

# 第十七届全国大学生 智能汽车竞赛



## 技 术 报 告

学 校： 东北林业大学

队伍名称： 东林 Ares-PTB-70

参赛队员： 刘 嵩

孙艺迪

张 靓

葛全益

钟涵海

带队教师：刘佳鑫

刘九庆

## 关于技术报告和学术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：

孙艺迪 张凯  
葛益 钟海

带队教师签名：

刘佳鑫 刘九庆

日

期：2022-8-15

# 目录

关于技术报告和学术论文使用授权的说明 .....	II
摘 要 .....	1
第一章 引 言 .....	2
第二章 主要硬件的选择 .....	3
2.1 主控芯片的选择 .....	3
2.2 传感器的选择 .....	3
2.2.1 摄像头 .....	3
2.2.2 编码器 .....	4
2.2.3 陀螺仪 .....	5
第三章 模型车机械安装 .....	6
3.1 传感器的设计安装 .....	6
3.1.1 摄像头支架安装 .....	6
3.1.2 编码器与电机的安装 .....	7
3.1.3 舵机安装 .....	8
3.1.4 电池安装 .....	9
3.1.5 Edgeboard 安装 .....	10
3.1.6 车壳装饰与安装 .....	10
3.2 系统电路板的固定及连接 .....	12
3.3 前桥调节机械结构设计及调整 .....	13
3.3.1 主销后倾 .....	13
3.3.2 主销内倾 .....	14
3.3.3 前轮前束 .....	14

3.4 轮胎处理技巧 .....	15
3.5 后桥的调整 .....	15
<b>第四章 电路设计说明 .....</b>	<b>17</b>
4.1 单片机系统 .....	17
4.2 主控设计方案 .....	18
4.3 电源模块设计 .....	18
4.4 编码器电路 .....	20
4.5 电机驱动电路 .....	20
<b>第五章 视觉处理 .....</b>	<b>22</b>
5.1 图像采集与应用 .....	22
5.1.1 图像的失真与校正 .....	22
5.1.2 图像二值化方案的优化 .....	23
5.1.3 赛道原始边界的获取 .....	23
5.2 传统赛道类型识别与处理 .....	24
5.2.1 直道的识别及处理 .....	24
5.2.2 环岛的识别及处理 .....	24
5.2.3 出入库的识别及处理 .....	25
5.2.4 坡道的识别及处理 .....	25
5.3 特殊元素的识别及处理 .....	25
5.3.1 数据集 .....	25
5.3.2 模型训练&模型导出 .....	27
<b>第六章 控制方案选择 .....</b>	<b>32</b>
6.1 数字 PID 控制 .....	32
6.1.1 位置式 PID（舵机的控制算法） .....	33

6.1.2 增量式 PID（驱动电机的控制算法） .....	33
第七章 开发工具、制作、安装、调试过程说明 .....	35
7.1 开发环境 .....	35
7.1.1 单片机 .....	35
7.1.2 Edgeboard .....	35
7.2 上位机 .....	36
第八章 模型车的主要技术参数说明 .....	37
第九章 结 论 .....	39
参考文献 .....	40
附录 A：核心算法子程序源代码 .....	41
程序源代码： .....	41



## 摘 要

本文主要介绍了基于摄像头的四轮循迹智能小车系统的原理、软硬件设计以及小车制作过程，对小车的系统介绍包括车模机械结构的调整、模块电路的设计、传感器信号的处理、控制算法以及整车调试的方法等。设计以 Edgeboard 和英飞凌公司的 TC264 为控制核心，采用 Edgeboard 获取数字摄像头 wx600v1 采集到的赛道元素信息，然后通过串口发送给单片机进行舵机的响应，增量式编码器获取小车的速度。为了提高小车的稳定性，对摄像头的数据进行二次处理实现小车的加减速行驶与转向。使用 ADS 集成编译环境和开发工具 CLion 编写代码，使用 IPS114 屏幕、拨码开关和按键作为辅助调试手段。经过大量的软硬件测试，实现了智能小车高速下的稳定循迹运动。

关键词：智能车 TC264 PID Edgeboard 视觉处理

# 第一章 引言

全国大学生智能汽车竞赛是以智能汽车为研究对象的创意性科技竞赛,是面向全国大学生的一种具有探索性工程实践活动,是教育部倡导的大学生科技竞赛之一。本竞赛以“立足培养,重在参与,鼓励探索,追求卓越”为指导思想,旨在促进高等学校素质教育,培养大学生创意性科技竞赛鼓励探索,追求卓越”为指导思想,旨在促进高等学校素质教育;培养大学生创意性科技竞赛鼓励探索,追求卓越”为指导思想,旨在促进高等学校素质教育,培养大学生创意性科技竞赛。

本竞赛过程包括理论设计、实际制作、整车调试、现场比赛等环节,要求学生组成团队,协同工作,初步体会一个工程性的研究开发项目从设计到实现的全过程。竞赛融科学性、趣味性和观赏性为一体,是以迅猛发展、前景广阔的汽车电子为背景,涵盖自动控制、模式识别、传感技术、电子、电气、计算机、机械与汽车等多学科专业的创意性比赛。本竞赛规则透明,评价标准客观,坚持公开、公平、公正的原则,保证竞赛向健康、普及、持续的方向发展。

完全模型组完全模型组需要参赛选手必须使用开源深度学习平台飞桨完成模型的训练、推理和部署,有人工智能视觉模型支持车模在赛道上以及赛道下完成相应的运行动作,作为新组别其挑战性更大、趣味性更强对参赛的选手有更大的提高。

本文将分九章对本队制作智能车的情况进行概述,其中第二章主要讲解硬件传感器的选择、第三章介绍车模机械安装方面的内容、第四章主要为硬件电路设计说明、第五第六章讲解软件设计的内容。



## 第二章 主要硬件的选择

### 2.1 主控芯片的选择

完全模型组别今年允许使用的 MCU 种类除了必选的 EdgeBoard FZ3B 以外允许使用的 infineon 微控制器有 TC264、TC212、TC377、TC364、此外还允许使用 infineon 出品的 *Aurix<sup>TM</sup>* 系列的 TC2××和 TC3××其他型号的单片机。对此经过对比考虑到虽然光电组别提取赛道信息较为复杂，运算量较大的，但规则要求视觉模型要部署在百度的 EdgeBoard 上同时兼顾队员芯片熟悉程度、一定程度降低开发难度。再加上对于开发成本的考虑我们最终选择采用 TC264 芯片模块。

### 2.2 传感器的选择

#### 2.2.1 摄像头

比赛所用的摄像头可以分为两类：一类为 CCD 摄像头，另一类为 CMOS 摄像头。两者信息读取方式不同，相较而言，CCD 电荷耦合器存储的电荷信息，需在同步信号控制下一位一位地实施转移后读取，电荷信息转移和读取输出需要有时钟控制电路和三组不同的电源相配合，整个电路较为复杂。而 CMOS 光电传感器经光电转换后直接产生电流（或电压）信号，信号读取十分简单。

且 CCD 电荷耦合器需在同步时钟的控制下，以行为单位一位一位地输出信息，速度较慢；而 CMOS 光电传感器采集光信号的同时就可以取出电信号，还能同时处理各单元的图像信息，速度比 CCD 电荷耦合器快很多。

CCD 电荷耦合器制作技术起步早，技术成熟，采用 PN 结或二氧化硅（SiO<sub>2</sub>）隔离层隔离噪声，成像质量相对 CMOS 光电传感器有一定优势。由于 CMOS 光电传感器集成度高，各光电传感元件、电路之间距离很近，相互之间的光、电、磁干扰较严重，噪声对图像质量影响很大，使 CMOS 光电传感器很长一段时间无法进入实用。近年，随着 CMOS 电路消噪技术的不断发展，为生产高密度优质的 CMOS 图像传感器提供了良好的条件

