

第十六届全国大学生 智能汽车竞赛

技 术 报 告

学 校：武汉大学

队伍名称：珞珈-蜗牛

参赛队员：陈宇航、秦浩滨、周梦阳、

张涛、王佳政

带队教师：专祥涛、路晓庆

关于技术报告和研究论文使用授权的说明

本人完全了解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：陈宇航、秦浩滨、周梦阳、张涛、王佳政

带队教师签名：杨飞、张荣

日 期：2022-8-18

引言

从 2006 年开始，全国大学生智能车竞赛已经成功举办了十六届了，为了实现竞赛的"立足培养、重在参与、鼓励探索、追求卓越"的指导思想，竞赛内容设置需要能够面向大学本科阶段的学生和教学内容，同时又能够兼顾当今时代科技发展的新趋势。

自动驾驶作为一个热门领域，智能车比赛紧跟时代，推出完全模型组这一组别，相较于传统竞赛组多了加油站，施工区，泛行区等实际元素，尽可能的模拟真实自动驾驶环境。

本文以 Edgeboard (FZ3B) 为上位机芯片，完成模型运行、元素识别传统巡线，TC264 作为下位机直接控制舵机、电机等外设，二者通过 USB 通信协作进行图像采集、图像预处理、模型预测、基本巡线、元素识别和外设控制，从而实现自动寻迹。本技术报告将简述小车设计过程，技术报告主要篇幅将集中于软件设计、控制算法设计与神经网络搭建部分。

关键词：模型预测，自动巡线，PID 控制，神经网络

目录

- 引言.....3
- 第 1 章 任务要求.....1
 - 1.1 设计要求.....1
 - 1.2 系统概述.....2
- 第 2 章 方案实现.....4
 - 2.1 编码器模块功能验证.....4
 - 2.2 陀螺仪模块功能验证.....4
 - 2.3 电机驱动模块验证.....5
- 第 3 章 硬件电路模块设计.....6
 - 3.1 电源管理部分系统.....6
 - 3.2 USB 转串口模块.....9
 - 3.3 传感器与人车交互模块.....9
 - 3.4 电机控制模块.....10
- 第 4 章 软件设计.....12
 - 4.1 程序主体流程.....12
 - 4.2 预处理.....13
 - 4.2.1 赛道边沿及中线提取.....13
 - 4.2.2 图像矫正.....14
 - 4.3 元素识别.....15
 - 4.3.1 识别可用几何特征.....15
 - 4.3.2 坡道识别.....15
 - 4.3.3 十字识别.....16
 - 4.3.4 环岛识别.....17
 - 4.3.5 车库识别.....18
 - 4.3.6 三岔路识别.....19
 - 4.4 控制算法.....20
 - 4.4.1 舵机控制.....20

4.4.2 电机控制	20
4.4.3 参数作用与整定	20
4.5 滤波算法	21
4.5.1 卡尔曼滤波	21
第五章 神经网络模型	21
5.1 任务概述与操作	21
5.1.1 采集数据	22
5.1.2 数据标注	23
5.1.3 训练模型	23
5.1.4 训练模型	25
5.2 小车自动行驶及标志物识别处理	26
5.2.1 标志物采集实现以及优化方案	26
5.2.2 标志物数据处理实现以及优化方案	26
5.2.3 标志物模型	28
第 6 章 遇到的问题及解决方案	30
6.1 硬件问题	30
6.2 软件问题	30
6.3 其他问题	31
第七章 总结与心得	32
致谢	33

第 1 章 任务要求

1.1 设计要求

车模使用 I 型车模。禁止不同型号车模之间互换电机、舵机和轮胎；禁止改动车底盘结构、轮距、轮径及轮胎；如有必要可以对于车模中的零部件进行适当删减；禁止采用其它型号的驱动电机，禁止改动驱动电机的传动比；禁止改造车模运动传动结构；禁止动舵机模块本身，但对于舵机的安装方式，输出轴的连接件没有任何限制；禁止改动驱动电机以及电池，车模前进动力必须来源于车模本身直流电机及电池；禁止增加车模地面支撑装置。在车模静止、动态运行过程中，只允许车模原有四个车轮对车模起到支撑作用。为了安装电路、传感器等，允许在底盘上打孔或安装辅助支架等。对车轮：参赛车模的车轮需要是原车模配置的车轮和轮胎，不允许更改使用其它种类的车轮和轮胎，不允许增加车轮防滑胶套。

比赛分为两个阶段：

预赛阶段：小车沿着赛道运行两周，按照比赛时间进行排名。

决赛阶段：小车沿着赛道运行两周，按照比赛时间进行排名，但加入更多识别预测元素。

按照不同赛题组别，竞赛车模主控电路采用百度的 EdgeboardFZ3B 板卡。如果增加额外的单片机，则需要使用 Infineon 公司单片机。如果所选用的传感器或者其它电子部件中也包含有微处理器，对此微处理器的种类和数量不做限制，但其不得参与对于赛道信息识别和处理、不参与车模运动决策与控制；针对某些特殊电子模块的允许使用或者禁止使用组委会将会另行公布相应器材清单进行说明。

1.2 系统概述

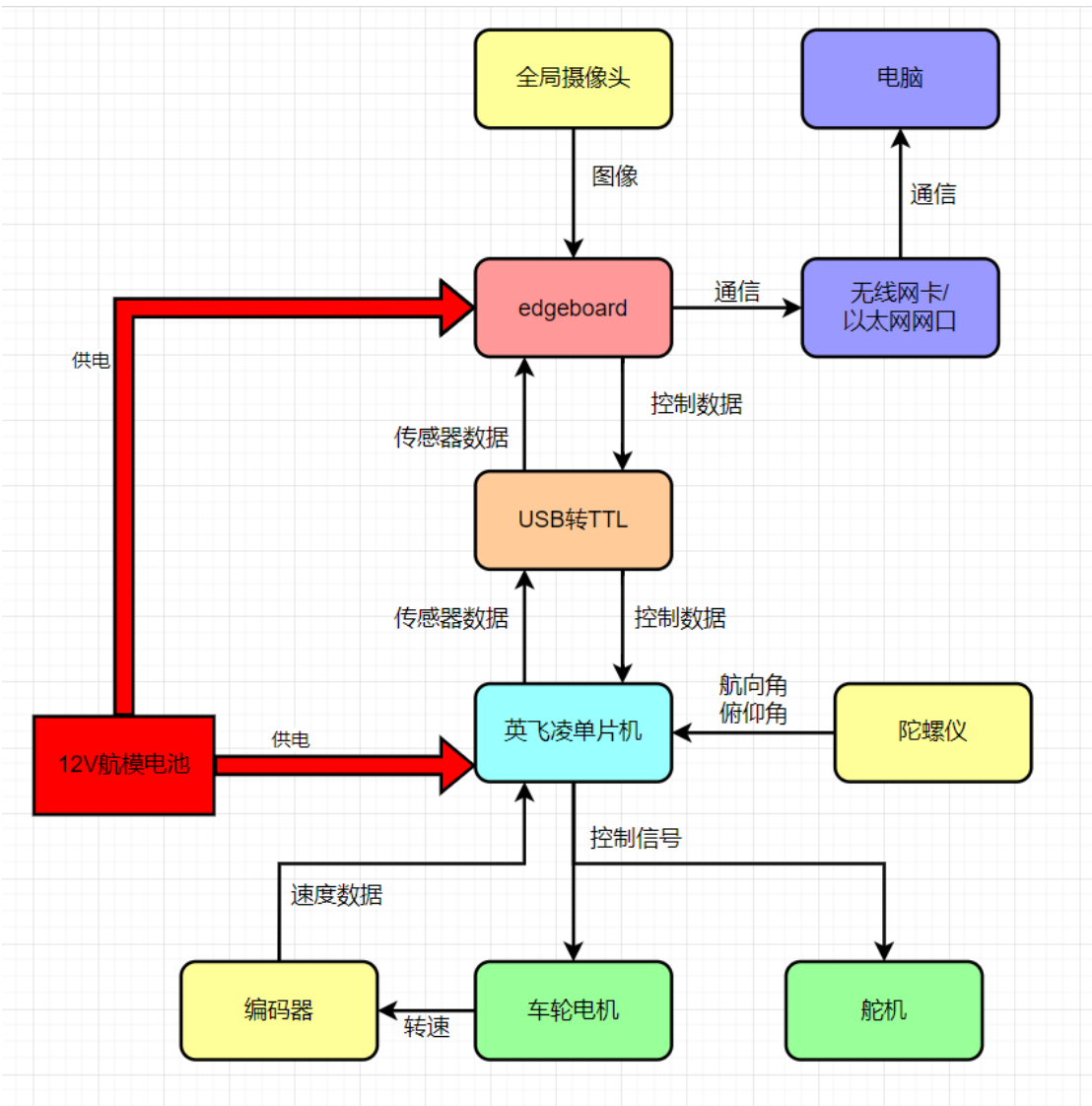


图 1-1 智能车系统总体结构

完全模型组赛事方要求赛道识别必须使用百度 EdgeBoard 计算卡（FZ3B）和百度深度学习框架的人工智能算法实现。因此摄像头并非连接在往届使用的英飞凌 MCU 上，而是连接在基于 Linux 系统的 EdgeBoard 上，循迹算法与元素识别都在 EdgeBoard 上完成，英飞凌的 MCU 仅用于车模的运动控制。

这就要求我们要能够熟练地，在 Linux 平台上构建一个 C/C++ 工程，在工程中完成循迹和识别任务，训练并部署神经网络模型，构建上下位机（EdgeBoard 和英飞凌 MCU）的通信方式，自主设计下位机 pcb 电路，焊接下位机电路板，搭建车模且安装和校正传感器，在下位机设计一套运动控制代码。这是一次综合性

