第十七届全国大学生

智能汽车竞赛

技 术 报 告

学　　校：　长沙理工大学

队伍名称：　华南第一反卷联盟

参赛队员：　蔡明达 杜均益 邓思露

朱海权 罗锦程

带队教师：　徐晓强　刘理

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：蔡明达

杜均益

邓思露

朱海权

罗锦程

带队教师签名：徐晓强

刘理

日　期：2022 年 8 月 13 日

摘要

本文详细介绍了长沙理工大学“华南第一反卷联盟队”在第十七届全国大学生智能汽车竞赛完全模型组中的系统方案。本次比赛采用大赛组委会统一指定的 I 型车模，以百度生产的 Edgeboard 作为主控制器进行赛道数据采集及图像处理，以 TC264 作为下位机控制器，通过串口来实现上下位机的通信，将上位机（Edgeboard）提取出的赛道信息实时传送给下位机进行赛车控制。比赛要求智能车识别白色赛道和车库坡道等多个元素，除此之外还要对 AI 元素作出相应处理。以最快速度完成整个比赛，智能车采用 CMOS 摄像头对赛道信息检测，根据每行的跳变点搜索中线；使用 PID 控制算法调节电机的转速和舵机的打角，实现智能车在运动过程中速度和方向的闭环控制；实验结果表明，该系统设计方案确实可行。

关键词：Edgeboard；TC264；图像处理；PID

目录

第一章 绪 论

1.1 智能车研究背景

本研究设计是为了参加第十七届全国大学生智能汽车竞赛而进行。全国大学生智能汽车大赛是受教育部高等教育司委托，由高等学校自动化专业教学指导委员会负责主办全国大学生智能车竞赛。该项比赛已列入教育部主办的全国五大竞赛之一。自第一届来，此项大赛已成为各大高校的热点赛事，比赛涉及控制、传感技术电子、计算机、机械等多个学科的专业知识，对学生的知识融合和动手能力的培养，对高等学校控制及汽车电子学科学术水平的提高，具有良好的推动作用。这一届大赛分为四轮电磁组、四轮摄像头组、平衡信标组、多车编队组和完全模型组等，本文介绍的是针对完全模型组比赛进行的智能车设计，采用委会统一提供的竞赛 I 车模，选用 Edgeboard 作为核心控制单元，选用摄像头传感器作为赛道检测的方式，并针对本届比赛规则，设计传感器检测方案、方向控制及速度控制算法等，最终实现一套能够自主识别路线的智能车控制硬件、软件系统。比赛向来以准确和速度为主要评判标准，因此，设计主要以准确稳定提取赛道信息以及快速控制方向和速度为主要设计标准。相对传统组别，今年新出的完全模型组还有特定的任务要完成，赛道元素更多样化，比如说泛行区、施工区和加油站的设计，在完成竞速目的的同时还需考虑准确完成任务。

1.2 方案总体介绍

系统是以检测特殊标志位和黑白赛道为基础，通过 Edgeboard 处理信号实现对车体控制，实现车体能够准确沿着预设路径寻迹。系统电路部分需要包括 Edgeboard 和单片机控制单元、电机驱动电路等部分，除此之外系统还需要一些外部设备，例如编码器测速、直流电机驱动车体、舵机控制转向打角等。根据以上系统方案设计，赛车共包括七大模块： 主控模块、传感器模块、电源模块、电机驱动模块、速度检测模块及辅助调试模块。

各模块的作用如下：

1．主控模块：作为整个智能车的控制中心，将采集特殊标志位信息、摄像头

信息、编码器等传感器的信号，根据控制算法做出控制决策，下发给下位机，完成对驱动直流电机和舵机的控制。

2．传感器模块：作为智能车获取赛道信息的关键，可以通过一定的前瞻性，提前感知前方的赛道信息，为智能车的控制中心做出决策提供必要的依据和充足的反应时间。

3．电源模块：将电池电源转化为合适的不同电压以驱动不同的模块。

4．电机驱动模块：驱动直流电机完成智能车的加减速控制和转向控制。

5．速度检测模块：检测反馈智能车后轮的转速，用于速度的闭环控制。

6．辅助调试模块：主要用于智能车系统的功能调试、赛车状态监控等方面。

1.3 本文结构

本文共分为六章。第一章主要是介绍了比赛的背景及智能车系统总体方案的介绍；第二章简要叙述了智能车系统主要部分机械结构的安装和调整；第三章重点介绍了系统中所涉及的硬件设计方案和原理；第四章是介绍了智能系统的软件算法包括图像处理以及电机舵机的控制策略；第五章对调试过程中的一些手段进行了讲解。

第二章 机械结构设计及调整

智能车所有的硬件、软件程序算法最终都是建立在机械结构的基础上来实现竞速的，其机械结构的设计和调整同样是智能车制作中一个不容忽视的环节。除了对车身姿态的影响外，机械性能影响车的加减速响应速度，运行的对称性和稳定性等。本章将主要叙述智能车系统主要部分机械结构的安装和调整。

2.1 舵机安装

舵机安装直接关系到是否能快速灵敏地转向，通过改变舵机的安装位置可以提高舵机的响应速度。通过分析舵机控制转向轮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。

舵机安装方式有立式和卧式两种，而我们采用的是立式安装。一方面舵机安装时要保证左右对称，这样可以保证舵机左右转向时力臂相等且处于最大范围，提升了舵机的响应效率和准确度。经理论分析，功率等于速度与扭矩的乘积，加大转向速度必然减少输出扭矩，扭矩过小会造成迟钝。立式安装把舵机架高，虽然增大了阻力，使力的作用减小，但增长了力臂，使得小车反应更加灵活。所以安装时必须考虑到舵机的响应速度与舵机扭矩之间的关系，获得最佳转向效果。

