第十七届全国大学生
智能汽车竞赛

技 术 报 告

辽宁工程技术大学
LIAONING TECHNICAL UNIVERSITY

学　　校：辽宁工程技术大学

队伍名称：光翼零队

参赛队员：张杨 陈嘉和 吴灿 谢辰晨

带队教师：王智勇 姜国强

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：

带队教师签名：

日　　期：2022年08月18日

目录

第一章 引言

全国大学生智能汽车竞赛是以“立足培养、重在参与、鼓励探索、追求卓越”为宗旨，鼓励创新的一项科技竞赛活动。竞赛要求在规定的汽车模型平台上，通过增加道路传感器、电机驱动模块以及编写相应控制程序，制作完成一个能够自主识别道路的模型汽车。参赛队员的目标是模型汽车需要按照规则以最短时间完成单圈赛道。智能汽车竞赛的赛道路面为宽度不小于45cm的白色面板，赛道两侧边沿有宽为25mm的连续黑线作为引导线。

在这份报告中，我们队伍将通过对车模设计制作整体思路、电路、调试等方面的介绍，详尽地阐述了我们的思想和创意，具体表现在电路的创新设计，以及机械结构的独特想法，而对EdgeBoard板卡及单片机具体参数的调试也让我们付出了艰辛的劳动。这份报告凝聚着我们的心血和智慧，是我们共同努力后的成果。

我们队伍的成员涉猎控制、模式识别、传感技术、汽车电子、电气、计算机等多个学科，这次对我们的知识融合和实践动手能力的培养有极大的推动作用，在此要感谢清华大学，十分感谢他们将这项很有意义的科技竞赛引入中国；也十分感谢辽宁工程技术大学电气与控制工程学院对此次比赛的关注与宣传，我们的成果离不开辽宁工程技术大学创新实践学院的大力支持及电气与控制工程学院王智勇老师等指导老师悉心的教导；还要感谢的是和我们一起协作的队员们，协助，互促，共勉使我们能够走到今天。

第二章 方案设计

2.1 系统概述

智能车主要由三个部分组成：检测系统，控制决策系统和动力系统。其中检测 系统采用摄像头，控制决策系统采用EdgeBoard作为上位机，英飞凌TC264主控板作为下位机。动力系统主要控制直流电机的转速。整体的流程为:通过摄像头来检测前方的赛道信息以及采集图片，并将赛道信息发送给上位机。同时，通过微型编码器构成的反馈渠道将车体的行驶速度信息通过下位机发送给上位机。根据所取得的赛道信息和车体当前的速度信息，由上位机做出决策，并通过与下位机通信将PWM信号控制直流电机进行相应动作，从而实现车体的转向控制和速度控制。

在实现深度学习在无人驾驶的应用，百度的AIstudio提供了免费的GPU 算力实现模型训练，还提供了移动端的飞桨框架，方便模型在移动端部署，第十七届全国大学生智能汽车竞赛完全模型组的赛题利用飞桨深度学习框架来实现模型训练与部署。这次比赛需要我们完成施工区、泛行区以及加油站等元素，在此完成的基础上，用时短者胜。

多次测试后，基本实现了Edgeboard与英飞凌TC264主控板的上下位机通信联调、基于PaddleLite的模型部署以及模拟路段下的控制量信息正确传送和实时的运动控制。根据智能车系统的基本要求，我们设计了总体方案图，如图1-1所示。

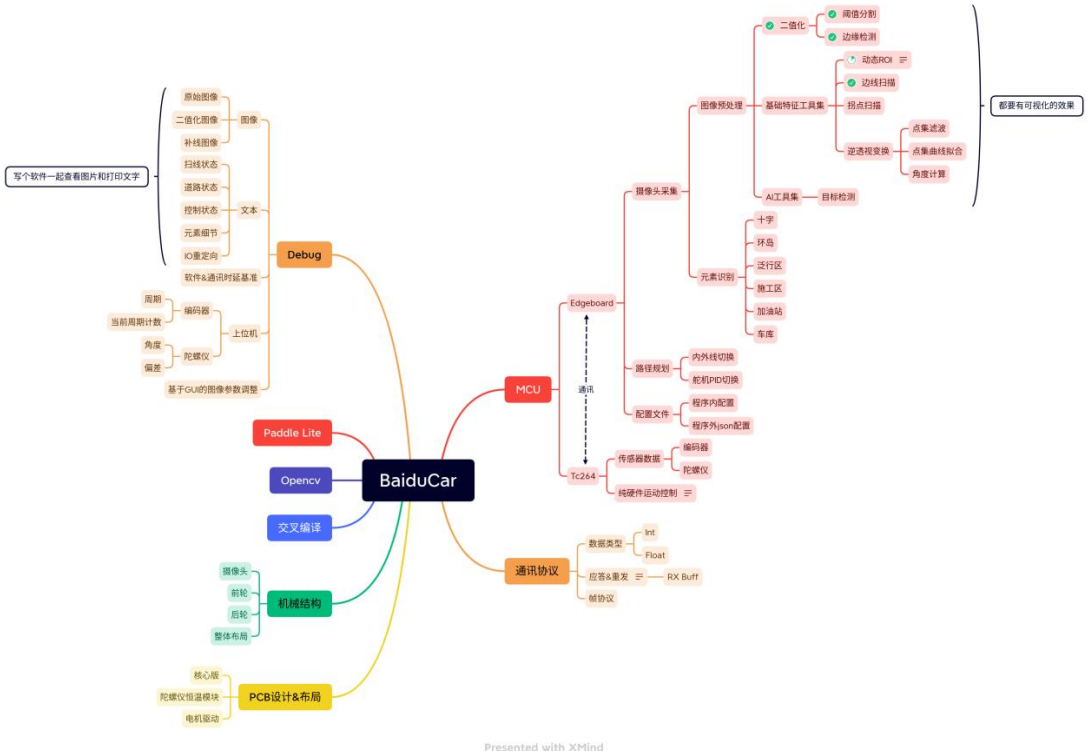


图1.1总体方案图

2.2 整车布局

（1）电池放于车模前部，合理分配重心，方便速度控制。
（2）舵机直立放置，以提高舵机响应速度。
（3）铜柱结合亚克力板固定Edgeboard板卡减小板卡晃动对导线接线处的影响。
（4）加装原厂的弹簧进行增强，以增加固定件的方式将悬挂固定。
（5）车模内部使用允许使用无WiFi模块用于车模的调试和发车。

