

第十一届“恩智浦”杯全国大学生 智能车竞赛技术报告



学 校： 华东交通大学

队伍名称： Jarvis 队

参赛队员： 周佳新

雷科

胡康荣

带队教师： 杨云

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: 邵志新
雷科
胡康荣

带队教师签名: 杨红

日期: 2016.8.13

摘 要

本课题是设计并实现一种基于 MK60FN5121MOVLP15 微控制器为核心的自主识别路径的智能车系统，通过硬件二值化高速摄像头检测跑道的路况，再由编码器检测模型车的车速，使用 PID 控制算法调节车模的舵机转向以及驱动电机的转速，完成对模型车运动速度和运动方向的闭环控制。设计目标是使智能小车在不同路况的赛道上自主循迹行进。为了使得调试更加方便快捷，我们使用了 NRF24L01 无线通讯模块、蓝牙模块等多种调试工具。经过长时间的调试，本系统已经可以有效、可靠地运行。

报告介绍了总体的设计方案，小车循迹原理，各个模块的硬件组成以及软件设计。硬件方面介绍小车电路系统板的制作、电路的原理以及选型。软件方面介绍智能车的各个模块的初始化，速度控制等。辅助设计工具有无线通讯模块调试工具。

关键字： MK60FN5121MOVLP15、摄像头、PID、编码器

Abstract

This topic is designed and implemented based on self-identification path MK60FN5121M0VLQ15 microcontroller as the core of the intelligent vehicle system, the detection of the runway by a linear camera road, and then attitude data collection model car with encoder detection model car speed. Cars using the PID control algorithm to control the turning of the steering gear and adjust the drive motor speed, complete closed-loop control of the model car movement speed and direction of movement. Design goal is to make smart car in different road conditions on the track of autonomous tracking. To make debugging easier and faster, we use a variety of debugging tools NRF24L01 wireless communication module, Bluetooth module. After a long period of commissioning, the system has been effective, reliable operation.

The report describes the overall design, the tracking principle of the car, each module consisting of hardware and software design. Introduction trolley system hardware circuit board production, as well as the principle of the circuit and selection. Software introduces each module initialization intelligent vehicle speed control. Aided design tools have wireless communication module debugging tools.

Keywords: MK60FN5121M0VLQ15, camera, PID, encoder.

目录

第一章 引言.....	1
第二章 系统基本原理分析及总体设计.....	3
2.1 系统需求分析.....	3
2.2 基本原理.....	4
2.3 设计思路.....	4
第三章 硬件模块的方案与实现	5
3.1 电源电路设计.....	5
3.2 电机驱动模块	6
3.3 摄像头图像采集电路	9
3.4 编码器电路.....	10
3.5 液晶显示接口电路.....	11
3.6 按键电路.....	11
3.7 蜂鸣器电路.....	12
3.8 指示灯电路.....	12
3.9 SD 卡电路.....	13
3.10 核心板接口电路.....	13
3.11 其它.....	14
3.12 小结.....	14
第四章 模型车机械结构调整与优化.....	15
4.1 B 车底板的固定.....	16
4.2 电池的安装.....	17

4.3 编码器的安装.....	18
4.4 摄像头的安装.....	19
4.5 舵机的安装.....	20
4.6 前轮的机械结构调整.....	22
4.7 主控板的安装.....	24
4.8 小结.....	25
第五章 智能车控制软件设计.....	27
5.1 主程序流程图.....	27
5.2 图像采集与处理.....	28
5.2.1 图像的采集.....	28
5.2.2 图像的处理.....	28
5.2.3 赛道曲率的计算.....	29
5.2.4 赛道分类.....	29
5.3 控制算法与函数.....	34
5.3.1 PID 控制算法介绍.....	34
5.3.2 分类进行转向舵机的方向控制.....	37
5.3.3 速度控制.....	38
5.4 小结.....	39
第六章 调试方案.....	41
6.1 IAR 集成开发环境.....	41
6.2 上位机.....	43
6.2.1 摄像头上位机.....	43
6.2.2 PID 调试助手.....	45
6.2.3 参数修改上位机.....	45
6.4 小结.....	46

第七章 模型车主要技术参数.....	47
第八章 总结.....	49
参考文献.....	51
附录 A 主控板原理图及 P C B	
附录 B 部分程序源代码	

第一章 引言

“恩智浦”杯全国大学生智能汽车竞赛是由教育部高等自动化专业教学指导分委员会主办的一项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。

智能车比赛起源于韩国，是韩国汉阳大学汽车控制实验室在飞思卡尔半导体公司资助下举办的以 HCS12 单片机为核心的大学生课外科技竞赛。后“飞思卡尔”智能车比赛更名为“恩智浦”智能车比赛。组委会提供一个标准的汽车模型、直流电机和可充电式电池，参赛队伍要制作一个能够自主识别路径的智能车，在专门设计的跑道上自动识别道路行驶，谁最快跑完全程而没有冲出跑道并且技术报告评分较高，谁就是获胜者。其设计内容涵盖了控制、模式识别、传感技术、汽车电子、电气、计算机、机械、能源等多个学科的知识，对学生的知识融合和实践动手能力的培养，具有良好的推动作用。

我们的智能车系统由微控制器模块、电源管理模块、路径检测模块、车速检测模块、直流驱动电机控制模块和调试模块组成。本系统以 32 位微处理器 K60 为控制核心，并采用 IAR Embedded Workbench 软件编程、J_LINK、NRF24L01、蓝牙等作为调试工具。运用摄像头进行道路信息采集，并采用 PWM 技术和 PID 自动控制技术来控制电机转速和舵机转向。各个部分经过微控制器模块的处理控制，能够以较快的速度在指定的轨迹上行驶，在进弯道、十字路口、坡道、小 S 等特殊轨迹之前能够提前识别并通过改变轮子和舵机上的 PWM 的占空比，达到稳定快速通过和减小路程的效果。

在本次竞赛的近半年准备过程中，我们团队对车模的结构制作方案和控制方案进行了比较深入的研究，硬件上经过几次的调整，使得小车结构的稳定性得到了很大的提高，为后期提速打好了坚实的基础。软件上，结构上的修改和参数上长期的调整，模型车已经可以在赛道上以较快的速度稳定的运行。

在准备比赛的过程中，因为智能车竞赛的知识涵盖自动化控制、颜色识别、传感技术、汽车电子、电气、计算机、机械等多个学科，我们都需要去学习，这次磨练对我们的知识融合、实践动手能力和团队协作能力的培养有极大的推动作用。在此要感谢华东交通大学对此次比赛的关注、支持，我们的成果离不开学校的大力支持及指导老师悉心的教导；还要感谢和我们一起协作的队员们，共勉使我

们能够走到今天。最后还要感谢清华大学卓老师、 以及论坛里面的各位战友。

第二章 系统基本原理分析及总体设计

2.1 系统需求分析

随着智能车竞赛已经举办了十届。在这十届竞赛以来，参赛队伍的实力不断地增强，模型车的速度已经达到了一个非常快的速度了。本次竞赛我队选择了摄像头组。摄像头组竞赛相对于其它组别赛来说有比较长的比赛时间，所以这么多届以来的图像处理和识别的经验积累和方法改进，使得这个组别的赛事竞争程度越来越激烈。此次竞赛摄像头配的车模是B车。B车只有一个大电机，是被动差速，同时车模的整体的机械结构对整车速度提升的影响也是非常大的。也正源于此，这对我们来说也就有了很高的挑战性。我们必须对车模进行比较全面深入的了解，清楚模型车所面临的各种问题。

模型车需要在规定的赛道上行驶，就需要能比较清晰地看到赛道的“眼睛”——一个好的摄像头。摄像头将赛道的情况扫描出来，对看到的图形进行处理，获得所需的图像，以及对赛道种类的识别，这些都是模型车能否快速行驶的基本条件。这是赛道识别控制。

智能车竞赛是一个竞速赛，就需要模型车以最快最稳的速度行驶。B车电机的动力相对来说还是蛮给力的，但是也要控制得好，不然整辆车就乱跑了。所以为了得知模型车的实时速度，我们加了一个编码器来获取电机的转速从而推得车的速度。根据编码器反馈得到的速度再通过一系列算法改变车的速度，就实现了对模型车速度的闭环控制了。这是对模型车的速度控制。

模型车识别出赛道后，处理器根据赛道种类的不同控制舵机的转向，同时舵机的转向也要配合整车的速度。根据车速的不同，舵机配合转多少角度。舵机控制得好在弯道就能跑得顺畅，跑得顺畅对模型车的整体速度提升也会有很大的帮助。这就是模型车的转向控制。

前面三个任务是模型车控制任务的最核心、最基本的控制任务。

最后，在模型车的调试阶段是模型车提速的重要阶段，只要前面的基础打好，剩下的就只是调试提速了，但是在调试的阶段，调试的手段至关重要，所以在调试

之前，就应该做好准备，准备好相应的调试工具和调试手段。比如各个模块的上位机以及通讯模块。

2.2 基本原理

前面讲到对模型车的控制问题。这些问题说到底其实就是对舵机和电机的控制，因为要让模型车能跑起来主要靠的就是电机和舵机。电机提供动力，舵机掌握方向。所以整个车的控制原理就是：摄像头采集赛道信息，之后微处理器对摄像头采集到的信息进行分析处理，然后再控制电机的转速，电机的转速又通过传感器反馈回微处理器，微处理器再对反馈回来的速度进行分析以使电机的速度根据不同种类的赛道进行实时变化，同时微处理器还控制舵机的转向和拐角，使得模型车能在弯道这样的情况下顺利通过。当然其中会用到很多算法，比如对图像处理的算法，涉及对图像的采样周期和处理方式的控制，对舵机和电机采用PWM 控制技术进行驱动，然后再根据PID算法对舵机的打角和电机的速度进行实时控制。在调试过程中，运用多种方式对模型车的参数进行实时修改，使车能跑得更好。

2.3 设计思路

根据摄像头组的规则、要求以及方便调试。我们设计的模型车功能电路共包括五个模块：主控模块、传感器模块、电机驱动模块、电源管理模块、调试模块。

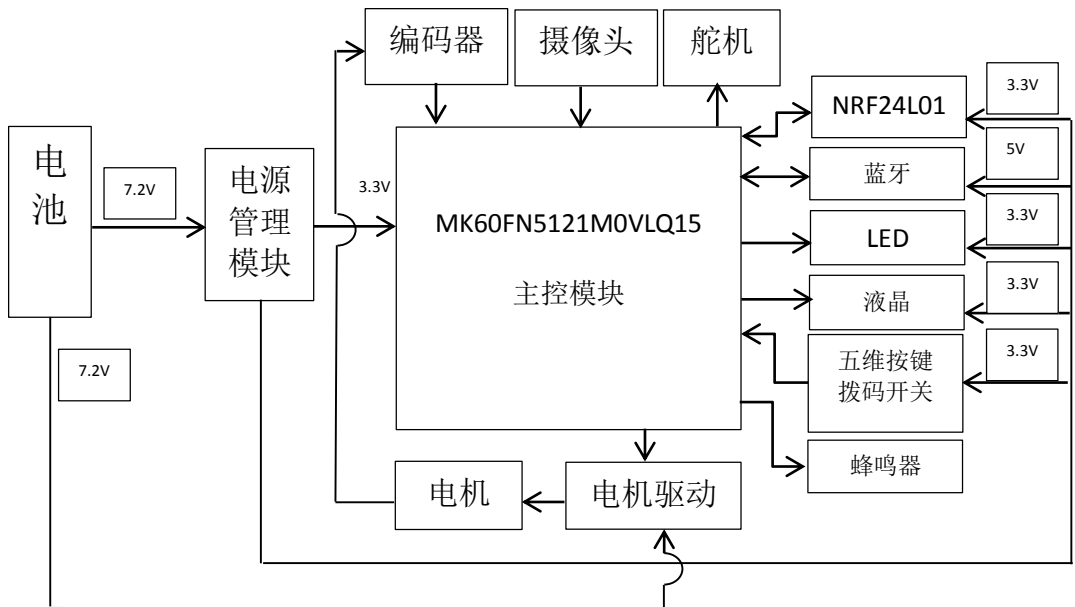


图 2.1 设计整体框图

第三章 硬件模块的方案与实现

硬件电路是整个系统中想当重要的一部分，是系统运行的基础，所以在每一个模块中电阻和电容的相关系数，都需要仔细斟酌。硬件电路的设计和制作是一个漫长的过程，更是一个充满乐趣的过程，也是一个不断完善的过程。如果电路刚开始设计考虑的不周全，可能就需要重新设计与绘制电路板，这样的话 会浪费很多的时间，所以一定要保证电路的稳定性，在此基础上提高电路板的集成度，这需要耗费很多的时间和精力去完成它。

3.1 电源电路设计

电源模块是智能车动力核心，电源模块是否稳定，各模块供电电流是否满足需求，这些将决定小车是否能够快速加速和减速和舵机响应的快慢，查阅电源模块的相关资料，我们选择了比较适合电源。比赛指定智能车电源只能使用指定型号的 7.2V 2000mAh Ni-Cd 电池供电。实际充满电时其电压可达到8.0V至8.4V左右，因此我们的供电模块在输入电压有浮动时输出电压要保持稳定。同时我们为了避免各个模块工作时对其他模块造成干扰，我们对每个主要模块进行单独供电。

我们现在需要 4 种电源，一种是为摄像头、液晶、LED、SD 卡、编码器等供电的3.3V 电源；一种是为蓝牙和驱动芯片供电的 5.0V 电源；一种是为舵机供电的6V左右的可调电源；另一种是为电机供电的 7.2V 电源；。前三种电源可供我们选择， 一种为开关稳压电源， 另一种为线性稳压电源。开关稳压电源功耗小，效率高；线性稳压电源，电路简单，输出纹波小。整体考虑，我们选择线性集成三端稳压电源。电机电源我们就直接取自电池两端。

5.0V 供电我们采用 LM2940，因为 LM2940 是压差小，输出纹波小的线性稳压 电源，当电源的电压比较低时也能为系统提供稳定的电源。另外由于若是直接从电池端电压直接稳压到 3.3V 电源的效率会变得很低，所以我们先将电池电压降为 5V 然后再通过稳压芯片降为 3.3V。3.3V电源供电， 我们采用TPS7333，因为 TPS7333 的压差小，当电源的电压比较低时，也可以为系统提供稳定的电源，而且 TPS7333 的输出纹波非常小，输出容量也满足各元器件的供电要求。为了使舵机的响应速度更快在车速度较快时能及时做出反应给电机供电，我们将对舵机

采用一个芯片单独供电，其实这也是为了防止舵机堵转时瞬时电流过大当与其它元器件共用一个电源芯片的情况下对其它元器件造成影响。舵机电源我们采用 LM1084 芯片所搭成的电压可调电路将输出电压转成 6V 左右，经实际测试，这种方法是很有用的。至于给电机供电，我们只是对电池经过滤波之后直接接到电机上的。下面是我们的电源模块原理图：

