

# 第十八届全国大学生智能汽车竞赛

## 技术报告



学 校： 兰州理工大学

队伍名称： 红柳狗熊岭

参赛队员： 郑嗣炜 谭森铭 黎森予 王丽霞

焦菲

带队教师： 林洁 张文旭

# 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： 郑嗣炜 谭森铭 黎森予 王丽霞 焦菲

带队教师签名： 林洁 张文旭

日 期： 2023年8月7日

## 摘 要

本文介绍了兰州理工大学红柳狗熊岭在第十八届全国大学生智能汽车竞赛完全模型组的详细制作方案。四轮车模为 I 车模，以 Paddlepi 作为上位机，以英飞凌公司生产的 TC264 作为下位机，实现了沿赛道寻线、维修区，断路区，谷仓区等任务，并追求高速稳定完成整个比赛。摄像头采用金乾象公司生产的 USB 全局曝光快门摄像头模组对赛道寻线，基于飞桨的 yolov3 模型对赛道上的 3D 模型进行检测并进入相应区域，通过编码器监测车的实时速度，使用 PID 控制算法调节电机的转速，实现了车模在运动过程中速度的闭环控制。

**关键词：**TC264, Edgeboard, I 车模、PID

## 目录

第一章 引言	3
1.1 智能车大赛介绍	3
1.2 第十八届完全模型组规则介绍	3
第二章 机械结构设计	5
2.1 整体布局	5
2.2 舵机安装	5
2.3 前轮倾角调节	5
2.3.1 前轮外倾	6
2.3.2 前轮前束	6
2.4 悬挂调节	6
2.5 陀螺仪	6
2.6 图像获取	6
2.7 车壳设计	7
2.8 车模主要技术参数	7
第三章 硬件电路的设计	8
3.1 电源方案	8
3.2 驱动方案	9
第四章 软件方案	11
4.1 图像处理方案	11
4.2 基于 YOLOV3 的模型识别	11
4.3 PID 运动控制算法	11
4.3.1 位置式 PID	13
4.3.2 增量式 PID	13
4.3.3 PID 参数整定	14
4.4 舵机保护与控制	14
4.5 电机 PI 控制	14
第五章 上位机与下位机开发环境	16
5.1 EdgeBoard 的开发环境	16
5.2 下位机开发环境	16
第六章 总结	17
6.1 车模总结	17
6.1.1 控制方面	17
6.1.2 结构方面	17
6.1.3 不足之处	17
6.2 工作综述	17
致谢	18
参考文献	19
附录	22

一、车模技术检查表 .....	22
二、车模照片 .....	25
1、车模外观照片 .....	25
2、电路板 PCB 图 .....	26
3、电路板原理图 .....	29

## 第一章 引言

### 1.1 智能车大赛介绍

全国大学生智能车竞赛受教育部高等教育司委托，由教育部高等自动化专业教学指导分委员会（以下简称自动化分教指委）主办全国大学生智能汽车竞赛。该竞赛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，为加强大学生实践、创新能力和团队精神的培养，促进高等教育教学改革。该竞赛以“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，培养大学生的综合知识运用能力、基本工程实践能力和创新意识，激发大学生从事科学研究与探索的兴趣和潜能，倡导理论联系实际、求真务实的学风和团队协作的人文精神，为优秀人才的脱颖而出创造条件。该竞赛由竞赛秘书处为各参赛队提供/购置规定范围内的标准硬软件技术平台，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。该竞赛融科学性、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。该竞赛规则透明，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。

### 1.2 第十八届完全模型组规则介绍

#### （1）赛道环境

完全模型组比赛赛道以室内循环赛道为基础，赛道材质，赛道规格均保持一致。在导引方式上完全保留室内循环赛道的导引方式，并在此基础上添加完全模型组任务导引标志和锥桶，引导车模完成完全模型组赛道任务。

#### （2）任务描述

完全模型组的基本比赛任务为：

本届完全模型组场地为智慧农业场景，模拟智能车辆在收货季节的运行和作业，车辆不但需要自主驶过常规道路，还要驶入特殊区域，如车辆维护区、农田区、动物出没区、坡道、粮仓区。农业场景下智能车辆会出现穿过农田区、驶过动物出没区、经过上下坡道等动作。农用车辆处于长时间高负荷工作状态，

需要频繁的进行维护和保养，通过车辆维护区时要求车辆停入指定区域，并有片刻的停车动作，完成车辆维护过程。满载丰收粮食的智能车辆需驶入粮仓，根据入口处的指示，到达粮仓内部，并从指定的出口驶出。