CS-3120 为我们组的舵机型号，具有较快的反应速度和大扭矩的优点，配合阿克曼转向能够更好的实现小车的实时控制。

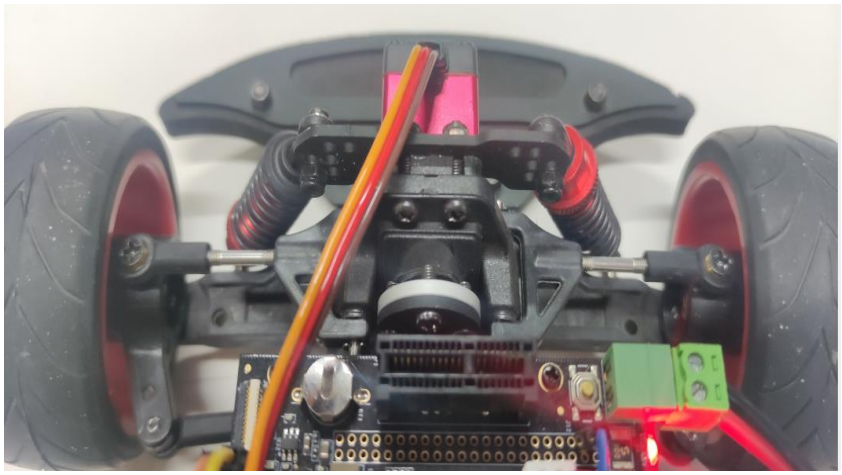


图 2.1 舵机安装和调整

2.2 摄像头的安装

对于摄像头的安装，需要在保证其前瞻性的基础上，考虑图像采集的稳定性。我们设想的摄像头安装位置有 3 种：底盘中间、底盘中间偏后和底盘最前沿。经过试验发现，如果将摄像头安装在最前面的话，由于车身的晃动，采集的图像质量不高，这对于图像的处理有着很大的影响，不利于电机和舵机的控制。此外，在前期的调试过程中，由于小车经常会冲出赛道，撞到实验室的物体，由此导致的摄像头晃动和碰撞，其连接件和底座可能会产生松动，导致摄像头位置的轻微变动，进而改变图像，对于图像识别和控制方面存在一定的隐患，这个方案最先被否决。最终我们选择将摄像头安装在中间偏后的位置，将其固定在后轮电机的上方。在确定了摄像头的大概高度和保证其采集的图像适配于我们的舵机控制方案的前提下，我们还用了热熔胶来对摄像头与碳素杆的连接处进行加固，以确保摄像头的位置和采集到的图像不会发生变化，固定摄像头的前瞻。在调试小车的过程中，车模还可能因为各种原因而发生侧翻。对此，在摄像头的连接处上方还预留有一定长度的碳素杆，防止车模侧翻时摄像头直接撞击赛道或地面带来的位置变动和损伤。 摄像头安装如图 2.2 所示：

图 2.2 摄像头的安装

2.3 编码器的安装

因为完全模型组的传动结构的限制，所以我们不能使用类似基础组别的编码器，我们使用的是赛曙科技提供的 CSPE5-500 型光电编码器，此编码器安装较为繁琐，其中码盘的安装需要格外注意，应同时将编码器的 PCB 板和码盘一起装到电机轴，然后通过顶丝将轴和码盘固定起来，实现同时传动，从而实现对电机的速度测量和控制。CSPE5-500 型光电编码器的安装如图 2.3 所示：

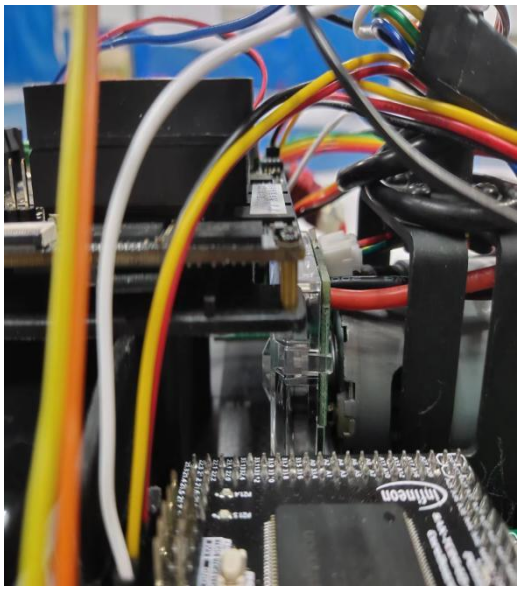


图 2.3 编码器的安装

2.4 系统电路板的固定及连接

因为 I 车模在底板的开孔个数有限，并且因为规则要求安装车壳，所以在设计 PCB 的时候要考虑车模两边的尺寸。我们根据 PCB 的大小，在车模底板开孔来实现两边电路板的固定。保证在运行过程中不会因为电路板的松动而出现意外情况。除此之外，因为底板为导电的碳纤维板，所以我们不能直接将电路板放到底板上。我们小车选择用铜柱进行自制电路板的固定。对于 Edgeboard，I 车模存在自带的安装模块。

我们的小车通过两个电路板完成下位机的相关功能。在设计时，将驱动模块和主控模块分隔开，能够保证电路板的安全性和稳定性。并且可以将小车的重心尽量维持在机械零点附近，能够有效的保证小车运行的稳定性。两个部分通过杜邦线进行连接，将单片机产生的 PWM 波通过杜邦线的方式传输到驱动电路中，实现对舵机和电机控制。

2.5 车壳的安装

车壳使用百度制作的车壳，外加贴画作为装饰，为了方便开关和调试，我们将车体上方和两侧开了一定大小的空隙。车壳图片如图 2.4 和图 2.5 所示:

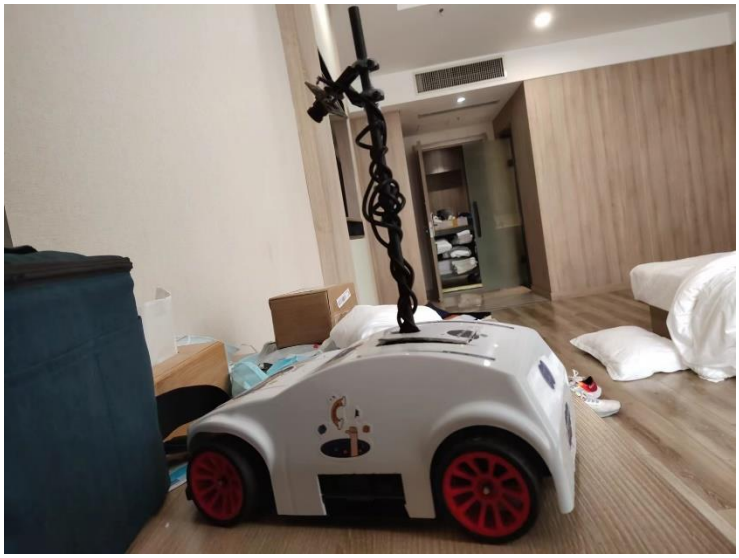


图 2.4 车模左视图

图 2.5 车模俯视图

第三章 硬件电路设计

硬件是基础，只有一个良好、稳定的硬件环境才能保证车能平稳快速的行驶。我们在整个系统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，PCB 的效果简洁。我们采用百度团队提供的下位机硬件模块，该系统采用模块化设计方式，主要包括下位机系统单片机最小系统模块、电源模块、电机驱动模块、串口通信模块、编码器测速模块、显示操作模块等部分。

3.1 下位机系统单片机最小系统模块

在本届比赛中，我们使用的下位机主控芯片为 Infineon 公司的 TC264DA，该款芯片属于英飞凌 Tricore 架构的 Aurix 系列单片机。这型单片机依靠 Infineon 公司在汽车电子领域深厚的底蕴，具有许多特别的优势（例如有多核并行处理能力，可以进行快速傅里叶变换，支持多种通信协议和丰富的通信接口等）。该款芯片的引脚数量为 144，双核架构，主频等级为 200MHz.这些丰富的系统资源可以更方便的连接外部电路，多个独立的 ADC 通道也使得外围的某些 ADC 模块的设计得到了简化，多种通讯接口也方便了扩展更多的外设和电路功。最高 200MHz 的频率也让该单片机具有更快的计算速度，使得可以最大的减小数据处理时间。对单片机的引脚利用如图 3.1 所示：