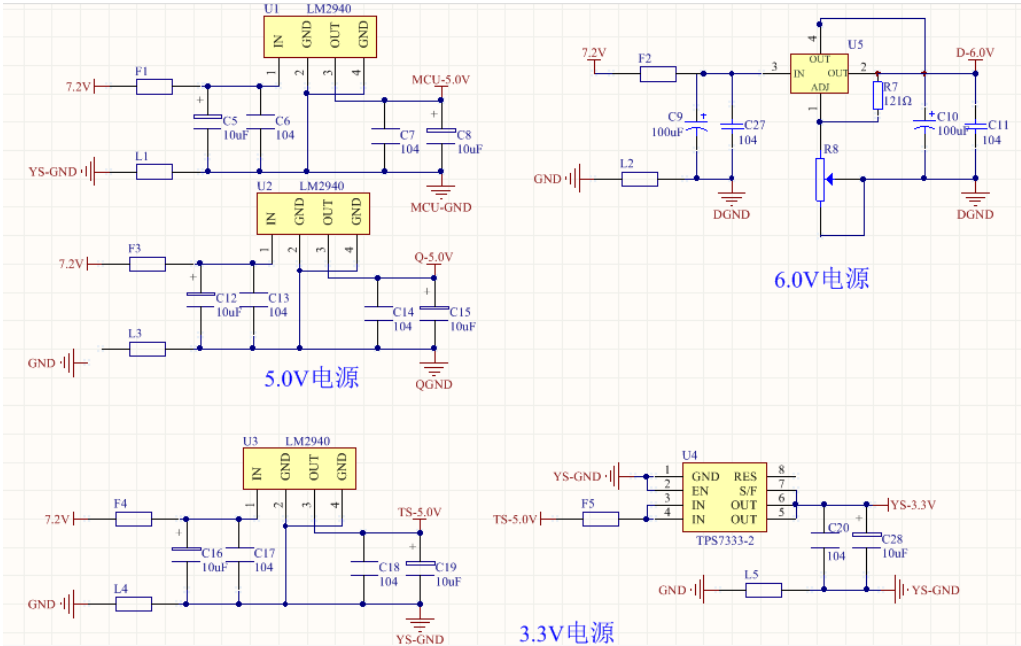


图 3.1 电源模块各电路

3.2 电机驱动模块电路

本次摄像头组由于B车只有一个电机，所以只需要驱动 1 个电机，驱动的设计尤为重要。常用的电机驱动有两种方式：一、采用集成电机驱动芯片；二、采用 N 沟道 MOSFET 和专用栅极驱动芯片设计。市面上常见的集成 H 桥式电机驱动芯片中，BTN7971 型芯片性能较为出色，该芯片具有完善的过流、欠压、过温保护等功能，使用 PWM 调制输入频率最高 25KHz，内部 MOSFET 导通电阻为 16 毫欧，具有最大 71A 的连续工作电流。使用集成芯片的电路设计简单，可靠性高。

由于分立的 N 沟道 MOSFET 具有极低的导通电阻，大大减小了电枢回路总电阻。另外，专门设计的栅极驱动电路可以提高 MOSFET 的开关速度，使 PWM 控制方式的调制频率可以得到提高，从而减少电枢电流脉动。并且专用栅极驱动芯片通常具有防同臂导通、硬件死区、欠电压保护等功能，可以提高电路工作的可

可靠性。

1. 专用栅极驱动芯片的选择:

IR 公司号称功率半导体领袖,所以我们主要在 IR 公司的产品中进行选择。其中 IR2104 型半桥驱动芯片可以驱动高端和低端两个 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。使用两片 IR2104 型半桥驱动芯片可以组成完整的直流电机 H 桥式驱动电路。由于其功能完善,价格低廉容易采购,所以我们选择它进行设计,如图3.2所示。

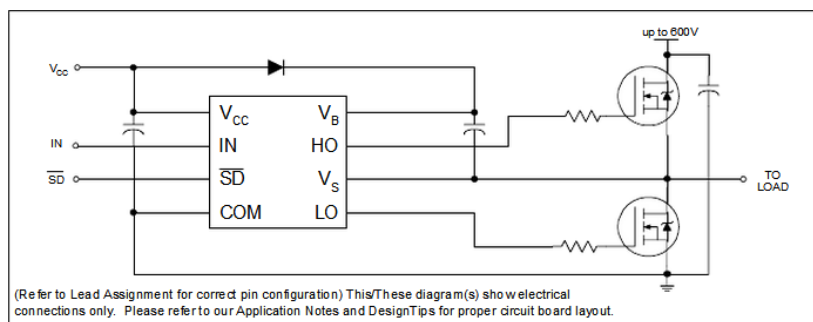


图 3.2 IR2104 应用图

2. MOSFET 的选择:

选择 MOSFET 时主要考虑的因素有:耐压、导通内阻和封装。智能车电源是额定电压为 7.2V 的电池组,由于电机工作时可能处于再生发电状态,所以驱动部分的元件耐压值最好取两倍电源电压值以上,即耐压在 16V 以上。而导通内阻则越小越好。封装越大功率越大,即同样导通电阻下通过电流更大,但封装越大栅极电荷越大,会影响导通速度。常用的 MOSFET 封装有 TO-220、TO-252、SO-8 等,TO-252 封装功率较大、而栅极电荷较小。于是我们最终选择了 IR 公司 TO-252 封装的 LR7843 型 N 沟道 MOSFET, $V_{DS} = 55$ 伏、 $R_{DS(on)} = 8.0$ 毫欧、 $I_D = 110$ 安。

3. 控制逻辑电路设计:

IR2104 的控制信号有两个管脚: IN 和 SD。IR2104 输入输出信号关系图如图 3.3 所示:

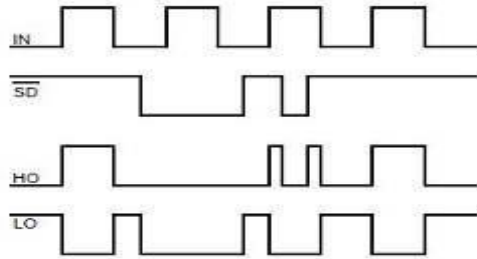


图 3.3 IR2104 输入输出关系

而当两片 IR2104 驱动如图 3.4 所示可逆桥式电路时，其真值表为表 3.1：

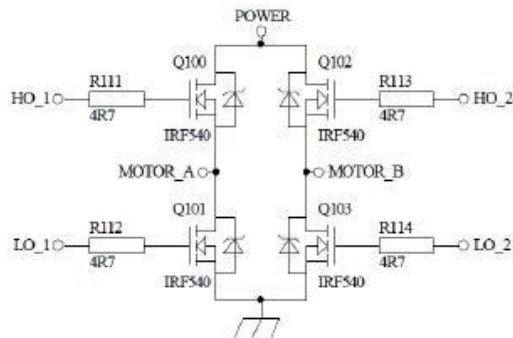


图 3.4 可逆桥式电路

表3.1 可逆桥式电路中IR2104输入输出信号真值表

状态	输入				输出			
	IN1	SD1	IN2	SD2	H01	L01	H02	L02
正转	H	H	L	H	H	L	L	H
反转	L	H	H	H	L	H	H	L
上桥臂制动	H	H	H	H	H	L	H	L
下桥臂制动	L	H	L	H	L	H	L	H
关闭	X	L	X	L	L	L	L	L

4. 驱动原理图如下：

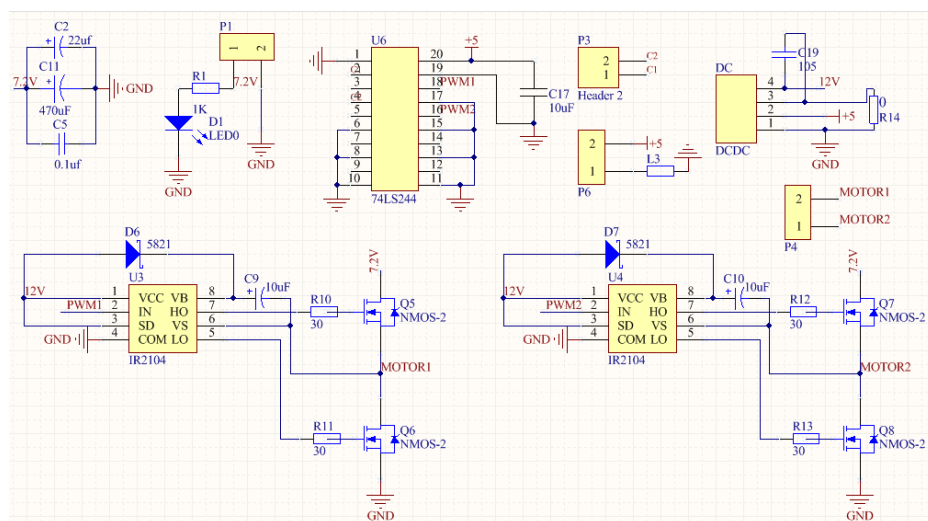


图 3.5 电机驱动电路

我们用两片IR2104 以及四个MOSFET组成的全桥的电机驱动电路，并且采用单极性的控制模式，PWM 的占空比在0%时电机速度为零，在0~100%时控制转速。另外为了是从MCU传过来的信号驱动力更强和隔离信号，我们特意用了一个74244芯片。事实证明，这种方法能比较好的起到上述提到的作用。

3.3 摄像头图像采集电路

(1) CCD 电荷耦合器件

CCD 是一种半导体器件,能够把光学影像转化为数字信号。CCD 上植入的微小光敏物质称作像素(Pixel)。一块 CCD 上包含的像素数越多,其提供的画面分辨率也就越高。CCD 的作用就像胶片一样,但它是把图像像素转换成数字信号。CCD 上有许多排列整齐的电容,能感应光线,并将影像转变成数字信号。经由外部电路的控制,每个小电容能将其所带的电荷转给它相邻的电容。构成 CCD 的基本单元是 MOS(金属-氧化物-半导体)结构。CCD 的基本功能是电荷的存储和电荷的转移。工作时,需要在金属栅极加一定的偏压,形成势阱以容纳电荷,电荷的多少基本与光强成线性关系。电荷读出时,在一定相位关系的移位脉冲作用下,从一个位置移动到下一个位置,直到移出 CCD,经过电荷-电压变换,转换为模拟信号。

(2) CMOS 光电传感器

CMOS 是一种高集成度大规模集成电路,仍使用光敏元件为感光器件,光敏元件在排列方式上与 CCD 电荷耦合器件相同,只是在光电转换后信息传送方式不同。CMOS 芯片内部提供了一系列控制寄存器,通过总线编程来对自动增益、自动曝光、白色平衡、 γ 校正等功能进行控制,编程简单、控制灵活。直接输出的电视机制式图像信号可以很方便地和后续处理电路接口,供处理器 AD 模块对其进行采集和处理器处理。

经过比较与选择,我们决定使用CMOS 摄像头OV7725作为路径识别的传感器。其接口如下图:

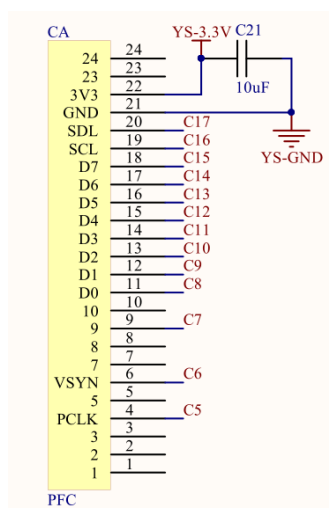


图 3.6 摄像头接口

3.4 编码器电路

为了使得模型车能以闭环控制系统运行,就需要测的模型车的运行速度,以使模型车在急弯以及长直到时速度不至于过快而冲出赛道。如果开环控制电机转速,速度的提升会比较难,通过速度检测,对模型车速 度进行闭环反馈控制,就可以使车模速度提升更快,控制也更稳定。

常见的速度检测传感器有,测速发电机,光电测速传感器,霍尔传感器和光栅编码器等。相比较而言,测速发电机的体积和质量比较大,用测速发电机会对模型车的机械结构产生比较大的影响。经过我们的多次比较最后我们打算使用龙邱的mini 512线编码器。其接口如下图:

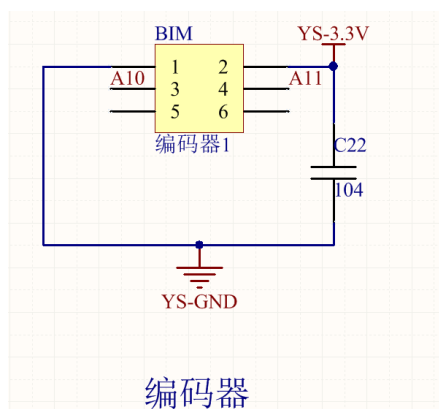


图 3.7 编码器接口电路

3.5 液晶显示接口电路

液晶屏具有显示内容直观，可显示图像和汉字的优点。在车身添加液晶屏模块可以使调试更加方便，因此本次车身上我们添加了液晶屏模块。在调试过程之中，我们需要实时的了解与掌握一些车的运行状态，比如当前赛道类型、还可以显示调试时需要修改的参数等等，调试时用液晶将这些参数显示出来，让我们实时的监测车的状态，从而做出判断，在很大程度上方便了对车的调试。下图为液晶显示接口电路：

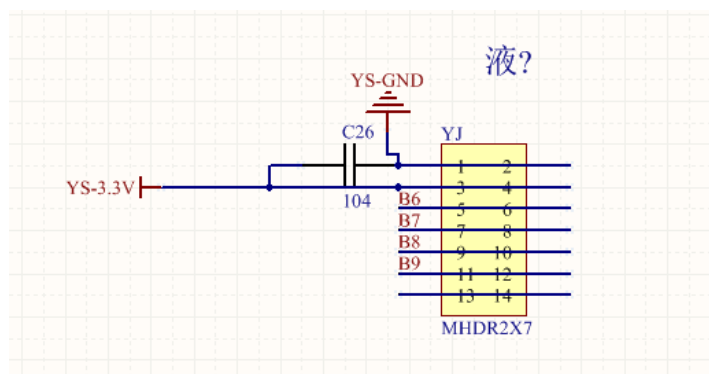


图 3.8 液晶显示接口电路

3.6 按键电路

有时候需要对参数作修改处理，如果每修改一个数据就下载一次程序的话，就会浪费时间，这时应用键盘，它就起到一个人机交互的作用。而且在比赛前的调试过程中，如果发现模型车不适应赛道，还可以临时对部分的参数进行修改，使得模型车的适应性更强。

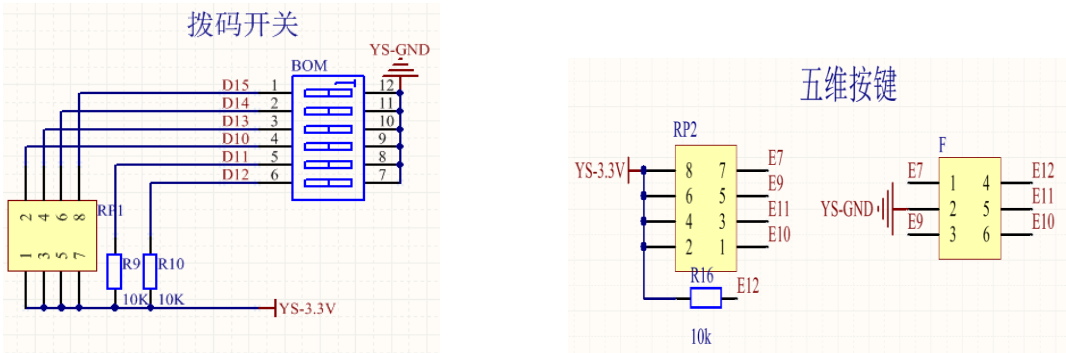


图 3.9 按键电路

3.7 蜂鸣器

蜂鸣器，这个可以用在模型车对赛道的识别上，这样就可以十分直观的看出来模型车是否识别赛道以及识别时间。这样就可以更加方便的调试。

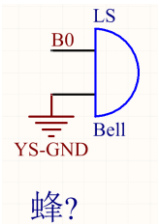


图 3.10 蜂鸣器电路

3.8 指示灯

指示灯电路跟蜂鸣器一样可以实时显示特定的某个参数的状态变化。

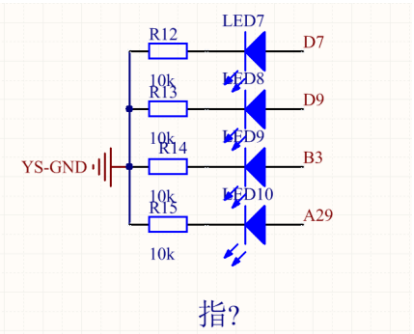


图 3.11 指示灯电路

3.9 SD卡电路

在调车的过程中，我们不能做到每个时刻都可以知道摄像头采集到的图像是什么，所以为了方便采集摄像头每个时刻的图像我们采用了SD卡存储图像信息，这样在车子跑完整个赛道后我们还可以进行查看分析车子在行驶的时候到底出了什么问题。