CCD 摄像头图像对比度高、动态特性好。但同时 CCD 摄像头的耗电也比较严重。CMOS 摄像头，体积小，图像稳定性较高，只需 3.3V 供电，耗电量低，但动态性能不如 CCD 摄像头好。智能车高速运行时，摄像头拍摄的图像可能会变得模糊。但是为了保证系统的稳定性和功能的实现我们最终选择了 CMOS 摄像头。考虑到摄像头选择与配置（帧率 (fps)，分辨率 (hvg)，饱和度 (saturation)，曝光 (exposure)，自动白平衡 (awb)）和镜头选择（光圈 (aperture)，广角 (wide)、焦距 (focal)）需要对特殊元素标识的识别我们选择高帧率全局曝光的彩色摄像头。



图 2.1 摄像头

### 2.2.2 编码器

编码器我们采用了官方的 CSPE5-500 型光电编码器，工作电压 3.3v-5.0v。集成度高、体积小、便于安装到电机尾部与电机组合整体。使用过程中稳定性高，简单方便。但是由于是 500 线的所以精度不高仅能支持简单的电机闭环使用。

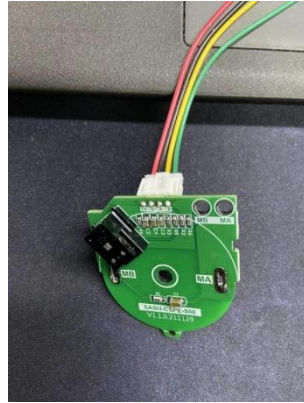


图 2.2 编码器

### 2.2.3 陀螺仪

初期我们尝试了两款 ICM-20602 六轴加速度传感器陀螺仪模块和 IMU963RA 九轴姿态加速度计陀螺仪磁力计传感器模块来解算车体的移动角度，但是由于后期摄像头识别准确率高，舍弃了惯性导航这一方案。

### 第三章 模型车机械安装

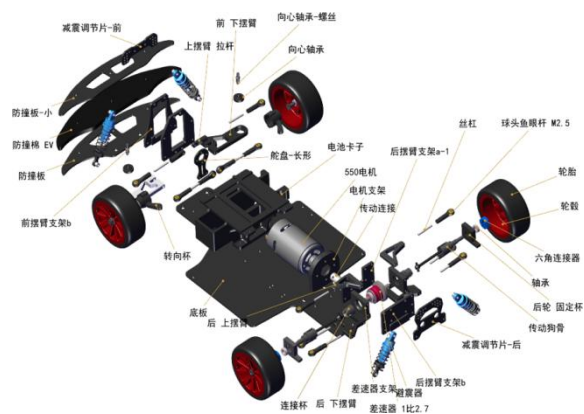


图 3.1 车模组装图

### 3.1 传感器的设计安装

这一部分较多地参考了飞桨官方开源的方法。

### 3.1.1 摄像头支架安装

- 1、先组装摄像头支架，先用圆头固定座螺丝和防滑螺母将碳纤杆固定座与固定支架连接，在将碳纤杆插入碳纤杆固定座中，并用顶丝固定；
- 2、用平头支架螺丝和防滑螺母将底盘与组装好的摄像头支架连接。



图 3.2 摄像头的固定

安装完成后我们也对摄像头的防护做了一些工作。固定了高度和角度，利用热熔胶对其进行了粘合加固。

### 3.1.2 编码器与电机的安装

为了实现电机闭环，提高控制精度，本车使用了 AB 相正交解码编码器，并对编码器传回数据进行了滤波处理。

在拆除完原始车模上的电机后首先需要将焊接编码器电路板和电机。使用电烙铁焊接编码器电路板，需要注意；

1. 焊接时编码器电路板与电机尾部凸起相互贴合。
2. 焊接时编码器电路板中间孔不能与电机轴相互干涉。

然后再是用电烙铁焊接电源线，需要注意

电源线的长度应恰好能与主板相连

焊接的位置与方向要准确

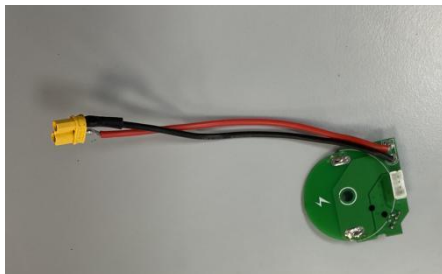


图 3.3 编码器的焊接

按照图示方向安装码盘，使用 0.8mm 内六角螺丝刀安装码盘顶丝。 注意；

1. 安装码盘时，黑色读取器会阻挡码盘，要先将码盘放入读取器槽中再与电机轴安装（码盘安装时可以轻微弯曲）。
2. 安装完码盘之后，码盘不能与读取器有干涉。

再完成以上操作后安装官方的编码器保护壳和信号线。确认无误后安装电机。

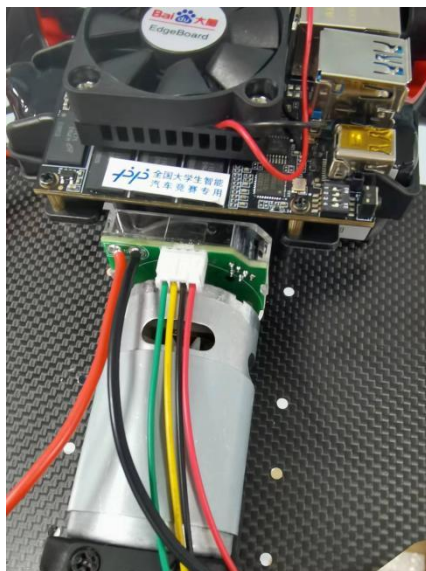


图 3.4 编码器与电机的安装

### 3.1.3 舵机安装

- 1 将前桥拆除，用十字螺丝刀将十字自攻钉从底板上拧下，此时底板与前桥分离；
- 2 如图所示将圆头十字和圆头十字四颗螺丝拆除；
- 3 金属舵盘固定，用 4 颗圆头十字螺丝将长型舵盘与金属舵盘连接；
- 4 将舵机嵌入到前摆臂支架中，注意舵机是处于前摆臂支架 与减震调节片之间的，然后用之前拆下的螺丝固定，再用舵机自带圆头十字 螺丝将金属舵盘与舵机固定(提前确定好舵机中值的大致位置)；
- 5 前桥固定，用十字自攻钉将底板与 前桥连接；