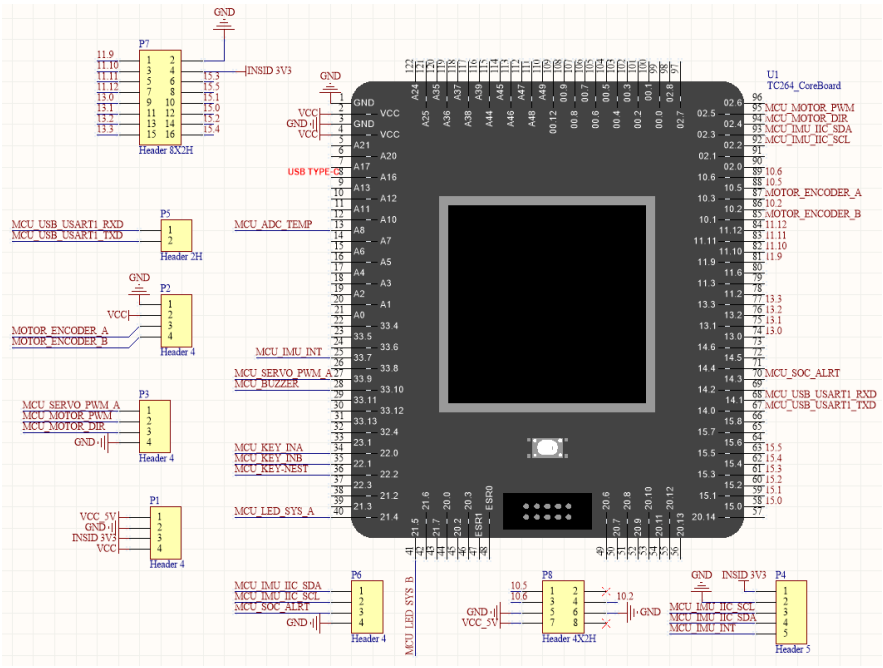


图 3.1 单片机引脚分配

3.2 电源模块设计

该系统中电源模块主要分为对系统中下位机控制电路各个模块以及上位机 Edgeboard 运行提供所需要的电源。设计中，除了需要考虑电压范围和电流容量等基本参数之外，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。可靠的电源方案是整个硬件电路稳定可靠运行的基础。

全部硬件电路的电源由一节 CB-22003 电池提供，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，本车模上存在 4 种供电电压：

(1) 智能车使用锂电池供电，正常使用时电压在 11.0~12.V。可通过开关直接用于电机供电以及对上位机 Edgeboard 的供电，供电模块如图 3.2 所示：

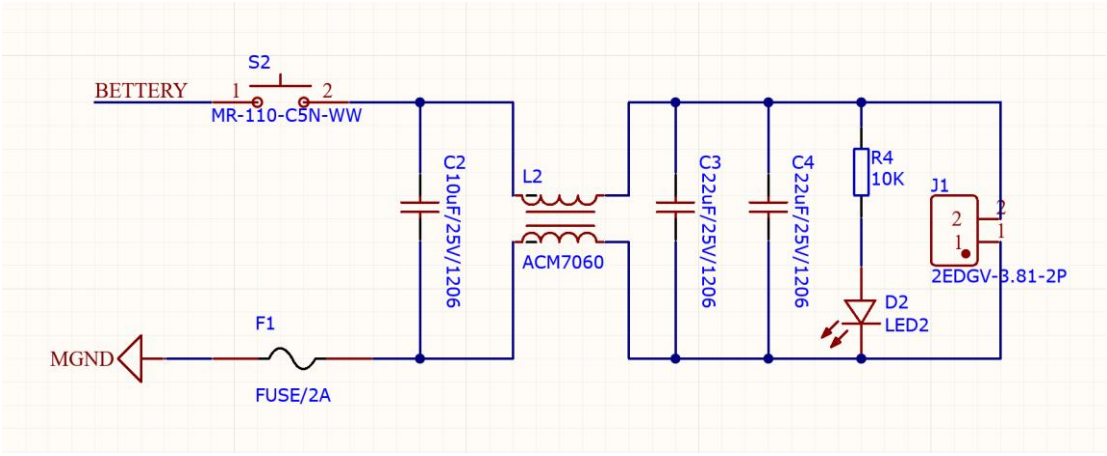


图 3.2 Edgeboard 直接供电

(2) 使用稳压芯片 TPS 5430DDAR 输出电压 7.4V，用于舵机 CS-3120 的供电。原理图如图 3.3 所示：

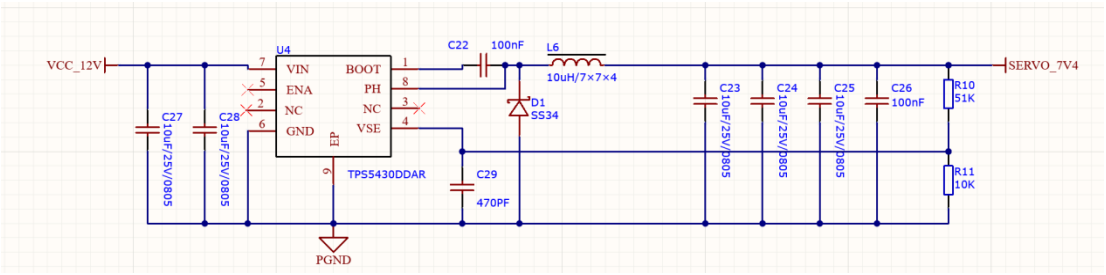


图 3.3 转压电路 12V-7V 原理图

(3) 使用稳压芯片 TPS5430DDAR 输出电压 5V，用于逻辑门 1G08，1G14 以及驱动芯片 IR2184PBF 等供电,原理图如图 3.4 所示：