强，创新性高、任务量大的实习任务。

智能车开始运行时，EdgeBoard 从摄像头采集图像，对图像进行推理，确定是否存在特殊标志，若存在特殊标志，转入对应任务的处理程序。同时 EdgeBoard 对图像进行二值化处理，寻找赛道边界，完成循迹任务。图像处理完毕后，根据任务需要会运算得到车辆转向的偏差和期望的行驶速度，EdgeBoard 将偏差和期望的行驶速度通过串口发送到下位机，下位机接受到数据后，通过控制算法控制电机和舵机，执行对应动作。同时下位机采集陀螺仪和编码器等传感器数据，处理后发送回 EdgeBoard，在接收到这些数据后进行下一步处理和反馈处理。

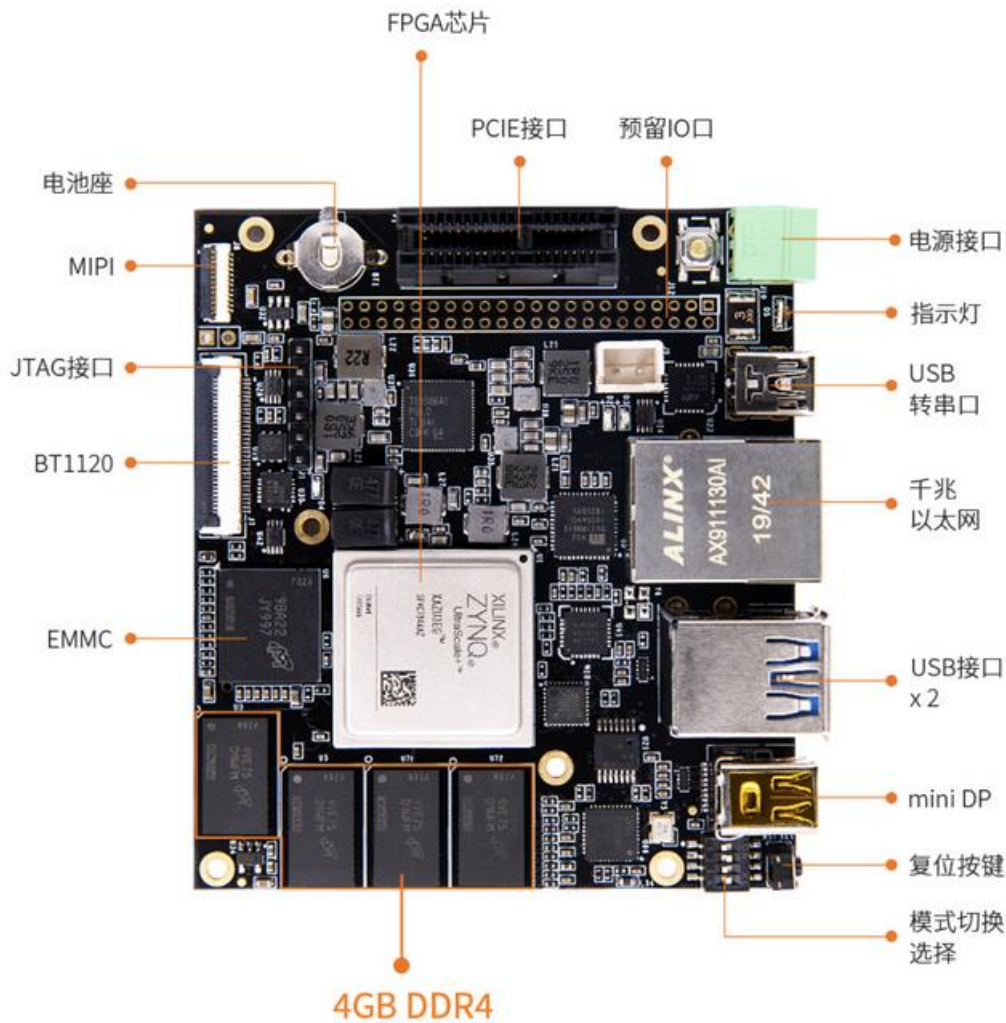


图 1-2 EdgeBoard 总体结构

EdgeBoard 是百度面向嵌入式与边缘部署场景打造的 AI 视觉硬件方案，可无缝兼容百度大脑丰富的工具平台与开放能力，具有高性能、低成本、使用简单等三大优点，可广泛适用于智能设备打造、科研教学、安防监控、餐饮卫生监控、工业质检、交通巡检、医疗诊断、智能零售等领域。

第 2 章 方案实现

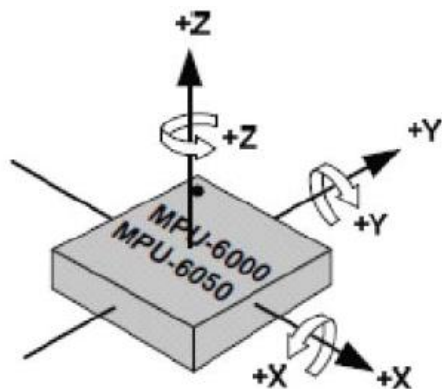
车模传感器包括有：速度传感器（编码器），车模姿态传感器（陀螺仪），电机驱动模块以及供电模块。小车的传感器相当于人的五官，它利用速度传感器来感知行进的速度，通过陀螺仪和加速度计获得车模倾角，检测坡道以及颠簸路面，而小车前方的摄像头就像人的两只眼睛，用来探知前方的路况，然后将信号传给单片机进行分析，判断下一步的行走方向。传感器与小车的其他部分，配合单片机共同完成对小车智能控制，实现小车的行走，自动循迹。

2.1 编码器模块功能验证

速度传感主要由编码器构成，本文车模用的是 1024 线编码器。通过编码器将旋转位移转换成一串数字脉冲信号，利用控制单片机的计数器测量在固定时间间隔内速度脉冲信号的个数可以反映电机的转速。将编码器架在后轮同步转动的齿轮上，测得编码器脉冲输出。

2.2 陀螺仪模块功能验证

车模姿态传感器使用了陀螺仪。为了最大程度上减少车模运行前后振动对于测量的干扰，通常尽可能的将陀螺仪固定在整个车模最底部比较平稳的位置。安装角度传感器电路板时应该尽量保证陀螺仪传感器水平安装。若是陀螺仪安装不能够保证水平，控制芯片无法得到正确的横滚角，从而无法输出正确的前叉打角。其中转动会带动陀螺仪转动。如果陀螺仪安装不是绝对的水平，那么这个转动就会在陀螺仪的 Z 轴方向存在一个分量。根据陀螺仪倾斜的方向不同，这个分量有可能是正，有可能是负。本文中的小车陀螺仪安装在车模的中轴线上。



2.3 电机驱动模块验证

当今最常用的电机调速方式为：采用 H 桥驱动方式，通过脉宽调制 PWM

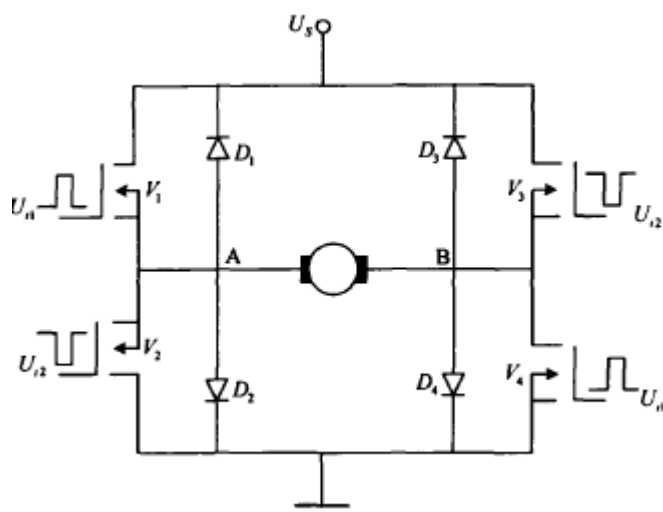


图 2-1 H 型双极可逆驱动系

来控制电动机电枢电压，实现调速。脉冲宽度调制（ Pulse-Width Modulation ，简称 PWM ），其实质是一种将数字量转换为模拟量的方法，此时模拟量的表现形式为周期固定、宽度可变的脉冲波。脉冲波所携带的信息用波形的占空比来编码，占空比是波形为高电平的时间与 PWM 信号的周期之比。H 型双极可逆驱动系统由四个开关管和四个续流二极管组成，单电源供电。在每个 PWM 周期内，当控制信号 U_{i1} 为高电平时，开关管 V_1 、 V_4 导通，此时 U_{i2} 为低电平，因此 V_2 、 V_3 截止，电枢绕组承受从 A 到 B 的正向电压；当控制信号 U_{i1} 为低电平时，开关管 V_1 、 V_4 截止，此时 U_{i2} 为高电平，因此 V_2 、 V_3 导通，电枢绕组承受从 B 到 A 的反向电压。

第3章 硬件电路模块设计

电路部分是硬件平台的重要组成部分，稳定的电路是后续软件顺利运行的基础。我们在设计电路之时对电路设计的所有环节都进行了兼容性设计，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。我们的电路的设计思想是在保证正确检测信号的前提下，尽可能使用精简、高效的电路。