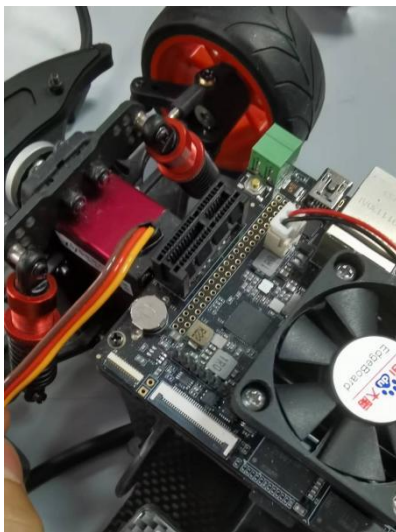


图 3.5 舵机安装

#### 3.1.4 电池安装

将电池嵌入电池卡子中，电池出线方向依据电路板电源插头的位置而放置，在运行过程中我们发现由于惯性电池会被拖拽，所以电池固定圈是十分必要的，依图所示固定上。

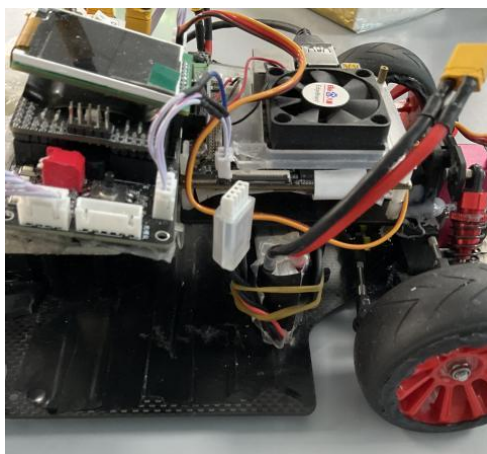


图 3.6 电池的固定

### 3.1.5 Edgeboard 安装

我们利用尼龙柱顺应 Edgeboard 设计的孔位将其固定在电池仓的上方官方附带的 Edgeboard 安装架，并安装好保护壳

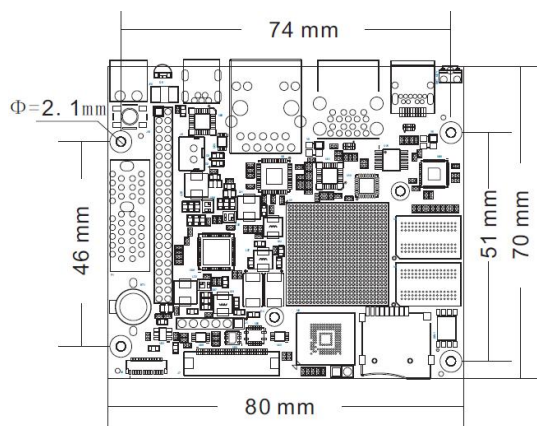


图 3.7 Edgeboard 尺寸

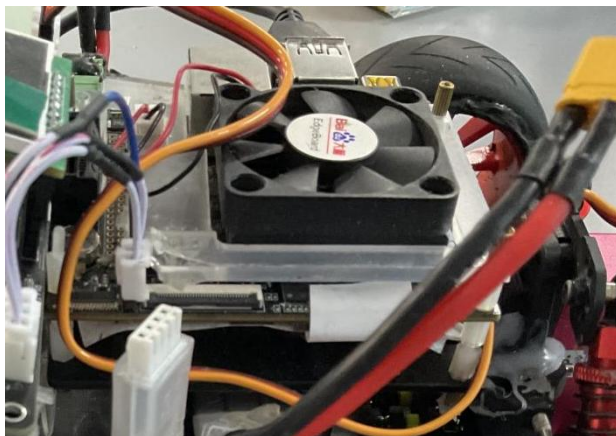


图 3.8 Edgeboard 安装

### 3.1.6 车壳装饰与安装

我们一开始采用的材料由于质量过大导致车壳对车模运行影响大，所以最后我们决定使用几乎没有质量的贴纸对车壳进行装饰。



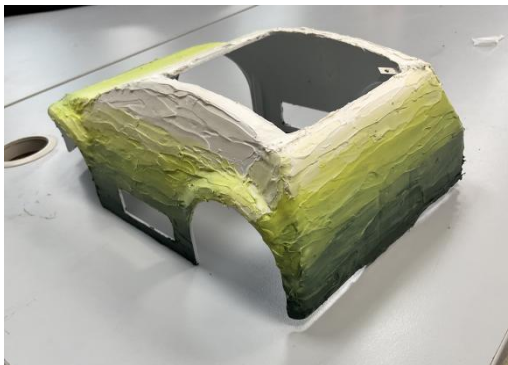


图 3.9 质量过大的车壳

用平头支架螺丝将底盘和车壳连接。



图 3.1.0 车壳与低盘的连接孔位

但是由于我们为了调车的便利，利用了按键和屏幕改变一些参数，所以我们为了能够快速调整按键、Edgeboard 接口方便插接、避免重复拆卸车壳等对车壳进行了一些切割，将车壳部分做成可活动的部分。如图所示。



图 3.1.1 车壳侧视图与俯视图

### 3.2 系统电路板的固定及连接

I 车模由于电机及编码器都在车尾部，重心偏后，所以我们在不影响舵机连杆运动的情况下将电路板尽量靠前安装，将重心保持在中间偏后。同时我们利用 solidworks 画了打印件来固定母板。EdgeBoard 和驱动则合理排布并与母版对应连接。

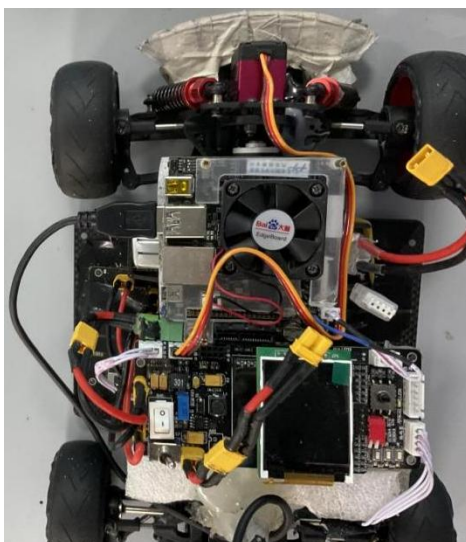


图 3.1.2 车模电路板大致排

### 3.3 前桥调节机械结构设计及调整

I 车模四轮独立悬挂参数可调。要改变智能车与地面的接触方式，调试出利于车转向、直线的四轮定位，可以通过调整前轮倾角各定位参数来实现。它的安装位置由主销内倾、主销后倾、前轮外倾和前轮前束四个项目决定。

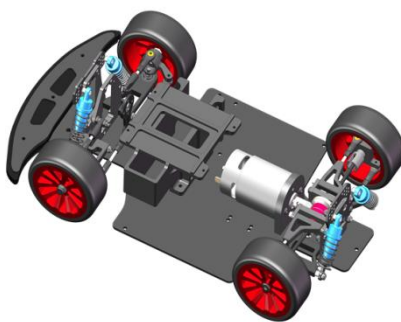


图 3.1.3 车模 3D 建模

#### 3.3.1 主销后倾

主销后倾是指在纵向平面内主销轴线与地面垂直线之间的夹角，向前为负，向后为正。它在车辆转弯时会产生与车轮偏转方向相反的回正力矩，使车轮自动恢复到原来的中间位置上。主销后倾角对保持直线稳定和转弯后的前轮自动回正的能力的加强有显著效果。但过大直线运行更稳定的同时也会增加转向的阻力，指定使用的舵机性能限制我们仅做了  $1^{\circ} \sim 2^{\circ}$  的修改。



图 3.1.4 主销后倾

### 3.3.2 主销内倾

主销内倾是将主销的上端向内倾斜。从车前方看去，主销轴线与通过前轮中心的垂线之间形成一个夹角，即主销内倾角。轮胎调整为倾斜以后直线行走的时候是轮胎内侧着地，而当过弯的时候，由于惯性车体会要向弯道外侧倾斜，而这时候的外侧轮胎如果倾斜角度事先调整得当则正好可以胎面着地，从而使车辆在弯道获得最佳抓地力。使车轮转向后能及时自动回正和转向轻便。即不处理则会使得直线轮胎接触面积大，处理后弯道轮胎接触面积大。