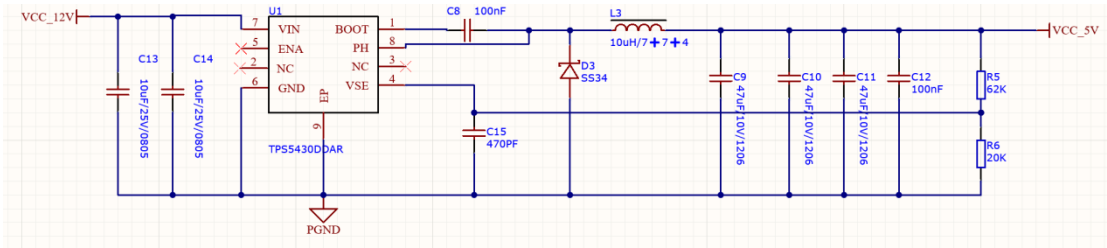


图 3.4 转压电路 12V-5V 原理图

(4) 使用稳压芯片 me6211c33m5g-n 输出电压 3.3V，用于编码器，有线串口，显示屏，单片机的供电，如图 3.5 所示：

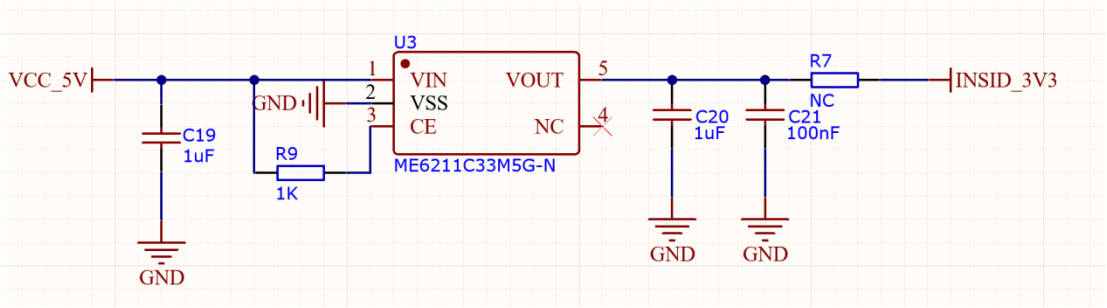


图 3.5 转压电路 5V-3.3V 原理图

3.3 电机驱动电路

在栅极驱动芯片选择方面，我们选择 IR2104 芯片，IR2104 芯片可以驱动高端和低端两个沟道 MOSFET，提供较大的驱动电流，并有硬件死区，防止同桥臂导通。使用两片 IR2104 可以构成一个 MOS 管全桥驱动电路，如图 3.6 所示：

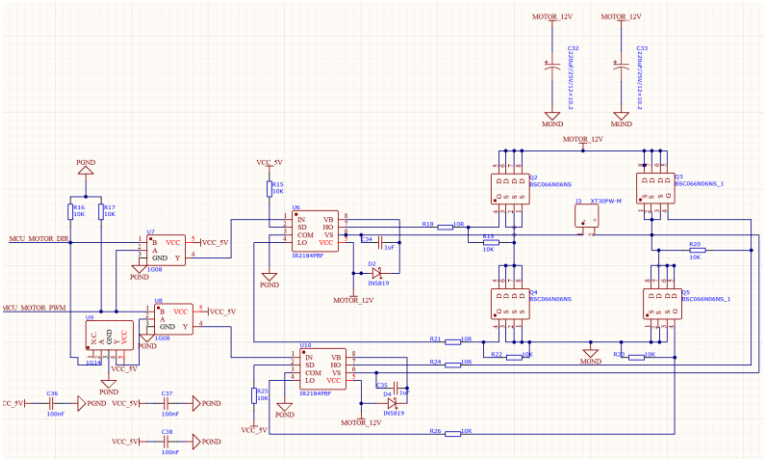


图 3.6 电机驱动原理图

3.4 串口模块

这里我们采用串口模块进行上下位机的通讯，上位机 Egdeboard 图像处理得到的数据通过串口模块发送到下位机上，下位机接受并做出相应处理。电路结构上我们采取直接将上位机串口通讯线 tx，rx 与下位机串口通讯线 rx，tx 直接通过杜邦线相连的方式进行连接。有关电路如下图：

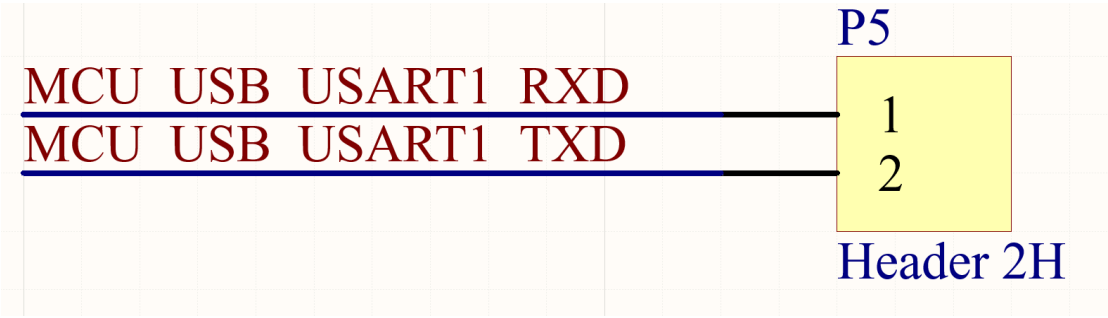


图 3.7 串口通信模块原理图

3.5 舵机控制电路

使用逻辑门 1G08 芯片用作电路隔离，能有效提高舵机使用寿命。

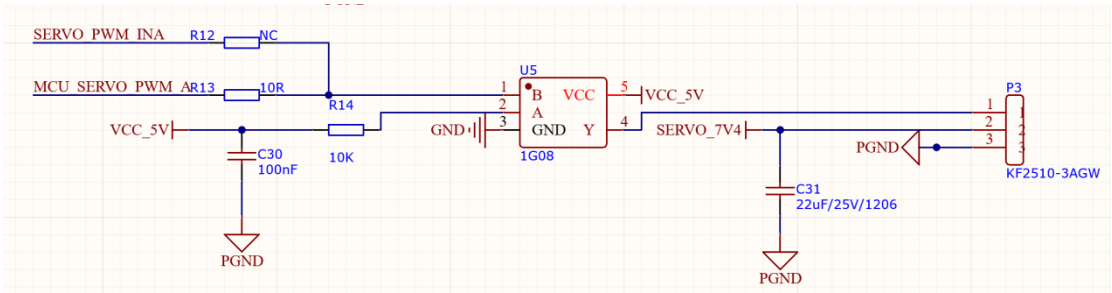


图 3.8 舵机控制电路

3.6 主控板

在本系统中我们的电路板的制作主要思想是分立工作、追求简洁，便于电路板的调试和安装等工作。主控由电源稳压电路，OLED 屏与按键交互电路，单片机系统以及各个传感器接口等组成。实物图如图 3.9 所示：

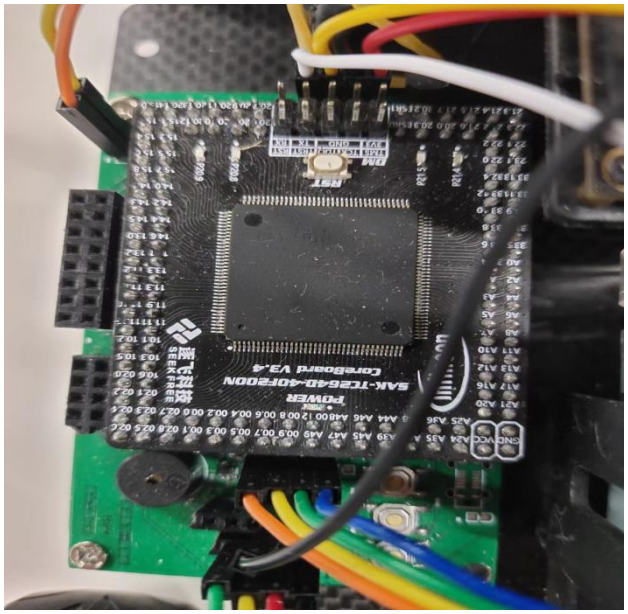


图 3.9 主控板实物图

3.7 驱动板

驱动板在除了原有的电机驱动电路外，还增加了电流环模块、编码器接口，主要目的是方便编码器的连线，以减少编码器连线过长导致的接触不良等问题，驱动板的实物图如图 3.10 所示：