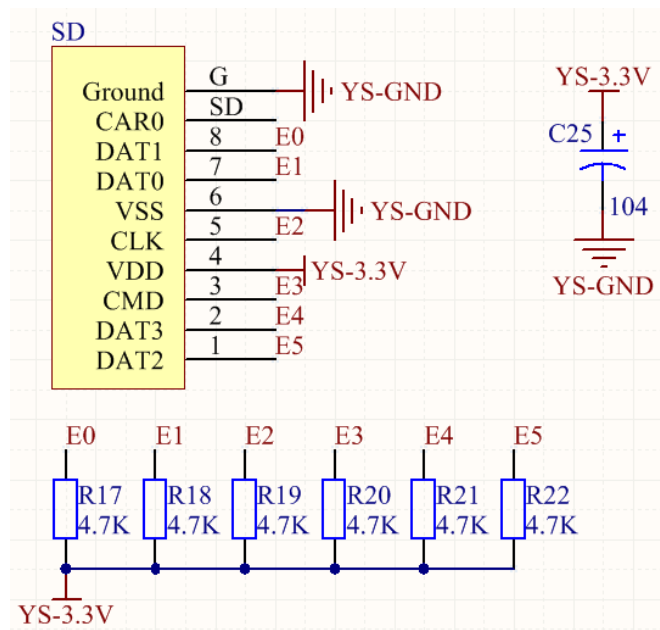


图 3.12 SD 卡电路

3.10 核心板接口电路

在硬件电路设计的时候考虑到电路的稳定性以及车模的结构局限性，我们打算使用现成的核心板，这样可以保证整个电路系统工作的稳定性，同时也方便更换。所以设计的时候，我们用适当的接口将核心板与主控板连接起来。

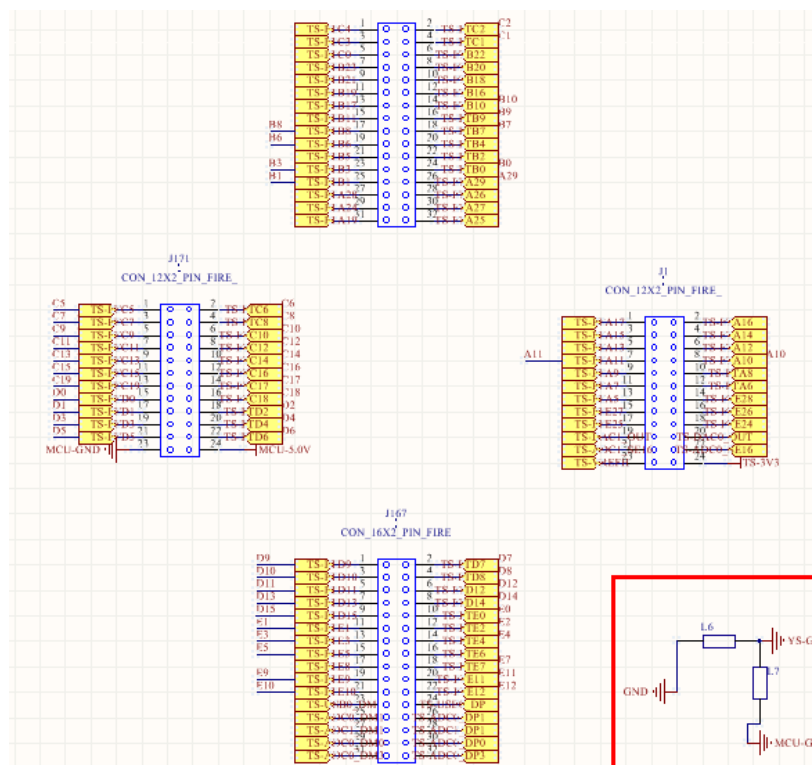


图 3.13 核心板接口电路

3.11 其它

在对小车的参数进行调整的时候，由于蓝牙的传输速度比较慢，无法在模型车运行的时候修改参数，所以我们就用 NRF24L01 无线模块进行修改，这样可以大大提高修改参数的效率，从而加快调车的进度。

我们给模型车配备了一个蓝牙，虽然蓝牙修改参数速度很慢，但是在图像处理的时候还是需要用蓝牙来发图的，把小车看到的图，发到电脑上的上位机，然后再来对着图片进行图片处理，这样一来也可提高对图片处理的速度及精度。

3.12 小结

在硬件设计的时候考虑到与模型车机械结构的契合型，我们板子的PCB 图进行了多次修改，由于做双面板的经验不是很足，所以第一次画板子的时候花的时间要稍微长一点。另外有些器件用得不合理的地方也经过了多方面的分析和修改，最终得到的板子耐用有效。这期间也要感谢同学和老师的帮助，对我提出的问题能及时正确的回答。

第四章 机械结构的调整与优化

智能车的整体参数，包括车体重心、高度、传感器排布方式等，都对整个智能车系统的稳定运行起着至关重要的作用。因此，对智能车机械系统的调节，有助于模型车更快更稳定的运行。

模型车的布局以精简、可靠、稳定为前提,通过对小车的布局，尽量保证小车左右机械结构对称，以及寻找一个合适的重心，保证模型车既能稳定快速前进，又能在转弯时流畅。

智能汽车各个统的控制都是在机械结构的基础上实现的，因此在设计 整个软件架构和算法之前一定要对整个模型车的机械结构有一个全面清晰 的认识，然后建立相应的数学模型，从而再针对具体的设计方案来调整赛 车的机械结构，并在实际的调试过程中不断的改进优化和提高结构的稳定性。本章将主要介绍模型车的机械结构和调整方案。

根据竞赛规则，各个竞赛组别所使用的车模均为组委会指定车模，摄像头组所使用的车模为 B 型，其初始的基本结构如下图所示：

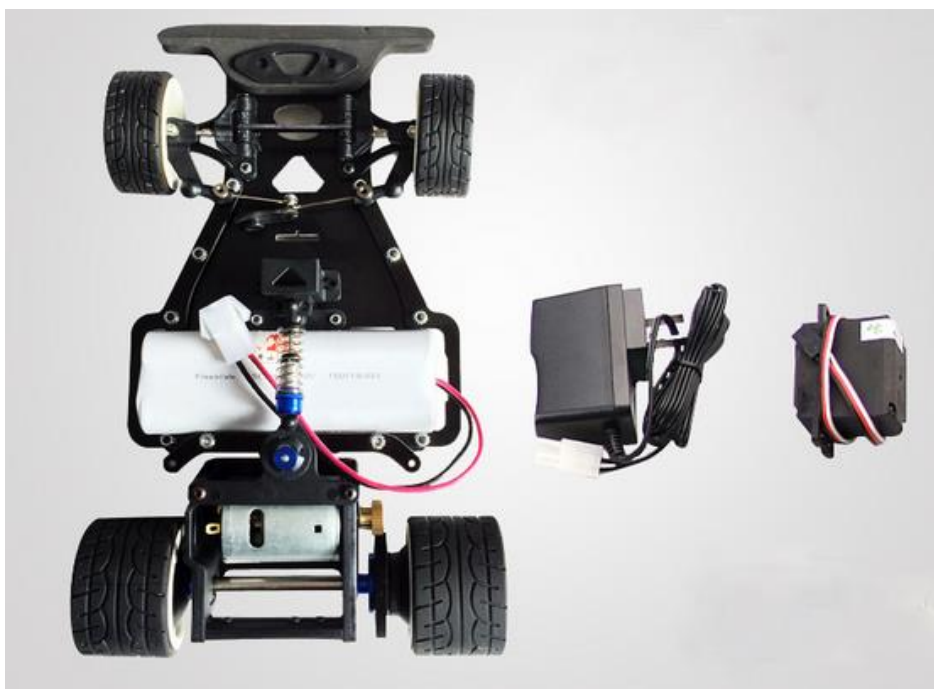


图 4.1 第十一届智能车竞赛摄像头组 B 型车模

4.1 B车底板的固定

B车模买来的时候后轮机械图前面的底板时可拆卸的。由于当时经验不足，我们总共设计了两套方案。一种是将后轮机械与底板牢牢固定，这样做可以使车身在上下坡和转急弯的时候不容易出现抖动，目的是使摄像头不会出现晃动，从而减少舵机乱摆的现象，缺点是转弯比较生硬；另一种是采用后轮机械与底板与摄像头之间进行动态平衡的机械形式，我们在后轮机械与底板的交界处采用三轴悬挂弹簧减震的方式来固定小车，这样做的优点是小车在转弯的时候比较平滑，车身不容易左倾或右倾减小了车轮的负担，使轮胎磨损更加均匀。下面是两种机械结构，最后我们选择了其中一种比较适合我们风格的机械结构。

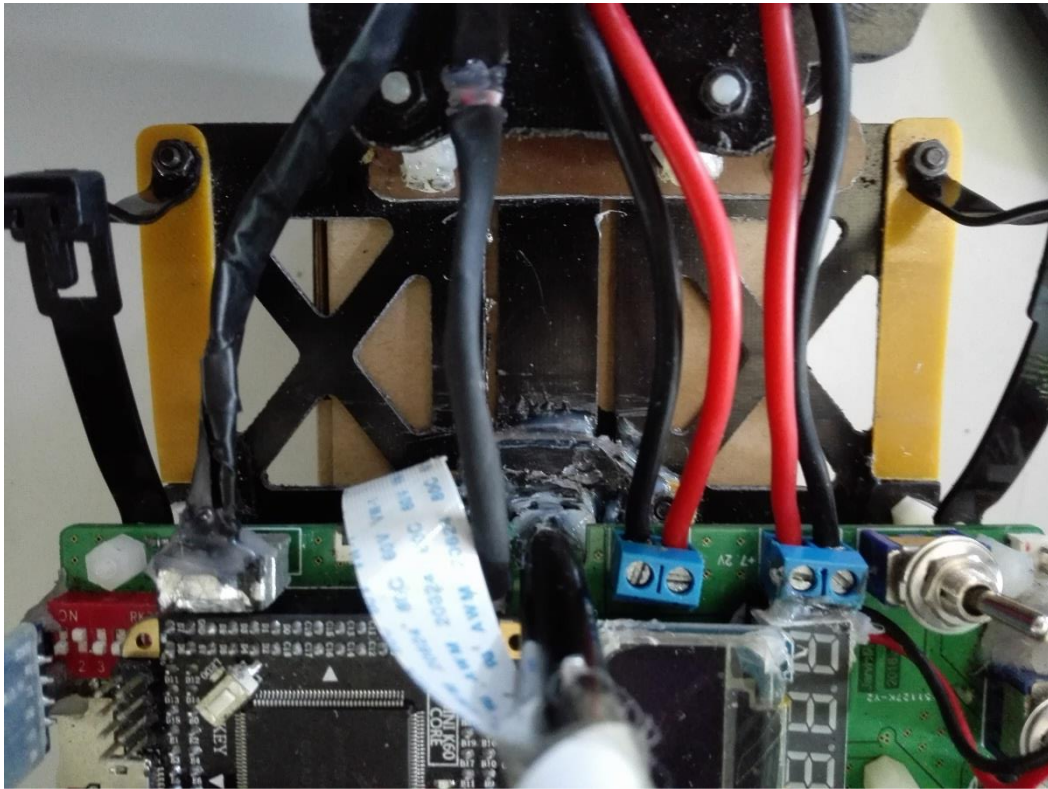


图 4.1 固定式

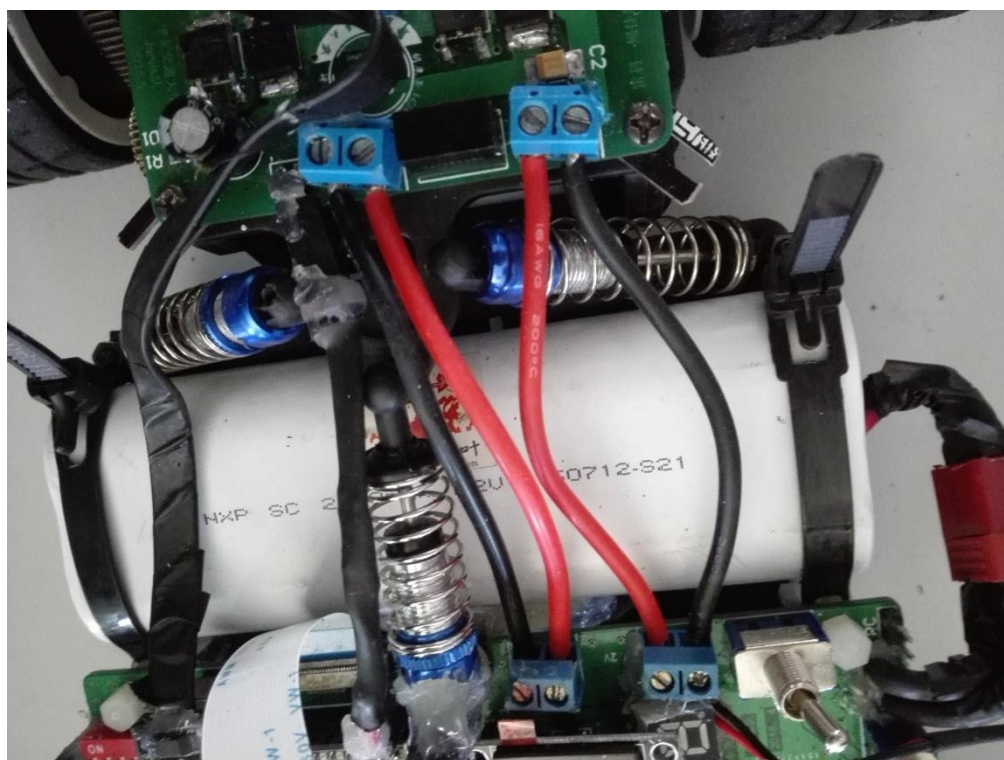


图 4.2 三轴悬挂式

4.2 电池的安装

电池是占模型车车重的主要部件，它的安装对重心的影响极大，车模重心太高在高速过弯时可能会出现侧翻，降低重心能增加过弯的稳定性。所有模型车的部件安装都要尽量保证小车的重心足够低，而电池是这么多零件中最重的，所以电池的安装位置对模型车有很大的影响。考虑到车子可能出现侧滑以及甩尾等现象，同时也为了配合摄像头杆子的位置，我们决定把电池放在模型车的后面。如图 4.2:

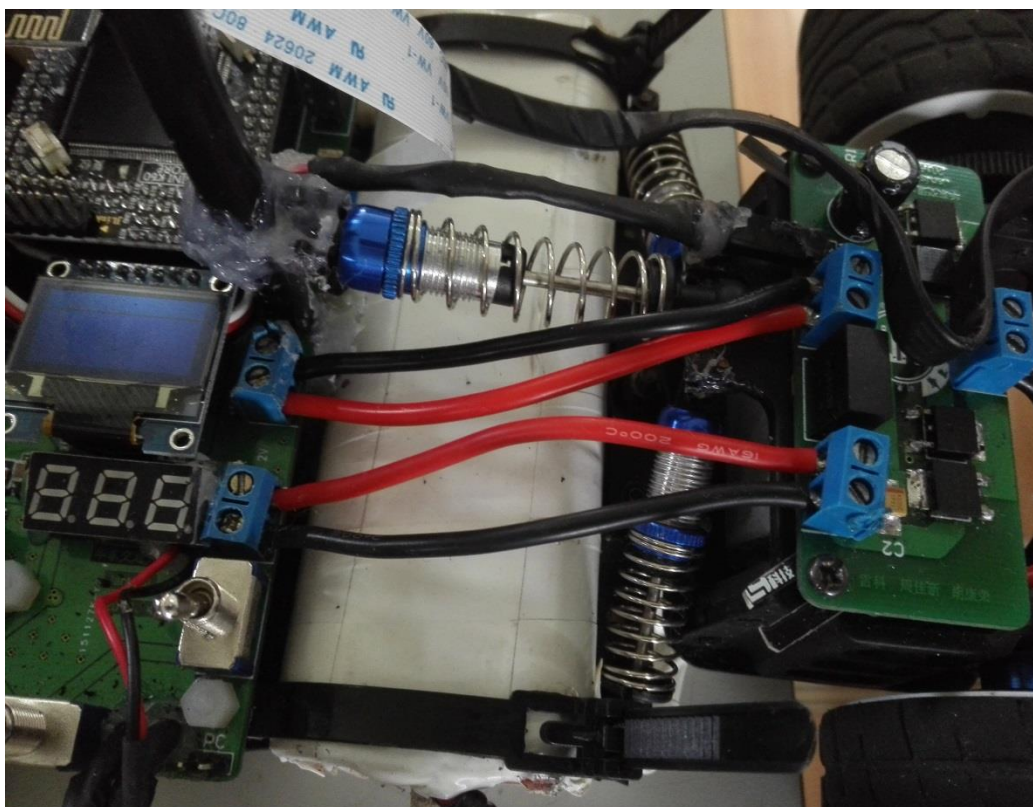


图 4.3 模型车电池安放位置

4.3 编码器的安装

选用可以 5V 工作电压的 512 线 MINI 增量式编码器进行速度的测量，保证测量的精度，也是为了后轮能劳抓地面。速度传感器用螺钉通过支架固定在后轮支架上，这样固定好之后，就有了较高的稳定性。然后调节编码器齿轮，使其与差速齿轮紧密咬合，增大测速的精确性，但是咬合过紧也增大了摩擦，减小了对电机做功的利用率，影响模型车的快速行驶，因此减小摩擦同时增强齿轮间的咬合是我们主要考虑的因素。具体的测量方法可以用一张A4纸来测量，以A4纸刚好能通过编码器齿轮与后轮齿轮之间的缝隙为基准，另外还可以用齿轮润滑油涂抹齿轮。经实际检验，效果可靠有效。编码器安装示意图如图 4.3 所示：

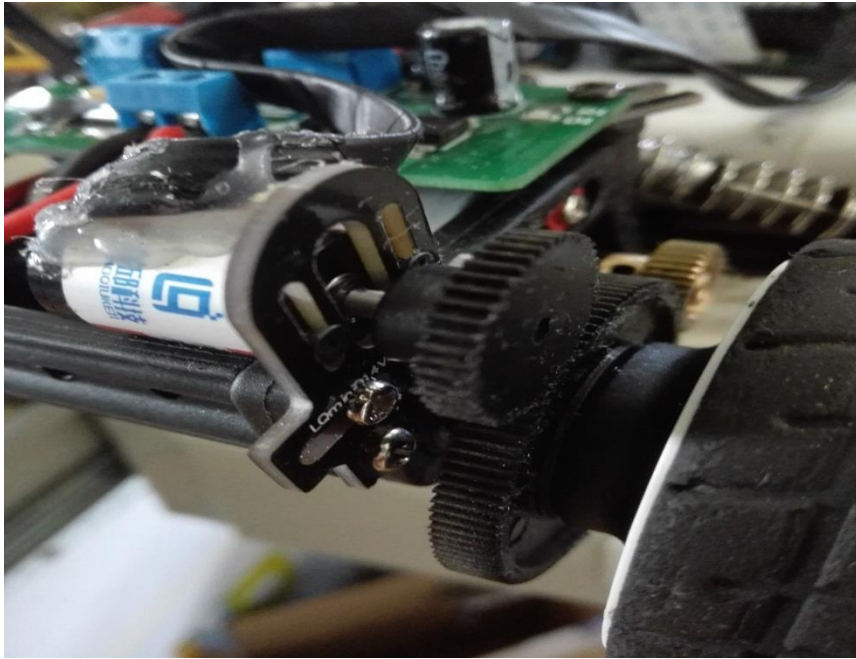


图 4.4 编码器安装位置图

4.4 摄像头的安装

摄像头为采集赛道边界模块，赛道宽 45cm，其中两边都有 2.5cm 的黑线，为了更好地采集到赛道两边的黑线，就必须使得摄像头的位置达到一定高度，而且有一定的倾角，这个决定了模型车的视野的大小。若视野过窄会导致模型车调整方向时的死区问题。另外还需要考虑到摄像头对模型车的重心的影响。

经过多次的测试与验证，模型车的摄像头安装位置如图 4.4：



图 4.5 摄像头的安装

4.5 舵机的安装

舵机转向使整个控制系统中延迟较大的一个环节，为了减小此时间常数，通过改变舵机的安装位置，而并非改变舵机本身结构的方法便可以提高舵机的响应速度。通过分析舵机控制转向的原理可以发现，在相同的舵机转向条件下，转向连杆在舵机一端的连接点离舵机轴心距离越远，转向轮转向变化越快。因此相当于增大了力臂长度，便可以提高线速度。因此我们在安装舵机时改变了舵机原有的安装方式，将舵机立起来安装在车的中轴线上，使得左右两轮的力臂一样长，并且适当调节转向两岸的长度，这样便使得小车转向更灵敏，有利于小车转向。

同时我们也采用了舵机主销角度改变的方法来增大车在转弯时的稳定。

主销（Kingpin），转向轮转向时的回转中心，转向轮的转向轴线的代名词，转向轮在转向时，是以主销为轴线向左右转动的。

舵机的摆放方式可以采用主销后倾的方法，从车的侧面看去，主销轴线与通过前轮中心的垂线之间形成一个夹角，即主销后倾角。主销后倾的作用是增加汽车直线行驶时的稳定性和在转向后使前轮自动回正。主销后倾主销后倾角的存在

可使车轮转向轴线与路面的交点在轮胎接地点的前方，可利用路面对轮胎的阻力产生绕主销轴线的回正力矩，该力矩的方向正好与车轮偏转方向相反，使车保持直线行驶。后倾角越大车的直线行驶性越好，转向后回复性也越好，但主销后倾角过大会使转向变得沉重；主销后倾角过小，当汽车直线行驶时，容易发生前轮摆振，转向盘摇摆不定，转向后转向盘自动回正能力变弱。后倾角一般在1—3度。具体可以根据自身来调节。

主销内倾，是将主销（即转向轴线）的上端向内倾斜。主销内倾的作用是使车轮转向后能及时自动回正和转向轻便。主销内倾不能过大，否则转向过于沉重。调整范围在0—10度。

主销后倾和主销内倾都有使转向轮自动回正的作用。但主销后倾的回正作用与车速有关，而主销内倾的回正作用与车速无关。因此，高速时主要靠主销后倾的作用，而低速时主要靠主销内倾的作用。

根据实际情况，我们将主销倾角设为 0 度

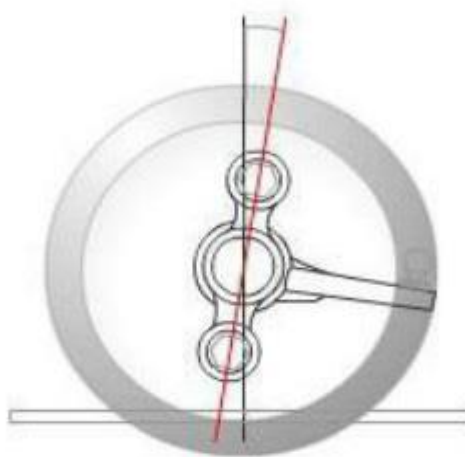


图 4.6 主销后倾图

考虑到实际买的舵机摆臂不够理想，我们自己用PCB做了一个。下面是舵机安装示意图：



图 4.7 舵机的安装

4.6 前轮的机械结构调整

为了使车子在过弯的时候能前轮受力更加科学，我们将前轮的车轮朝向稍稍做了调整。

前轮的朝向主要有两种一个是前轮外倾，一个是前轮前束。

前轮外倾，车轮在安装后，其端面向外倾斜，即车轮所处平面和纵向垂直平面间的夹角。轮胎呈现“八”字形张开时称为负外倾，而呈现“V”字形张开时称正外倾。作用是为了提高车轮工作时的安全性。由于主销与衬套之间、轮毂和轴承等处都存在着装配间隙，这些间隙在不同程度上影响着车轮正常工作。当车轮有一定外倾角时，轮胎的中心线与地面相交于A点，转向主销轴线与路面相交于B点。这两点之间的距离称为偏移。由于车轮在转向时，是绕主销轴线的中心和半径转动的，因此在转向主销周围因受到轮胎滚动阻力的作用，会产生一个大力矩，而增加转向力。偏移越大，产生的力矩也愈大。当具有外倾角时，可使偏移量减小，所以能减少转向力。此外，当车轮外倾时，在垂直载荷的作用下会产生一施加于芯轴上的分力，使车轮向内压在轴承上，以防止车轮甩脱。

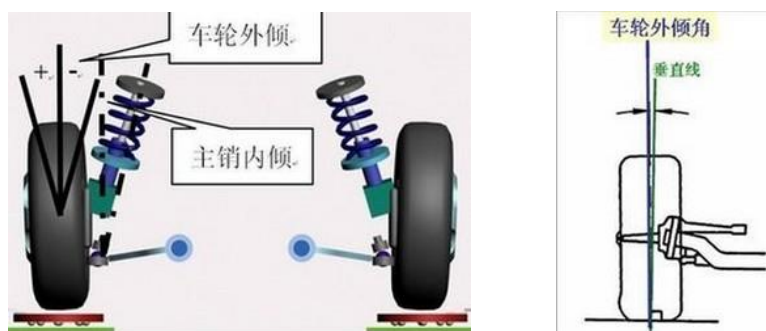


图 4.8 车轮外倾示意图

前轮约束，是使汽车两前轮的前端距离小于后端距离。其距离之差叫做前束值。从汽车的上面往下看，左右两个前轮形成一个开口向后的“八”字形。作用：

①前轮外倾有使前轮向外转向的趋势，前轮前束有使车轮向内转向的趋势，可以抵消因前轮外倾带来的不利影响，使车轮直线滚动而无横向滑拖的现象，减少轮胎磨损。

②悬架系统铰接点的变形，也使前轮有向外转向的趋势，也要靠前轮前束来补偿。前轮前束值一般不大于8mm，其值由调节横拉杆的长度来保证。现代轿车的前束值有缩小的趋势，前轮外倾为负值时前束也为负值，即变为前轮后束。

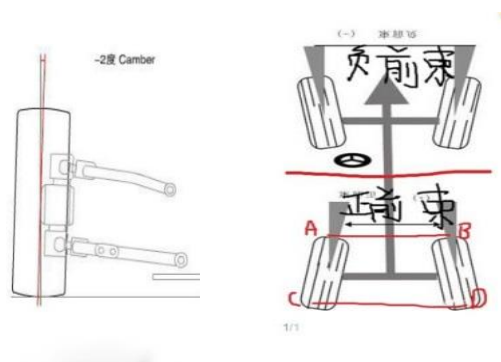


图 4.7 前轮前束示意图

最后我们采用上述两种方法对前轮进行了调整。结果如下图：



图 4.8 前轮机械结构图

4.7 主控板的安装

主控板当时设计的时候是根据机械结构来确定形状与大小的，由于我们所用的核心板比较大，因此整个板子看起来比较细长，但是也能比较好地契合车模。同时也能起到稳定车的作用。驱动板固定在车模后方，有固定后轮支座的作用。

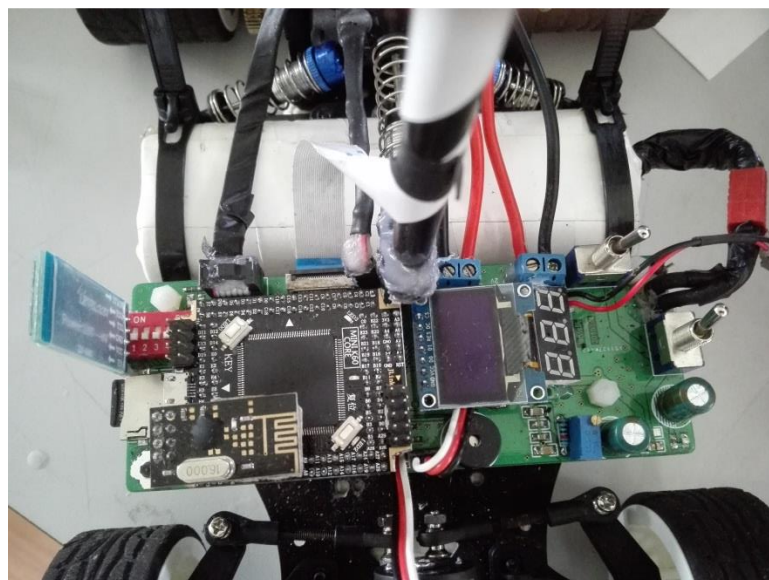


图 4.9 主控板的安装

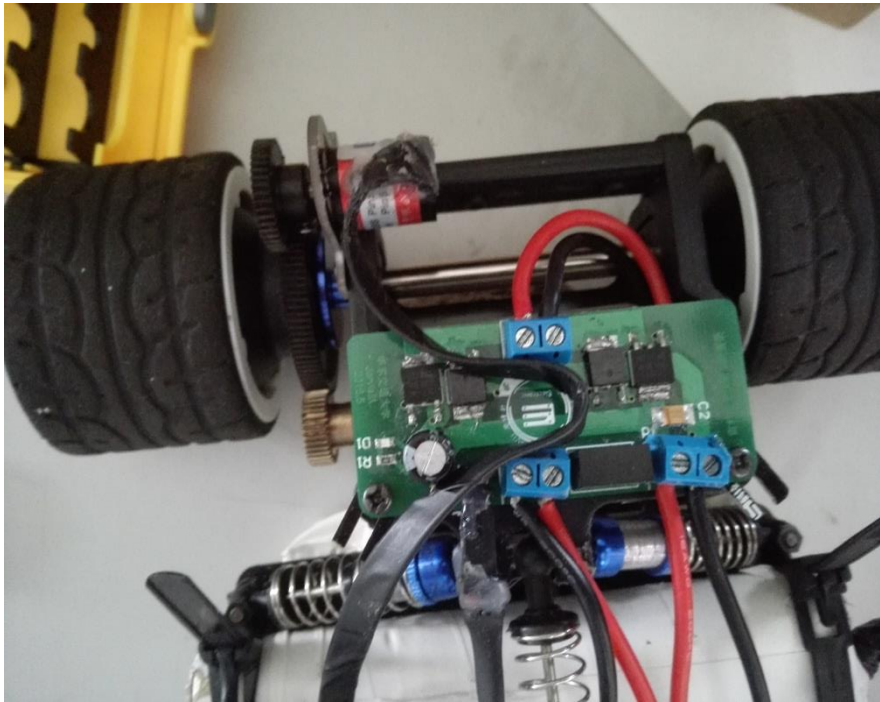


图 4.10 驱动板的安装

4.8 小结

在实际对车模结构的调整与优化过程中我们遇到过很多麻烦，比如某个齿轮了，多级位置装得有点儿偏了，以及摄像头杆子固定的位置与理想的有偏差等等。但是在后面调车的过程中调车感觉一步步变好，逐渐能迅速指出车模结构上的问题，并对车模进行相应的调整以达到最优的适合车跑的机械结构。

