就是根据电感与电磁线位置偏差 Error 进行打角，我们选用位置式 PD 对双电机进行控制。在这之中，我们用了差比和、差比积等等算法，发现差比和是呈线性变换的，而差比积是呈指数函数变化的。我们希望差速转弯较为圆滑而不是猛转弯，这样是提高不了速度的。而且两个电感计算的偏差并不是呈一次函数变化的，因此我们采用四个水平电感。找出车身与中心线的位置关系选用不同的电感组合方式计算出偏差，再进行限幅。这样计算出来的偏差再利用上位机看就是呈线性变化的了。

当遇到坡道时，小车不进行处理，很容易飞坡，最近几年坡道接的都是急弯或者中 s，很容易冲出赛道。因此我利用陀螺仪判断车头仰角增大到一定条件就判定位坡道进行减速处理，防止冲出赛道。

4.4.2 电机速度控制

电机的控制就是对小车速度进行控制和规划。

速度的规划我们采用常见的“入弯减速，出弯加速”方案进行速度规划，速度不一定要与打角对应，即没必要让打角与速度大小一一对应。

电机速度控制思路我们采用的是增量式 PI，可以瞬间提速到目标速度。但是增量式 PI 有缺陷是速度偏差为 0 时就无法增速了，还有就是积分饱和的影响。

因此我们对此，加入抑制积分饱和处理方式，如限制积分饱和，积分分离等等。

4.5 PID 控制算法

在工程实际中，应用最为广泛的调节器控制规律为比例、积分、微分控制，简称 PID 控制，又称 PID 调节。PID 控制器问世至今已有近 70 年历史，它以其结构简单、稳定性好、工作可靠、调整方便而成为工业控制的主要技术之一。当被控对象的结构和参数不能完全掌握，或得不到精确的数学模型时，控制理论的其它技术难以采用时，系统控制器的结构和参数必须依靠经验和现场调试来确定，这时应用 PID 控制技术最为方便。即当我们不完全了解一个系统和被控对象，或不能通过有效的测量手段来获得系统参数时，最适合用 PID 控制技术。PID 控制，实际中也有 PI 和 PD 控制。本系统采用的就是舵机基于位置式 PD 控制，电机基于增量式 PID 控制。PID 控制器是一种线性控制器，它根据给定值与实际输出值构成控制偏差。将偏差的比例(P)、积分(I)和微分(D)通过线性组合构成控制量，对被控对象进行控制，故称 PID 控制器，原理框图如图 4.2 所示。