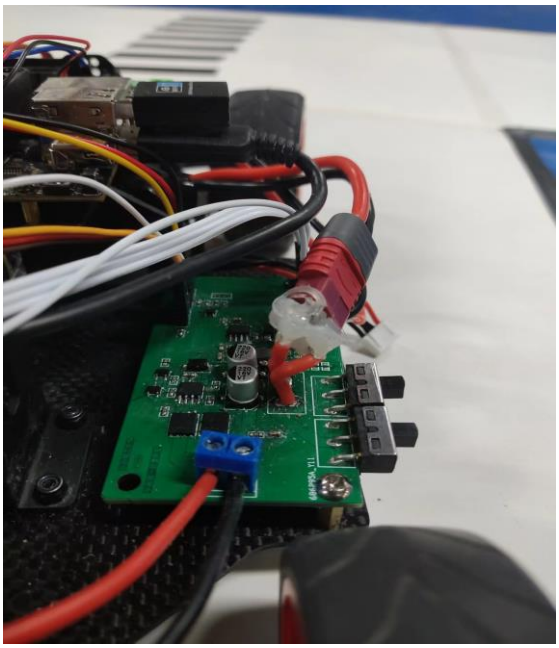


图 3.10 驱动板实物图

3.8 传感器的选择

3.8.1 摄像头的选择

CMOS 摄像头体积小，图像稳定性较高，只需 3.3V 供电，耗电量低；CCD 摄像头图像对比度高、动态特性好，但供电电压比较高。我们需要选用的摄像头，既要能支持 Edgeboard 免驱使用，且其分辨率和帧率要能够满足小车高速行驶状态下对图像的采集分析需求；而且还需要考虑小车的续航要求。选用 CMOS 摄像头对于供电模块的要求要比 CCD 摄像头要低，其性能虽不如 CCD 摄像头，但对于我们小车的需求而言也完全够用。综合考虑下来最终决定选用赛曙的 S320 CMOS 摄像头。

3.8.2 编码器的选择

光电编码器是一种通过光电转换将输出轴上的机械几何位移量转换成脉冲或数字量的传感器，其获取信息准确、精度高、应用简单，是目前应用最多的测速传感器之一。

采用增量式 500 线光电编码器，其供电电压为 3.3V 或 5V。增量式编码器转轴旋转时，有相应的脉冲输出，其旋转方向的判别和脉冲数量的增减借助后部的判向电路和计数器来实现。其计数起点任意设定，可实现多圈无限累加和测量。还可以把每转发出一个脉冲的 Z 信号，作为参考机械零位。编码器轴转一圈会输出固定的脉冲，脉冲数由编码器光栅的线数决定。我们选用的编码器分辨率高，抗干扰能力强，其与电机组成一个整体测速后，准确度高，稳定性好。

第四章 软件控制设计

高效的软件程序是智能车高速平稳自动循线的基础。我们设计的系统采用CMOS 摄像头进行赛道识别，图像采集及处理是整个软件的核心内容。在智能车的转向和速度控制方面，我们使用 PID 控制算法，使智能车能够稳定快速在赛道中循迹并完成比赛。

4.1 赛道中心线提取及优化处理

4.1.1 原始图像的特点

在 Edgeboard 采集图像信号后需要对其进行处理以提取主要的赛道信息，同时，由于交叉道、起跑线的存在，光线不均匀、赛道外反光点、赛道远处图像不清楚的干扰，图像效果会大打折扣。因此，在软件上必须排除干扰因素，对赛道进行有效识别，并提供尽可能多的赛道信息供决策使用。

在图像信号处理中我们提取的赛道信息主要包括：赛道位于图像的位置、赛道边界点信息，赛道宽度信息，赛道中心线位置，赛道类型识别，AI 图标识别。

由于摄像头自身的特性和远小近大的视觉原理，图像会产生梯形式变形，这使得摄像头看到的赛道信息和实际真实的赛道信息有所区别。经过摄像头采集的信息默认为彩色，首先我们通过 OpenCV 的自带函数来将其变为灰度图像。得到一个二位矩阵。

摄像头返回的这个矩阵中，每一个像素点都有一个从 0 至 255 的灰度值，值越大表示该像素点越亮，相反，越暗的像素点值就越小。同色间的灰度差很小，而不同色间的灰度差很大，我们使用 OpenCV 自带的二值化函数来进行灰度图像的二值化处理。

4.1.2 赛道搜线算法

Edgeboard 在对原始图像进行二值化之后，得到赛道信息，然后用搜线算法来对其进行处理，从而确定出赛道两边的黑线在哪儿，并计算出相对应的赛道中线，从而指导舵机转向和电机的加减速。搜线算法的基本思想如下：

(1) 首先从最底部中间开始搜线；

(2) 遍历图像，得到底部几行的中点后，后面几行的起点由上一行的中点开始搜起，来寻找左右黑白跳变点；

(3) 搜线完毕后，就得到了处理好的能够进行赛道分析的完整的矩阵。

4.1.3 赛道中心的计算

得到完整的赛道信息矩阵后，即可通过一定的算法计算出赛道的中线，由PID控制算法的思想，我们利用智能车的图像中心与赛道的实际中心的偏移量来控制舵机的转动和电机的驱动力度。赛道中线算法即为左右边线相加除以2。

4.2 出入车库策略

4.2.1 入车库位置判定

我们小队通过图像识别来判定车库位置，主要通过对斑马线最低行的判断来限定开始入库时车的位置，搜索车库的左上或右上角点，来往右下角或者左下角拉线，这种判断方法在低速时很稳定，对入库时机把握良好，速度越快，对应入库时机会有所偏差。

4.2.2 出入车库方案

对于出库，我们通过 AI 识别车库，因为出库只需要出一次，不用在意出库后因为模型识别不稳定而导致判断错误，左右出库分别对应右拉线和左拉线，能够实现较为稳定的出库动作。入库我们应用传统图像处理，速度慢的情况下可以实现较为稳定的入库动作。