第五章 智能车控制软件设计

5.1主程序流程图：

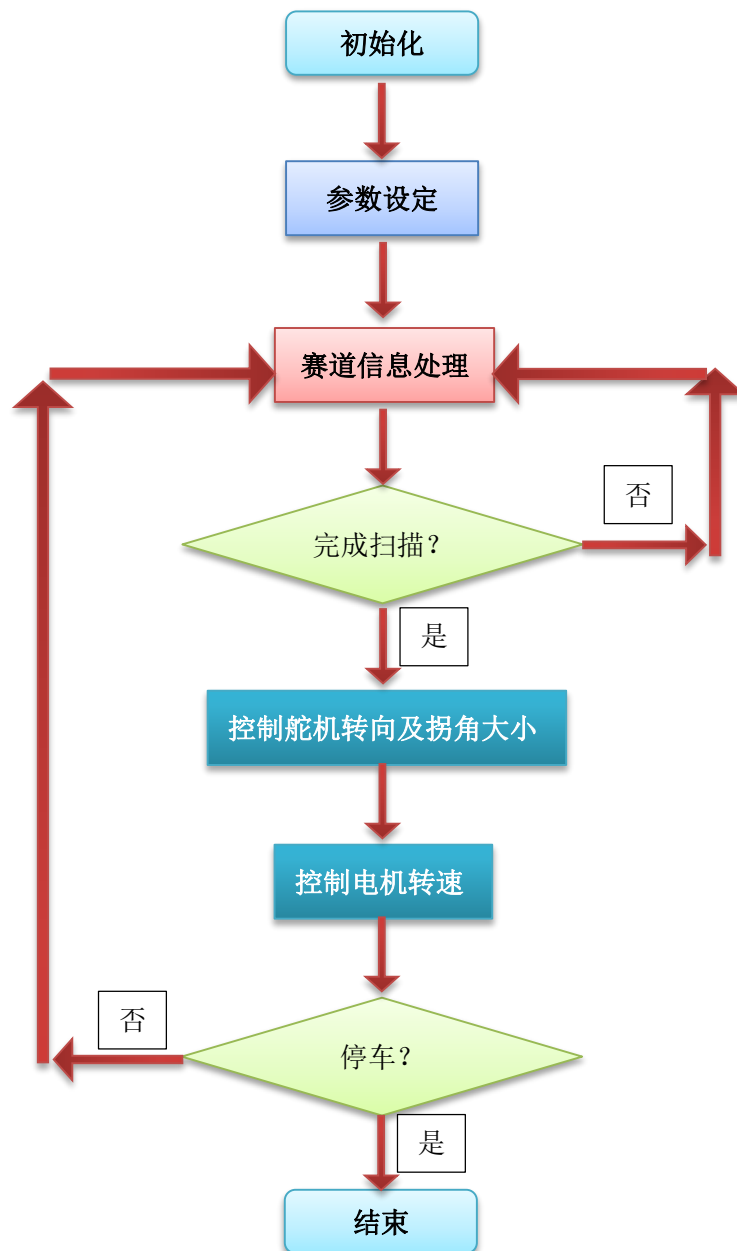


图 5.1 程序流程图

5.2 图像采集与处理

5.2.1 图像的采集

由于我们采用了 k60 芯片和带有硬件二值化的 ov7725 的摄像头,我们可以采用 dma 采集赛道图像,dma 采集效果比一般的读取 io 口效率高,并且 OV7725 没有噪点,效果好。图像采集过程中 dma 独占内存,无法进行图像处理,我们选用的摄像头采集速度可以达到 150 帧,于是我们抛弃了一张图片,变成每秒钟 100 张,这样我们就有了充足的时间进行赛道信息处理和车体姿态控制。由于 ov7725 可以通过 sccb 配置寄存器,改变采集图像的大小,简化了采集过程,开始我们选用了 80*60 的图像大小

经过一段时间发现效果并不理想,赛道信息比较粗糙,而且存在较为严重的反光现象,对赛道信息提取和识别有较大干扰,于是我们通过 sccb 配置寄存器把图像扩大到了 160*120,并去掉图像最左右和最靠近车的部分,赛道的细节部分得到了很好的展现,对之后车的提速有很大的作用。

5.2.2 图像的处理

在采集出图像信号后需要对其进行处理以提取赛道信息,同时,由于杂点、交叉线、起跑线、光线、赛道连接处、赛道上的杂物,以及赛道以外图像的干扰,得到的图像有很多干扰,软件上需要排除干扰因素,对有效赛道进行识别,并提供尽可能多的信息供决策使用。图像信号处理中提取的赛道信息主要包括:

- (1) 每一行的赛道左、右边线位置以及宽度;
- (2) 每一行的赛道中心位置;
- (3) 赛道的曲率;
- (4) 赛道的有效行;
- (5) 赛道类型;

摄像头组的核心部分为左右边线的提取,从而推算出中心线的位置和赛道曲率,引导车子的行进方向。

边线提取,中心线推算思路:

1. 近处图像较远处图像可靠,所以边线由近到远提取;

2. 近十行先从中间向两边进行边沿搜索，找到边沿，确保近处图像有效；
3. 之后的行采用边沿跟踪，根据上一行的黑线位置动态地确定本行黑线的搜索范围，搜索黑线，节省时间；
4. 若连续几行的左右线都找不到，判断为十字弯，进行左右线的延伸，直至再找到有效的边线；
5. 若左右边线均找到，该行中心线直接为左右边线等权重加权；若某一边线丢失，则根据前面有效行的赛道宽度和另一边线的位置对中心线进行推算；
6. 推算完中心线后可对中心线进行一定的修正，更符合实际。

5.2.3 赛道曲率的计算

在开始处理图像时，我们参考了学长的赛道类型识别方法，当时在实际调试过程中我们发现此方法并不够有效，出现误判的情况较多。在查找有关资料后我们决定引入赛道曲率这个变量。

首先，在提取完赛道边线后，计算出赛道的中线；

其次，使用一段弧线上的三点进行曲率计算。根据赛道有效前瞻的不同，我们动态的从赛道可靠行内提取出四个点，其中两个关键位置的点重复利用，利用三个间距较大的点（A、C、D点）计算整体的曲率，靠近赛道顶部的三个点（A、B、C）校正距离计算赛道前端曲率。将两个曲率值综合，就能够较准识别赛道。

5.2.4 赛道分类

一. 处理思路：

计算完中心线后，根据有效行大致把赛道类型分为弯道以及入弯、直道、等等经过对赛道信息的整理分析，分类思路如下：

当有效行大于25时，判断为弯道；反之有效行较小，前瞻较大而判断为其他类型。然后再结合赛道曲率进行更为细致的判断。

对于特殊的赛道类型还需要结合其他信息进行比较判断，例如判断左右边线的情况可以识别出障碍以及十字路等。

二. 直道

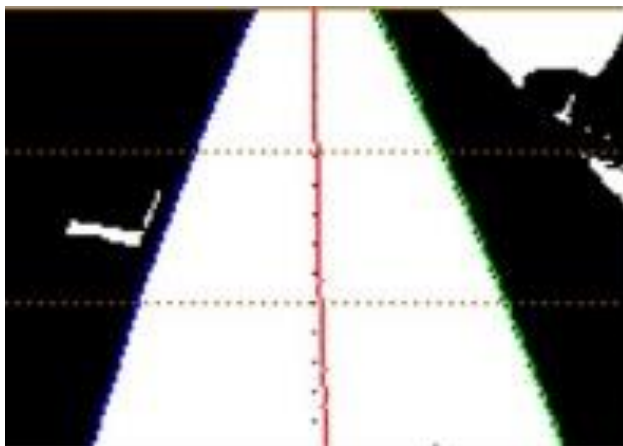


图 5.2 直道图

直道是最基本的赛道类型，通过提取赛道边线后计算出的中线不会出现较大的弯曲现象，而且有效行也会在图像顶部附近，经过对图像的分析，我们发现当有效行大于25，赛道整体曲率半径很大，即可以认为当前赛道类型为直道。

三. 入弯

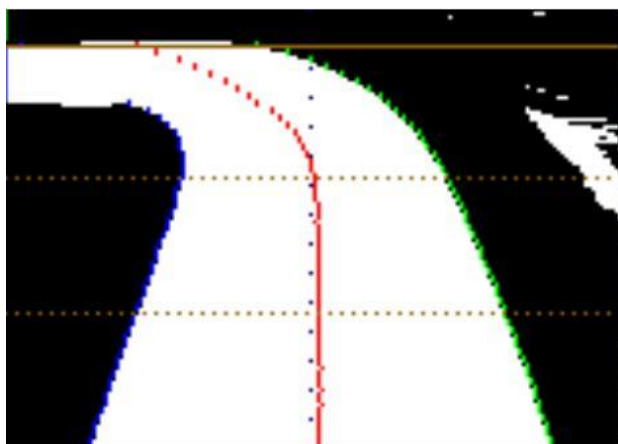


图 5.3 入弯图

由于赛道中存在弯道因素，所以也相应的存在入弯的这种情况。之所以要判断该类型，是由于舵机弯道打角要比直道大很多，而且识别出入弯道，也有利于弯道的控制。之前我们进行了赛道的曲率计算，入弯的特点为底部和中部的偏差都较小，而顶部的偏差很大。同时中线的顶部的曲率很大，而整体的中线曲率要小于直道。判断出入弯后，可以适当调整舵机提前打角，调整小车入弯的姿态。

四. 弯道

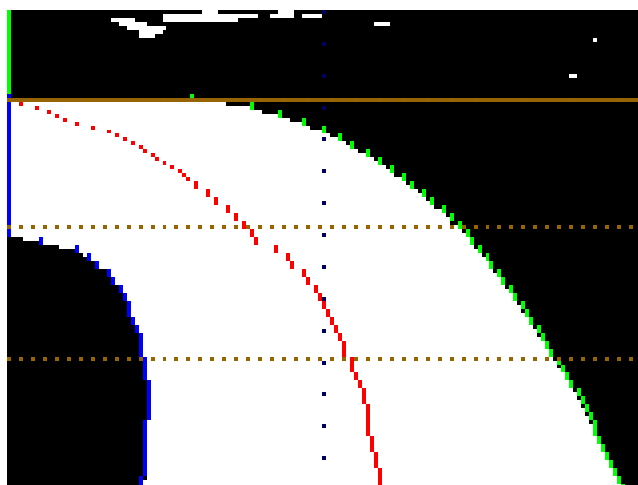


图 5.4 弯道图

对于弯道而言，是限制小车速度的主要因素之一，小车整体速度很大程度上取决于弯道的速度。而弯道的速度影响因素很多，轮胎，前轮的倾角，入弯的姿态，弯道的打角等等；所以识别出弯道很有必要，专门给弯道时舵机合适的打角参数有利于过弯。在弯道里，有效行较大，而且顶部偏差很大，很容易识别出来

五. 十字

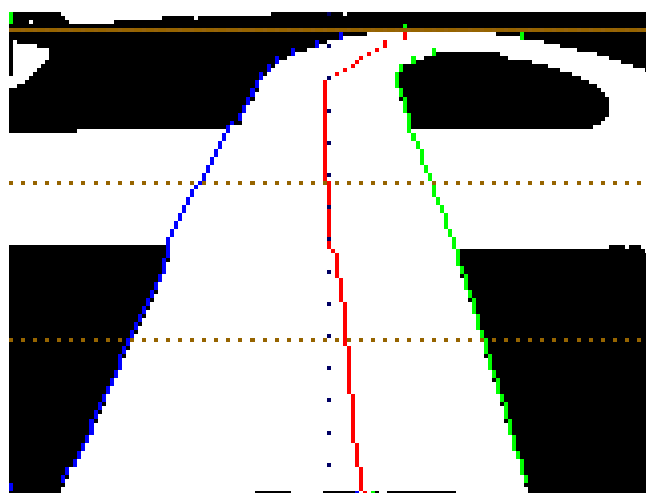


图 5.5 十字图

十字赛道类型很考验赛车的稳定性，所以特需要进行特定的识别，通过摄像头拍回来的图像可以看出，存在大量行左右边线丢失的情况——可以判断出现十字路；然后进行补左右边线，即等待边线又重新出现时，对边线进行连线。然后当作直道或者入弯处理。经实际调试发现，这种方法可以很好决解十字带来的

干扰。

六. 小S

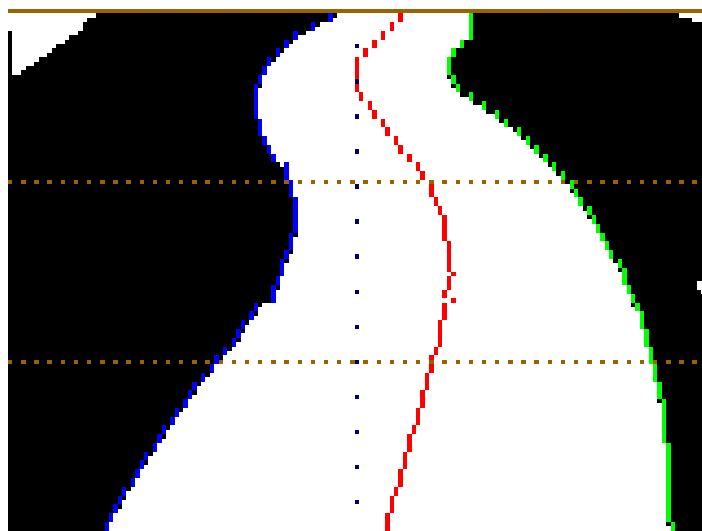


图 5.6 小 S 图

在小S里，如果不对其进行特定的识别，虽然不会出现小车乱跑的情况，但小车在小S 里会出现入弯和弯道的情况，小车会在小S里“扭来扭去”，对于竞速的智能车比赛来说，这是绝对要避免的，而根据左右边线的变化情况识别则可以很好判断出小S赛道类型。对于小车的提速有很大影响。