第三章 比赛任务

3.1 完全模型组比赛任务

3.1.1 比赛任务以及要求

(1)车 模

车模采用I型车模。要求自行设计制作车模外壳。车模制作完之后，对于车模尺寸没有限制。

(2)微控制器与传感器

车模的赛道元素检测识别需要只能使用百度EdgeBoard计算卡（FZ3B赛事定制版）且只允许使用1块。车模运动控制单片机使用Infineon单片机。视觉模型部署在百度的EdgeBoard。

要求模型算法必须使用百度Paddle框架搭建，即必须使用百度深度学习框架的人工智能算法实现。考虑到学生自身设备可能有限，组委会提供统一线上算力平台AI Studio（https://aistudio.baidu.com/）便于大家训练模型。

车模作品中只允许最多使用2个摄像头对赛道进行识别，并且摄像头必须直连到EdgeBoard计算卡（可通过HUB拓展USB接口直连数量）上用于赛道及其元素的检测。车模作品中允许使用其他非摄像头类传感器进行环境的辅助检测，车辆姿态和运动控制的反馈，但不得用于赛道元素（直道，弯道，坡道等）和赛道标志的识别。选用的传感器或者其它电子部件中不得包括独立的微处理器，超声波传感器除外。

(3)比赛赛道

a.基础赛道的标准元素

b.完全模型组特有元素:“泛行区”、“施工区”、“加油站”、“坡道”

(4)比赛任务

a.循迹任务：车模在给定时间内，队伍车模按照组别任务完成比赛，比赛次数不限。最终成绩取比赛成绩中最优两次成绩的平均值。如果在规定时间内只完成一次比赛，则取此次比赛成绩的1.5倍为最终成绩。

b.识别任务：车模在运行时识别泛行区、施工区、加油站等特殊区域。

3.1.2 第十七届完全模型组赛道特有元素

比赛在基础赛道外设置有脱离基础赛道的任务区域，这些任务区域由可以移动的锥桶在基础赛道的边缘临时搭建而成。

图3.1锥桶示意图

竞赛所用锥桶由塑料材质制作而成，外表面为红色纯色，无任何标志。锥桶的底部直径74mm，高度74mm。在赛道搭建的过程中锥桶为可移动状态，车辆在运行过程中触碰到锥桶，导致锥桶偏离所在位置时，车模按冲出赛道处理。

（1）泛行区

两个岔路口围成的区域为泛行区，在完全模型组别中小车行驶不受三岔路两条线路内部闭合的道路边界线的限制，内部闭合的道路边界线特定区间内不铺设路肩。参赛队自行选择最优的路线从三岔路的入口行驶到出口即可。三岔路口的入口车道中心处放置有识别定位标志，表示前方为泛行区。泛行区的蓝底布区域出口和入口中心连线的指定范围内放置有禁止通行的标志（位置随机），车辆需要绕过此标志进行通行。车辆在蓝底布部分通过泛行区，奖励减时间5秒，车辆撞到禁止通行的标志后通过泛行区既不惩罚也不奖励。路肩的铺设要求和标志放置如图所示。

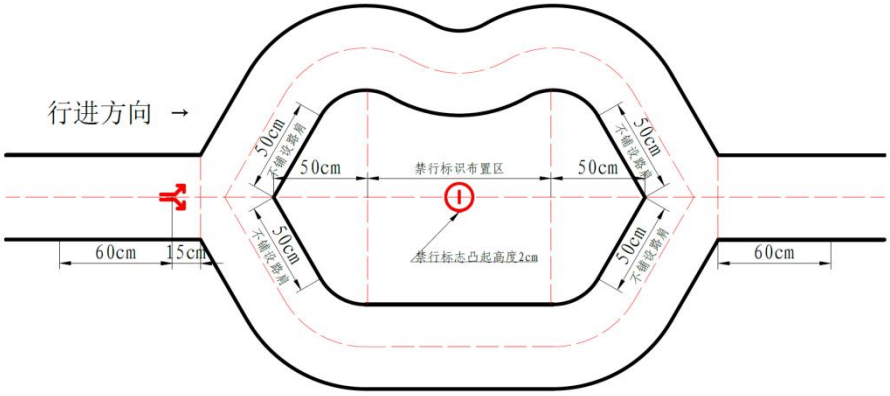


图3.2泛行区示意图

（2）施工区

赛道中选择一段直道，设置施工绕行区域。施工绕行道路采用锥桶围成，从赛道的边缘黑线区处开始放置（入口处黑线边缘不放置），组成一条临时通行车道。车辆行驶到施工区的时需要离开常规的赛道从搭建的绕行车道通行。施工区域前方70厘米处（标志中心到锥桶中心线的距离）放置施工标志指示车辆前方为施工区需要进入临时道路进行绕行。