4.3 模型训练

4.3.1 数据集的采集

对于数据集的采集我们通过手推车的方式进行。在Pycharm中编写好代码移植到Edgeboard中，通过相关操作将图像采集出来。

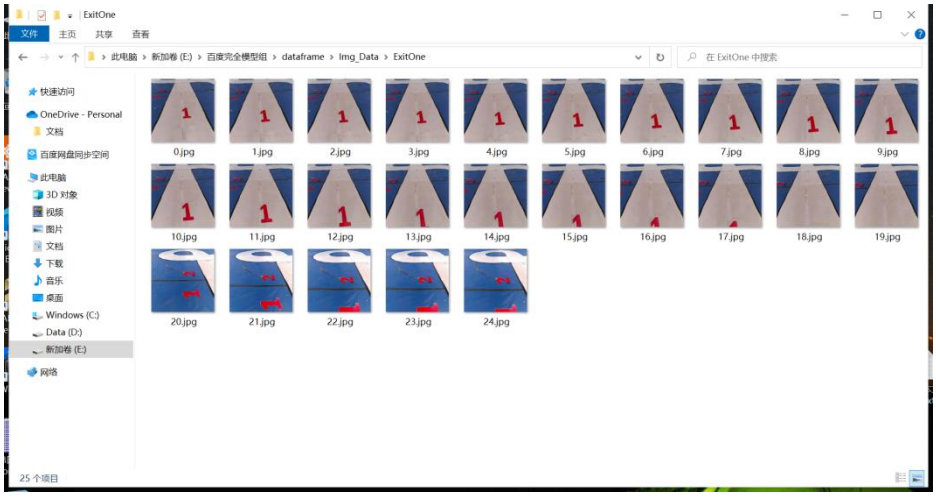


图 4.3.1 数据集的采集

4.3.2 数据集的标注

对于数据集的标注，我们组采用了 labelImg 软件来进行标注，对于标注的数据集我们通过数据清洗来达到可以进行端到端训练的格式。

4.3.3 模型的训练

模型训练通过百度 AIStudio 网站来进行训练。训练完成之后将模型部署在 Edgeboard 中，通过 Vscode 编写相关代码，以及应用 PaddleLite 相应的 API 来实现模型的预测。通过实时部署发现出去斑马线，其余模型的预测精度在百分之 60%到 99%不等。精度可观。

4.4 模型实时处理

模型实时处理为完全模型组的比赛要求，泛行区的识别处理我们小组通过 PaddleLite 的 API 首先进行元素识别，然后通过目标检测的框图可以识别模型的具体位置，但因为实际的处理都在二值化图像中，所以我们选择通过图像缩放的方式将模型在彩色图像中的位置转化为二值化图像中的位置，通过补线，重新搜线等一系列操作来完成泛行区的识别。对于施工区和加油站，思路都是通过锥桶的中心点来进行边线的规划，然后将通过锥桶赛道分为三个部分，分别为进入锥桶，锥桶巡航，出锥桶。通过跑模型进行赛道外车道线的识别。

4.5 上下位机通信

本组小车通过 Edgeboard 自带 UART 接口实现小车上下位机的通信。通过设

置帧头，标志位，然后一帧一帧传送数据，我们传送的是速度和角度值以此来使得 Edgebaord 能够快速的控制 TC264 来实现舵机转向和电机的驱动。然后通过校验位和帧尾完成一次数据的传送。

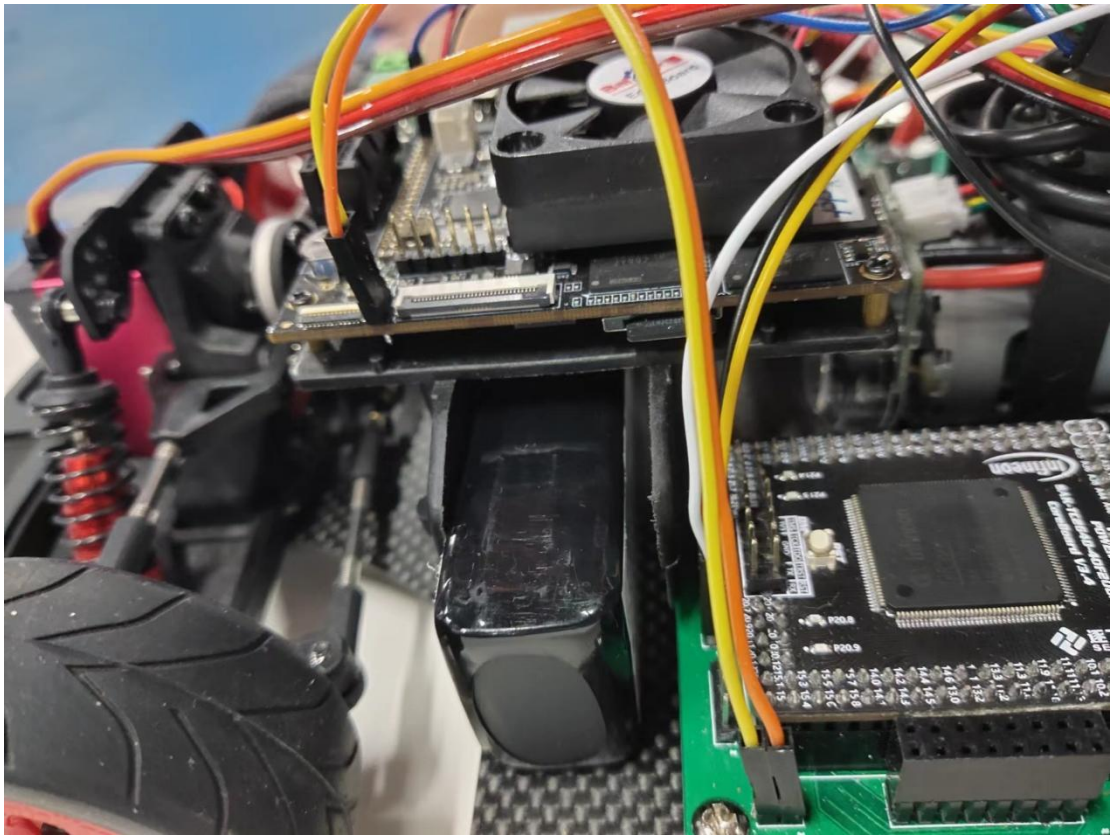


图 4.5 上下位机通信

4.6 PID 控制算法介绍

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。

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 4.6 所示。

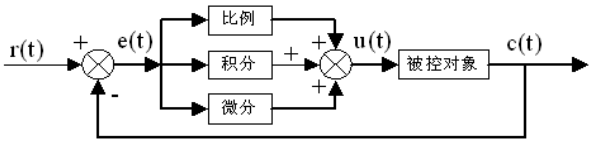


图 4.6 PID 控制器原理框图

4.6 转向舵机的 PD 控制算法

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。

对于舵机的闭环控制，我们采用了位置式 PD 控制算法，根据往届的技术资料和实际测试，将每场图像中线上部分中点加权平均值与舵机 PD 参考角度值构成非线性关系。

我们将直到和弯道区分开，分别应用不同的 PD 参数，通过区分直道弯道，也可以达到直道加速，弯道减速的效果。以此来使的小车更加平稳的运行。

4.7 驱动电机的 PI 控制算法

对于速度控制，我们采用了增量式 PI 控制算法，基本思想是直道加速，弯道减速。通过调节参数让电机的控制达到快速准确。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。

在调试参数的时候，我们通过 Json 配置文件来进行更改参数，不用编译烧录，较为方便。

第五章 系统调试

下位机软件开发工具选用的是 AURIX Development Studio，是英飞凌公司于 2019 年底推出的的集成开发环境，支持英飞凌 TriCore™内核 AURIX™ 系列 MCU；ADS 是一个完整的开发环境，包含了 Eclipse IDE、C 编译器、Multi-core 调试器、英飞凌底层驱动库（low-level driver,iLLD),同时对于编辑、编译及调试应用代码没有时间及代码大小的限制。

对于上位机开发则是通过 vscode 进行远程连接，来对 Edgeboard 中的代码进行实时修改，通过路由器将 Edgeboard 和本机连接到同一个局域网即可。