七. 障碍物

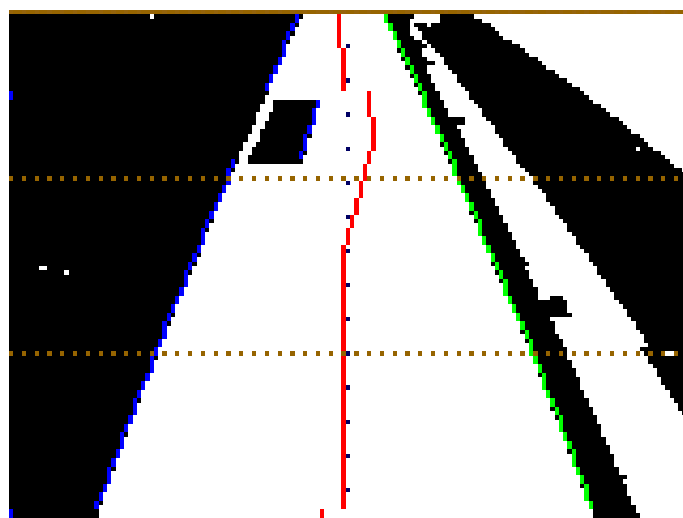


图 5.7 障碍物图

障碍物首先出现在第九届飞思卡尔智能车的比赛赛道上,对于 B 车而言,如

果不特别处理的话,会直接撞到障碍物上,即使小车没有冲出跑道,也会因此耽误很多时间。根据障碍的特点,边线会出现突变情况,并且赛道宽度会出现急剧变小情况;根据这个特点可以很好的识别到障碍物。

八. 起跑线

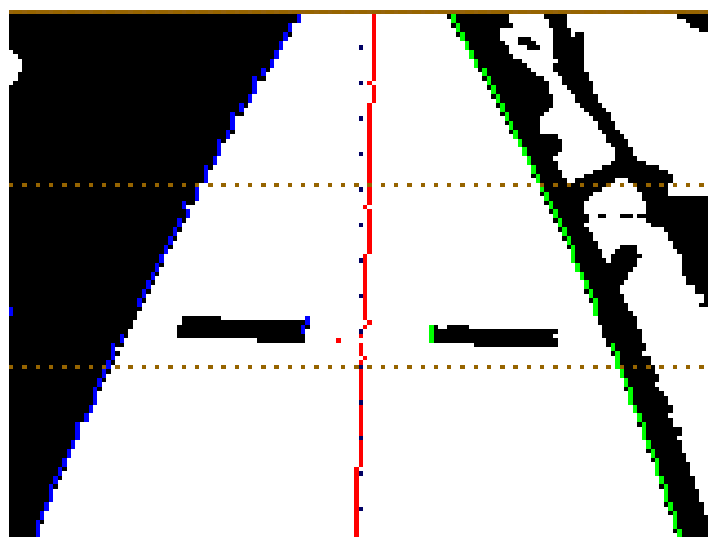


图 5.8 起跑线图

全国大学生智能车竞赛规定,智能车必须能够识别起跑线,并在跑完一圈以后能够自动停止在3米以内的赛道中,否则在本来的时间基础上加上1秒,如果加上1秒那必须将速度加大很多才能挽回。因此,对于起跑线的精准识别显得至关重要。我们通过对图像的分析发现,起跑线的颜色从左到右依次为“黑”→“白”→“黑”→“白”→“黑”→“白”→“黑”,因此我们可以利用这个特征来识别起跑线。即当这部分图像中出现这样的颜色规律时就判定为是起跑线。

九. 坡道

坡道对于智能车的稳定运行是一个不小的挑战,如果速度过快通过坡道时会出现飞坡的情况,当小车在空中时摄像头拍到的图像将是乱的,路径也没有参考价值,对后续小车的运行来说是很大的干扰因素。但坡道的识别也较为容易,即对赛道的宽度进行判断即可,宽度超过一定阈值,既可以认定当前赛道为坡道。

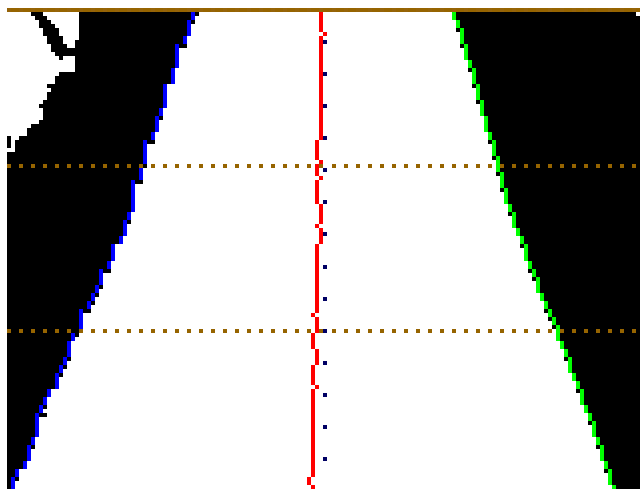


图 5.9 程序流程图

5.3 控制算法与函数

5.3.1 PID 控制算法介绍

智能车主要采用增量式PID控制和位置式PD控制，使智能车能够在赛道上平稳快速的行驶。

一、经典PID控制介绍

在工程实际中，应用最为广泛的调节控制规律为比例、积分、微分控制，简称PID控制，又称PID调节。PID控制器结构简单，稳定性好、工作可靠、调节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通过有效的测量手段来获取即不能得到精确的数学模型时，系统控制器的结构和参数必须依靠现场调试和经验来确定，这时应用PID控制技术最为方便。PID控制器是一种线性控制器，它根据给定值与实际输出值之间形成偏差，将偏差的比例（P）、积分（I）、微分（D）通过线性组合构成控制量，对被控对象进行控制。如图5.9。

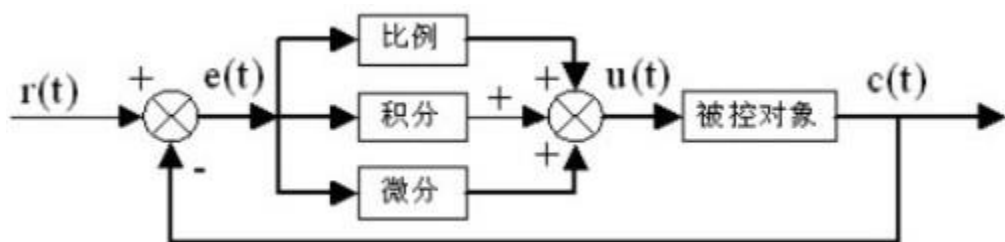


图 5.9 PID 控制器原理框架图

单位反馈 e 代表理想输入与实际输出的偏差，将偏差信号 e 送入控制器，控制器算出偏差信号 e 的微分值和积分值，将计算出来的微分值和积分值与原来的误差信号 e 进行线性组合，得到输出量 u 。计算公式如公式（5-1）所示。

$$u(t) = K_P \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_D \frac{de(t)}{dt} \right] \quad (5-1)$$

其中 K_P 、 K_i 、 K_d 分别成为比例系数、积分系数、微分系数； $r(t)$ 、 $e(t)$ 、 $u(t)$ 分别为控制器的输入信号、偏差信号、输出信号。

在PID控制系统中，比例调节是最简单的调节方式，它能够快速、无滞后的是被控参数稳定在给定值附近，并且能及时克服扰动干扰，一般增大比例系数 K_P 可以减小上升时间，但比例控制不能消除稳态误差，并且比例系数 K_P 过大时会导致系统不稳定。

二. 位置式 PID 算法

在计算机控制系统中，使用的是数字 PID 控制器。数字 PID 控制算法可以分为位置式 PID 和增量式 PID 控制算法。

由于计算机控制是一种采样控制，它不能像模拟控制那样连续输出控制量进行连续控制，而只能根据采样时刻的偏差计算控制量，数字 PID 控制需要将微分项和积分项进行离散化处理。离散化处理方法为：采样周期为 T ，采样序号为 k ，则离散时间 kT 对应着连续时间 t ，得到离散化的 PID 表达式由公式（5-2）给出：

$$u(k) = K_P \left\{ e(k) + \frac{T}{T_i} \sum_{j=0}^n e(j) + \frac{T_D}{T} [e(k) - e(k-1)] \right\} \quad (5-2)$$

其中， $u(k)$ — 第 K 次采样时微机输出；

$e(k)$ — 第 K 次采样时的偏差值；

$e(k-1)$ — 第 $K-1$ 次采样时的偏差值。

位置式 PID 控制是将输出量 $u(k)$ 直接作用于执行结构， $u(k)$ 的值与执行机构的位置时一一对应的。由于位置式 PID 每次输出都与过去的状态有关，计算时要对 $e(k)$ 进行累加，工作量大，并且当计算机出现故障时，输出量 $u(k)$ 变化幅度很大，从而引起执行机构位置的大幅度变化，很有可能造成严重的生产事故，不能用于实

际的生产。

三. 增量式 PID 算法

增量式 PID 控制是将控制量的增量 $\Delta u(k)$ 作为数字控制器的输出, 控制器的第 $k-1$ 个采样时刻的输出值由公式 (5-3) 给出:

$$u(k-1) = K_p\{e(k-1) + \frac{T}{T_i} \sum_{j=0}^{k-1} e(j) + \frac{T_d}{T} [e(k-1) - e(k-2)]\} \quad (5-3)$$

将式 (5-2) 与式 (5-3) 相减并整理得, 可以得到的增量式 PID 控制器的增量 $\Delta u(k)$, $\Delta u(k)$ 由公式 (5-4) 给出:

$$\Delta u(k) = K_p[e(k) - e(k-1)] + K_i e(k) + K_d[e(k) - 2e(k-1) + e(k-2)] \quad (5-4)$$

式中: $K_i = K_p \frac{T}{T_i}$ 积分系数;

$$K_d = K_p \frac{T_d}{T} \text{ 微分系数。}$$

此控制器输出的控制量的增量 $\Delta u(k)$ 与采样周期、比例系数、积分时间常数和微分时间常数有关。当计算机控制系统采用恒定的采样周期 T , 确定 K_p 、 K_i 、 K_d , 则只需要使用前后三次测量的偏差值, 就可以求出控制量。

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。

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。

5.3.2 分类进行转向舵机的方向控制

一. 整体思路

对与舵机的控制，我们主要采用赛道中心线偏差作为输入，对不同的赛道类型设定不同的比例系数。

二. 直道的方向控制算法

对于赛道中的直道，是方向控制中最好处理的一种情况，因为只要小车没有偏离赛道，就可以不进行偏转，而当智能车偏离赛道时也只需要一个较小的偏转，让智能车能缓慢回归赛道就可以了，具体的方法是计算扫描到黑线的偏离中心线的平均值，再将这个数分别乘上比例系数，加上舵机偏向中心位置时需要给出的高电平的值，作为 PWM 波的高电平送入给舵机，就可以实现直道上的智能车方向控制了。

三. 入弯道的方向控制算法

在道路中遇到弯道时需要转弯，转向角只受弯道的弯曲程度影响，但是距离大弯多远就开始转弯却需要受到当前速度的影响，也不能等到了弯道才开始转弯，而是需要提前转弯，即入弯赛道类型。在入弯中小车提前打角，而且是越早越好调整自己的姿态，为过弯道做准备。

四. 弯道的方向控制算法

小车在弯道中控制方法则需要打角较大，因为舵机在转向时有一定的延时，舵机转 60 度大约需要 110 毫秒，而智能车的最快速度可以达到 10 米/秒，如果智能车当时运行的速度就是 10 米/秒，那么当舵机转过 60 度时智能车已经走过 1 米多了，但由于在入弯赛道中小车姿态提前地调整，弯道中的角度变化比较小，而且在弯道中也需要一直保持打角。因此设定偏差的比例系数都要比其他赛道类型要大，在弯道里小车速度也要是整个赛道中最小。

五. “小 S”赛道的方向控制算法

“小 S”一般是指“S 型”道最靠左的黑线和最靠右的黑线之间相差的距离大于 60 厘米，这是因为全国大学生智能车竞赛中的赛道为 60 厘米，冲出赛道就算违规，因此“小 S”道是不能直接冲的，但是我们可以转尽量小的弯来使智能车通过的路线为最短，而且转更小的弯也可以使得速度不至于减少太多，实现转尽量小弯的方法是设定的偏差比例系数相对于弯道要小，但要大于直道，利用这种算法可以实现智能车走最少的路程，转最小的弯。

六. 起跑线的方向控制算法

起跑线的作用是让智能车可以在跑完两圈以后自动停车，也即是第三次检测到时，当检测到的次数不足三次时，应该直接忽略，当检测次数到达三次时就应该直接停车，也不应该影响到智能车的方向控制。

5.3.3 速度控制

一. 速度控制的意义

要想智能车以最短的时间跑完整个赛道，不但要求优越的方向控制算法，让智能车尽量走更近的路，转更少的弯。而且对于速度控制算法也有很高的要求，智能车在行驶过程中，不能够完全匀速，在直道上应该提速，而在弯道上应该减速。

在智能车沿着黑线行驶的过程中，速度不能够恒定不变，而应该根据道路的具体形状进行适当的调整。当然，最好的方法是让智能车始终以最快的速度通过整个赛道，但是如果速度始终调得太高，当小车遇到急弯或者是圆道时必然无法转过来，就会冲出赛道。因此不能够始终给最快的速度。然而，如果速度调得始终过低，就无法以最短的时间通过赛道。因此，速度应该要随着检测到的道路形状变化而变化。例如当前方道路为直道或类似直道时，应该将 PWM 波提得最高，让通过主电机的电流达到最大，从而让智能车以最快的速度行驶。当进入弯道时，应该将 PWM 波降到最低，让通过主电机的反向电流达到最大，从而让智能车以最短的时间将速度降到理想状态。

二. 速度控制的算法设计

智能车速度控制的总体思路与智能车方向控制的思路完全相同，也是将道路的形状进行分类，对于不同的类给以不同的速度。而在某一时刻给出智能车的理想运行速度以后，采用 PID 控制的算法实现对智能车的速度调节，以最短的时间使智能车达到这一理想速度。从而实现希望智能车的速度完全由道路情况所控制。

三. PID 参数调节

比例控制能迅速反应误差，从而减少误差，但是比例控制不能消除静态误差， K_p 的加大，会引起系统的不稳定；积分控制的作用是，只要系统存在误差，积分控制作用就不断地积累，输出控制量以消除误差，因而，只要有足够的时间，积分控制将能完全消除误差，积分作用太强会使系统超调加大，甚至使系

统出现振荡现象；微分控制可以减小超调量，克服振荡，使系统的稳定性提高，同时加快系统的动态响应速度，减少调整时间，从而改善系统的动态性能。而在该智能车系统调速的过程中，由于快速性是最重要的指标之一，因而应主要考虑其快速性，在调试的过程中，把 K_p 参数调得较大，这是为了增大速度调节的快速性，将 K_D 参数调得较为适中，因为除了快速性以外，最重要的就是要减少系统的超调量，而将 K_I 参数调得较小，因为在该速度调节中，并不需要达到静态，由于理想速度就在时时刻刻发生变化，所以也不可能达到静态。实验结果表明， K_p 参数取 0.8， K_D 参数取为 0.4，而 K_I 参数取得尽可能小，大约为 0.01 时，可以取得较好的效果。

5.4 小结

本章从整体的算法流程、赛道信息的获取及赛道类型的识别、智能车的速度及方向控制等着手，详细介绍了我们智能车软件部分的相关设计思路及实现方法。对于摄像头而言，首要的任务是赛道信息的获取，以及从赛道信息中获得的更为重要、有标识性的内容。其次便是对小车的控制，入弯的姿态、速度等等，都是十分重要的。