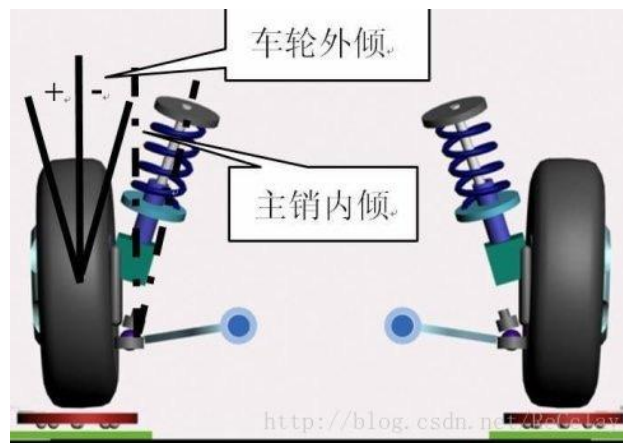


图 3.1.5 主销内倾

### 3.3.3 前轮前束

车轮前束是指两轮之间的后距离数值与前距离数值之差，也指前轮中心线与纵向中心线的夹角。从上往下看，两个车轮指向的方向在前端指向内称为车轮前束，指向外的则称为车轮后束。前轮前束的作用是保证汽车的行驶性能，减少轮胎的磨损。由于前轮外倾使轮子滚动类似圆锥滚动，导致车轮向外的趋势，但由于舵机拉杆的存在，车轮不可能向外滚开，则增加了轮胎外侧的磨损。前轮前束可以抵消车轮外倾的副作用。而后束则可以使得过弯道更加灵活。对于这一处理方式我们将前轮前束的值控制在  $0 \sim 10\text{mm}$ 。

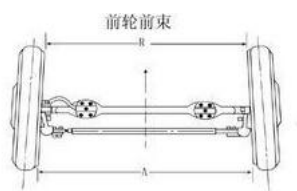


图 3.1.6 前轮前束

### 3.4 轮胎处理技巧

车模运行速度不断提高后，轮胎的抓地力的重要性逐渐提高。由于只能使用指定官方的 I 模型车轮胎 CICR-ITY 橡胶轮胎，我们需要对轮胎进行一些处理才能获得更好的抓地力。

我们将轮胎简易安装在轮毂上，固定在台钻上用砂纸磨去一部分花纹。砂纸应该从低到高逐渐过渡，打磨过程中应不时注意花纹深浅程度和轮胎形变。打磨完成后在轮毂上缠上一些宽度厚度合适的海绵胶，然后再在轮胎和轮毂之间垫官方提供的海绵。最后使用硅橡胶密封轮胎和轮毂的接缝处。



图 3.1.7 未经处理的轮胎与处理过后的轮胎

### 3.5 后桥的调整

### 3.5.1 差速器

I 车模齿轮差速器和主减速器为一体式结构，全金属材质，结构紧凑，坚固耐用。由于采用的单电机后驱的方式，所以仅能通过机械差速的方式改变两个后轮的速度。因为速度没有达到转弯对差速要求高的程度，所以我们对后轮的差速影响调节得较小，即转弯仅有轻微两轮差速。



图 3.1.8 差速器

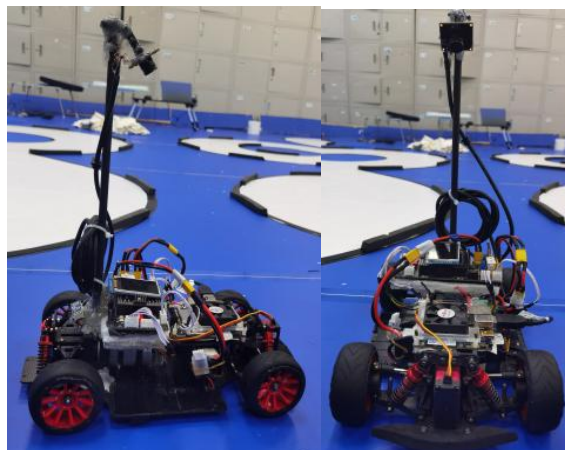


图 3.1.9 整体外观

## 第四章 电路设计说明

对于硬件电路部分，一稳定、简洁、可靠的硬件是车模正常运行的前提，抗干扰性能一定要高。单片机供电电压一定要稳定，硬件电路是智能车的基础，只有打好基础才能继续软件方面的工作，保证系统的正常运行，经过比赛之后理解到，良好的硬件基础直接决定智能车运行的基础速度和上限速度。在尝试使用多种硬件电路模块后，最终确定了一套可靠、高效的电路系统，并可满足强劲的性能需求。我们设计过程中严格按照规范进行。本着可靠、高效的原则，在满足各个要求的情况下，尽量使设计的电路简单。尽量减少元器件使用数量，缩小电路板面积，可使车体正常安装车模，减轻整车重量，降低车模重心。

## 4.1 单片机系统

单片机最小系统为智能车系统核心控制部件，我们选用了逐飞科技制作的 TC264 核心板进行开发使用。

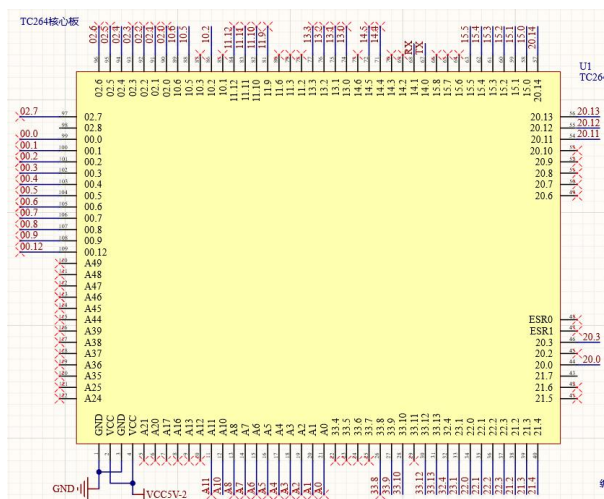


图 4.1 芯片原理图



## 4.2 主控设计方案

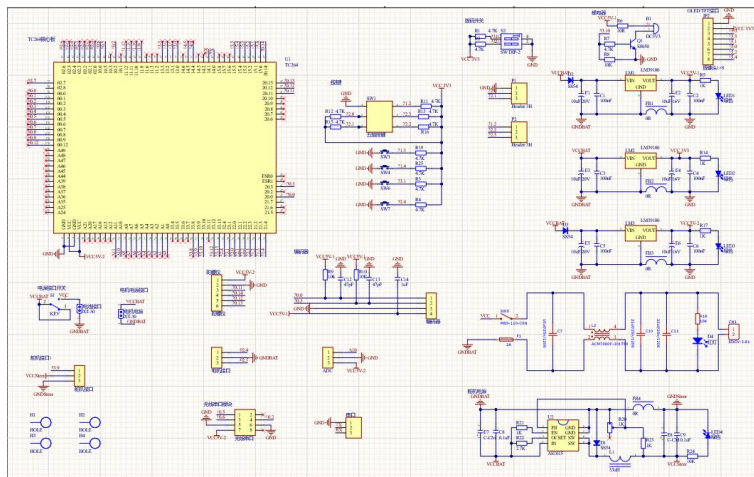


图 4.2 主控设计方案

## 4.3 电源模块设计

电源模块为系统其他各个模块提供所需要的电源。本赛项使用电池电压较高，设计中需要重点考虑电压范围和电流容量还要在电源转换效率、降低噪声、防止干扰和电路简单等方面进行优化。

全部硬件电路的电源由锂聚合物电池（格式电池）提供（额定电压 11.1V，满电电压 12.6V）。

由于电路中的不同电路模块所需要的工作电压和电流容量各不相同，因此电源模块应该包含多个稳压电路，将充电电池电压转换成各个模块所需要的电压。为满足需要，本电路板设置 4 种不同电压：

（1）智能车使用锂电池供电正常使用时电压在 11.1~12.6V，直接用于电机和算力板供电。保险丝防电流过大，共模滤波器抑制差分信号电路的共模噪声。为方便调试，将总开关与算力板供电开关采用电路并联，互不干扰。