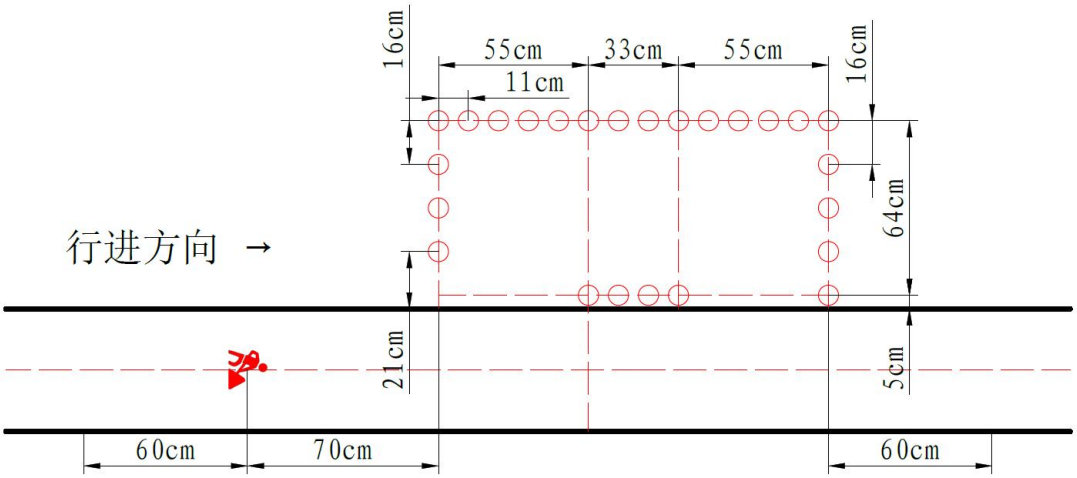


图3.3施工区示意图

图中每个红色圆圈代表1个小锥桶，实际摆设中存在锥桶中心不在同一直线以及间距也不完全相等，误差在锥桶半径范围内。车辆未进入施工绕行道路1次罚加时5秒，车辆每次成功通过绕行区奖励减时间5秒。

（3）加油站

赛道中设置有加油站区域，加油站区域由红色的锥桶围成（入口处黑线边缘不放置锥桶），分为1个入口和2个出口。2个出口分别标有1，2两个数字。加油站入口处的赛道中央贴有驶出时出口数字,识别到出口数字后从对应的出口驶出即可。在车模出发后，由现场裁判员随机确定指示牌的数字，车模运行两圈，两圈的数字不相同。车辆识别到数字后驶出加油站，当车模离开加油站之后，再由裁判员将标牌更换。加油站区域前方70厘米处赛道中心贴有加油站标志，指示车辆前方为加油站需要进入加油站后驶出。加油站元素仅在全国赛上使用。

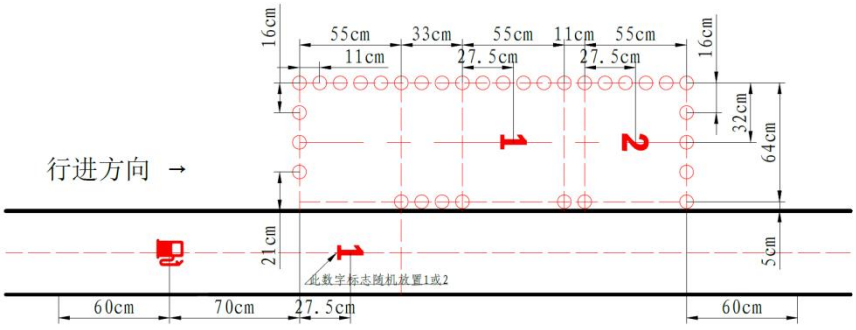


图3.4加油站示意图

（4）坡道

基础赛道中设置有坡道，为了契合人工智能自动驾驶的技术路线，因此在坡道前方15厘米处贴放坡道的标志，供人工智能模型的识别，便于提前预知坡道从而进行速控操作。

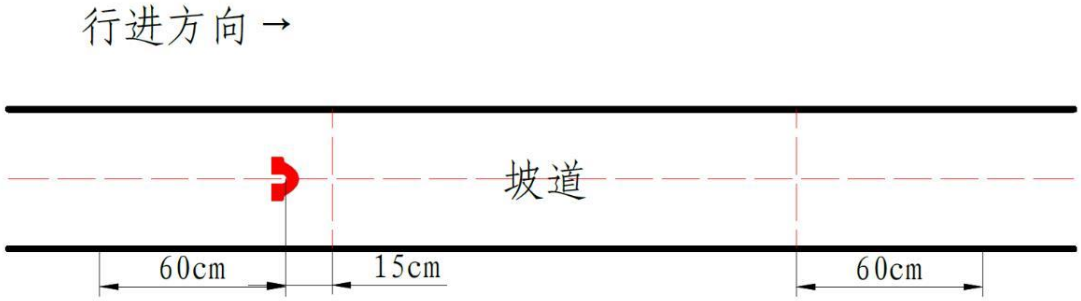


图3.5坡道示意图

3.1.3 第十七届完全模型组赛道普通元素

（1）十字路口

十字路口没有任何限制，在十字路口中心店两条相交的赛道中心线是垂直即可。两个十字路口中心距离，以及十字路口距离三岔路口中心的距离不小于45厘米。

完成十字路口任务标准：

①从十字路口一侧进入，直线通过十字路口；

②不允许直接左拐与右拐；

车模通过同一十字路口有两种情况：

①如果车模从十字路口通过后，没有途径其它赛道元素(坡道、环岛、三岔路口)直接回到十字路口(这种情况称为**十字圆环路口**)；

②如果从十字路口通过后，途径其它赛道元素(坡道、环岛、三岔路口）之后又回到同一十字路口通过；

以上两种都被视作通过十字路口两次

图3.6十字圆环的小型十字路口

（2）环岛

要求环岛赛道中心线的直径大于等于 50 厘米。环岛中心线与赛道中心线是 相切关系。

完成环岛任务标准：进入环岛绕行至少一周以上驶出环岛；

对于在环岛内多次绕行不进行判罚。不允许车轮进入环岛内侧区域，否则认 为是比赛任务失败。

（3）坡道

只规定坡道的最高定点距离地面不小于10厘米。对于坡道的宽度、长度、形状以及制作材质不做任何要求。通过坡道任务标准：车模在通过坡道最高点时，车轮必须距离地面超过10厘米。