这次电路采用信号处理、主控、驱动分体的电路板设计，避免了单块电路板的占地面积过大以及便于模块化检修替换。

3.1 电源管理部分系统

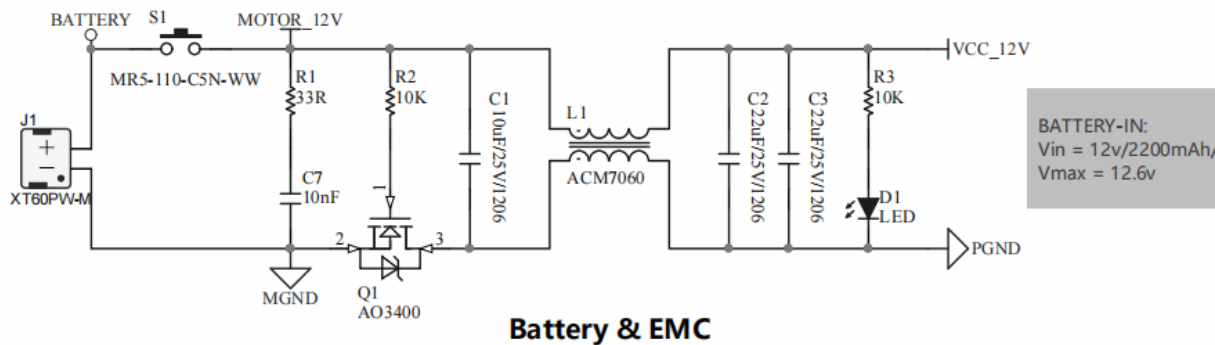


图 3-1 电源管理

本次竞赛我组采用格氏 12.6V 电池，此模块正是为了对这种相对比较高电压大电流的电池进行保护。

由上图可见，此模块主要以下几部分构成：

①以共模滤波器 ACM7060 为主体搭建，其首要目的是将 BATTERY 和 MGND 进行隔离，得到 VCC_12V 和 PGND，再对主板进行供电。

②在共模滤波器的主体上，增加了三个滤波电容 C1、C2、C3 消除纹波。

③由于电池电压较高，增加防反接 MOS 管 AO3400，避免电源反接造成的危险。

④R1、R2、C7 构成延时上电功能，可以满足防止突然上电，导致后面负载烧坏的反应。同时可以减小后级负载太大，导致前端电压被拉低，造成系统出现故障。

⑤R3、D1 构成电源指示灯。

(1) EdgeBoard 供电系统

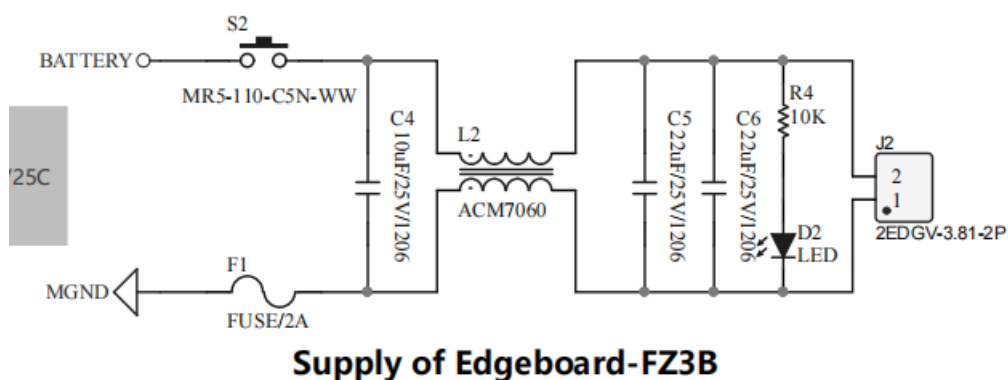


图 3-2 EdgeBoard 供电管理

由上图可见，此模块与主电源管理模块结构类似，所以不再赘述。唯一的不同点是取消了延迟上电和防反 MOS 管，增加了 2A 的自恢复保险丝。

(2) 5V 供电电路

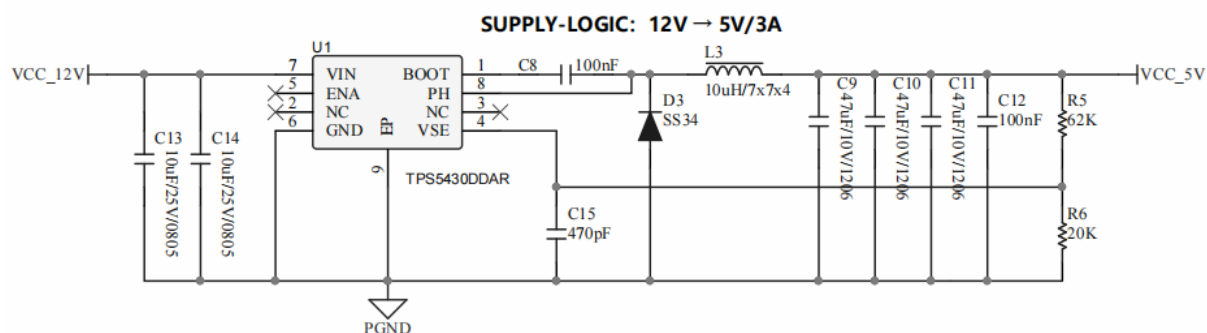


图 3-3 5V 供电电路

此电路是基于 TPS5430DDAR 建立的降压电路。

输入电压的范围：10.8V-----19.8V

输出电压：5V

输入纹波电压：300mV

输出纹波电压：30mV

输出电流额定值：3A

(3) 3.3V 供电电路

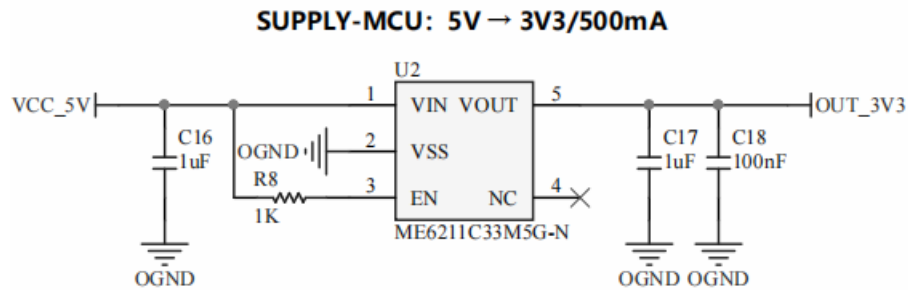


图 3-4 3.3V 供电电路

此电路是基于低压差线性稳压器 ME6211C33M5G 建立的降压电路。

(4) 舵机供电电路

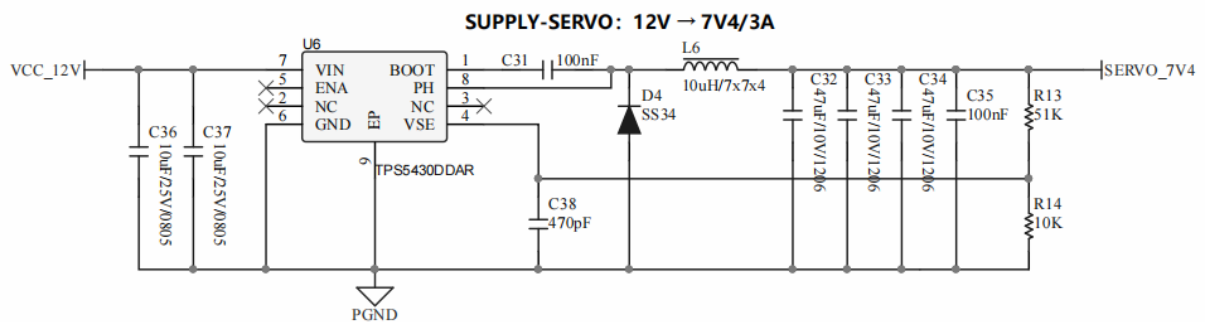


图 3-5 舵机供电电路

与主板的 5V 供电电路相同，也是基于 TPS5430DDAR 建立的降压电路，通过调整输出端 R13、R14 的比值来控制输出电压，从而控制舵机的响应速度。

3.2 USB 转串口模块

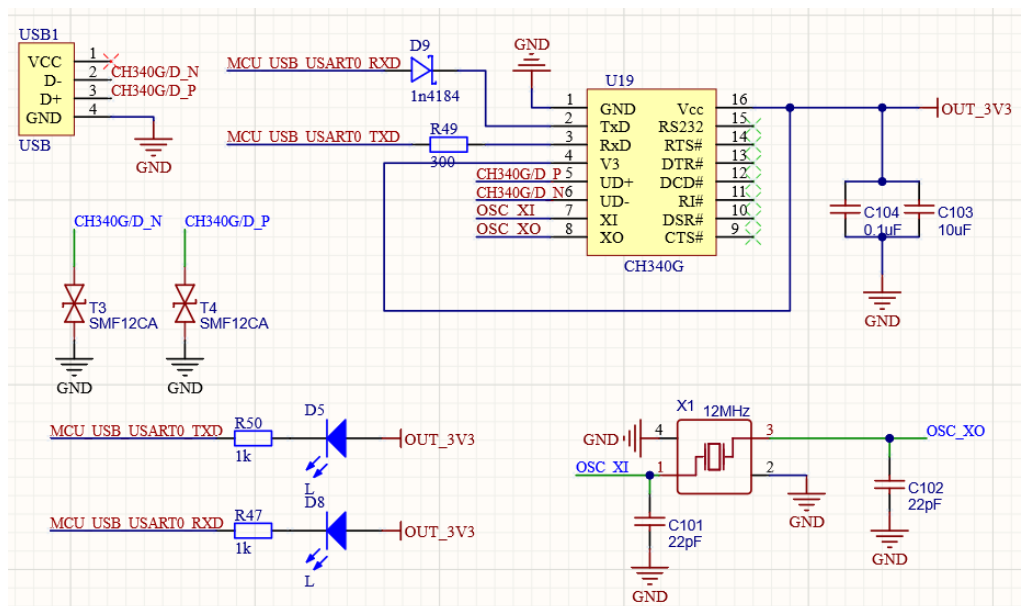


图 3-6 USB 转串口电路

在这里，上位机EdgeBoard通过传输信号，经转换芯片CH340G处理后传输给英飞凌MCU，完成上下位机的通信。其中D9为防止电流倒灌损害MCU。R49为CH340G与MCU之间的限流电阻。T3、T4为瞬态抑制二极管，防止浪涌。

3.3 传感器与人车交互模块

液晶 OLED 用于显示摄像头拍摄到的实时图像便于调整参数。五向开关与 MCU 相连，通过上下左右进行参数调整并在 OLED 上面显示。陀螺仪在智能车进入施工区时进行惯性导航。预留一个串口和一个无线串口作为冗余，便于调试和后期增加新设备。