(2)

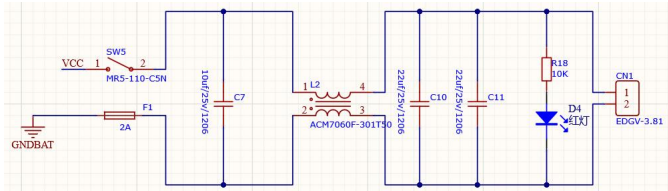


图 4.3 电源模块原理图

(3) 转压模块利用 LM39100 转压芯片（耐压-0.3V 到 20V），设置两路均输出 5V 电压。参考芯片手册绘制此芯片所需电路图，设立两路地信号，经器件消耗的接入 GND，未经器件消耗的接入 GND BAT，多加 0  $\Omega$  电阻隔离模拟地信号和数字地信号，防干扰，分析每路电流情况分别为核心板，编码器，蜂鸣器等模块供电。

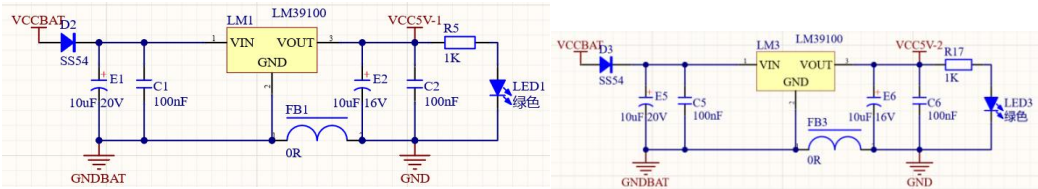


图 4.4 转压模块原理图

(3) 使用稳压芯片 LM39100 输出电压 3.3V，主要为按键部分，显示屏供电。

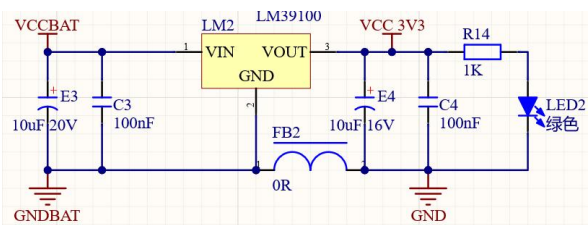


图 4.5 稳压模块原理图

(4) 使用 AS1015 芯片相关电路为舵机供电，受限于本组舵机及车体结构，为提高舵机打脚速度，将电压升至舵机最大耐受边缘。(约为 6.5v-7.0v)

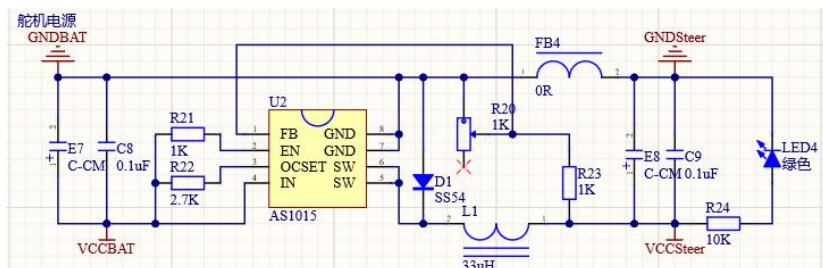


图 4.6 舵机电压原理图

#### 4.4 编码器电路

编码器模块采用官方提供方案，电容接地滤波，电阻分压接入：

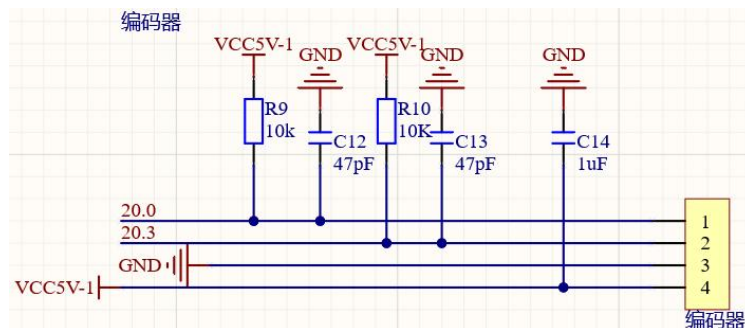


图 4.7 编码器电路原理图

#### 4.5 电机驱动电路

IRLR7843 是用于电机驱动应用的集成大电流半桥。它包含一个 p 沟道高侧 MOSFET 和一个 n 沟道低侧 MOSFET，其中一个封装有集成驱动芯片。由于 p 通道高压侧开关，因此无需电荷泵，从而将电磁干扰降至最低。集成驱动芯片

易于与微控制器接口，该芯片具有逻辑电平输入、电流检测诊断、转换率调整、死区时间生成和过热、过压、欠压、过流和短路保护功能。LR7843 提供了一个成本优化的解决方案，用于具有非常低的板空间消耗的受保护的大电流 PWM 电机驱动器。在 IRLR7843 前级，增加了 74HC244 缓冲电路，来防止电机电流反冲烧坏。

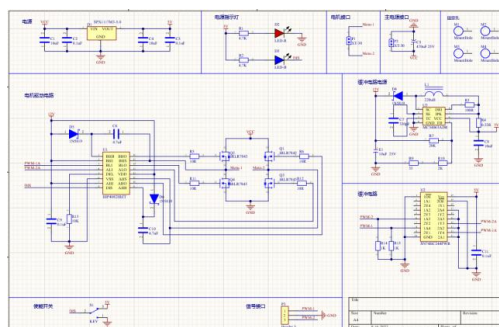


图 4.8 驱动电路原理图

4.6 显示屏采用 1.8 寸液晶屏, 3.3V 供电。

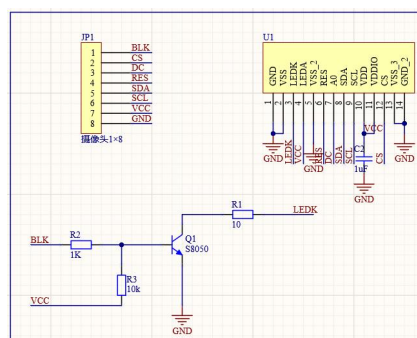


图 4.9 显示屏原理图

## 第五章 视觉处理

良好的硬件可以支持各个模块的实现，良好的软件则可以支持模块实现的稳定性。由于完全模型组比赛要求的特殊性，我们选用了彩色摄像头来获取赛道信息。我们对比了传统巡线和深度学习寻线的效果，最后决定采用传统循线的方法

### 5.1 图像采集与应用

我们有用 OpenCV 对图像进行了一些处理，例如图像的标定与矫正、图像的预处理（去噪、压缩等）、赛道搜索等等。普通赛道的循迹是采用的二值化图像类似于摄像头四轮循迹方法，灰度特征明显，赛道元素分布明显，利用二值化，边缘提取等方法确定赛道元素。通过边界数组求得中线的结果。我们是根据赛道中线进行循迹，所以中线的处理也非常重要。一般情况是用中线=（左边线+右边线）/2 来进行计算。

使用 OpenCV 在检测到红色阈值时再开始识别各种元素的标识。

#### 5.1.1 图像的失真与校正

由于摄像头针对赛道平面所采集到的图像是一幅符合透视原理的梯形图像，图像会产生桶形失真，这使得摄像头看到的信息与真实赛道信息有所差异，我们在尝试了逆透视算法，将获取的图像进行了矫正。