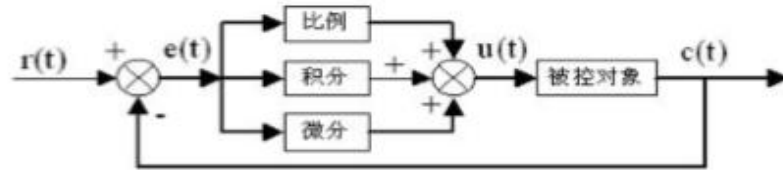


图 4.2 PID 控制原理图

在计算机控制系统中，使用的是数字 PID 控制器，控制规律为：

$$e^k = r^k - c^k \quad (\text{公式 4.1})$$

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

简单说来，PID 控制器各校正环节的作用如下：

比例环节：及时成比例地反映控制系统的偏差信号，偏差一旦产生，控制器立即产生控制作用，以减少偏差。

积分环节：主要用于消除静差，提高系统的无差度。积分作用的强弱取决于积分时间常数，越大，积分作用越弱，反之则越强。

微分环节：能反映偏差信号的变化趋势(变化速率)，并能在该偏差信号变得太大之前，在系统中引入一个有效的早期修正信号，从而加快系统的动作速度，减小调节时间。

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

4.5.1 位置式PID

位置式 PID 中，由于计算机输出的 $u(k)$ 直接去控制执行机构(如阀门)， $u(k)$ 的值和执行机构的位置(如阀门开度)是一一对应的，所以通常称公式(4.2)为位置式 PID 控制算法。位置式 PID 控制算法的缺点是：由于全量输出，所以每次输出均与过去的状态有关，计算时要对过去 $e(k)$ 进行累加，计算机工作量大；而且因为计算机输出的 $u(k)$ 对应的是执行机构的实际位置，如计算机出现故障， $u(k)$ 的大幅度变化，会引起执行机构位置的大幅度变化，这种情况往往是生产实践中不允许的，在某些场合，还可能造成严重的生产事故。因而产生了增量式 PID 控制的控制算法，所谓增量式 PID 是指数字控制器的输出只是控制量的增量 $\Delta u(k)$ 。

4.5.2 增量式PID

当执行机构需要的是控制量的增量(例如：驱动步进电机)时，可由式(4.2)推导出提供增量的 PID 控制算式。由式(4.2)可以推出式(4.3)，式(4.2)减去式(4.3)可得式(4.4)。

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

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

$$\text{式中 } \Delta e(k) = e(k) - e(k-1); K_I = K_p \frac{T}{T_I}; K_D = K_p \frac{T_D}{T};$$

公式(4.4)称为增量式 PID 控制算法，可以看出由于一般计算机控制系统采用恒定的采样周期 T ，一旦确定了 K_P 、 T_I 、 T_D ，只要使用前后三次测量值的偏差，即可由式(4.4)求出控制增量。

增量式 PID 具有以下优点：

(1) 由于计算机输出增量，所以误动作时影响小，必要时可用逻辑判断的方法关掉。

(2) 手动/自动切换时冲击小，便于实现无扰动切换。此外，当计算机发生故障时，由于输出通道或执行装置具有信号的锁存作用，故能保持原值。

(3) 算式中不需要累加。控制增量 $\Delta u(k)$ 的确定仅与最近 k 次的采样值有关，所以较容易通过加权处理而获得比较好的控制效果。

(4) 但增量式 PID 也有其不足之处：积分截断效应大，有静态误差；溢出的影响大。使用时，常选择带死区、积分分离等改进 PID 控制算法。

4.5.3 PID 参数整定

运用 PID 控制的关键是调整 K_P 、 K_I 、 K_D 三个参数，即参数整定。PID 参数的整定方法有两大类：一是理论计算整定法。它主要是依据系统的数学模型，经过理论计算确定控制器参数；二是工程整定方法，它主要依赖工程经验，直接在控制系统的试验中进行，且方法简单、易于掌握，在工程实际中被广泛采用。由于智能车系统是机电高耦合的分布式系统，并且要考虑赛道的具体环境，要建立精确的智能车运动控制数学模型有一定难度，而且我们对车身机械结构经常进行修正，模型参数变化较为频繁，理论计算整定法可操作性不强，最终我们采用了工程整定方法。此外，我们先后实验了几种动态改变 PID 参数的控制方法。

第五章 系统调试工具

5.1 在线调试

开发工具使用的是 IAR Rmbedded Workbench 开发环境，如图 5.1 所示。

它可以为单片机 KEA128 提供与之配套的应用程序开发模块。在目标程序的下载方面。通过 J-link 与单片机之间的连接下载程序。IAR 具有强大的在线调试功能，可以满足智能车软件系统的需要，软件界面如图 5.1 所示。