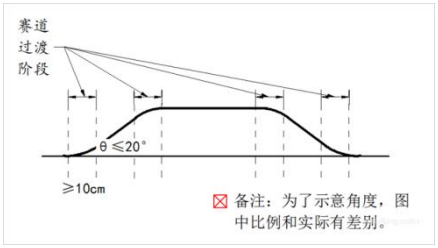


图3.7坡道示意图

在满足上述要求的同时，车模是否从坡道上掉落不进行判罚。

（4）车库

车库允许放置在赛道任意位置。对于车库的形状没有任何限制。只要保证车 模在发车时，车模的中心线垂直与赛道中中心线，车模车轮在赛道边缘之外即可。

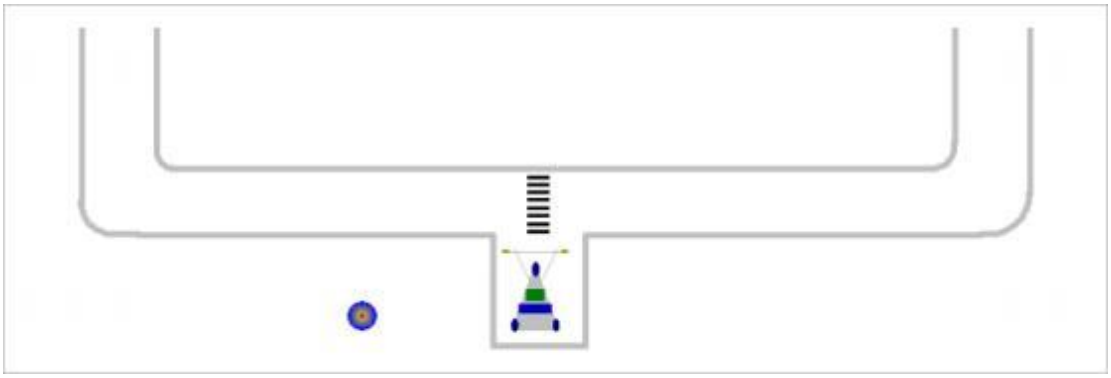


图3.8车模发车时与赛道中心线垂直并在赛道之外

完全模型组赛道各元素罚时标准见下表3.9：

加罚种类	加罚时间(s)	判断标准
入库失败	15	有一个或者以上的车轮中线在车库范围之外。
未进环岛	30	车模没有进入环岛运行。总的罚时等于未进环岛次数乘以30。
未进入施工区	5	车模未进入施工区元素，从直道通过则加罚5秒。总的罚时等于通过次数乘以5。

表3.9完全模型组赛道各元素罚时标准

完全模型组赛道各元素奖励时间标准见下表3.10：

奖励种类	奖励减时间(s)	判断标准
成功通过泛行区	5	车模从三岔路的入口进入蓝底布到另一侧三岔路出口驶出蓝底布，并未碰到禁行标志则奖励减5 秒时间；
成功通过施工区	5	车模每次成功通过施工区则奖励减5 秒时间，如果车模在行驶过程中碰到锥桶，但锥桶未移动位置则认为成功通过；

表3.10完全模型组赛道各元素奖励时间标准

第四章 硬件设计

4.1 硬件设计方案

整个智能车控制系统是由三部分组成的：Edgeboard上位机、下位机、电机驱动电路板以及电源管理班板。 最小系统板可以插在主板上组成信号采集、处 理和电机控制单元。同时为了减小下位机以及电机驱动电路给上位机带来的干扰， 我们把上位机，下位机和电机驱动分开来。

上位机主要电路如下：Edgeboard主芯片采用Xilinx公司的XAZU3EG-1SFVC784I芯片。PS端挂载了4片DDR4（4GB，64bit），1片8GB eMMCFLASH 存储芯片和1片256Mb的QSPI FLASH、2个USB接口（1个USB3.0，1个USB2.0）、1个MINI DP接口、1路千兆以太网接口、1个USB串口、1路PCIE接口、1路TF卡接口、1个44针扩展口、1路MIPI接口，1路BT1120接口和按键LED。

下位机主要电路包括如下：电源稳压电路、最小系统板插座、电机驱动器接口、测速接口、OLED、电源接口、指示灯、开关等。

4.2 传感器选择

本设计中，传感器分为两部分：摄像头传感器、速度传感器。

4.2.1 摄像头

摄像头的工作原理是：景物通过镜头(LENS)生成的光学图像投射到图像传感器表面上，然后转为电信号，经过A/D(模数转换)转换后变为数字图像信号，再送到 数字信号处理芯片(DSP)中加工处理，再通过一定的传输方式给其他设备,由于加入识别这个任务,再加上赛道环境在一定程度上是未知,所以我们采用 摄像头识别的识别方式。

市面上常见的摄像头主要是分为两种:数字摄像头和模拟摄像头,数字摄像头 主要有OV7725,OV2640, OV5640,OV7670,MT9V032等,模拟摄像头主要有MT9V136等。数字摄像头通常会有DVP接口,其数据口通信电压一般为3.3v远低于一般模拟摄像头需要的12v,采用模拟摄像头会大大增加电路的安全性, 就算是操作失误,也不会造成大范围的硬件损坏。

事实证明智能车对于图像的分辨率要求并不是很高,但是对于其动态性能要求十分高,特别是在小车高速行驶时,图像变化会很大,再加上赛场环境的复杂的光照条件,过亮过暗,有无阳光直射,灯光的颜色和强度都会影响到摄像头的成像,所以 选择一款满足条件摄像头就显得尤为重要。