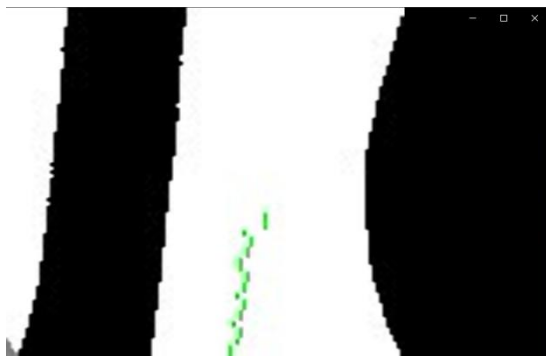


图 5.1 二值化图像

### 5.1.2 图像二值化方案的优化

摄像头直接获取到的是灰度图像，也即每个像素点的取值范围在 $[0, 255]$ ，直接利用该信息具有一定的处理难度，也会延长程序的调试周期，因此我们决定先使用大津法对图像进行二值化，再对二值图像做后续处理。

### 5.1.3 赛道原始边界的获取

考虑到当车体位于赛道内时图像相对单调，综合各算法的复杂程度与应用范围，我们决定采用简单的边缘追踪算法——八邻域边缘追踪算法。

## 5.2 传统赛道类型识别与处理

根据提取到的赛道中线，可将非任务区赛道分为以下几个元素，并根据赛道元素进行相应的控制。

### 5.2.1 直道的识别及处理

直道的左右边界对称度高，赛道宽度稳定。对于赛道中的直道，车速可以适当提高，因为只要小车基本不会偏离赛道，就可以不进行大幅度打角，让车模能够回到赛道中线即可。

### 5.2.2 环岛的识别及处理

环岛的特征比较明显，一侧为直道，一侧有圆形区域。我们采用传统的识别方法并进行改良。当一侧为直道且一侧丢线后进入预识别，当丢线一侧有圆环后检测为环岛，然后检测换到上方的拐点后进行补线。详细步骤如下：

- ①检测到一侧为直道，另一侧丢线，预判定圆环方向
- ②检测到丢线侧有线且赛道宽度逐渐递减
- ③找环岛上拐点补线
- ④检测到一侧丢线（右环岛右丢线，左环岛左丢线）
- ⑤补线出环



图 5.2 环岛图

### 5.2.3 出入库的识别及处理

因为完全模型组的赛道具有单向性，所以出库可以提前设定方向，固定打角补线出库。入库方向与出库方向一致，由黑白跳变点识别斑马线和库尖角，据此补线入库，然后电机停转使车模停在车库内

### 5.2.4 坡道的识别及处理

由于坡道设定在泛行区的一侧，执行泛行区任务后是不会经过坡道的，对于这一元素不进行识别和处理。

## 5.3 特殊元素的识别及处理

### 5.3.1 数据集

完全模型组除了普通循迹以外还有其特殊的元素：泛行区、施工区、加油站.....对于这部分元素的处理我们采用深度学习的方法。图像的含义不在图像本身而在于如何去观测。电脑对图像是毫无概念的，需要人的先验知识去打上标签。数据集的采集就显得尤为重要，我们对同一标识静态采集大于 400 张图片动态采集 200 张图片作为数据源。

在采集数据集的难点：

1、小车运行过程中会产生动态模糊，当速度到一定的程度后，动态模糊的影响较大。因此在采集数据集时，应单独对动态模糊图像进行采集。

2、在采集数据集时尽可能的复现比赛场景，即标识所处的背景应贴合比赛场景。同时施工区和三叉标识容易误判，但是结合赛道的一些特征后便可区分。

对于第一点对应的数据增强也可解决一些问题。

然后进行数据标定,我们使用 Labelimg 制作数据集。Labelimg 是一款开源的数据标注工具，可以标注三种格式。

1. PascalVOC 标签格式，保存为 xml 文件
2. YOLO 标签格式，保存为 txt 文件
3. CreateML 标签格式，保存为 json 文件

数据集的标注步骤如下：

1. 将采集到的照片放在一个文件夹内然后打开 labelimg,

## 第十七届节全国大学生智能汽车邀请赛技术报告

2. 点击打开目录，找到需要标注的数据集文件夹，点击选择文件夹。
3. 这时文件夹的图片就会显示中间界面，文件夹下的所有图片会显示在右下角
4. 点击改变存放目录，我们可以选择文件夹作为标记后图像的输出文件夹，用以存放标记好的图片
5. 将键盘输入法切换为英文，按“W”键，labelimg 中会出现下图中的十字线
6. 按住鼠标左键将小狗纳入方格区域内，注意方格区域要尽可能的小
7. 松开鼠标，在文本框中输入目标名称，例如 Construction\_area，点击 ok
8. 依次点击保存，下一个图像
9. 点击下一个图像后，我们可以按照上面的步骤继续对数据集图片进行标注

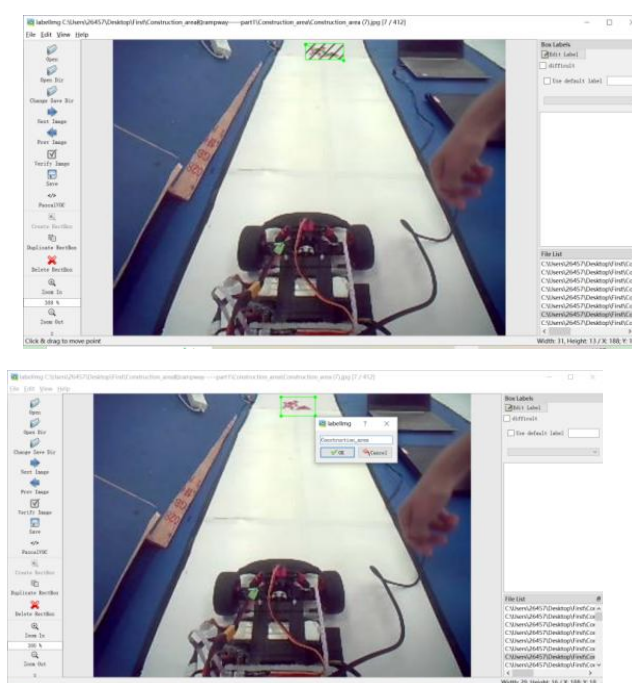


图 5.3 数据集采集



```
<?xml version="1.0"?>
<annotation>
  <folder>Construction_area</folder>
  <filename>WIN_20220612_13_36_20_Pro.jpg</filename>
  <path>C:\Users\26457\Desktop\out_png\Construction_area\WIN_20220612_13_36_20_Pro.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>320</width>
    <height>240</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>Construction_area</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>158</xmin>
      <ymin>5</ymin>
      <xmax>186</xmax>
      <ymax>15</ymax>
    </bndbox>
  </object>
</annotation>
```

图 5.4 xml 文件内容

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物围成的临时路段。	
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。	

图 5.5 特殊元素标识

5.3.2 模型训练&模型导出

将数据集打包至 aistudio，只需要将打包来的图像放入单独一个文件夹，将生成的 xml 文件放入另一个文件夹中，对于模型的训练。我们开始时采用是官方的模型进行训练，但是效果并不理想。

## 第十七届节全国大学生智能汽车邀请赛技术报告



图 5.6 官方训练模型

模型选取：

MobileNet 系列：

由于 Edgeboard 的计算资源十分的有限，在使用 yolov3，以原生的 Darknet 做主干特征提取网络时在 Edgeboard 的帧率仅仅只能到 3FPS，这在本次比赛中是完全不够用的。因此我们将目光转向轻量级神经网络 Mobilenet，虽然 PaddDetection 0.5 版本也支持 GhostNet，但 GhostNet 在数据表现上不如 Mobilenetv3