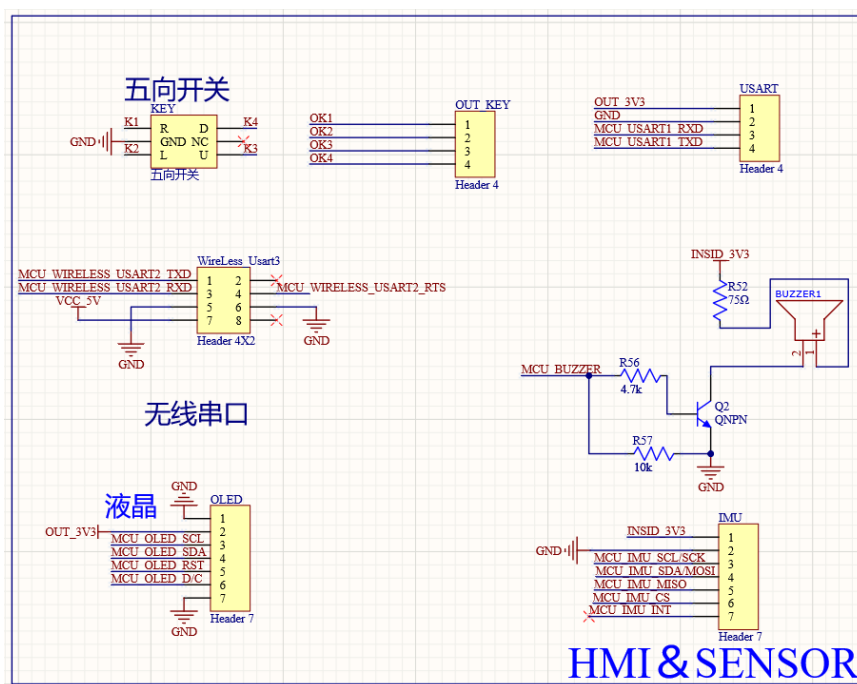


图 3-7 传感器与人车交互模块

3.4 电机控制模块

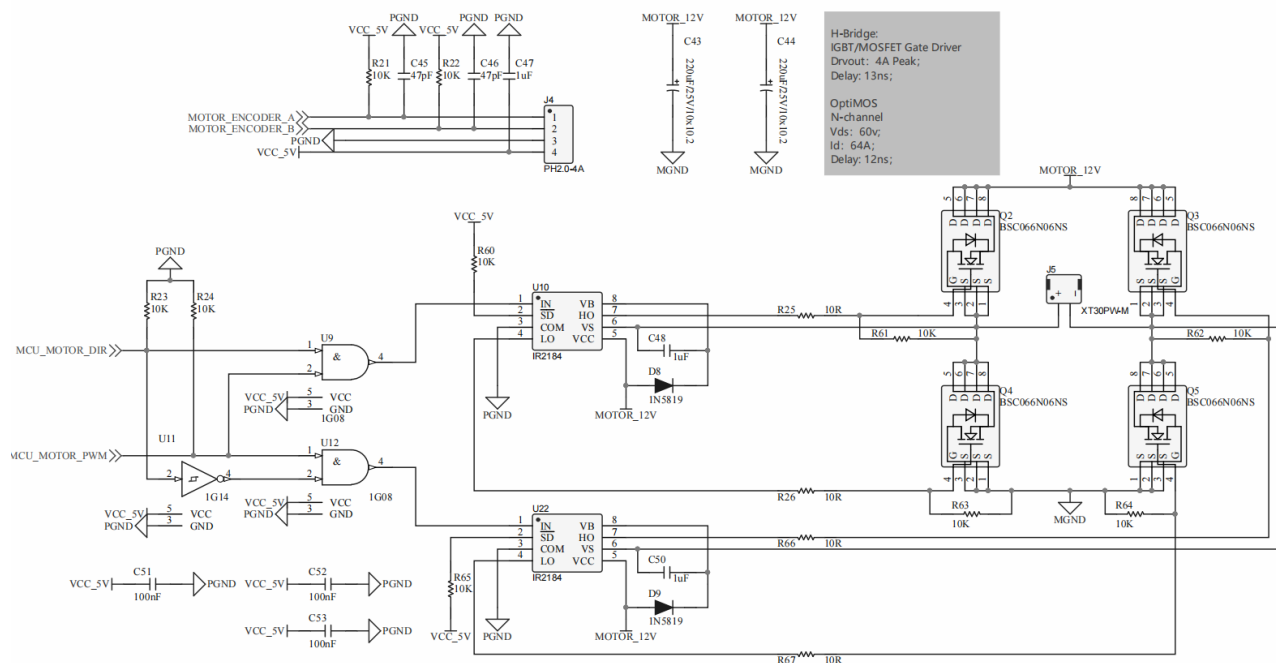


图 3-8 电机控制模块

电机驱动电路是用 MCU 输出的 PWM 信号，通过两个半桥电路驱动电机，使用

IR2104 半桥驱动芯片。

因为下桥臂MOS源极接地，想要导通只需要令其栅极电压大于开启电压 V_{th} 。而上桥臂MOS源极接到负载，如果上桥臂MOS导通，那么其源极电压将上升到H桥驱动电压也就是MOS的供电电压，此时如果栅极对地电压不变，那么 V_{gs} 可能小于 V_{th} ，又关断。因此想要使上桥臂MOS导通，必须想办法使其 V_{gs} 始终大于或一段时间内大于 V_{th} (即栅极电压保持大于MOS管的电源电压+ V_{th})。

下图是IR2104S的内部原理框图。此类芯片的内部原理基本类似，右侧两个栅极控制脚(HO和LO)均是通过一对PMOS和NMOS进行互补控制。

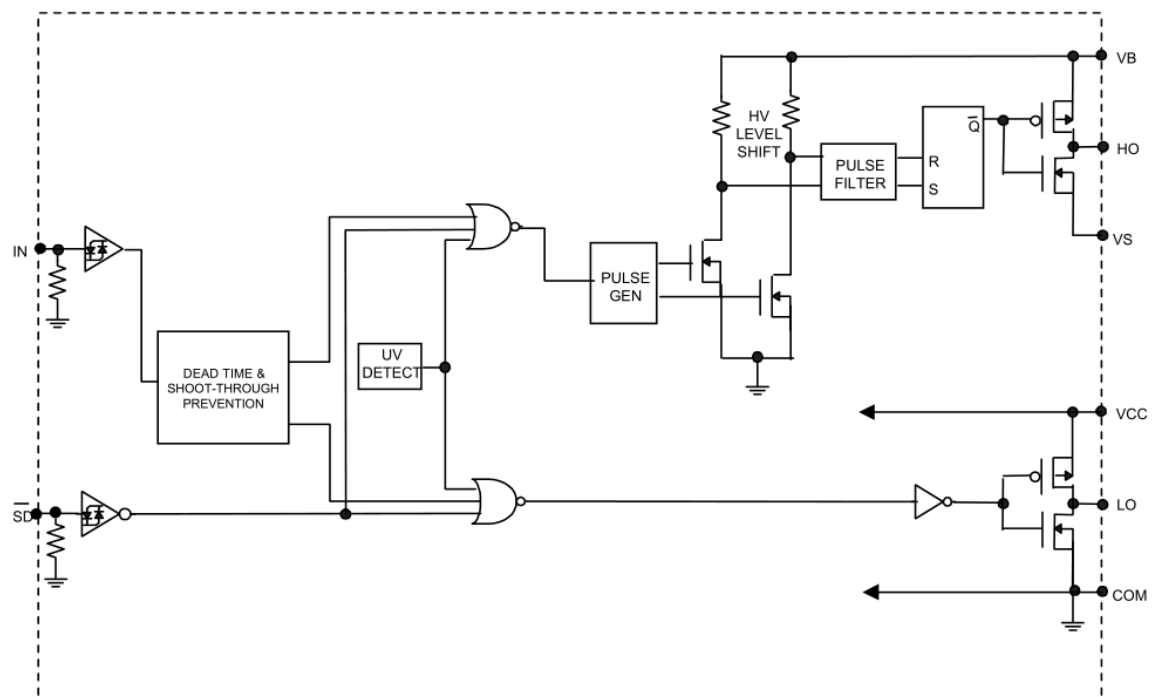


图 3-9 IR2104 内部结构

第 4 章 软件设计

要想小车能够快速稳定地对赛道进行循迹，那么就需要设计一套逻辑严谨、稳定优化的控制算法结构。识别的准确，算法的好坏直接决定了小车的速度上限，所以图像的采集与处理也就是整个算法的核心部分，在本章我将具体对其进行简单分析。

4.1 程序主体流程

第一次启动时会进行各硬件外设的初始化操作，其次是延时发车，等待延时结束后，程序开始正式进入循环阶段。这里的循环阶段主要检测图像是否采集完成，当然也可以将图像的识别以及处理放在摄像头的 DMA 中断里，一旦 DMA 采集完成触发中断即表示一帧完整的图像已经采集完成，这时候就可以开始数据处理了，数据处理主要分为图像预处理、元素识别、转角控制与期望速度计算。由于总钻风摄像头采集的原始图像是灰度图，所以我们组别在对图像进行处理之前先对该灰度图像进行了二值化处理，二值化处理的关键要当阈值的确定，实践证明动态和静态的阈值均有其应用价值。

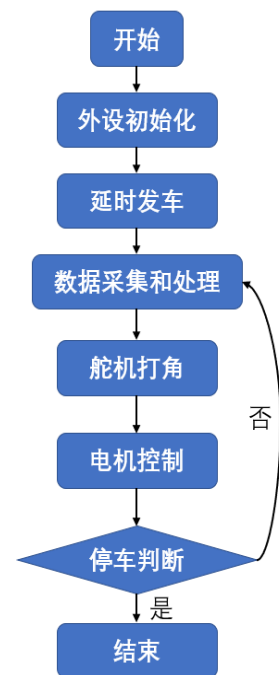


图 4-1 程序主体流程

图像解压完毕之后就可以进行预处理了，预处理主要就是寻找赛道边线，以及计算边界的斜率等等。得到这些信息之后，我们就可以开始进行对赛道元素的识别，具体的识别思路可以参考后面章节，之所以需要识别赛道元素，是因为不同的元素需要不同的路径规划以及不同的行驶速度。识别出赛道元素并执行相应操作之后我们便得到了当前期望的打角与速度，通过小车当前的状态并结合合适的控制算法得出需要输出给舵机和电机驱动模块的 PWM 占空比。