KS2A543是一款彩色全局曝光摄像头,作为一个低压的COMS能通过USB2.0接口提供图像。

图4.1 KS2A543摄像头及其参数

4.3 下位机

单片机最小系统板使用逐飞科技出品的TC264系统板，为减少PCB电路板的空间，主板上仅将本系统所用到的引脚引出，包括信号接口、外部中断接口、若干普通IO接口等。

4.3.1电源管理部分

电源部分电路图如图4.2所示：

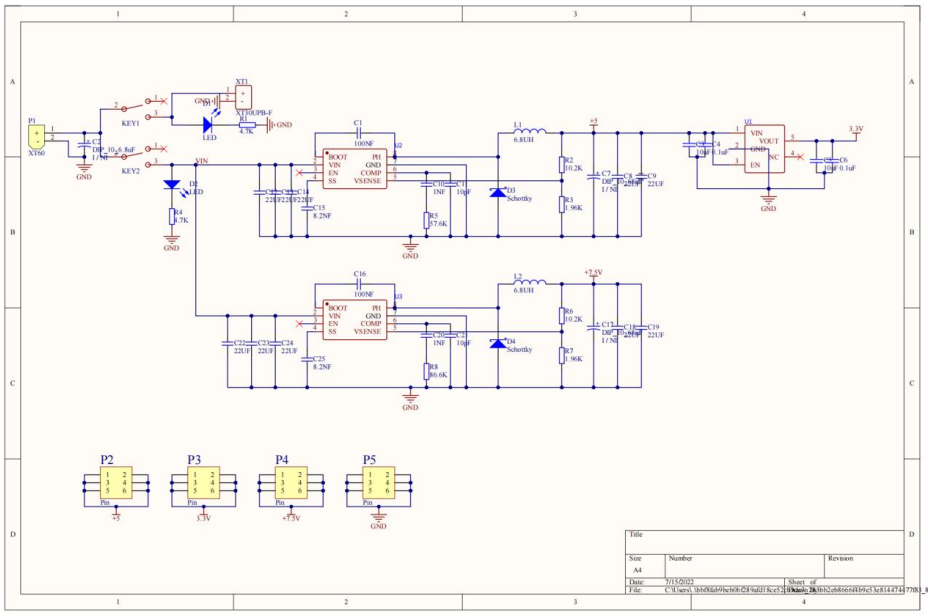


图4.2 电源管理电路

4.3.2电机驱动板

电机驱动板为一个由分立元件制作的直流电动机可逆双极型桥式驱动器，其功率元件由四支N沟道功率 MOSFET管组成，额定工作电流可以轻易达到100A以上，大大提高了

电动机的工作转矩和转速。该驱动器主要由以下部分组成：PWM号输入接口、逻辑换向电路、死区控制电路、电源电路、上桥臂功率 MOSFET管栅极驱动电压泵升电路、功率 MOSFET管栅极驱动电路、桥式功率驱动电路、缓冲保护电路等。

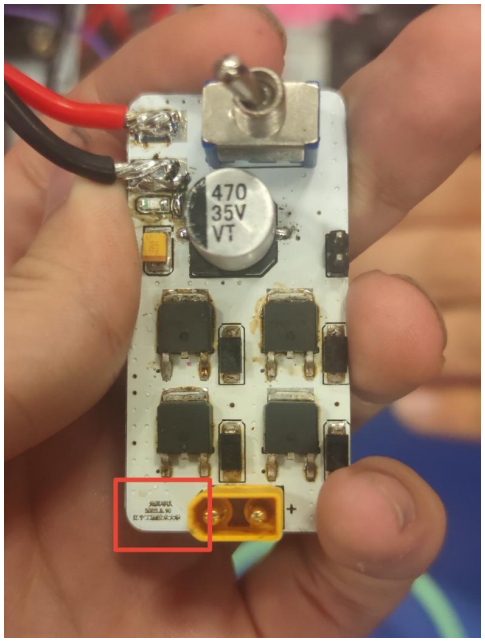


图4.3 电机驱动板实物

第五章 软件系统设计

5.1 软件控制程序的整体思路

软件的控制，就是让控制车模在符合比赛规则前提下，以最快最稳定的速度跑完整个赛道。不论上哪种方案，软件的总体框架总是相似的，我们追求的就是稳定至上，兼顾速度，以我们的方案，软件上就是图像采集、图像处理、识别横断，速度控制以及速度反馈以及计算偏差。

5.2 图像处理

采用OpenCV对采集回来的图像进行处理。主要分为以下步骤：

1.图像的灰度化以及二值化。

2.图像的最大连通域的选取。

3.图片的中线求取，中心扩散搜索，找到黑点记录黑点当前列值；两边都有直接求和平均求出中心值；只找到一边时，根据图像修正值确定中心值。

图5.1 二值化前的图像

5.3 电机以及舵机控制

PID控制主要有三部分组成，比例、积分、微分。比例控制是一种最简单的控制方式。其控制器的输出与输入误差信号成比例关系。偏差一旦产生，调节器立即产生控制作用使被控量朝着减小偏差的方向变化，控制作用的强弱取决于K_P。当仅有比例控制时系统输出存在稳态误差。

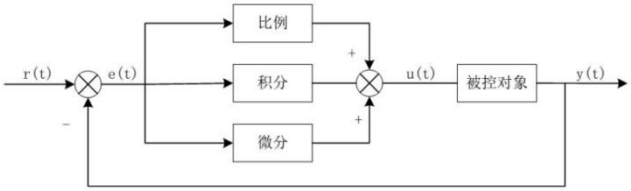


图5.2 PID控制示意图

5.4 识别任务处理

MobileNetV1 是一款专门应用于低算力设备的轻量化网络，旨在提高深度网络在有限硬件条件下的实时性能，故命名为MobileNet。研究表明MobileNet在ImageNet上的分类精度仅仅比 VGG－16低1% ，但是参数量却是VGG－16 的1/33。MobileNetV1最大的创新点是采用深度可分离卷积替代标准卷积，而且每层卷积后面都使用了BN层和ReLU激活函数，目的是加快网络训练同时避免反向传播时候的梯度消失和梯度爆炸。