选手制作的车模完成从车库出发沿着赛道运行两周。车模需要分别通过道路设置的各种元素,识别道路旁的标志完成特殊路段通行。

比赛时间从车模驶出车库到重新回到车库为止。如果车模没有能够停止在车库内停车区内，比赛时间加罚 5 秒钟。对于未完成的任务会通过相应的加罚时间叠加在比赛时间上。

以下特殊路段通过失败时发车手可手动援救，重新将车辆置于特殊路段之后启动，该路段视为通过失败，按规定罚时，且计时不停止；通过各区域时明显碰撞锥桶视为破坏农田，每次罚时 5 秒。

## 第二章 机械结构设计

本章主要讲述我们的机械结构。

### 2.1 整体布局

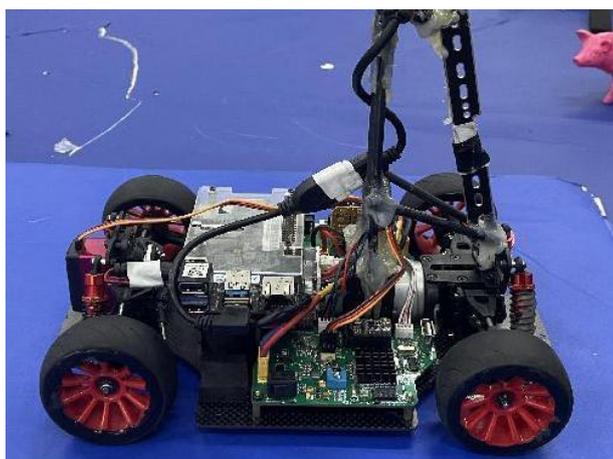


图 2.1 车模整体结构

为了保证车模重心，主控与电源驱动安装在车体两侧，edgeboard 居中放置

### 2.2 舵机安装

舵机转向是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，需要对舵机安装方式和位置进行研究和改进。通过分析舵机控制转向的原理可知，在相同转向条件下，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。这相当于增大力臂长度，提高线速度。

舵机安装方式有立式和卧式两种，通过比较得立式安装效果更优。舵机安装时须保证左右对称，以保证舵机左右转向时力臂相等且处于最大范围，提升舵机响应速度。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，故安装时须考虑到转向机构的响应速度与舵机扭矩之间的关系，获得最佳转向效果。

### 2.3 前轮倾角调节

为使汽车直线行驶稳定，转向轻便，转向后能自动回正，减少轮胎和转向

系零件的磨损等，在转向轮、转向节和前轴之间须形成一定的相对安装位置，叫车轮定位。本系统所采用的智能车通过四只轮胎与地面接触，后轮同轴受到限位，无法调整，与智能车的前进方向保持平行，因此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，只能通过调整前轮倾角各定位参数来实现。它的安装位置由前轮外倾和前轮前束两项目决定。

### 2.3.1 前轮外倾

前轮外倾是指前轮安装后，其上端向外倾斜，于是前轮的旋转平面与纵向垂直平面间形成一个夹角，称之为前轮外倾角，其主要作用是使转向轻便，使车轮紧靠轮毂内轴承，以减少外轴承及轮毂螺母的负荷，有利于安全行驶。一般前轮外倾角为 $1^{\circ}$ 左右，但对于有高速、急转向要求的车辆，前轮外倾角可减小甚至为负值。

### 2.3.2 前轮前束

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。前轮在滚动时，其惯性力自然将轮胎向内偏斜，如果前束适当，轮胎滚动时的偏斜方向就会抵消，轮胎内外侧磨损的现象会减少。

## 2.4 悬挂调节

通过调节悬挂，调整车模底板尽可能低，并增加配重，防止翻车。

## 2.5 陀螺仪

我们选用了逐飞科技公司的 ICM-20602 模块作为姿态传感器。并通过自定义帧将数据传到上位机。

## 2.6 图像获取

我们使用了一个金乾象公司生产的 100 帧摄像头，经过长时间的尝试，既能保证寻线的准确性，又可以保证模型的识别率。摄像头通过螺丝和 AB 胶固定在碳杆上，以确保车模在运行过程中摄像头的稳定性。