4.2 预处理

4.2.1 赛道边沿及中线提取

赛道边沿的提取是所有识别以及控制的基础，也是感知赛道信息最直接的方式，可以说这是一切后续程序的基础，所以找到一个准确的赛道边界是非常重要的。在提取赛道边沿之前，我们先来看看原始图像的特点，由于图像透视的原理，摄像头所采集的图像是近大远小的，所以根据这个特点，我们对赛道边沿

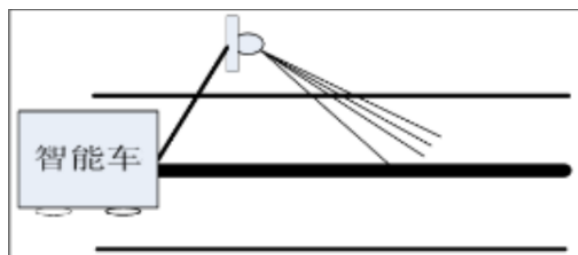


图 4-2 小车循迹示意图

提取时最好还是先从近几行开始入手，在得到最近几行准确的赛道边界之后，根据近处的边线给定远处边线的寻找范围，通过阈值判断找出下一行的准确边界，以此由近及远得到完整的赛道边界，当左右边界相等或者图像搜索完成后即可对边界进行保存用于后续处理。

既然说到了边沿提取那就不得不谈一下这方面的未来趋势，由于现在很多组别都在采用灰度摄像头，而将灰度图像二值化势必会丢失很多重要的信息。所以很明显，直接对原始灰度图像进行处理可能会比目前的方法更加准确且不易



图 4-3 二值化图像

失真，对于灰度图像的边界寻找来说可以直接对各像素点灰度值的差比和来寻找边界，当差比和达到一定阈值时即可认为识别到了边界。使用这种方法可以有效地抑制光线不均匀带来的干扰，换句话说，对于现场环境的依赖程度会大大减弱，间接增加了小车循迹时的稳定性。

在得到边沿之后即可开始对中线进行求取，先将左右边线取中即可得到原始的中线

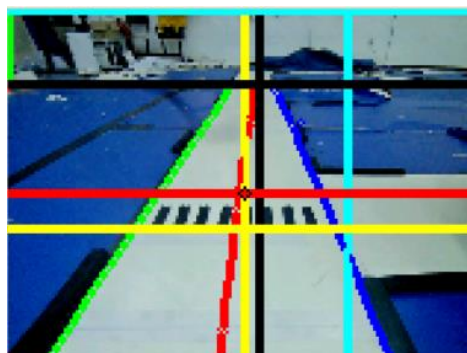


图 4-4 中线提取

位置，但事实中我们可以发现这时的中线会有很多噪点，这时我们就需要开始对中线进行一次简单的滤波。滤波的方法很多，只要能让中线变得圆滑且不失真严重都应该是满足要求的，由于数据是离散的，很多方法其实不是很实用，在这里我们采用最小二乘法来拟合这些离散的中线，然后根据拟合出来的曲线得到滤波后的真实中线，由图可知，这种方法来滤除噪点或者异常点的效果是非常好的。

4.2.2 图像矫正

图像校正是指对失真图像进行的复原性处理。引起图像失真的原因有:成像系统的象差、畸变、带宽有限等造成的图象失真;由于成像器件拍摄姿态和扫描非线性引起的图象几何失真;由于运动模糊、辐射失真、引入噪声等造成的图像失真。图象校正的基本思路是，根据图像失真原因，建立相应的数学模型，从被污染或畸变的图象信号中提取所需要的信息，沿着使图象失真的逆过程恢复图象本来面貌。

实际的复原过程是设计一个滤波器，使其能从失真图象中计算得到真实图象的估值，使其根据预先规定的误差准则，最大程度地接近真实图象。摄像头在拍摄图像的时候，最理想的位置是能垂直于拍摄平面，这样才能保证图像能按原来的几何比例重现。然而在智能车的实际运用中，受到智能车车体结构的限制和车体控制需摄像头有一定视野距离的要求，摄像头一般与地平面成一定角度安装。所以这时我们看到的图像是会有一些畸变的，这样也会产生一些问题：垂直线被拍摄成斜线导致斜率计算错误，远处的弯道被压缩导致曲率计算错误等等。所以我们必须对图像进行桶形变换和逆透视变换加以矫正，为此我们编写了一套 MATLAB 程序得到矫正矩阵，在程序中得以计算出每个像素点所对应的真实世界坐标值，具体效果可见图片展示。

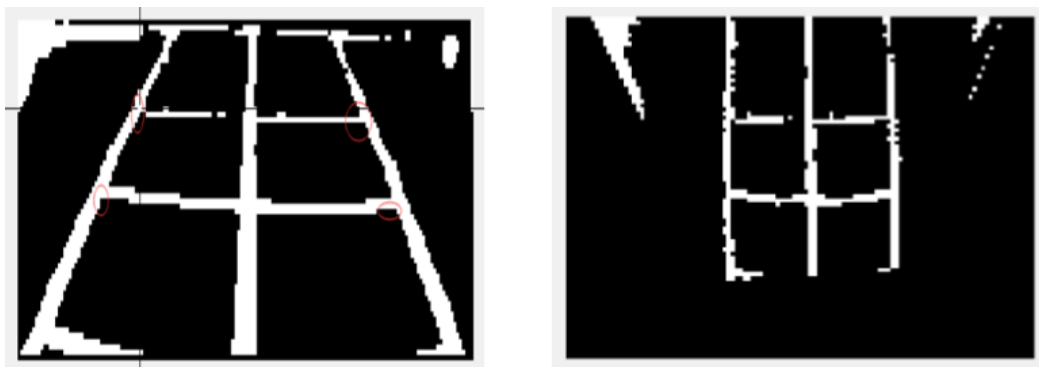
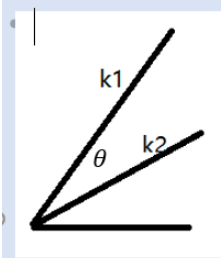


图 4-5 图像校正

4.3 元素识别

4.3.1 识别可用几何特征

用于识别的几何特征还是比较多的，常用的有图像赛道边沿的跳变行和斜率、赛道的真实宽度、特殊角点的角度、以及中线特殊点距离车头的距离等等。合理使用这些特殊的几何特征可以让元素识别变得更加简单直接。赛道的真实宽度可以由矫正后得出，角点的角度可以通过以下公式计算得出，该公式局限于必须先进行逆透视变换才能使用，得到角点的角度之后可以很容易的区分很多相似元素，例如双十字和环岛之间可能互相误判的情况就可以避免。



已知两条直线的斜率为 k_1, k_2 ，求两条直线之间的夹角

$$\tan \theta = \frac{k_2 - k_1}{1 + k_1 * k_2}$$

图 4-6 几何方法

4.3.2 坡道识别

坡道的识别比较明显，首先坡道图像和长直道图像很相似，所以检测坡道的

第一个判断条件就是必须赛道已经延伸到了图像尽头，我们检测坡道的主要依据是图像有明显异常的赛道宽度，这个很好理解，相当于赛道立起来之后逆透视矫正出来的赛道真实宽度会有明显增加。

识别到坡道之后需要对其处理，否则上坡后的杂乱图像会导致很多问题，我们采取的策略是上坡之前记录陀螺仪的状态，解算得到当前角度，使用陀螺仪进行坡道上的方向控制，再通过检测车身俯仰角判断坡道是否结束，辅以路程判断，实践表明效果优于硬核的舵机限幅控制。当所获取的图像拟合出来的赛道宽度比真实赛道宽度大一定范围，且两边线斜率符合一定变化范围，判定为坡道，并进行延时，在延时的时间段内使智能车直走。



图 4-7 坡道识别

4.3.3 十字识别

十字路口的图像其实特征还是比较明显的，除了入环阶段就只有这里会有一些全白行，当然这一特征点是完全不够识别十字路段的，十字路口的另一个特征就是拥有四个直角角点，是否是直角角点可以根据前面的几何特征识别里面的公式进行计算，判断四个角点的角度都在一定范围内即可认为识别到了十字，识别十字之后将远处的图像作为转向依据即可很好地驶过十字，避免在十字里抖动浪费时间。



图 4-8 十字识别

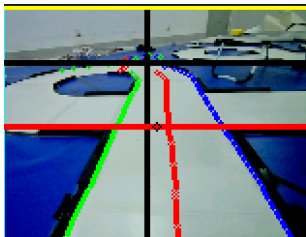


图 4-9 十字处理

车辆行驶十字的过程分为初入十字，处于十字中心处，出十字等多过程。十字元素类型分为中间十字，斜入十字等多类型。十字的处理：识别十字（屏蔽圆环）—>判断十字类型—>判断十字过程。

对于斜十字，从下向上纵向搜索，不在是横向搜索。纵向搜索的好处在于可以通过边界的像素分布看出当前搜索的边界属于左边界还是右边界，通过区分左右边界后可以很容易的找到十字缺口处的拐点。找



图 4-10 斜十字识别

到两拐点后接下来就没有什么难点了，接下来我的处理方法是重新计算两个拐点之间的中线，简单来说就是一个拟合直线的过程，这样就可以得到一个很好地中线。在采用了这种算法之后在十字路口基本就再也没有出错的时候了，但是建议选择广角镜头这样能够看到近处更大的范围，有助于十字搜索判断。上述方法即是本人采用的十字处理方案，这里只是一种情况，其他情况类似，通过自己的实验很容易把条件约束起来，因为每个人遇到的情况都有所不同，但是方法可以类似。