第六章 开发工具说明

6.1 Visual Studio Code工具介绍

Visual Studio Code（简称“VS Code”）是Microsoft在2015年4月30日Build开发者大会上正式宣布一个运行于MacOSX、Windows和Linux之上的，针对于编写现代Web和云应用的跨平台源代码编辑器，可在桌面上运行，并且可用于Windows，MacOS和Linux。它具有对JavaScript，TypeScript和Node.js的内置支持，并具有丰富的其他语言（例如C++，C#，Java，Python，PHP，Go）和运行时（例如.NET和Unity）扩展的生态系统。

该编辑器也集成了所有一款现代编辑器所应该具备的特性，包括语法高亮（syntax high lighting），可定制的热键绑定（customizable keyboard bindings），括号匹配（bracket matching）以及代码片段收集（snippets）。Somasegar 也告诉笔者这款编辑器也拥有对Git的开箱即用的支持。Microsoft Docs（微软文档）提供了相应的学习教程帮助用户在Visual Studio Code中登陆GitHub。

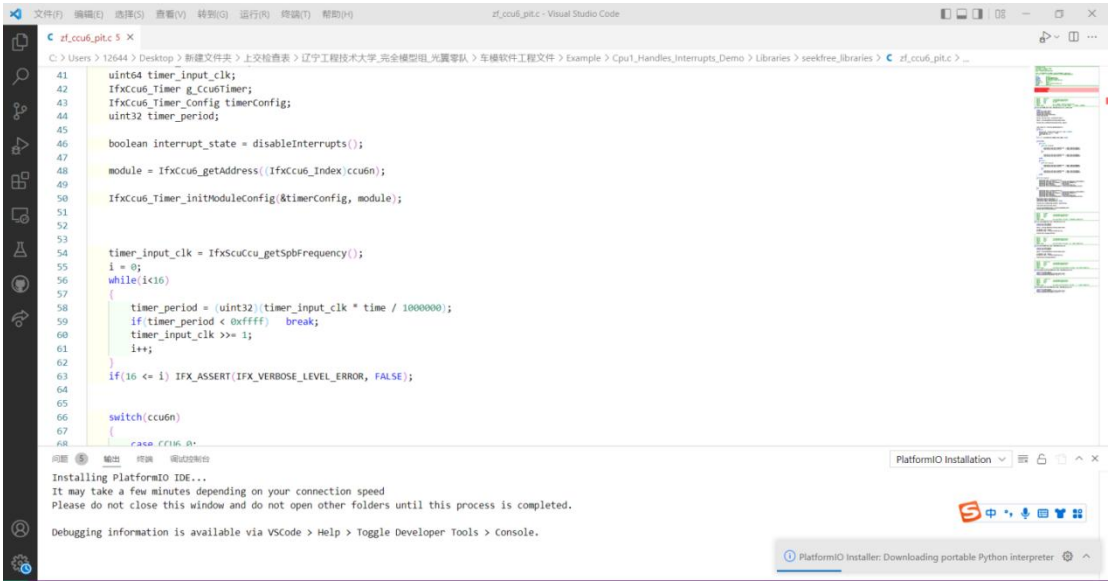


图6.1 Visual Studio Code平台示意图

6.2 AURIX Development Studio工具介绍

AURIX™ Development Studio（以下简称ADS）是英飞凌公司于2019年底推出的免费的集成开发环境，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。

AURIX Development Studio作为一款免费的编译器，配上下载器后可以进行debug，不过并没有用过debug，感觉debug没有看着代码用脑子想好用，如果经常写bug的话可以多用debug模式去追踪各个变量的变化情况和时序，毕竟多