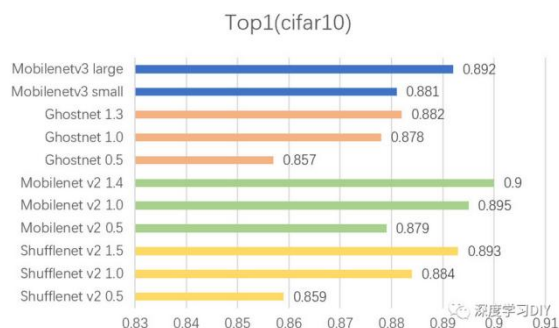


图 5.7 cifar

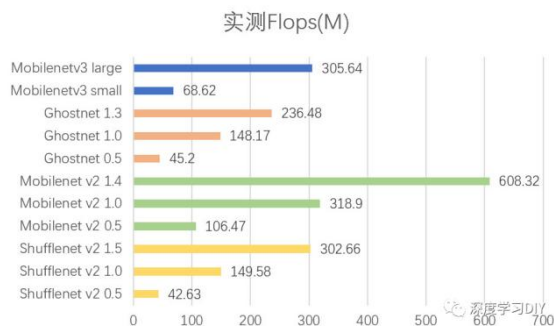


图 5.8 浮点运算量对比

Flops 即浮点运算量,可以一般化为运算速度,可以很明显看到 Ghostnet1.0 在具有更大运算量的同时仍比 mobilenetv3\_small 的精度要小。因此我们选择使用 Mobilenet 系列, Mobilenet 系列采用分组卷积和逐点卷积的方式去替代了传统卷积, 以此达到减少参数数量的效果。



图 5.9 卷积方法对比

我们开始选择了 Mobilenetv3。经过一段时间的调试后,发现对于边缘计算设备, Mobilenetv1 的表现效果要远远好于 Mobilenetv3。在实验 SSD\_MobileNetv1 和 SSD\_MobileNetv3\_small\_FPn 两个部署情况对比:前者运算速度仅仅需要 20-22ms,而后者的运算则需要 40-60ms 之间。

Yolov3 目标检测原理是以 anchor 的形式去不断微调框框,最后再做 nms 非极大值移植。因此我们需要对数据集做相应的 anchor size 的调整

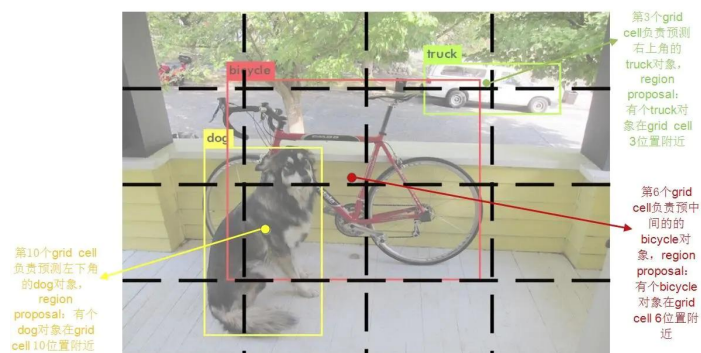


图 5.1.0 Yolov3 目标检测

NMS 的关键:

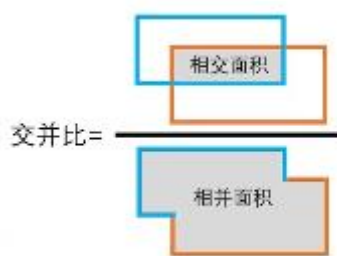


图 5.1.1 交并比

如果采取Yolov3\_MobileNetv1 结构训练,首先是对数据集 Anchor 做 Kmeans 聚类算法。生成的 anchor 框,按照顺序修改 yml 文件中的 anchor size。然后配置 yml 文件进行训练。已随项目的 PaddleDetection 放于 yolov3\_mobilenetv1\_eb.yml 中。最后是模型训练然后模型导出。

SSD 目标检测原理如下:

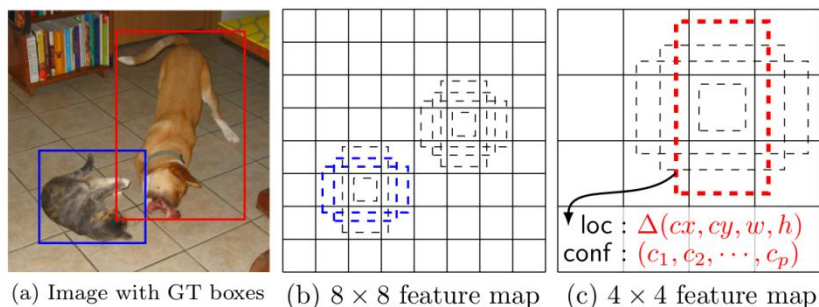


图 5.1.2 SSD 目标检测原理

SSD 相较于 yolo 则是基于每个像素点去做锚框，每一个像素点都会生成 anchor 去看是否存在物体。因此其也是一个 anchor based 的结构。我们在做自己数据集的时候，要注意修改 anchor size 的大小：即 `max_sizes: [[], 150.0, 195.0, 240.0, 285.0, 300.0]` `min_ratio: 20` `min_sizes: [60.0, 105.0, 150.0, 195.0, 240.0, 285.0]` 这几个参数 因此在设置 yml 文件的 `num_classes` 时要类别+1（背景类）

如果使用 SSD 模型训练，太轻量的模型：精度低，容易误判。

需要注意的一点是，在使用 `paddleslim 1.1.1` 等相关功能时，PARL 的版本需要 `paddle` 的版本在 1.8.5 以上，因此我们需要自己 `pip install` 到相应的版本

模型部署后可能会遇到一些问题，例如加上模型之后延迟变高。我们也遇到了这个问题。这可能是是由于模型太大或是使用的尺寸太大或是代码逻辑太过于冗余。对于这个问题可以通过使用轻量级目标检测算法，但是由于我们经过实践操作发现轻量级误判高，所以最终我们最终选择在逻辑结构和尺寸上进行了改进。

## 第六章 控制方案选择

### 6.1 数字 PID 控制

我们使用位置式 PD 算法控制舵机打脚，使用增量式 PI 算法控制电机稳速。

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 (Proportion)、积分 (Integral) 和微分 (Derivative) 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，原理框图如图所示<sup>[1]</sup>。

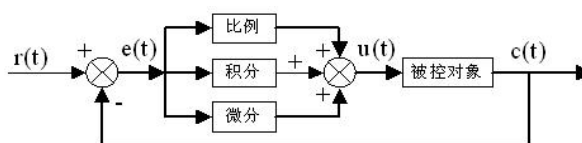


图 6.1 PID 控制器原理框图

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (\text{公式 1})$$

其中  $e(t)$  为被控量与给定值的偏差。 $K_p$  为比例系数， $T_i$  为积分时间常数， $T_d$  为微分时间常数， $T$  为采样周期。

简单说来，PID 控制器各校正环节的作用如下：

比例环节 Proportion（比例）：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 Integral（积分）：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 Derivative（微分）：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。

### 6.1.1 位置式 PID（舵机的控制算法）

$$u_k = K_p * e_k + K_i \sum_{j=0}^k e_j + K_d (e_k - e_{k-1}) \quad (\text{公式 2})$$

其中：

$$K_p \rightarrow P \quad K_i \rightarrow I \quad K_d \rightarrow D \quad e \text{数组} \rightarrow \text{error 数组}$$

$$e_k \rightarrow \text{本次期望与实际差值} \quad e_{k-1} \rightarrow \text{上次期望与实际差值}$$

位置式 PID 中，由于计算机输出的  $u(k)$  直接去控制执行机构， $u(k)$  的值和执行机构的位置是一一对应的，所以通常称公式 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。