第六章 调试方案

前面几章一直是在为系统制定方案以及方案的细化。但整个系统的完善主要还是在系统的现场调试。在细分的每个模块中，大部分都涉及有众多参数，

对这些参数的确定就需要软硬件联合调试。而这过程就需要一整套开发调试环境和工具。包括程序源代码的编辑以编译环境，参数调节与设定工具。

6.1 IAR 集成开发环境

IAR Embedded Workbench 是一套完整的集成开发工具集合：包括从代码编辑器、工程建立到 C/C++编译器、连接器和调试器的各类开发工具。它和各种仿真器、调试器紧密结合，使用户在开发和调试过程中，仅仅使用一种开发环境界面，就可以完成多种微控制器的开发工作。

如同 Windows 操作系统其它一般的软件安装一样，单击 setup.exe 进行安装，你将会看到如图 6.1 的界面。

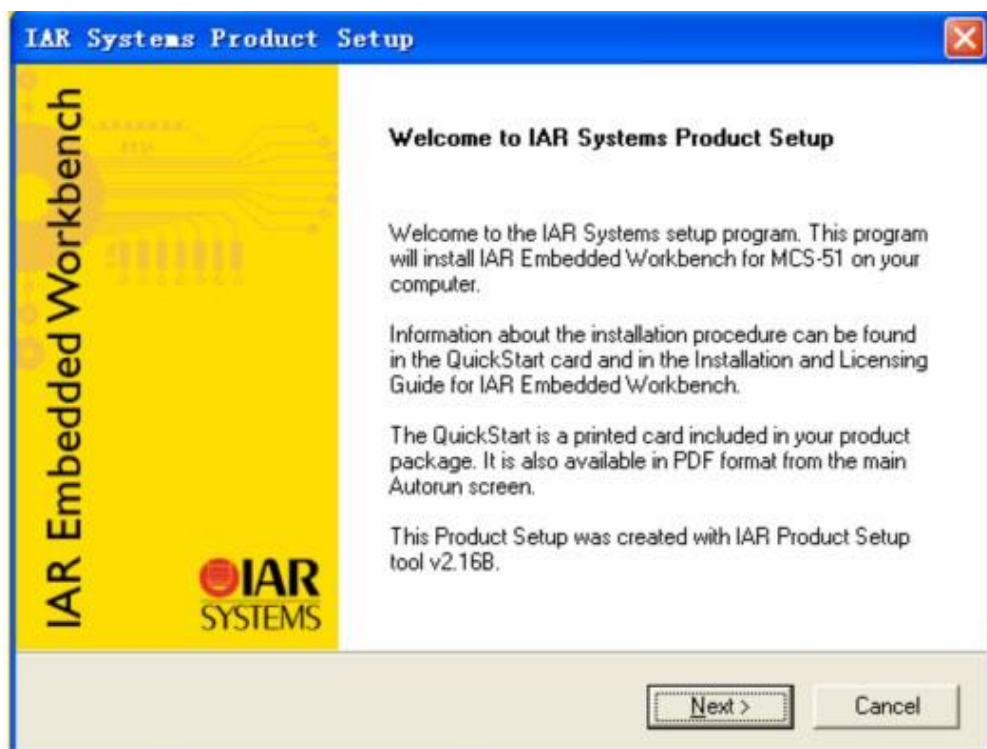


图 6.1 安装开始



图 6.2 认证序列

输入的认证序列以及序列钥匙正确后，单击“NEXT”到下一步，如图 4 所示，在你将选择完全安装或是典型安装，在这里我们选择完全安装。

当进度完成后，它将跳到下一个界面，到此我们便完成了安装的步骤。



图 6.3 安装完成

仿真调试编，译好后接下来就是调试程序了。首先你需要连接你的硬件平台才能进行调试。在计算机与 ZigBee 硬件系统连接前，你需要确保你已在你的计算机上安装了必要的仿真器驱动。

安装完成仿真器驱动后，通过 USB 接口把 ZigBee 开发系统与计算机连接。选择菜单 Project\Debug 或按快捷键 CTRL+D 进入调试状态，也可按工具栏上按钮进入调试。进入调试后，整体窗口如图 6.4：

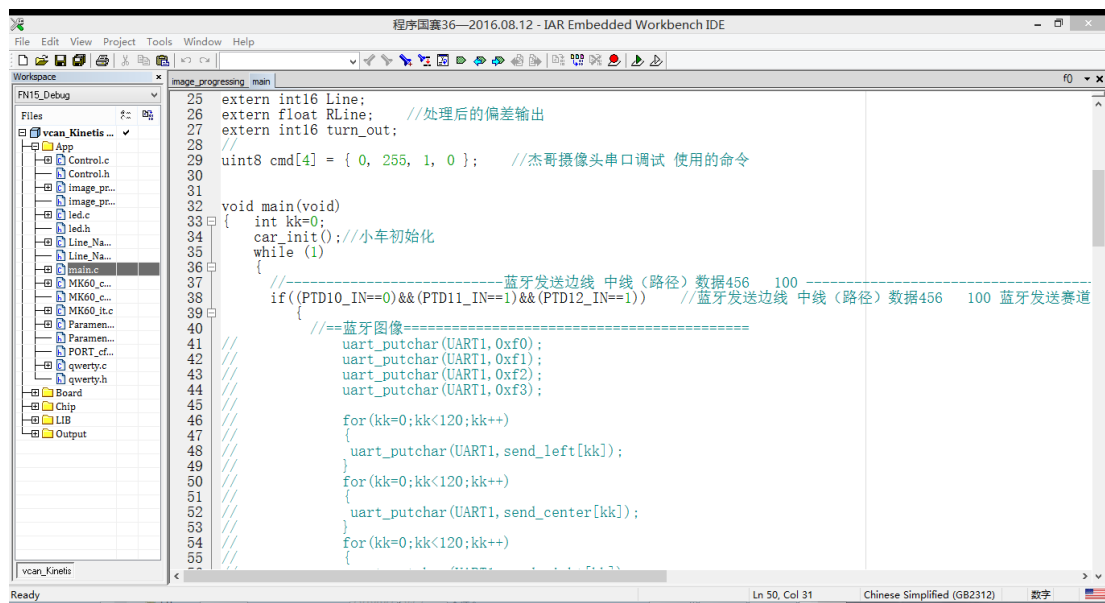


图 6.4 调试界面

6.2 上位机

我们用到的上位机有三个：摄像头上位机两个，PID调试助手，参数修改神器。所有上位机波特率都为 115200。

6.2.1 摄像头上位机

由于我们使用的是摄像头，需要实时的查看小车实际看到的赛道信息，方便于我们对图片的分析。我们使用 Visual Studio 开发的摄像头上位机，可以接收 NRF24L01发来的图片信息，同步执行算法分析图片。在调试的后期，发现仅仅使用这个上位机并不能完全分析到小车在实际运行的过程中看到的图像，我们小组又将上位机进行改进，开发了通过蓝牙接收数据只看图像边界线的上位机，进而方便我们的调试。

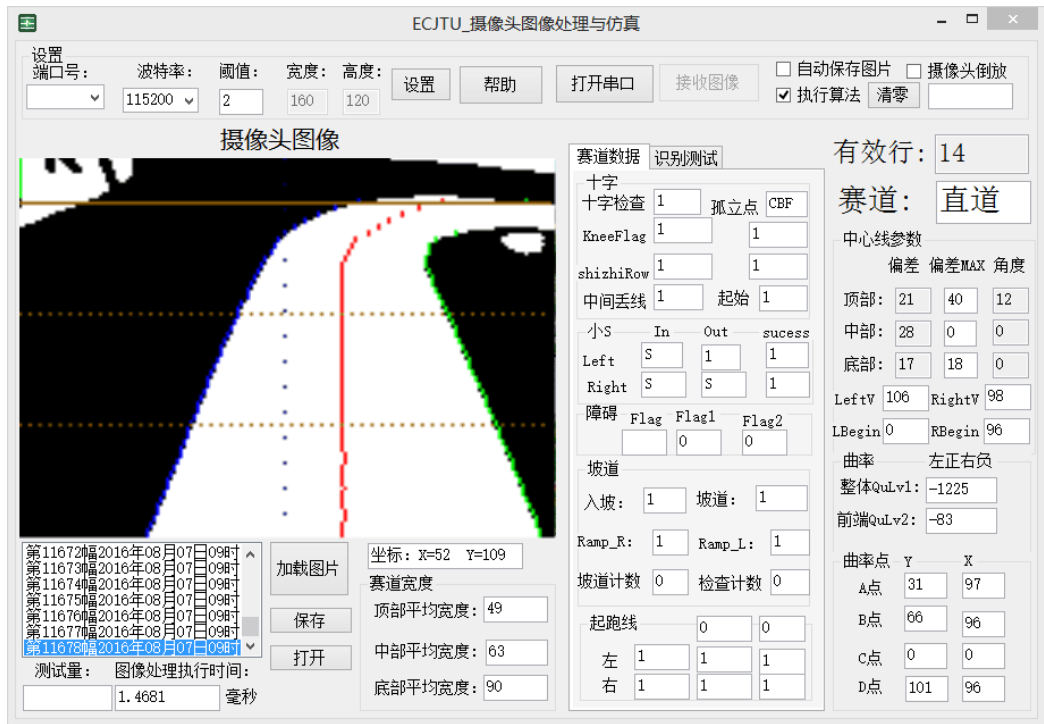


图 6.5 用 NRF24L01 接收整张图像的上位机



图 6.6 用蓝牙接收下位机处理的边线的上位机

6.2.2 PID 调试助手

由于会对模型车结构进行修改，导致车子在调试过程中参数经常要修改，有时候调车调得速度上不去的时候，会考虑到是不是车子的PID 没调好而去调PID，为了使调试过程中各参数动态变化能直观显现出来，我们特意开发了一个能通过蓝牙接收PID 调节过程中各参数值的上位机，就是这个PID调试助手。

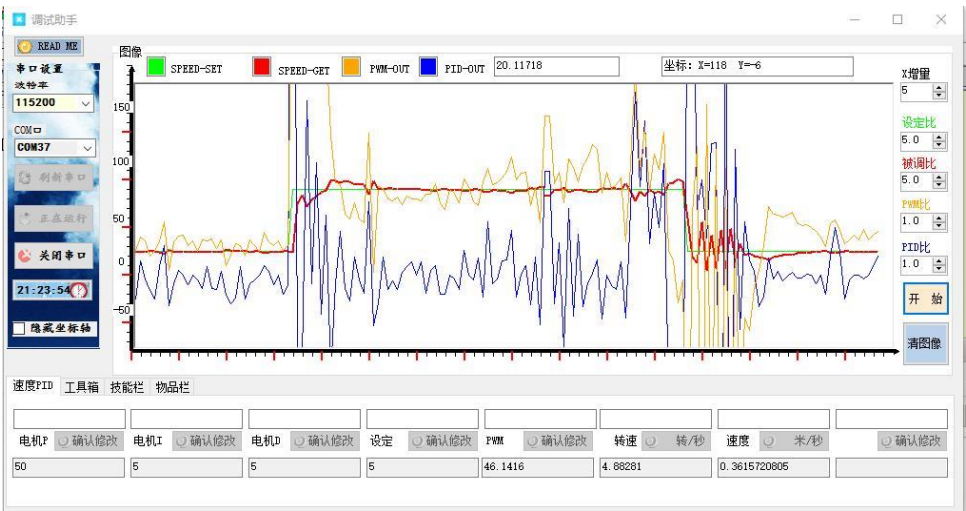


图 6.7 PID 调试助手

6.2.3 参数修改上位机

由于小车的各项参数较多，每次为了改参数而去下载程序，会使得在调车的过程十分不便。因此，我们利用 NRF 将数据发送给小车，可以使得小车在行走的过程中动态地修改参数，而不影响小车的正常运行。



图 6.8 参数修改神器

6.2.4 小结

通过组委会提供的 IAR 编译软件的在线调试功能，可以得到大量的信息，为智能汽车的调试提供了很大的帮助。

在智能汽车的调试过程中，有针对性的开发了一系列便于人机交互的上位机系统，通过简单明了的可视化界面直观的显示智能汽车的各种状态对调试有很大帮助。监视小车路径的图像处理上位机，同时也具有算法仿真的功能，为小车图像处理程序的开发提供了方便，使程序编写更加高效。显示小车 PID 调节时的各项参数的上位机，调节 PID 时也更有方向性；我们也开发了具有参数修改功能的上位机，时参数修改变得更加分方便等等。

第七章 模型车主要技术参数

智能汽车相关技术参数如表7.1 所示：

表 7.1 智能汽车技术参数

项目	参数
车模质量	约1000 克
车模尺寸(毫米)	长290/宽180/高375
车模平均电流（匀速行驶）(毫安)	约3100
电路电容总量（微法）	1600
传感器种类及个数	摄像头/1个 编码器/1个
赛道信息检测空间精度（毫米）	2
赛道信息检测频率（次/秒）	50
主要集成电路数量	5

第八章 总结

小车，对很多男生来说这可是一个童年中不可或缺的“圣物”，小时候动画片中播放的“四驱兄弟”，对很多人来说是接触小车的经典中的经典代表作。很多人心中有一个“小车梦”，现在我们终于有了圆这个梦的机会了。我们现在做的是比动画片中更厉害的智能车。智能车，“能自己跑的车”，很多人一听就感觉逼格高了不少。带着对智能车无限的向往和喜爱，我们三个对小车充满无限憧憬的男生开始了对智能小车的“探索之旅”。为了配合我们的设计理念，我们给我们的队伍取名为“Jarvis”。看过《钢铁侠》这部电影的人都不会陌生，Jarvis是钢铁侠开发出的一套人工智能系统，而智能车的“智能”就体现在“人工智能系统”方面。尽管我们做不到像“钢铁侠”里面的Jarvis系统那么厉害，但是我们也希望我们开发的小车智能系统能做到最好，能做出属于我们的“Jarvis”。

做智能小车是一件很辛苦的事儿，但是出于对小车的喜爱，我们还是坚持了下来，并取得了不错的成绩。在智能车这场征程中，我们有过第一次车跑起来的开心，有过车被撞坏的心疼，也有过一直苦恼的问题得以解决的喜悦，甚至有过一个问题得不到解决苦闷一个星期的烦恼。仿佛人生的大起大落就是这么来的。智能车，成为我们人生中难忘的一笔。

在本届智能车制作和研究过程中，从电路板的制作到赛道的搭建软件的编写，从中我们学习了硬件的制作，软件程序的编写学习，加上自身在学校课堂学习到的电子电路知识，以及汽车机械结构原理知识，对车模的安装有自己的独特的见解，对车子的机械，硬件，软件设计结合有了一整套的方案。在制作的过程中我们感受到了模型车第一次跑完赛道的喜悦；感受到了当图像处理遇到问题时的困惑；感受到模型车撞坏的难过。也许也正是因为这些欢乐、伤心，才是这次比赛的真正意义。

首先，硬件是基础，硬件的搭建就像大楼的根基，没有稳定的基础，谈做车就是无本之木，所以第一步就是把相关的硬件做稳定。在硬件电路及机械方面我们在前期花费了大量时间来分析、测试。然后确定我们的硬件电路。不仅如此，我们在中期的调试中还通过调试发现问题，进而改进硬件电路及机械结构。确保硬件、机械方面为模型车的提速打好最好的基础。在于此同时，我们还在开发用于调试的上位机，以及一些调试手段为下一步调试做好准备。

其次，软件是灵魂，有了强壮的体魄，更加需要智慧的灵魂，在硬件支托下，我们通过仔细的分析卓晴老师在第七届比赛时公布的一个培训视频，以及其他人的技术报中软件的设计思路，然后初步搭建我们的程序框架。最后在不断地调试中不断地改

进、升级我们的程序。在调试过程中，我们经常在智能车论坛里，找一些车友交流技术，共同提升。从网上找到许多的控制算法的文章，不断研究最后开发属于适合我们小车的控制算法。

第三，机械上的调整优化。在速度达到一定数值之后，模型车的速度就会达到机械的极限，无论再怎样调节参数或者调整算法都很难让速度提高一个层次，但同样的算法在另一部机械结构有些不同的车上使用却能够跑出更好的效果，这时我们就要通过调整机械了，例如加润滑剂，车身重心的调整，轮胎调整，整车质量的调整，镜头角度的更换等等，这些微小的调整可能会使得车子的性能带来更大的进步。

总的来说，我们在实验室做智能车的半年，有很多地方做的还不够好，一是车模的重心较高，二是车模在出弯时，前轮会抖一下，这是因为图像处理的时候没有处理好，三是前轮机械结构不是很对称。这些问题我们会努力改进，让车子跑得更加顺畅。我们在做智能车的这半年，学到了很多终生难忘的专业基础知识，增强了实践的能力，提高发现问题、解决问题的能力，加强了团队合作的意识。

作为刚上大二的大学生，在杨云老师的辛勤指导下，我们从什么都不懂的孩子走向了能独立分析问题，能代表学校进入全国总决赛。激动的心情中怀着无限感激。感谢华东交通大学电气学院和创新基地在调试场地上的支持和帮助；感谢我们的指导老师的辛勤付出；感谢所有帮助过我们的学长学姐，没有你们的帮助，我们还会绕更多的弯路；同时也感谢那些在我们比赛的时候仍时刻关注着我们的学长们，即使你们已经毕业，远离了智能车的战场，但你们对智能车的热爱仍是我们无法追赶的。我不禁想起了其中一位学长的QQ签名：Keep Moving!!!