4.3.4 环岛识别

环岛识别比较复杂，可以分为很多阶段进行，但第一步是得先能够准确识别到环岛，环岛最基础的特征主要是一边与长直道相同，另一边斜率变化会比较大，根据这个特点会比较

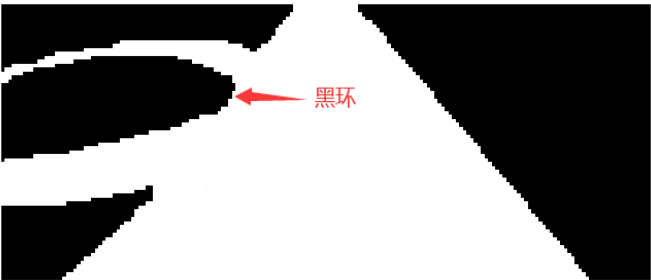


图 4-11 环岛识别

容易且准确的识别到环岛，识别到环岛之后要对环岛中间最具特征性的黑环进行匹配，我们这里采用的是斜率匹配法。在准确匹配到黑环之后，便可根据黑环位置补出期望路径的中线位置，对所得中线矫正滤波之后，便可得到比较理想的入环路径，且路径在最小二乘法的滤波之下会



使小车以一个圆滑的姿态进入环岛。

进入环岛之后的图像和普通弯道并没有区别，所以这里并不做任何处理，当即将出环是会发现有一边边界会

出现斜率突变的现象，这时就需要

开始执行出环处理了，如果不做处理可能会导致小车直接反方向打出或者直接冲出赛道。我们出环的处理方式与入环相似，还是匹配圆环中间的黑环，匹配之后对所得中线进行滤波，是小车得到圆滑的出环路径，是小车度过这段危险路段之后基本不需要什么处理就可完整驶出圆环，只需简单滤波即可。

环岛是难度较大的元素，须考虑大小环岛的图像差异，将环岛处理分为以下几步：发现环岛（直走）—>过环岛中间（补线打角）—>入环岛（正常巡线）—>出环岛（补线打角）。

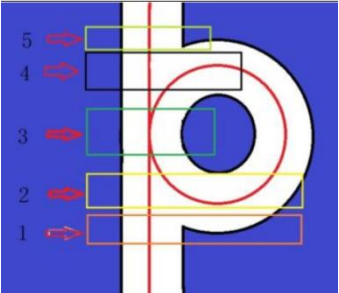


图 4-13 环岛处理过程

4.3.5 车库识别

车库的特征明显且独特，最直接识别车库的方法就是寻找斑马线，在赛道范围内寻找每一行的跳变点个数，正常赛道的跳变点个数是肯定小于一个阈值的，一旦大于这个阈值我们就可以认为识别到了斑马

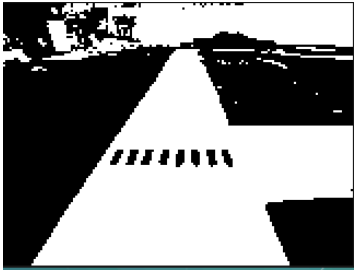


图 4-14 三岔路识别

线，进而可以相信是被到了车库。

识别到车库之后我们需要进行一些处理才能让其完美驶入车库，再次期间我们需要结合陀螺仪姿态和编码器测得路程来综合计算当前位置。大致流程如下，检测到斑马线所在行之后通过逆透视矫正即可计算其距离车头的真实距离，当到达一定距离之内使即可以开始执行入库操作。这里我们入库的原理类似于惯性导航，通过陀螺仪得到的角度以及加速度信息并辅以编码器测得的路程距离便可以粗略推测出目前的大致位置，当陀螺仪累计角度基本达到九十度附近时，即可判

断车子已经入库，开始停车程序。利用惯性导航的原理有一个大的缺点就是不能长距离使用，否则误差会累计的越来越大，但入库操作只需要简单的空间与时间复杂度，所以实际使用效果良好。

智能车比赛时首先要执行的是出库，当车辆放在车库靠一边时，如右出库，则将车辆放在车库的左侧，方便出库，首先设定一个固定的目标编码值，车辆启动后舵机打角随编码值不断增大，并进行相应限幅，达到一定编码值时，车身在赛道上摆正，停止给定舵机脉冲值，舵机转动根据图像控制。斑马线的标志位是有多个黑白块交界，当车辆在远处时看到车库标志位就给电机一个较低的期望值，防止在跑道时过高的速度使车辆在车库处无法正确。入库时，在靠近车库时则给舵机一个固定的脉冲值，从此时开始记录电机的编码值，达到一定编码值判断车辆已经进入车库，给舵机一个中值，使车轮摆正，并让车辆停止。

4.3.6 三岔路识别

可用以下方法判断。一、两边线出现拐点，且拐点上下边线斜率符合一定数组，二、图像中间偏上出现明显黑块，可通过判断中间列黑点数量进一步判断。三岔路最明显的特征就是左边界向左延伸、右边界向右延伸，并以此形



图 4-15 三岔路识别

成一个中部的上角点、两个下角点。因此可寻找左边界向左跳变、右边界向右跳变的点作为下角点的判断依据，可寻找黑白跳变的点作为上角点的判断依据，离开三岔路的图像与进入三岔路的图像一致，因此判断方法类同。此外，由于在三岔路分界点两条电磁线分别向左、右支路延伸，即，每条支路减少至只有一条电磁线，电感值明显减小，而在离开三岔路时，两条电磁线汇合至一起，因此，也可以根据电感值的变化判断进入和离开三岔路。

由于进入三岔路时自然循迹得到的中线会使车行驶出界或反向行驶，因此还需要进行中线的处理，补出期望的中线行驶路径，离开三岔路时也需要如此处理。进入三岔路之后的图像和普通赛道并没有区别，所以这里并不做任何处理。

此外，按照每次经过三岔路需经过不同支路的要求，需要交替地补出向左支路、右支路行驶的期望路径。

4.4 控制算法

4.4.1 舵机控制

舵机控制是控制算法中最重要的方面之一。一旦舵机控制出错，智能汽车就极有可能出界犯规了。因此，平时调车的主要任务之一就是观察舵机，使之控制合理。本队舵机的控制采取传统的 PD 控制。

4.4.2 电机控制

本队算法中，电机速度控制采用了 PID 棒棒控制的方法。其控制的思路大概如下：当实际速度与设定速度相差过大时，采用棒棒控制。如果二者相差在一定范围内，则采用 PID 进行控制。

4.4.3 参数作用与整定

为了更好地应用 PID 算法，首先必须对算法的参数有深刻的理解。下面阐述了 PID 参数的作用。

(1)、比例 P:比例项部分是对预设值和反馈值差值的放大倍数。显然比例 P 越大,电机转速回归到输入值的速度将更快,调节灵敏度就越高。从而加大其值,就可以减少从非稳态到稳态的时间,但同时也可能造成电机转速在预设值附近振荡的情形。

(2)、积分 I:应引入积分 I 解决上述问题。值在时间上进行累加。当差值不是很大时,为了不引起振荡,可以先让电机按原转速继续运行;此时将这个差值用积分项累加;当这个和累加到一定值时,再一次性进行处理,从而避免了振荡现象的发生。可见,积分项的调节存在明显的滞后,而且 I 值越大,滞后效果越明显。

(3)、微分 D:微分项部分其实就是求电机转速的变化率,也就是前后两次差值的差而已。也可以说,微分项是根据差值变化的速率,提前给出一个相应的调节动作。可见微分项的调节是超前的;并且 D 值越大,超前作用越明显;可以在一定程度上缓冲振荡。比例项的作用是放大误差的幅值,而目前需要增加的是“微分项”,它能预测误差变化的趋势,这样,具有“比例+微分”的控制器,就能够提前使抑制误差的控制作用等于 0,甚至为负值,从而避免了被控量的严重超调。

调节过程：

- (1)、在输出不振荡时，增大比例增益 P ;
- (2)、在输出不振荡时，减小积分系数 I ;
- (3)、在输出不振荡时，减小微分系数 D .

我们先得到电机在定值输入下的响应曲线，使用 MATLAB 仿真得到大致的 PI 参数，再通过试凑法对 I 和 D 进行细调。

4.5 滤波算法

4.5.1 卡尔曼滤波

滤波算法我们主要对期望速度使用卡尔曼滤波进行滤波，可以有效滤除噪点，平滑曲线，卡尔曼滤波是一种利用线性系统状态方程，通过系统输入输出观测数据，对系统状态进行最优估计的算法。由于观测数据中包括系统中的噪声和干扰的影响，所以最优估计也可看作是滤波过程。

数据滤波是去除噪声还原真实数据的一种数据处理技术，卡尔曼滤波滤波在测量方差已知的情况下能够从一系列存在测量噪声的数据中，估计动态系统的状态。由于它便于计算机编程实现，并能够对现场采集的数据进行实时的更新和处理，卡尔曼滤波滤波是目前应用最为广泛的滤波方法，在通信，导航，制导与控制等多领域得到了较好的应用。

第五章 神经网络模型

5.1 任务概述与操作

本次比赛中要求我们自主训练一个神经网络模型，并部署到 edge board 上。通过运行这个模型，检测并识别赛道上标志物（例如施工区、三岔路、禁止通行、坡道、加油站等），并在识别到标志物后完成对应任务。

序号	名称	说明	图示
1	泛行区标志	表示前方三岔路口围成的泛行区域，内部区域包括蓝色底布均可行驶。	
2	禁止通行标志	放置在泛行区域进出口连线上，车辆需要绕过此标志进行通行。（此标志高出距离地面有 2cm 高度其余均紧贴）。	
3	施工区标志	表示前方为施工区，需要绕行赛道外障碍桩围成的临时路段。	
4	坡道标志	表示道路前方有坡道。	
5	加油站标志	表示前方为加油站，车辆需要驶入加油站并按照指定的出口驶出加油站。	
6	加油站出口数字标志	加油站设置有“1”和“2”两个出口，并在出口地面贴有对应的“1”和“2”数字标志。比赛时加油站的入口处会随机放置车辆需要驶出时的出口数字。	