对于舵机的闭环控制，我们采用了位置式 PID 控制算法。首先我们将积分项系数置零，我们发现相比稳定性和精确，基本没有必要使用  $K_i$  参数。因为  $K_i$  系数的功能是消除稳态误差，但由于舵机基本没有稳态误差所以这一系数我们最终将其置 0。

经不断调试，我们最终采用了动态  $K_p$ ，静态微分项系数  $K_D$ ，使得小车在直道上几乎没有震荡，也得到了较为理想的转向控制效果。

### 6.1.2 增量式 PID（驱动电机的控制算法）

当执行机构需要的是控制量的增量时，可推导出提供增量的 PID 控制算式。

$$\Delta u[n] = K_p \{e[n] - e[n-1]\} + K_i e[n] + K_d \{e[n] - 2e[n-1] + e[n-2]\} \quad (\text{公式 3})$$

其中

$K_p \rightarrow P$        $K_i \rightarrow I$        $K_d \rightarrow D$        $e$ 数组  $\rightarrow error$ 数组

$e[n] \rightarrow$  本次期望与实际差值

$e[n-1] \rightarrow$  上次期望与实际差值

$e[n-2] \rightarrow$  上上次期望与实际差值

由于编码器精度影响，测得速度会有一定变化，在实际运行中也有较明显的波动，我们选取了简单的滤波方法对数据进行了一些处理。经过仿真与测试，对于速度控制，我们最终采用了PI控制算法，同时对于速度分配基本思想是直道加速，弯道减速。在实际测试中，我们发现小车直道和弯道相互过渡时加减速比较灵敏，与舵机转向控制配合得较好。



## 第七章 开发工具、制作、安装、调试过程说明

### 7.1 开发环境

在对程序进行开发和软硬件联调的过程中需要一整套的软件开发与调试工具。

#### 7.1.1 单片机

单片机程序的开发是在 IAR Embedded Workbench 下进行的，包括源程序的编写、编译和链接，并最终生成可执行文件。包括集成开发环境 IDE、处理器专家库、全芯片显示工具、项目工程管理器、C 交叉编译器、汇编器、链接器以及调试器。使用 英飞凌 TriCore TC264 DAP miniWiggler 下载器来下载程序，把编译好的程序下载到单片机里运行。这款下载器体积小易携带，不掉固件。

#### 7.1.2 Edgeboard

部署 Edgeboard 首先要求使用 Edgeboard 之前确保网线与电脑连接，并设置局域网静态 ip: 192.168.1.111 网关: 192.168.1.1。

对于 Edgeboard 我们可以采用以下几种方法进行部署：

##### (1) 利用 VsCode 远程连接修改程序

由于 FinalShell 的文件管理没有关键词检测以及缩进对比，因此修改代码以及维护十分困难。因此我们选择使用 Vscode 进行远程连接进行代码管理及修改

##### (2) SSH 工具 Putty/Xshell

##### (3) C++ 推理 makefile---cmake

CMake 主要有两大功能：

①在配置和生成各大平台的工程（vs 的 vcxproj，Qt 的 Pro）：比如设置输出目录，设置编译对象的 debug 后缀，设置源码在工程中的那个文件夹（Filter），配置需要依赖的第三方的头文件目录，库目录等等属性

②生成 makefile 文件计算机编译源文件的时候是一条指令一条指令的发送给编译器执行的，这样效率很低下，所以就产生了一种文件，把所有的命令写到一个文件中，这个文件就是 makefile。CMake 生成了这个 makeFile

之后，各大平台的编译器都会拿到这个 makeFile 然后解析它。将他的命令解析出来一条一条执行。

由于 Edgeboard 是一个四核的处理器，支持我们开多个线程，但是同时处理的线程只有 4 个。因此我们也利用的这个特性去做了程序优化。

## 7.2 上位机

采用匿名上位机软件获取波形图，通过不断调整参数和调试波形，我们得到了适合的当前小车的电机闭环参数，让电机响应快而且速度稳定。如图。

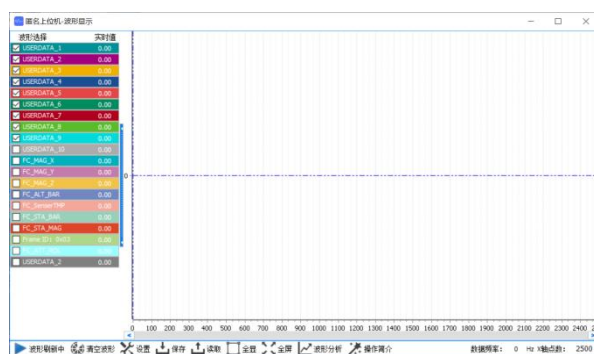


图 7.1 上位机图像

## 第八章 模型车的主要技术参数说明

表 8.1 模型车主要技术参数

主要技术参数说明项目	参数
车模类型	I 车模
车模整体尺寸	长：31.6cm 宽：19cm 高：36cm(含摄像头碳纤杆)
传感器种类、规格(型号)数量	编码器： CSPES-500 型光电编码器
舵机型号	CS-3120 数字舵机
微处理器型号	TC264 一个
其它可编程器件	Edgeboard 一个
电池的种类、规格	可充电锂离子电池 11.1V 2200mV



## 第九章 结 论

全国大学生智能汽车竞赛涵盖了控制、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。这次经历是十分宝贵的

在此份技术报告中，我们主要介绍了我们制作车模的大致思路，包括机械，电路，以及视觉和控制算法的创新思想。

最后在这里感谢飞桨官方开源的资料帮助我们进行开发。感谢学校提供场地与各方面的支持。

我们相信通过这段时间的辛勤努力和付出，我们一定能在此次比赛中取得满意的成绩！

## 参考文献

- [1] 赵鹏, 钱美, 李帅波. 单片机 PID 电机转速与位置控制系统设计[J]. 工业仪表与自动化装置, 2022(04):24-27+44. DOI:10.19950/j.cnki.cn61-1121/th.2022.04.005.

## 附录 A：核心算法子程序源代码

## 程序源代码：

```
void PID_Increase_motor(int16 target,int16 now)
{
    motor_pid_lastError=motor_pid_Error;

    motor_pid_Error=target-now;

    motor_Increase +=(kp_m * (motor_pid_Error - motor_pid_lastError)+ ki_m*
motor_pid_Error);

    motor_Increase =range_protect(motor_Increase,-499,499);
}

void PID_Place_Servo(int16 target,int16 now)
{
    kp_s=kp_s1+kp_s2*(ROW-valid_line)/valid_line_param;

    servo_pid_lastError=servo_pid_Error;

    servo_pid_Error=target-now;

    servo_position= kp_s*servo_pid_Error +
kd_s*(servo_pid_Error-servo_pid_lastError);

    servo_position=range_protect(servo_position,-Servo_Limit,Servo_Limit);
}
```

```
void uart_getdata_if(void)

{

    int i=0;

    message_type=0;

    if(uart_query(UART_1, &getdata))

    {

        DataBox[0]= DataBox[1]; //帧头

        DataBox[1]= DataBox[2]; //error

        DataBox[2]= DataBox[3]; //error

        DataBox[3]= DataBox[4]; //类型*

        DataBox[4]= DataBox[5]; //有效行

        DataBox[5]= DataBox[6]; //元素

        DataBox[6]= getdata;    //帧尾

    }

    if( DataBox[0]==0x17 && DataBox[6]==0x16)

    {

        message1=DataBox[1];
```



```
message1=message1<<8;

message2=DataBox[2];

Getmessage=message1+message2;

valid_line = DataBox[4];

message_type=DataBox[3];

for(i=0;i<7;i++) {DataBox[i]=0;}

getdata_deal();

}

}
```

## 第一届全国大学生智能汽车邀请赛技术报告