参考文献

- 【1】. 卓晴, 黄开胜, 邵贝贝. 学做智能车. 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. 北京: 清华大学出版社. 2004
- 【3】. 童诗白, 华成英. 模拟电子技术基础[M]. 北京. 高等教育出版社. 2000
- 【4】. 余志生. 汽车理论. 北京: 机械工业出版社. 2006 年 5 月.
- 【5】. 胡寿松. 自动控制原理. 北京: 科学出版社. 2001年 2 月第 4 版.
- 【6】. 蔡庆楠, 蔡兴旺, 潘锦洲. 上海大学S.U.L. 挑战者队技术报告.
- 【7】. 李杰, 郭和山, 邹璇. 华东交通大学和一队技术报告.
- 【8】. 王日明, 廖锦松, 申柏华. 《轻松玩转ARM Cortex-M4微控制器-基于Kinetis K60》. 北京: 北京航空航天大学出版社. 2014. 9
- 【9】. Bradley Jones , Peter Aitken ,Dean Miller. 21天学通C语言/第七版. 北京: 北京航空航天大学出版社. 2014. 11

附录 A 主控板原理图及 PCB 图

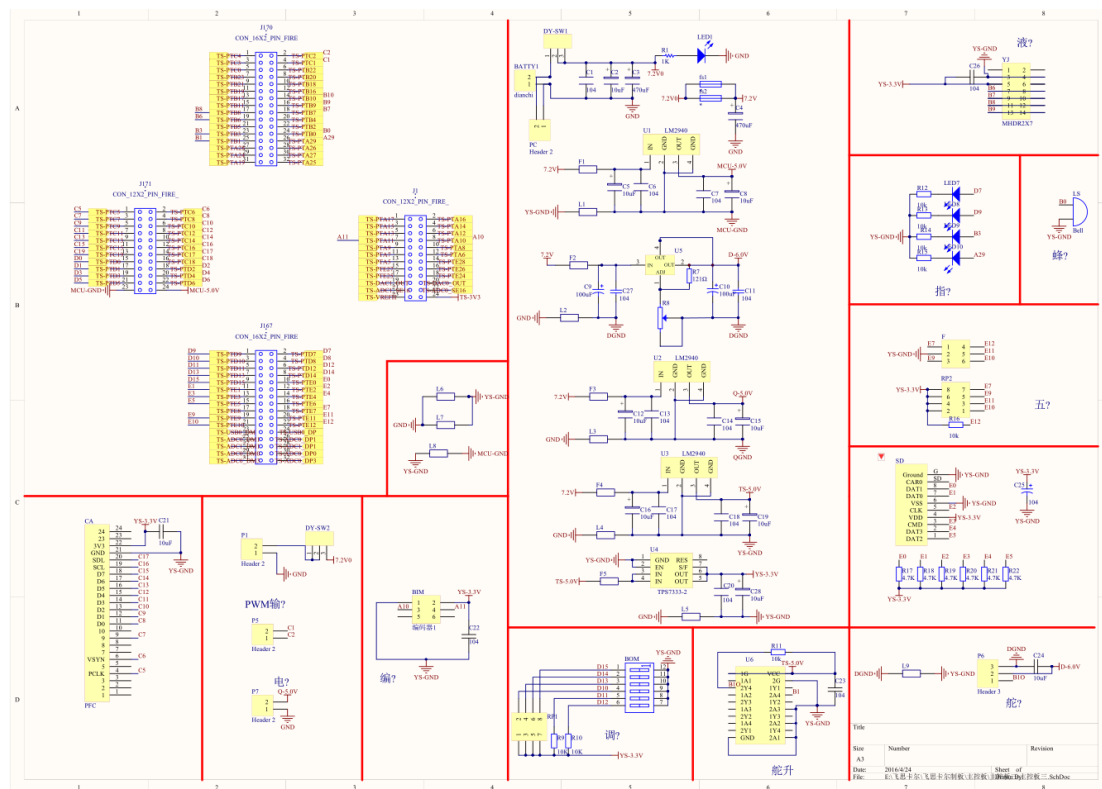


图 A-1 主控板原理图

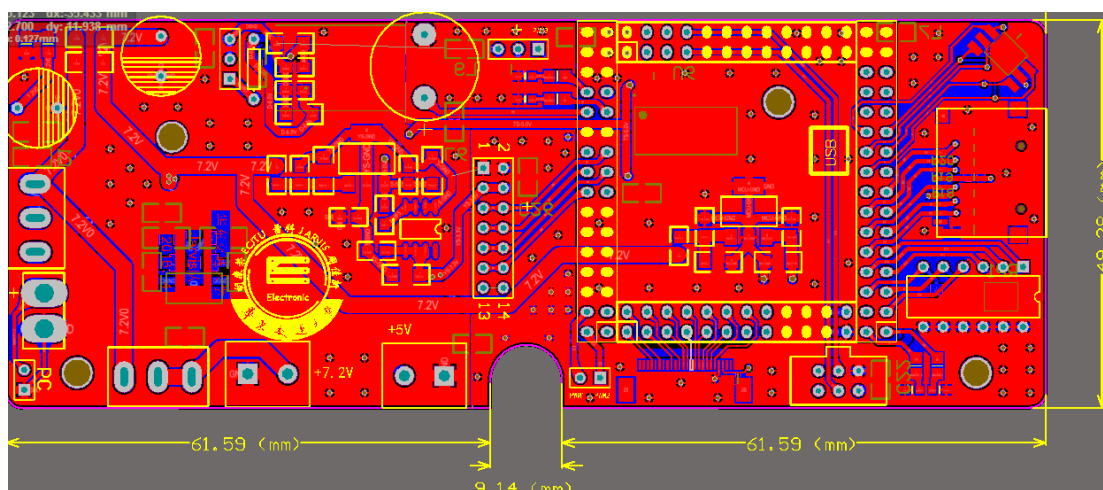


图 A-2 PCB 图

附录 B 部分程序源代码

```
void car_init()//小车初始化
{
    Ring_init();//蜂鸣器初始化管脚

    led_init(LED_MAX);//LED 全部初始化

    uart_init(UART1,115200);//调试信息打印

    SW_init();//拨码开关初始化

    qwerty_init();//五维按键初始化

    flash_init();//初始化 FLASH
    DELAY_MS(10);

    Init_AnalysisImg_Var();//图像数据初始化

    DELAY_MS(10);

    LCD_Init() ;//LCD 初始化
    DELAY_MS(10);

    Parament();//参数初始化
    // ParameterToFlash(0);
    DELAY_MS(10);

    if(PTD12_IN ==0)//拨码开关 654    000
    {
        //FlashToParameter(0);//从 flash 中读取参数
        Parameter_set(0);
        LCD_CLS();    //清屏
    }

    ftm_quad_init(FTM2); //编码器 FTM2 正交解码初始化
    DELAY_MS(10);

    ftm_pwm_init(FTM1, FTM_CH1,300,Paramenter.Turn_Center); //初始化 舵机
    PWM//400//392
    DELAY_MS(10);

    nrf_init();//初始化 NRF24L01+ ,等待初始化成功为
    DELAY_MS(10);

    ftm_pwm_init(RS_FTM, RS1_CH,RS_HZ,0); //电机
    ftm_pwm_init(RS_FTM, RS2_CH,RS_HZ,0); //电机初始化
    DELAY_MS(10);
```

```

    pit_init_ms( PIT1 , 20 );//定时器 PIT1 中断图像计算//舵机转向//加上上位机发送
    pit_init_ms( PIT0 , 4 );//定时器 PIT0 中断电机控制
    DELAY_MS(10);

    camera_init(imgbuff1,imgbuff2);//摄像头初始化
    DELAY_MS(10);

    set_vector_handler(PORTE_VECTORn, PORTE_IRQHandler); //设置 PORTE 的中断服务函数
    数为 PORTE_VECTORn
    DisableInterrupts; //禁用总中断
    NVIC_SetPriorityGrouping( 4 ); //设置优先级分组,4bit 抢占优先级,没有亚优先级
    NVIC_SetPriority(PORTC_IRQn, 0 ); //配置优先级(摄像头场中断)
    NVIC_SetPriority(DMA0_IRQn , 1 ); //配置优先级(摄像头 DMA)
    NVIC_SetPriority(PIT1_IRQn , 2 ); //配置优先级(图像计算)
    NVIC_SetPriority(PIT0_IRQn , 3 ); //配置优先级(舵机控制)
    NVIC_SetPriority(PORTE_IRQn, 4 ); //配置优先级(NRF2401)

    DELAY_MS(10);
    EnableInterrupts;
    enable_irq(PORTC_IRQn);
    enable_irq( DMA0_IRQn);

    DELAY_MS(2500);//发车延时
    Paramenter.qipaoxian_check=0;//初始化起跑线检测
    Paramenter.qipaoxian_stop_flag=0; //初始停车标志

    ftm_pwm_duty(RS_FTM, RS1_CH,0);
    ftm_pwm_duty(RS_FTM, RS2_CH,(uint32)Controlvar.Pwm_Out);//设置电机初始占空比
    DELAY_MS(10);

    enable_irq(PIT1_IRQn);
    enable_irq(PORTE_IRQn);
    enable_irq(PIT0_IRQn);//电机控制 使能

}
/*-----
*功能: 计算舵机 pid
*输入:
*输出:
*-----*/
int duoji_PD(float offse)
{
    float duoji_p=0;
    float duoji_d=0;
    float erro=0;//误差
    float erro_last=0;

```

```

float fP=0;
float fD=0;//
float PD_out=0;
//-----//
duoji_p=Controlvar.Duoji_P;
duoji_d=Controlvar.Duoji_RD;
erro_last=Controlvar.Erro;
//-----//
erro=offse;

Controlvar.Erro=erro;//储存误差

fP=duoji_p*erro;//P

fD=duoji_d*(erro-erro_last);//D

PD_out=fP+fD;//输出

if(PD_out>90){PD_out=90;}
if(PD_out<(-90)){PD_out=(-90);}

return (int)PD_out;
}
/*-----
*功能： 计算速度增量式 PID
*输入： 设定速度 speed_set
*输出： PWM 波占空比
*-----*/
void PID_Calculata(float speed)
{
    float erroo[3] = {0,0,0};//误差数组
    float derro = 0.0;

    float speed_p = 0.0;//比例 P
    float speed_i = 0.0;//积分 I
    float pwm_out_old = 0.0;

    float fp = 0.0;

    //-----变量保护-果然还是要用-----//

    erroo[1] = Controlvar.Error[1];//前次
    pwm_out_old=Controlvar.Pwm_Out_Old;
    //-----变量保护-结束-----//
    erroo[2]=speed-Controlvar.Speed_Get;//偏差计算
    if(start_pid_flag<=200||erroo[2]>=20) //大范围加速用比较慢的 pid 避免
超调太多 start_pid_flag<=200||
    {

```



```

speed_p = Controlvar.Speed_PS;
speed_i = Controlvar.Speed_IS;
}
if(erroo[2]<20&&erroo[2]>=0) //小范围加速
{
speed_p = Controlvar.Speed_PU;
speed_i = Controlvar.Speed_IU;
}
if(erroo[2]<0) //减速 用这个 {
speed_p = Controlvar.Speed_P;
speed_i = Controlvar.Speed_I;
}
// if(erroo[2]<-8) //减速 用这个
// {
// speed_p = Controlvar.Speed_P;
// speed_i = Controlvar.Speed_I;
// }

Controlvar.Error[0]=erroo[1];
Controlvar.Error[1]=erroo[2];
//-----积分计算-----//

if((erroo[2]<OFFSE_I_MAX)&&(erroo[2]>OFFSE_I_MIN)) //积分范围
{
Controlvar.Speed_cal_I=speed_i*erroo[2];
//-----你觉得要写限幅你就写-----//

if(Controlvar.Speed_cal_I<SPEED_I_MIN) Controlvar.Speed_cal_I=SPEED_I_MIN; //积分
结果限幅

if(Controlvar.Speed_cal_I>SPEED_I_MAX) Controlvar.Speed_cal_I=SPEED_I_MAX;
if(Controlvar.Speed_Get==0&&Controlvar.Speed_Set<11) Controlvar.Speed_cal_I=0;
//积分清零

}
else Controlvar.Speed_cal_I=0; //积分范围之外 不积分

//-----比例计算-----//
derro=erroo[2]-erroo[1]; //两次偏差的差值
fp=speed_p*derro; //比例结果

if(derro<OFFSE_P_MIN) {fp=speed_p*OFFSE_P_MIN;} //两次偏差差值限幅

if(derro>OFFSE_P_MAX) {fp=speed_p*OFFSE_P_MAX;}

//-----终于输出数值了-----//

```

```

Controlvar.Pid_Out=(fp+Controlvar.Speed_cal_I);//pid 计算输出    比例+积分
//
Controlvar.Pwm_Out_New=pwm_out_old+Controlvar.Pid_Out;//PWM 输出    =上一次 pwm
输出+pid 输出    平滑输出

Controlvar.Pwm_Out_Old=Controlvar.Pwm_Out_New;

//-----输出限幅-----//
Controlvar.Pwm_Out = Controlvar.Pwm_Out_New;
if( Controlvar.Pwm_Out<PWM_OUT_MIN) {Controlvar.Pwm_Out = PWM_OUT_MIN;
Controlvar.Pwm_Out_New= PWM_OUT_MIN;}

if( Controlvar.Pwm_Out>PWM_OUT_MAX) {Controlvar.Pwm_Out = PWM_OUT_MAX;
Controlvar.Pwm_Out_New= PWM_OUT_MAX;}

}

```