除此之外，可通过 VNC 来连接 Edgeboard 来进行图像的实时监控，可以对实际调车有很大的帮助。VNC 图像如图 5.1 所示：

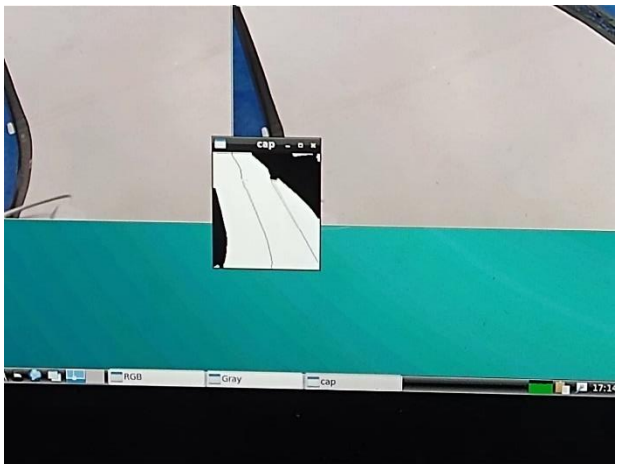


图 5.1 上位机调试图像

总结

自报名参加第十七届全国大学生智能汽车竞赛以来，我们小组成员从查找资料、设计机构、组装车模、编写程序一步一步的进行，最后终于完成了最初目标，定下了现在这个设计方案。

在此份技术报告中，我们主要介绍了新组别的规则，机械结构，软件架构和特殊元素的处理思想。在机械结构方面，我们介绍了车模的舵机和电机传动模块以及编码器，摄像头的安装；在电路方面，我们以模块形式分类，我们自己进行引脚分配，分模块化进行电路板的处理；在软件方面，我们使用过 Vim 和 Vscode 等编辑器进行配置和编写代码，语言使用和 C++和 Python，并且学会了使用百度 PaddlePaddle 的一些接口 API；在调试方面，我们学习了 PID 控制算法来控制小车，学习了分段 PID、模糊 PID。经过一年的学习，智能车竞赛让我们组受益匪浅。

现在，面对即将到来的大赛，我们已经做好了充分的准备，以最饱满的热情迎接比赛的到来，能站在最高赛事和其他学校进行学术交流也是一件无比荣幸的事情。这份技术报告，是我们组一年的总结，也是我们付出的证明。希望我们组的技术报告对以后参加智能车的成员能够有所帮助。

参考文献

[1]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.

[2]张军. AVR 单片机应用系统开发典型实例. 北京：中国电力出版社，2005.

[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，2001.

[4]阎石. 数字电子技术基础 [M] . 北京: 高等教育出版社，2000.

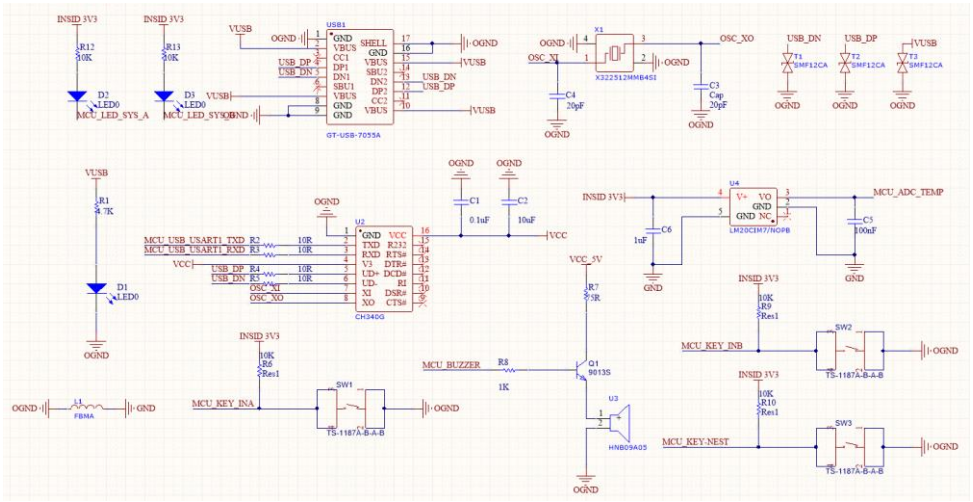
[5]谭浩强著. C 程序设计. 北京：清华大学出版社，2003.

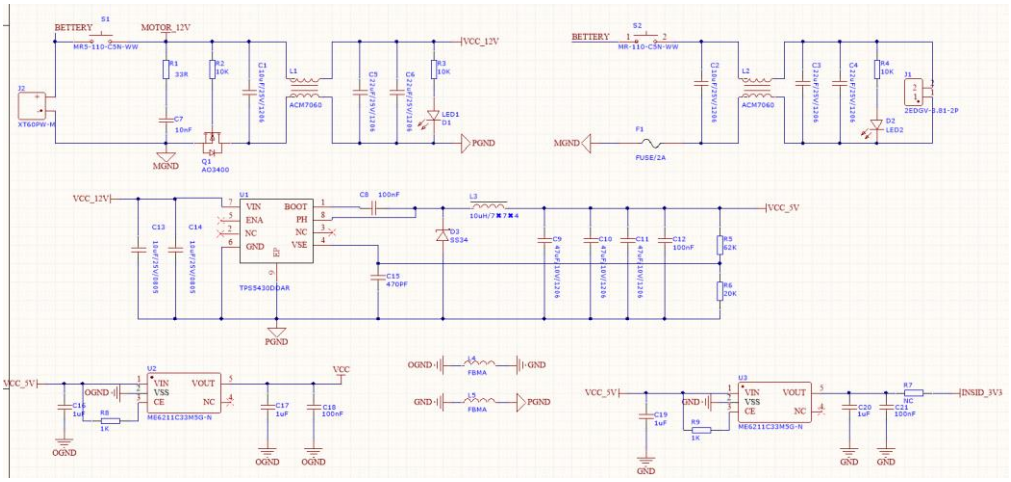
[6]塔里克·拉希德著，林赐译.Python 神经网络编程.

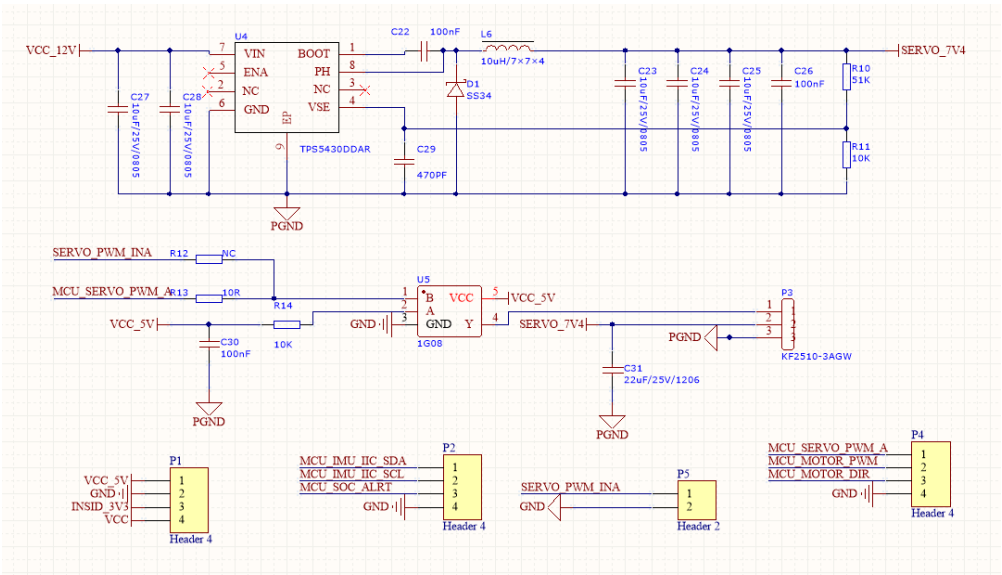
[7][美]史蒂芬·普拉达，C++ Primer Plus 第六版

[8]周志华，机器学习.