核单片机的时序和单核有着很大的区别，其中断函数和单核并不是完全等价的概念。

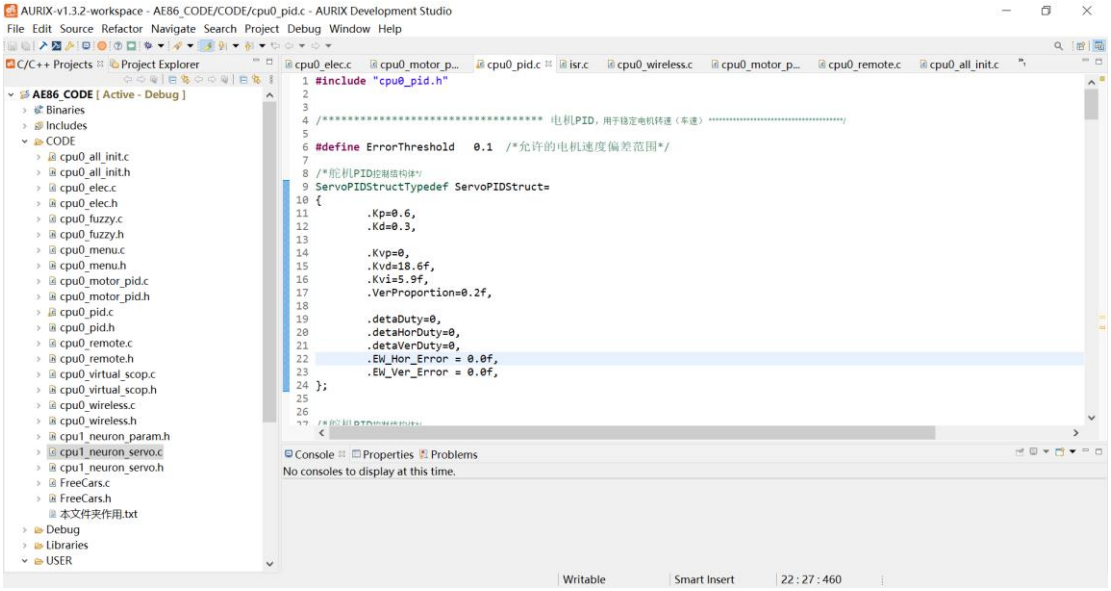


图6.2 AURIX Development Studio平台示意图

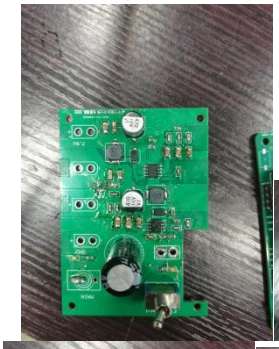
参 考 文 献

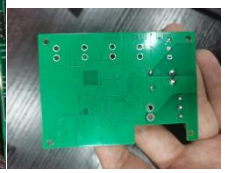
[1]于祥，《深度学习与飞浆 Paddle Paddle Fluid 实战》，人民邮电出版社，2019 .12 (1)
[2]刘靖宇,刘延飞,丛铭智,顾祎陆,何展博.一种基于EdgeBoard的无人驾驶交通信息识别系统设计[J].电子技术与软件工程,2020(23):199-201.
[3]泽德 A.肖，《笨方法学 python3》，人民邮电出版社，2018.6(3)
[4]Joseph Redmon, Ali Farhadi，YOLOv3: An Incremental Improvement，Computer Science，2018.4
[5]Joseph Redmon ,Ali Farhadi, YOLO9000: Better, Faster, Stronger, Computer Science，2016.12
[6]黄小平，王岩，《卡尔曼滤波原理及应用》，电子工业出版社，2017.1(1)
[7]Min Lin, Qiang Chen , Shuicheng Yan，Network In Network，In Proc. ICLR，2014.3
[8]Angela Fan，Edouard Grave，Armand Joulin，REDUCING TRANSFORMER DEPTH ON DEMAND WITH STRUCTURED DROPOUT，In Proc. ICLR，2019.9
[9]郑忠慧，车载全景视觉系统的研究与实现[D]，哈尔滨工业大学，2017.6

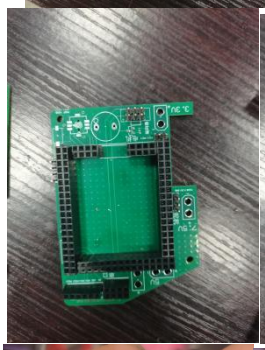
附录 A：主要技术参数

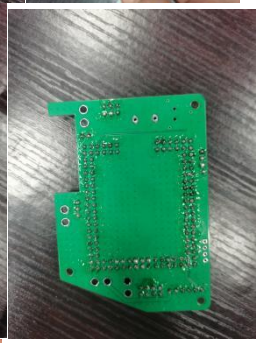
项目	参数
赛题组	完全模型组
车号	I车模
车模几何尺寸（长、宽、高）（毫米）	320*190*380
传感器种类及个数	摄像头×1、微型编码器×1、九轴陀螺仪模块
新增加伺服电机个数	0
摄像头检测精度	640×360