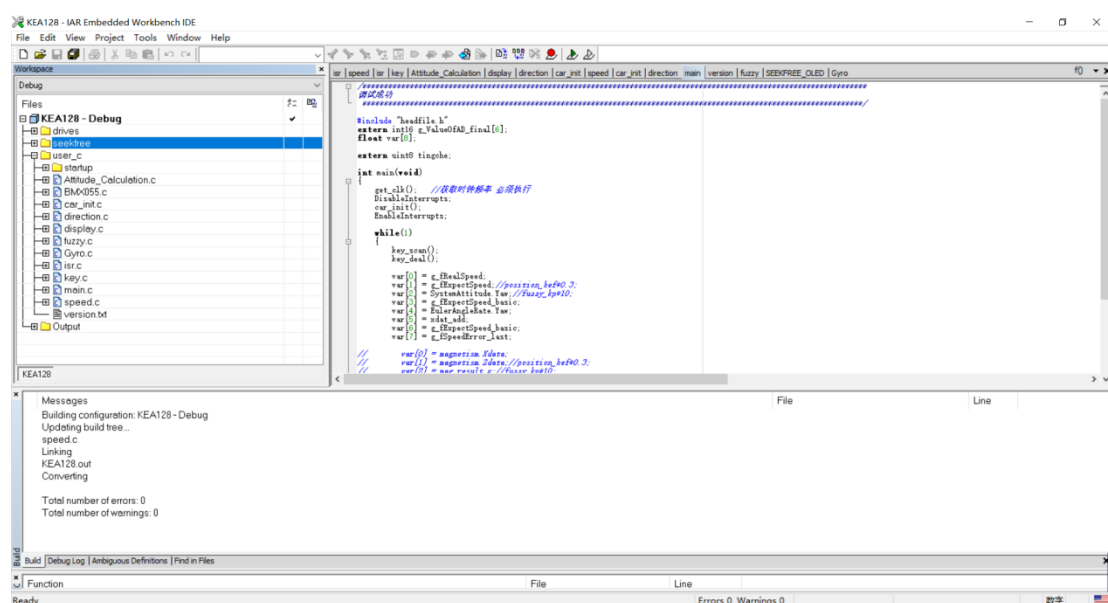


图 5.1 IAR 开发环境图

5.2 现场调试

在基本程序完成之后还需要在程序进行微调。我们使用拨码开关和蓝牙模块进行上位机调试。

比赛时我们试车后根据经验使用按键和拨码开关进行调试。

- (1) 各种形式赛道的速度参数；
- (2) 电机 PD 控制的 Kp 和 Kd 等参数。

5.3 调试过程

车模在运行时，我们在调参数的过程中怎么保证调试的参数符合我们想要条件呢？我们使用 Bluetooth UART Board 蓝牙模块配合上位机进行观察车模运行状态。



图 5.2 蓝牙调试模块



图 5.3 上位机调试界面

第六章 车模主要参数及说明

表 6.1 车模主要参数表

项目	参数
组别	电磁三轮组
车模几何尺寸（长、宽、高）mm	477、250、162
传感器种类与数量	电感/6 个 编码器/2 个
新增伺服电机数量	0
赛道信息检测（ms）	5
主要集成电路种类/数量	主控板 1 块

结论

“恩智浦”杯全国大学生智能汽车竞赛已经成功举办了十二届，该竞赛涵盖了控制、模式识别、传感技术、电子、电气、计算机、机械等多个学科，培养了学生的知识融合和实践动手能力。

此份技术报告中，我们主要介绍了准备比赛时的基本思路，也包括机械结构及重要的控制算法的创新思想。

在机械结构方面，从最初的洞洞板，到一体板；从双杆到单杆，再到双杆，我们的机械结构一改再改。另外，PCB 制图中元器件放置的位置，布线的方式，我们都做了相应的优化和改善，例如衰弱等长布线之间的干扰我们采用了波浪排布法，减少驱动和主板之间的干扰我们使用了分区敷大块铜且连接点采用 0 欧磁珠或者 0 欧姆电阻，采用单点共地，减少干扰的方法。

在软件方面，我们使用 C 编程，利用比赛推荐的开发软件调试程序，经过小组成员不断讨论、改进，终于设计出属于适应能力强比较稳定的程序，在这套算法中，我们结合路况调整车速，做到直道加速、弯道减速、圆环减速，保证比赛完成的前提下时间最短。

2017 年的 12 月份就开始准备比赛，其中经历了很多问题，光一个环岛问题处理了得有一两个月，其中一步步成长，每个队友都全身心的投入，才有了今天的的成绩，希望能在国赛有个很好的发挥，取得令人满意的成绩。