附录 A 电路板原理图


INSID_3V3
VUSB
OGND
USB1
GT-USB-7055A
X1
X322512MMB4SI
OSC_XI
OSC_XO
USB_DN
USB_DP
C1
C2
C3
C4
C5
C6
U2
CH340G
U4
LM20CIM7/NOPB
MCU_ADC_TEMP
MCU_USB_USART1_TXD_R2
MCU_USB_USART1_RXD_R3
VCC
VCC_5V
MCU_BUZZER
MCU_KEY_INA
MCU_KEY_INB
MCU_KEY_NEST
SW1
SW2
SW3
TS-1187A-B-A-B
Q1
9013S
U3
HNB09A05
D1
LED0
FBMA


BETTERY
MOTOR_12V
MR5-110-C5N-WW
MR-110-C5N-WW
VCC_12V
PGND
MGND
ACM7060
FUSE/2A
U1
TPS5430DDAR
SS34
U2
ME6211C33M5G-N
VCC_5V
VCC


VCC_12V
U4
TPS5430DDAR
SERVO_7V4
PGND
SERVO_PWM_INA
MCU_SERVO_PWM_A
U5
1G08
P1
P2
P3
P4
P5
MCU_IMU_IIC_SDA
MCU_IMU_IIC_SCL
MCU_SOC_ALRT
MCU_MOTOR_PWM
MCU_MOTOR_DIR
KF2510-3AGW
Header 4
Header 2

附录 B 程序源代码

```
cv::Mat Binary(cv::Mat frame)
    {
        cv::Mat OTSU_IMG;
        cv::Mat gray;
        cv::Mat Binary_IMG;

        cv::cvtColor(frame, gray, cv::COLOR_BGR2GRAY);

        cv::threshold(gray, OTSU_IMG, 0, 255, cv::THRESH_OTSU);
        resize(OTSU_IMG, OTSU_IMG, cv::Size(160, 180));

        cv::Mat element = cv::getStructuringElement(cv::MORPH_RECT,
cv::Size(3, 3));
        dilate(OTSU_IMG, OTSU_IMG, element, cv::Point(-1, -1), 1);
        erode(OTSU_IMG, OTSU_IMG, element, cv::Point(-1, -1), 2);

        threshold(OTSU_IMG, Binary_IMG, 0, 255, cv::THRESH_OTSU);
        return Binary_IMG;
    }

    void Basic_ScanLine(cv::Mat& image, int BMX_Flag)
    {
        int rem_last_middle, rem_last_left, rem_last_right;
        int rows = 0;
        int cols = 0;
        rem_last_middle = (image.cols) >> 1;
        img.IMG_ROWS = image.rows - 1;
        img.IMG_COLS = image.cols - 1;
        /***NomalScanLine***/
        memset(img.right_find, 0, sizeof(img.right_find));
        memset(img.left_find, 0, sizeof(img.left_find));
        int test = 0;
        for (rows = img.IMG_ROWS; rows >= 0; rows--)//行
        {
            for (cols = rem_last_middle; cols >= 3; cols--)//列,左边
线
            {
                if ((int)image.at<uchar>(rows, cols) == 255 &&
                    (int)image.at<uchar>(rows, cols - 1) == 0 &&
                    (int)image.at<uchar>(rows, cols - 2) == 0 &&
                    (int)image.at<uchar>(rows, cols + 2) == 255)
```

```
                {
                    img.left_find[rows] = 1;
                    img.Left_Line[rows] = cols + 1;//记录左边线
                    if (img.Left_Line[rows] > 130)
                    {
                        corner_left++;
                    }
                    break;
                }
                else img.Left_Line[rows] = 0;
            }

            for (cols = rem_last_middle; cols < img.IMG_COLS - 3;
cols++)//列，右边线
            {

                if ((int)image.at<uchar>(rows, cols) == 255 &&
                    (int)image.at<uchar>(rows, cols + 1) == 0 &&
                    (int)image.at<uchar>(rows, cols + 2) == 0 &&
                    (int)image.at<uchar>(rows, cols - 2) == 255)
                {
                    img.right_find[rows] = 1;
                    img.Right_Line[rows] = cols - 1;//记录右边线
                    if (img.Right_Line[rows] < 40) corner_right++;
                    break;
                }
                else img.Right_Line[rows] = 159;
            }



            img.Middle_Line[rows] = (img.Left_Line[rows] +
img.Right_Line[rows]) >> 1;
            rem_last_middle = img.Middle_Line[rows];
        }

        for (cols = img.IMG_COLS - 1; cols >= 5; cols--)
        {
            for (rows = 80; rows >= 3; rows--)
            {
                if ((int)image.at<uchar>(rows, cols) == 255 &&
                    (int)image.at<uchar>(rows - 1, cols) == 0 &&
                    (int)image.at<uchar>(rows - 2, cols) == 0)
                {
```

```
                    img.Up_Line[cols] = rows + 1;//记录上边线
                    break;
                }
                else img.Up_Line[cols] = 1;
            }
        }
        static uint8_t g_ucFlagRoundabout = 0;
        if (g_ucFlagRoundabout == 0 && Exit_Two == 0 && Gas_Station
== 0)//
        {
            RoadIsRoundabout(image, &g_ucFlagRoundabout);
        }

        if (g_ucFlagRoundabout)
        {
            Circle_Flag = 1;
            RoundaboutProcess(image, &g_ucFlagRoundabout);
            cout << "g_ucFlag: " << (int)g_ucFlagRoundabout << endl;
        }
        if (Start == 0)
            Car_Start(image, BMX_Flag, &Start);
        BMX_JUDE(image);
        if (g_ucFlagRoundabout == 0)
            Heigher_ScanLine(image);
        static uint8_t model_flag = 1;
        if (Sancha_flag == 1)
        {
            Flooed_Process(image, &model_flag);
        }


        if (Mid_ == 0)
            for (rows = 0; rows <= 179; rows++)
            {
                img.Middle_Line[rows] = (img.Right_Line[rows] +
img.Left_Line[rows]) >> 1;
                image.at<uchar>(rows, img.Middle_Line[rows]) = 0;
            }
    }
```