完成这样一个任务主要分为以下几步，采集数据，标注标签，训练模型，部署模型。接下来对这些步骤进行详细说明。

5.1.1 采集数据

为达到更好的识别效果，训练的数据就要和实际运行的数据尽量接近。一开始我们采用摄像头在静态下采集标志图片的方式，获得训练数据。但在实践中发现，当小车高速运行时，图像往往会产生动态模糊，和静态下图片有较大差异，导致训练出的模型不尽如人意。为此，我们采用多种方案进行改进。首先，造成拖影现象的原因是卷帘快门摄像机曝光时间长，于是我们换用了全局摄像头以提高快门速度，减小曝光时间，获得更清晰的图像。同时我们改进采集方案，令小车一边运行，一边采集图像。用动态过程中采集的图像进行训练，取得了更好的效果，提高了识别率，也降低了误识别率。并且我们在接近标志时进行适当减速，在速度降低后，也能提高图像的清晰度。



图 5-1 静态图像



图 5-2 运动拍摄图像

5.1.2 数据标注

因为最终要完成的任务（如施工区），需要我们确定车距离标志的位置，方便最后转入锥桶围成的区域。因此，我们决定训练一个目标检测模型，通过目标检测返回的标注框在图像中的位置，逆透视变换为实际离小车的距离。

首先将各个图片随机打乱并统一按顺序命名，并以大概 8:2 左右的比例划分训练集和验证集，对这些图片进行标注。

数据标注我们采用的是 labeling 软件，标注为 voc 格式数据。

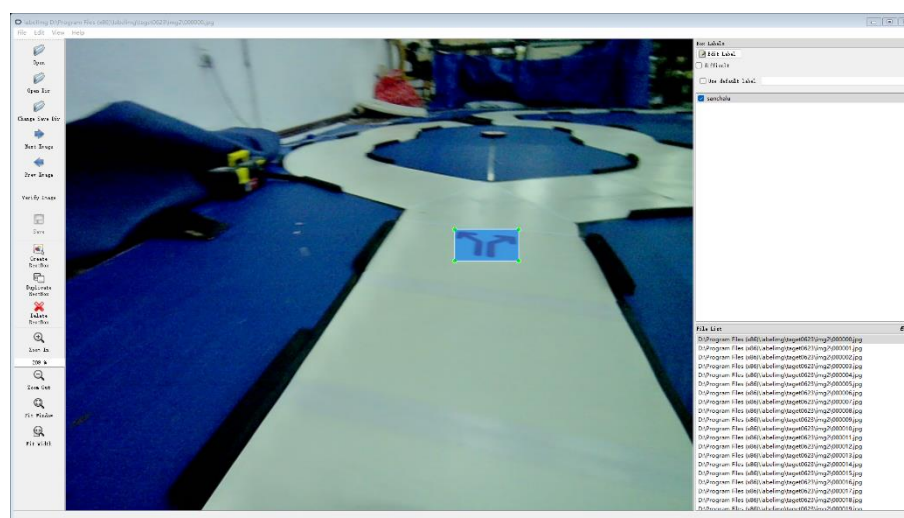


图 5-3 标注软件

标注过程为使用一个矩形框框出标志所在位置，并为其打上标签，确定其类型。标注时要注意标志应该完整，不过于小。标注时要使矩形框刚好能够框住整个标志，以优化训练效果。

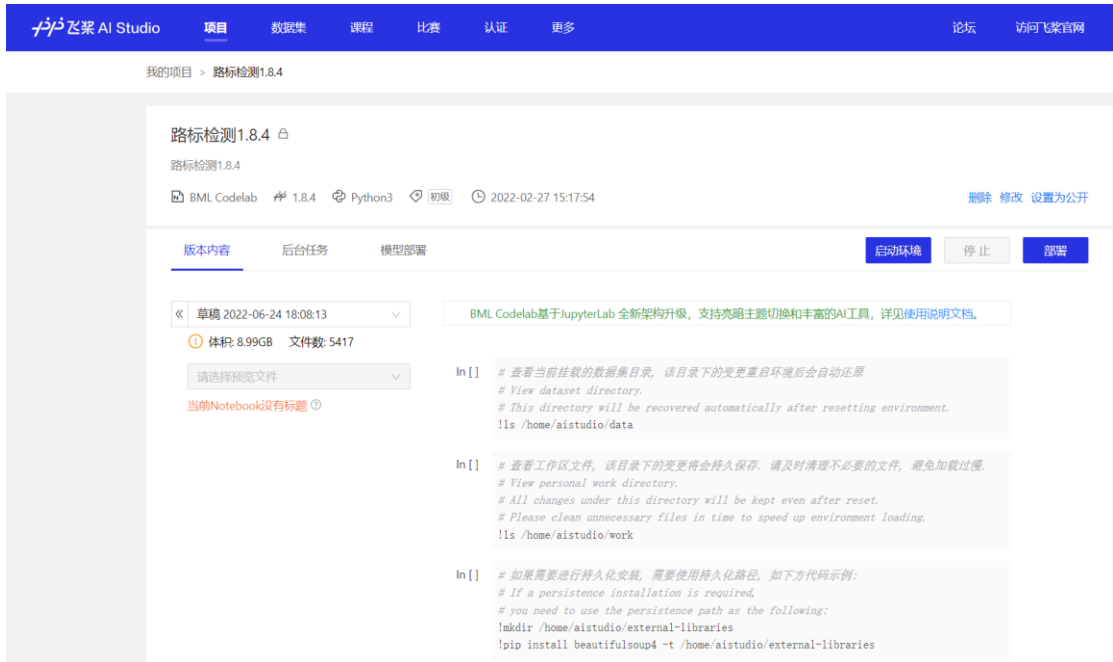
标注完成后会得到一个 xml 文件，这个文件记录了标签的各个数据，有了这个文件，才能够进行下一步训练。

5.1.3 训练模型

因为这次比赛是基于百度的 edgeboard 硬件，因此，我们的模型训练也在百度提供的 aistudio 平台上，使用百度的 paddle 框架完成。

在模型训练之前，我们需要先确定所选择的模型结构。可供选择的模型有轻量化适合部署在移动设备上的 monilenet—ssd, 识别准确的 yolo3 等等。作为竞速组别，程序运行的帧率，与小车的控制性能密切相关。我们希望运行帧率尽可能提高，因此，我们倾向于选择更为轻量，推理速度更快的模型。又因为本次识

别任务的背景为白色赛道和蓝色背景布，并不复杂，且各个标志区别较为明显，属于相对容易识别的任务。因此，我们最终选定了相当轻量化的 mobilenet-ssd 模型。经过测算，只运行 mobilenet-ssd 推理模型时，edge board 的处理速度大概能达到 30fps。



在 aistudio 上训练模型，我们需要将图像和 xml 文件一一对应并打包上传。然后在训练路径上将所有数据添加进去，同时修改相应的配置文件，调节学习率，训练轮次，分辨率等参数，以期达到更好的训练效果。



图 5-4 学习率对数曲线

5.1.4 训练模型

在 aistudio 上完成训练后,可以得到初步的模型文件。在 aistudio 上可以对模型进行验证,以确定模型的识别效果。模型确认无误后,可以导出 model, params, config, 三个文件,对应模型,参数和配置文件。将这几个文件导入到 edgeboard 中,再对应更改一些配置参数,就可以在程序中调用这个模型,并得到识别结果了。



图 5-5 预测结果 1



图 5-6 预测结果 2

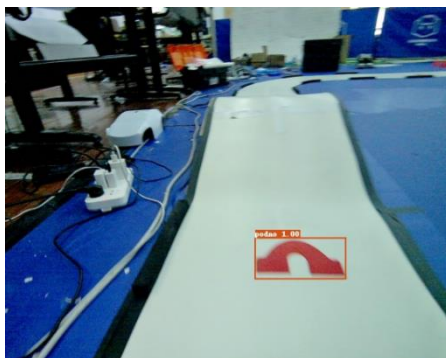


图 5-7 预测结果 3

调用时需要用 c++代码创建一个检测类中的对象,将图像通过这个类中的方法进行推理,推理过程在一个独立的线程中运行,待识别结束可以从这个对象中获取推理结果。但要注意使用锁和信号量对数据进行保护。

得到的识别结果是一个结构体数组,包含(类型 type,得分 score,目标框左上角点的横纵坐标 x、y,目标框的高和宽 height、weight)。我们可以自由选择目标得分的阈值,来调整识别的准确率。同时为了防止偶然因素引起的误判,我

们还加入了消抖滤波，只有连续多次推理结果高于阈值时，才采信这个结果。

我们还可以根据目标检测框在图像中的位置，进行逆透视变换，得到目标距离小车的位置，从而方便进一步完成任务。实践中，我们把车停在距离标志 $X\text{cm}$ 的位置上，在电脑上显示目标框的中心的 Y 坐标 ($Y=y+0.5*\text{height}$)，记录二者，变换不同距离多次记录，得到多组 X , Y 的数据，使用一个多项式拟合 X 和 Y 。从而得到目标框位置 and 实际距离的变换公式。

5.2 小车自动行驶及标志物识别处理

5.2.1 标志物采集实现以及优化方案

标志物数据集的采集是基于原车道线数据集采集程序的，一共尝试了两种数据集采集的方法。一种是让小车在手柄的遥控控制下沿车道线行驶动态的采集数据集，另一种是将小车的速度调整为 0，然后在有标志物处对标志物进行全方位多角度的数据集采集。通过以上两种方法能够得到丰富度非常好的数据集。标志物数据集经过采集打标后直接进入网络训练效果一般，训练出的模型鲁棒较差，不具有泛化性。所以我们用了更科学和系统的的方法对所采集的数据集进行了

人工纠错，纠错后再进入网络模型训练，效果相对更好，模型更加稳定，准确率和召回率也更容易调整到较为平衡的值。为了获得更高的模型得分，我们还尝试了模型融合，试图将 YOLOv3 和 SSD 模型融合，但部署有些问题，因此并未采用该方法。