参考文献

- [1] 卓晴, 黄开胜, 邵贝贝等, 《学做智能车——挑战“飞思卡尔”杯》[C]. 北京: 北京航空航天大学出版社, 2007.
- [2] 邵贝贝, 单片机嵌入式应用的在线开发方法[M], 北京: 清华大学出版社, 2004. 10
- [3] 陶永乐, 新型 PID 控制及其应用 (第二版), .北京: 机械工业出版社, 2002.
- [4] 谭浩强 .C 程序设计 (第三版). [M]. 北京: 清华大学出版社, 2005. 7.
- [5] 李仕伯, 马旭, 卓晴. 基于磁场检测的寻线小车传感器布局研究. [J]. 电子产品世界, 2009, 16(12). 41-44.
- [6] 康华光 .电子技术基础 模拟部分 (第五版) . [M]. 北京: 高等教育出版社. 2008. 1.
- [7] 康华光 .电子技术基础 数字部分 (第五版) . [M]. 北京: 高等教育出版社. 2008. 1.

附录：程序及初始化代码

```

/**
 * @file   main 函数
 * @note   主函数
 * @date   2018
 * @author 大学生电子工程协会
 */

#include "headfile.h"

extern int16 g_ValueOfAD_final[6];
float var[8];

extern uint8 tingche;

int main(void)
{
    get_clk(); //获取时钟频率 必须执行
    DisableInterrupts;
    car_init();
    EnableInterrupts;

    while(1)
    {
        key_scan();
        key_deal();

        var[0] = g_fRealSpeed;
        var[1] = g_fExpectSpeed;           //position_bef*0.3;
        var[2] = SystemAttitude.Yaw;       //fuzzy_kp*10;
        var[3] = g_fExpectSpeed_basic;
        var[4] = EulerAngleRate.Yaw;
        var[5] = xdat_add;
        var[6] = g_fExpectSpeed_basic;
        var[7] = g_fSpeedError_last;

        // var[0] = magnetism.Xdata;
        // var[1] = magnetism.Zdata;         //position_bef*0.3;
        // var[2] = mag_result_x;            //fuzzy_kp*10;
        // var[3] = mag_result_z;
    }
}

```

```

//      var[4] = EulerAngleRate.Yaw;
//      var[5] = xdat_add;
//      var[6] = zdat_add;
//      var[7] = tingche;

      vcan_sendware_V(var, sizeof(var));
    }

}

/*****
*
* @file      中断文件
* @company   皖西学院
* @author    大学生电子工程协会
* @version   v2.0
* @Software  IAR 7.7 or MDK 5.23
* @Target core S9KEA128AMLK
* @Taobao    https://seekfree.taobao.com/
* @date      2017-11-6
*****/

#include "isr.h"

////////////////////////////////////

BMX055Datatypedef    BMX055_data;
EulerAngleTypedef    SystemAttitude;      //姿态角
EulerAngleTypedef    SystemAttitudeRate;  //姿态角速度
AttitudeDatatypedef  GyroOffset;

float AccZAngle = 0;
float QZAngle   = 0;
float angle_Yaw = 0;

```

```

float integral_angle;

int angle_speed_racord[10];
uint8 z;
uint8 pit_num;

uint8 breakcar_tim, breakcar_flg, breakcar_flg_ramp, breakcar_flg_ramp_tim;
uint8 stop_flag, stop_journey_flag, stop_num, stop_flag_2, stop_journey_flag_2;
int32 mag_xdat[10], mag_zdat[10];
int32 xdat_add, zdat_add, mag_result_x, mag_result_z;
int32 value_x, value_z, value_x_b, value_z_b;
int32 xdat_save[10], zdat_save[10];
////////////////////////////////////

uint8 Flag_Stop = OFF;           //OFF 停车
uint8 Flag_Speed = ON;           //ON 增加速度控制
uint8 Flag_Direction = ON;       //ON 增加方向控制

void PIT_CHO_IRQHandler(void)
{
    uint8 fi, fj, i, k;
    int32 temp2;

    static uint8 IsAttitudeinit = 0;

    gpio_set(D5, 1);

    magnetism.Xdata = BMX055_data.MAGXdata;
    magnetism.Zdata = BMX055_data.MAGZdata;

    value_x = (int32)filter(magnetism.Xdata, value_x_b);
    value_z = (int32)filter(magnetism.Zdata, value_z_b);
    magnetism.Xdata = value_x_b = value_x;
    magnetism.Zdata = value_z_b = value_z;