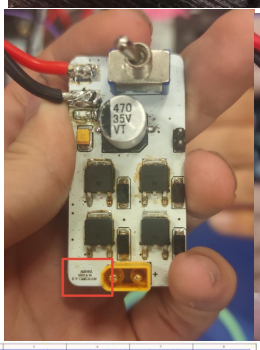
附录 B：原理图及PCB

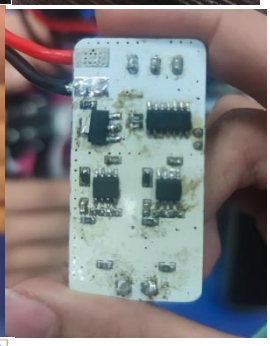


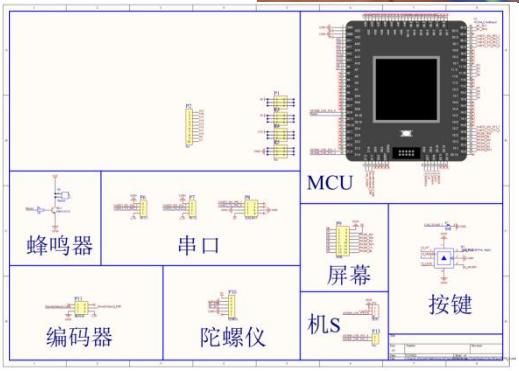


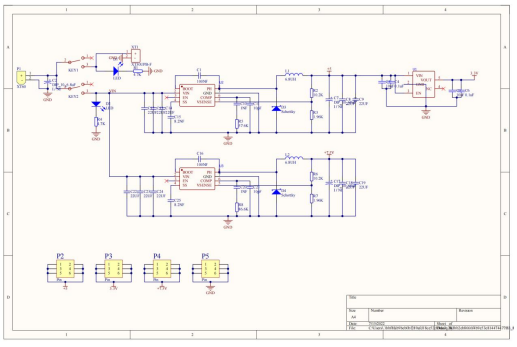












附录 C：部分代码

```c
#include "online_pid.h"
int online_pid_init(const char* dev) {
  edgeboard_recv_init();
  return 0;
}

uint8_t online_pid_menu(void) {
  int first, second;
  uint8_t type;
  printf("0. 调整电机 PID\n");
  printf("1. 调整舵机 PID\n");
  printf("2. 设置电机 Target\n");
  printf("3. 设置舵机 Current\n");
  printf("输入你选择的编号:");
  scanf("%d", &first);
  printf("0. 调整比例系数\n");
  printf("1. 调整积分系数\n");
  printf("2. 调整微分系数\n");
  printf("输入你选择的编号\n");
  scanf("%d", &second);

  // 选项有限不考虑溢出
  if (first < 0 && first > 3) {
    return 0;
  }
  if (second < 0 && second > 2) {
    return 0;
  }

  // 计算 type
  switch (first) {
    case 0:
      // 电机
      type = (uint8_t)0xC1 + second;
      break;
    case 1:
      type = (uint8_t)0xD1 + second;
      break;
    case 2:
```

```
      type = (uint8_t)0xB1;
      break;
    case 3:
      type = (uint8_t)0xB2;
      break;
  }
  return type;
}

float online_pid_check_input(void) {
  float data;
  char check;
  while (1) {
    printf("输入新的参数值:");
    scanf("%f", &data);
    printf("您输入的值为:%f\n", data);
    printf("请问是您刚才输入的值吗?\n 输入:y 确认数值");
    scanf("%c", &check);
    if (check == 'y') {
      return data;
    }
  }
}

void online_pid_send_data(int fd, uint8_t type, float data) {
  float_2_unit32_union data_union;

  switch (type) {
    case 0xB1:;
    case 0xB2:
      online_pid_send_data_uint16(fd, type, (int16_t)data);
      break;
    case 0xC1:;
    case 0xC2:;
    case 0xC3:;
    case 0xD1:;
    case 0xD2:;
    case 0xD3:
      data_union.f_data = data;
      online_pid_send_data_uint32(fd, type, data_union.u_data);
```

```
      break;
    default:
      break;
  }
}
void online_pid_send_data_uint32(int fd, uint8_t type, uint32_t data) {
  uint8_t send_buff[1 + 1 + 32 / 4 + 1] = {0};
  send_buff[0] = 0xA0;
  send_buff[10] = 0xF0;
  send_buff[1] = type;
  for (int i = 2; i < 2 + 8; i++) {
    send_buff[i] = data & 0xf;
    data >>= 4;
  }
  uart_send_buff(fd, send_buff, 1 + 1 + 32 / 4 + 1);
}
void online_pid_send_data_uint16(int fd, uint8_t type, uint16_t data) {
  uint8_t send_buff[1 + 1 + 16 / 4 + 1] = {0};
  send_buff[0] = 0xA0;
  send_buff[6] = 0xF0;
  send_buff[1] = type;
  for (int i = 2; i < 2 + 4; i++) {
    send_buff[i] = data & 0xf;
    data >>= 4;
  }
  uart_send_buff(fd, send_buff, 1 + 1 + 16 / 4 + 1);
}

float online_pid_paras_uint32_data(int fd) {
  uint8_t recv_data;
  uint32_t data = 0;
  uint8_t crc_array[8];
  for (int i = 0; i < 8; i++) {
    while (!uart_read_bit(fd, &recv_data))
      ;
    printf("%u\n", recv_data);
    crc_array[i] = recv_data;
    data |= recv_data << (i * 4);
  }
  // CRC16
```

```
  uint16_t crc16_data = crc16(crc_array, 8);
  uint16_t recv_crc16_data = 0;
  for (int i = 0; i < 4; i++) {
    while (!uart_read_bit(fd, &recv_data))
      ;
    recv_crc16_data |= recv_data << (i * 4);
  }
  if (recv_crc16_data != crc16_data) {
    return (float)-20000;
  }
  float_2_unit32_union data_union;
  data_union.u_data = data;
  return data_union.f_data;
}
float online_pid_paras_uint16_data(int fd) {
  uint8_t recv_data;
  uint32_t data = 0;
  uint8_t crc_array[4];
  for (int i = 0; i < 4; i++) {
    while (!uart_read_bit(fd, &recv_data))
      ;
    crc_array[i] = recv_data;
    data |= recv_data << (i * 4);
  }
  // CRC16
  uint16_t crc16_data = crc16(crc_array, 4);
  uint16_t recv_crc16_data = 0;
  for (int i = 0; i < 4; i++) {
    while (!uart_read_bit(fd, &recv_data))
      ;
    recv_crc16_data |= recv_data << (i * 4);
  }
  if (recv_crc16_data != crc16_data) {
    return (float)-20000;
  }
  int16_2_unit16_union data_union;
  data_union.u_data = data;
  return (float)data_union.data;
}
```

```
void online_pid_set_pid_params(int fd, pid* pid_data_array[]) {
  uint8_t read_data;
  if (uart_read_bit(fd, &read_data) && read_data == 0xA0) {
    // 读到头
    // gpio_set(P20_9, 1);
    while (!uart_read_bit(fd, &read_data))
      ;
    float temp;
    uint8_t type = read_data;
    switch (type) {
      case 0xB1:;
      case 0xB2:
        temp = online_pid_paras_uint16_data(fd);
        break;
      case 0xC1:;
      case 0xC2:;
      case 0xC3:;
      case 0xD1:;
      case 0xD2:;
      case 0xD3:
        temp = online_pid_paras_uint32_data(fd);
        break;
      default:
        return;
    };
    while (!uart_read_bit(fd, &read_data))
      ;
    // CRC16 校验不对退出程序的权宜之计
    if ((int)temp == -20000) {
      return;
    }

    if (read_data == 0xF0) {
      switch (type) {
        case 0xB1:
          // 设置电机 Target
          pid_data_array[0]->target = temp;
          break;
        case 0xB2:
          // 设置舵机 Current
```

```
            pid_data_array[1]->current = temp;
            break;
          case 0xC1:
            pid_data_array[0]->k_p = temp;
            break;
          case 0xD1:
            pid_data_array[1]->k_p = temp;
            break;
          case 0xC2:
            pid_data_array[0]->k_i = temp;
            break;
          case 0xD2:
            pid_data_array[1]->k_i = temp;
            break;
          case 0xC3:
            pid_data_array[0]->k_d = temp;
            break;
          case 0xD3:
            pid_data_array[1]->k_d = temp;
            break;
          default:
            return;
        }
      }
      // gpio_set(P20_9, 0);
    }
  }
```