5.2.2 标志物数据处理实现以及优化方案

由于我们在标志物的训练方面针对性的采用了三种不同的方案，所以从数据处理 部分开始就有了很大的不同。方案一采用了比赛官方提供的参考项目，使用的是标注单个交通标识的数据集。这一方案所采用的数据集就只需要按照每种单独的交通标志物进行分类采集，再利用软件 LabelImg 进行标志物标识，自动生成 xml 文件，然后直接替换掉原项目中的数据集就可以开始训练了。方案二

是对方案一做出了解答质疑性质的修正。在方案一训练后进行测试验证的过程中我们发现了一个频繁出现的问题，对于一张包含有多个检测目标的图片，有时模型甚至一个标志物都不能够检测出来，有时候只能检测到一个标志物，这种情况会随着训练的不断进行有所改善。为了验证和解答疑问，我们又尝试了将一张图片中能够明显识别出的标志物全部标识出来的标记方法。仅替换数据集进行训练又会出现目标标注框过多的问题，相较于一张图片只有一个框的标记方法，多个标记框的训练也让模型按照很多目标同时出现的情况，去尽可能完整的标注它认为的需要检测的所有可能的目标。这里就需要对训练网络中的各种参数配置以及数据处理代码做出适当的修改。

在对于比较难检测的标志进行单独训练时，也像车道线训练项目的模型那样采用了 CNN 的神经网络对训练集进行训练，其中的网络结构根据自身的需求，是可以做出相应的修改的，然后通过训练、测试、部署到小车上，来检验是否取得了很好的效果，最终决定是否做出相应的更改。例如在实验过程中，对于标志物是否存在，需要训练后的模型输出为 0 或者 1 表示该标识是否存在。为了获得区分度更加明显的效果，对于这种输出需求为是否的模型，在输出层采用了 sigmoid 激活函数。

采取了这一激活函数后，可以通过测试看到分数区分度的两极分化非常明显，但在数据集很少、训练次数非常有限的情况下，把是判断为否、否判断为是的错误也非常夸张，所以处理不好就会出现非常极端化的错误。在通过增加合适的训练集规模，完成足够轮次的反复训练后，在测试集上的结果是既有明显两极分化的分数分布，又有正确无误的判断，可以说是网络模型的构建上已经取得了很好的效果。但令人遗憾的就是由于该模型相较于原版可部署的车道线模型，所做的修改也算是有一定工作量的，最后导致模型部署到小车上后，会一直报数据类型不对应的错误，考虑到小车自身携带的资源也是非常有限的，所以部署成功可能会有困难，所以最终也放弃了这一模型。

方案三是为解决有些特定标志物不能很准确识别的问题。体现到小车行驶的实际情况下，就是即使训练出的模型能够在电脑上进行很好的目标检测，但部署到小车上以后就总是不能够识别出来。这个问题的解决方法就是先对图片进行一些图像处理，将处理后的图片作为数据集，建立新的网络，然后用处理后的图片

去训练得到模型，与之相关的就是在小车自动行驶的代码中也需要对图片做出相应的处理。此外，我们还对 YOLOv3 模型进行了一些修改。我们调整了一些超参数，比如对参与 loss 计算的交并比的大小的调整和对衰减比的调整等。并且修改了模型采用的优化器，将 RMSprop 优化器改为 Adam 优化器（相对更好的选择），但优化效果一般。我们还尝试了不同类型的聚类算法，最后采用了基于模型和网络的聚类算法。

5.2.3 标志物模型

标志物的第一种训练方式总体上来说就是对原例程进行了替换数据集然后直接训练。首先是数据集采集，可以采集动态行驶过程中的数据集和静止的全方位的数据集。然后接着就是数据集处理，将采集的所有图片按照不同标识物进行分类挑选，再按照从 0 开始的顺序顺次重命名。所有类别的图片按照各自不同的文件夹放置图片，然后通过 LabelIm 标识出标志物检测框并生成 xml 文件。最后将图片和 xml 文件打包作为数据集替换原数据集就可以直接进行训练了，训练到 loss 降低效果不明显后还可以降低最大学习率再进行训练。标志物的第二种方式相较于第一种方式，主要区别就是将原标注单个检测目标框的数据集，改为标注图片中能够清楚识别的所有目标的数据集。但是如果只是改变数据集标注框的数量，后面在预测时就会出现很多预测框，并且其中会有很多低得分以及重复标注的检测结果。为解决这一问题，经过对神经网络配置参数以及数据读取器的修改，在电脑上能够看到预测效果有很大改善。

修改的部分就是将每张图片中标志物的最大可能值修改为 2，将目标得分过滤值变为 0.2，在读取器中加入同一张图同一目标只能有一个标识框。最终在电脑中测试能够有很理想的预测结果。但实际部署到小车上以后发现模型并没有体现出相较于第一种有明显的优化效果，所以最终没有采用该模型。标志物的第三种方式本质上是对前面方式的弥补措施。在实际测试过程中，对于限速标志和红绿灯的识别总是会出现模型在小车上部署后效果很差，但是在电脑上检测效果都很好，为解决这一问题，我们重新对图片进行了一些针对性处理，然后用新的网络训练并部署到小车上，取得了较好的效果。较难的是部署这一过程，需要对网

络进行深度理解，然后更改相应代码，才能实现。

第 6 章 遇到的问题及解决方案

6.1 硬件问题

(1)、上下位机通信和核心板与外设连接时接口不兼容，尤其是 USB 接口数量不足。

通过增加拓展坞解决问题。

(2)、Edgeboard 板子发热严重，增加固定铜柱高度会导致车身重心偏高，不利于车身行驶的稳定。

通过改变上下位机摆放位置，不在采用官方推荐布局。将上位机 Edgeboard 板子置于下位机 TC264 上方，便与散热。

(3)、比赛要求必须配备车壳，由于官方给定车模车壳高度太低，开的孔位不方便对车进行调试。

通过自己根据本队车身结构特点自行 3D 建模打印专属车壳。

6.2 软件问题

(1)、运行预测模型时帧率会大大降低，帧率的稳定性也会下降，导致控制迟缓，不稳定。

通过在正常赛道关闭预测，同时不断检测赛道中红点（元素标志为红色）数目，当红点数目达到一定阈值后（即遇到元素标志）开启预测解决问题。

(2)、赛道元素加油站需要绕过锥桶并且行驶在蓝布上，利用惯性导航这一方案对进入施工区的车身位置和车身姿态要求很高，但是由于车模行驶会受多种因素影响，难以保证车身进入施工区时的位置和姿态不变，导致车模无法顺利绕过锥桶完成任务。

通过快进入施工区减速，然后在长直道校正车模姿态，进入施工区后惯性导航的相关参数通过车身进入位置和姿态调整，依次克服车模进入施工区是位置和姿态随机性的影响，从而解决问题。

(3)、上下位机通信时由于通信花费时间造成了超中断，严重影响了程序运行和车模行驶。

通过改变通信的时机，错开下位机接受数据做控制和下位机发送数据中断序

列，大大减少了中断服务程序的执行时间，解决了超中断的问题。

6.3 其他问题

车模本身由于结构原因，导致底盘偏低，过坡道是会磕到底盘，由于比赛要求不能更改车模本身结构，导致只能通过调整减震缓解磕碰，没办法从根本上解决问题。

第七章 总结与心得

全国大学生智能汽车竞赛已经成功举办了十六届。这次的实践过程，涉及到了控制、模式识别、传感技术、电子、电气、计算机、机械等多个学科，大大提高了我的知识融合和实践动手能力。我主要从软件方面入手，一步一步构建小车的软件系统，最终与硬件的同学合作，制作出一个外形美观，运行稳定的小车，能够稳定完成循迹和特定任务。

当然在备赛的过程也存在一些问题，我自身还是存在一些不足。例如，准备初始阶段，由于自身知识积累不够，遇到很多困难，走了不少弯路，也犯了不少错误。我们在系统建模与仿真方面的能力还是略显不足。但我们在不断学习中、在实践中熟练操作 MATLAB, C++ 工程结构搭建, Linux 系统使用, 逐渐对建模、仿真等有了进一步的了解。相对于其他传统组别，我们多了许多全新的模块，由于没有任何可以参考的设计成果，我们刚开始无从下手，进度缓慢。但是后来在与团队的合作中，我们一步一步自己摸索，从零开始，逐渐解决问题，在这个过程中极大地锻炼了我们的能力。

正所谓实践是检验真理的唯一标准。在课堂上，我们学习了很多理论知识，但是如果我们没有知行合一，我们永远无法真正成长，真正把知识变为能力。参加智能车比赛就是将我们在课堂上学的理论知识运用到实战中。作为自动化专业的学生，我们必须有强大的动手能力，能解决工程应用过程中的实际问题，拥有一定的自我创新能力，我们才能够在竞争愈发激烈的今天，建立自己的独特优势，为自己获取一席之地。

致谢

制作智能汽车是一个长期的工程，我们队并不是在孤军奋战。本文所涉及的，不仅仅是我们队辛勤劳作的结晶，同时还有许多人智慧的光芒。感谢一年中陪我们在实验室一起奋斗的同学，感谢他们跟我们提的好建议、好想法、好思路；同时也要感谢武汉大学电气与自动化院自动化系给我们提供一个自由学习的好环境，学校教务部也给我们提供了调试场地等诸多方便；还有指导老师在一年之中给我们悉心的指导和督促，在这里都表示衷心的感谢。也要感谢教育部高等学校自动化专业教学指导分委员会引进了这样富有活力的竞赛，还有清华大学自动化系老师们多年来为智能汽车竞赛做出的贡献。此外，还要感谢华中科技大学为本届华南赛区竞赛提供场地和竞赛环境，组织培训裁判和志愿者等多方面的努力。衷心祝愿这个融创新、智慧、汗水于一炉的高水平竞赛越办越好，越办越有活力。