```

```

if(fj == 2)
{
    fj = 0;

    if(stop_num == 0)
    {
        mag_xdat[0] = (int32)magnetism.Xdata;
        mag_zdat[0] = (int32)magnetism.Zdata;
        for(fi=9; fi>=1; fi--)
        {
            mag_xdat[fi] = mag_xdat[fi-1];
            mag_zdat[fi] = mag_zdat[fi-1];
        }

        for(i=0; i<9; i++)
        {
            for(k=0; k<9-i; k++)
            {

                if(mag_xdat[k] > mag_xdat[k+1])           //前面的比后面的大  则进行交换
                {
                    temp2 = mag_xdat[k+1];
                    mag_xdat[k+1] = mag_xdat[k];
                    mag_xdat[k] = temp2;
                }
                if(mag_zdat[k] > mag_zdat[k+1])           //前面的比后面的大  则进行交换
                {
                    temp2 = mag_zdat[k+1];
                    mag_zdat[k+1] = mag_zdat[k];
                    mag_zdat[k] = temp2;
                }
            }
        }

        xdat_add = 0;
        zdat_add = 0;

        for(fi=0; fi<10; fi++)
        {

```

```

        xdat_add += mag_xdat[fi];
        zdat_add += mag_zdat[fi];
    }
    xdat_add = xdat_add / 10;
    zdat_add = zdat_add / 10;
}
}
if(stop_flag_2 == 2)
{
    if((magnetism.Xdata < mag_result_x-520) || (magnetism.Xdata > mag_result_x+520) ||
        (magnetism.Zdata < mag_result_z-520) || (magnetism.Zdata > mag_result_z+520))
    {
        Flag_Speed = OFF;
        Flag_Direction = OFF;
        LeftDead = 0;
        RighDead = 0;
        DELAY_MS(10);
        stop_car(-500,-500);
        DELAY_MS(450);
        stop_car(-0,-0);
        stop_flag_2 = 3;
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

PIT_FLAG_CLR(pit0); //清除中断标志

static uint16 j = 0;
pit_num++;
if(pit_num == 5)
    pit_num = 0;

switch(pit_num)
{
    case 0:
    {
        fj++;
        BMX055_data.GYROXdata = (BMX055_data.GYROXdata - GyroOffset.Xdata) * 0.030517578;
        BMX055_data.GYROYdata = (BMX055_data.GYROYdata - GyroOffset.Ydata) * 0.030517578;
        BMX055_data.GYROZdata = (BMX055_data.GYROZdata - GyroOffset.Zdata) * 0.030517578;
    }
}

```



```

BMX055_data.ACCXdata *= 0.001953125;    //////////4 / 2048
BMX055_data.ACCYdata *= 0.001953125;
BMX055_data.ACCZdata *= 0.001953125;

Acc.Xdata = BMX055_data.ACCXdata;
Acc.Ydata = BMX055_data.ACCYdata;
Acc.Zdata = BMX055_data.ACCZdata;
Gyro.Xdata = BMX055_data.GYROXdata;
Gyro.Ydata = BMX055_data.GYROYdata;
Gyro.Zdata = BMX055_data.GYROZdata;

//TestUsing = pit_time_get(PIT1);
if(IsAttitudeinit == 0)
{
    Quaternion_init();                //姿态解算初始化
    IsAttitudeinit = 1;
}
else
{
    Attitude_UpdateGyro();            //快速更新
    Attitude_UpdateAcc();             //深度融合更新

    SystemAttitude.Pitch = -EulerAngle.Roll / PI * 180;    //转为角度
    SystemAttitude.Yaw = EulerAngle.Yaw / PI * 180;
    SystemAttitudeRate.Pitch = -EulerAngleRate.Roll / PI * 180;
    SystemAttitudeRate.Yaw = EulerAngleRate.Yaw / PI * 180;

    //为了更精准的获得俯仰角，再次进行互补滤波
    float AccZ = 0, AccZAdjust = 0;
    AccZ = -Acc.Zdata;
    if (AccZ > 1)
        AccZ = 1;
    if (AccZ < -1)
        AccZ = -1;
    AccZAngle = asinf(AccZ) * 180 / PI;
    QZAngle = SystemAttitude.Pitch;
    AccZAdjust = (AccZAngle - SystemAttitude.Pitch);
    SystemAttitude.Pitch += (-Gyro.Xdata + AccZAdjust) * PERIODS;

```

```

        integral_angle += EulerAngleRate.Yaw;
        if(integral_angle > 720)
            integral_angle = 720;
        if(integral_angle < -720)
            integral_angle = -720;

    }
}
break;
case 1:
{
    DirectionControl();

}
break;
case 2:
{
    Circle_judge();

    if(breakcar_flg == 1)
    {
        breakcar_tim++;
        if(breakcar_tim > 25)
        {
            breakcar_flg = 0;
            breakcar_tim = 0;
        }
    }

    if(breakcar_flg_ramp == 1)
    {
        breakcar_flg_ramp_tim++;
        if(breakcar_flg_ramp_tim > 100)
        {
            breakcar_flg_ramp = 0;
            breakcar_flg_ramp_tim = 0;
        }
    }
}

```

```

        j++;
        if(j >= 2)
        {
            j = 0;
            SpeedControl();
        }

        BMX055_MAGXdata(&BMX055_data, 1);
    }
    break;
    case 3:
    {
        BMX055_DataRead(&BMX055_data, 0);

    }
    break;
    case 4:
    {
        DirectinControlOutput();
        rampway();/////坡道
        PWMOut();          //最终 PWM 输出

        BMX055_MAGXdata(&BMX055_data, 1);
    }
    break;
}

gpio_set(D5, 0);
// GyroControl();
// AngleControl();
////////////////////////////////////

// Angle_Calcu();

/* ftm_pwm_duty(ftm2, ftm_ch2, 0);//you
   ftm_pwm_duty(ftm2, ftm_ch3, 500);//you

```

```

    ftm_pwm_duty(ftm2, ftm_ch5, 0);
    ftm_pwm_duty(ftm2, ftm_ch4, 500);*/

}

void PIT_CH1_IRQHandler(void)
{
    PIT_FLAG_CLR(pit1);

}

void IRQ_IRQHandler(void)
{
    CLEAR_IRQ_FLAG;

}

void KBIO_IRQHandler(void)
{
    CLEAN_KBIO_FLAG;

}

void UART0_IRQHandler(void)
{
}

/*
中断函数名称，用于设置对应功能的中断函数
Sample usage:当前启用了周期定时器 通道0 得中断
void PIT_CH0_IRQHandler(void)
{
    ;
}
记得进入中断后清除标志位

FTMRE_IRQHandler

```

PMC_IRQHandler
IRQ_IRQHandler
I2C0_IRQHandler
I2C1_IRQHandler
SPI0_IRQHandler
SPI1_IRQHandler
UART0_IRQHandler
UART1_IRQHandler
UART2_IRQHandler
ADC0_IRQHandler
ACMP0_IRQHandler
FTM0_IRQHandler
FTM1_IRQHandler
FTM2_IRQHandler
RTC_IRQHandler
ACMP1_IRQHandler
PIT_CH0_IRQHandler
PIT_CH1_IRQHandler
KBIO_IRQHandler
KB11_IRQHandler
Reserved26_IRQHandler
ICS_IRQHandler
WDG_IRQHandler
PWT_IRQHandler
MSCAN_Rx_IRQHandler
MSCAN_Tx_IRQHandler
*/