## 2.7 车壳设计



车壳在原厂车模车壳的基础上通过钻头锉刀进行精细加工，再使用中国象棋上拆下的合页和实验室随处可见的洞洞板形成电池仓门，方便拿取电池。

## 2.8 车模主要技术参数

智能车主要技术参数包括物理尺寸、电路指标等，具体参数见表 2.1。

表 2.1 智能车主要技术参数

模型车基本参数	长	32 cm
	宽	18 cm
	高	38 cm
	车重	2216g
电池容量		2300mah
传感器	编码器	1 个
	彩色摄像头	1 个
	陀螺仪	1 个
除了车模原有的驱动电机、舵机之外伺服电机个数		0 个
微控制器型号和数量	TC264	1 个

### 第三章 硬件电路的设计

这辆小车电路部分主要的模块包括：电源管理模块、传感器模块、驱动模块以及其他调试模块。各模块的总体设计原则是：稳定可靠且符合车模自身结构。

#### 3.1 电源方案

电源设计的首要目标是电源的可靠性，整个下位机和上位机的工作完全由电源供电的可靠性决定，电源供电不稳定会引起损耗、单片机复位、传感器损毁等严重问题。特别是舵机电源，为保证舵机的快速响应我们采用了 2236 进行供电，过流达 6A。

我们采用格氏 3S 锂离子动力电池为整个系统供电，电池电压保持在 11.7~12.6V，电池连接在电源驱动板上通过三个接口分别向电机、Edgeboard 板卡、主控板供电，为了避免电机对电源造成影响，我们为其分割了多个地并通过逻辑门进行隔离防止损害单片机。

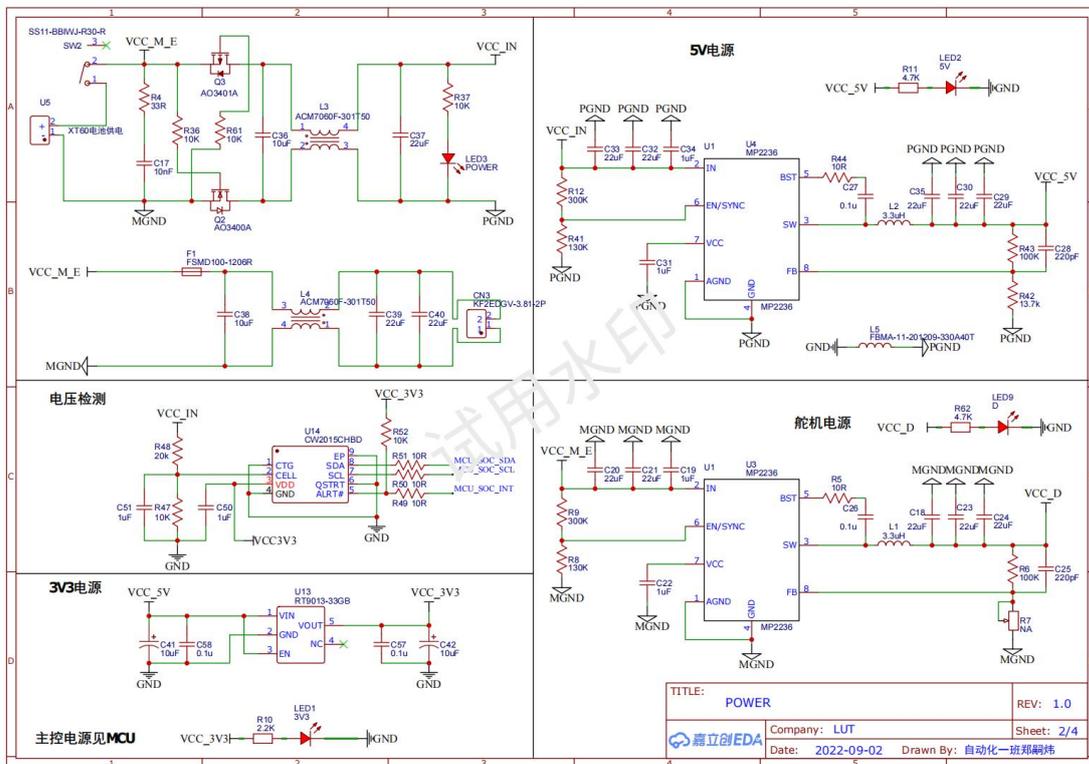


图 3.1 电源管理原理图

### 3.2 驱动方案

IR2104+MOS 管搭建的电机驱动电路，足以满足我们对于电机控制的要求，我们选用 HYG038N03 国产 MOS 作为驱动用的 MOS 管，实现转速闭环的需求，下面是电路原理图。

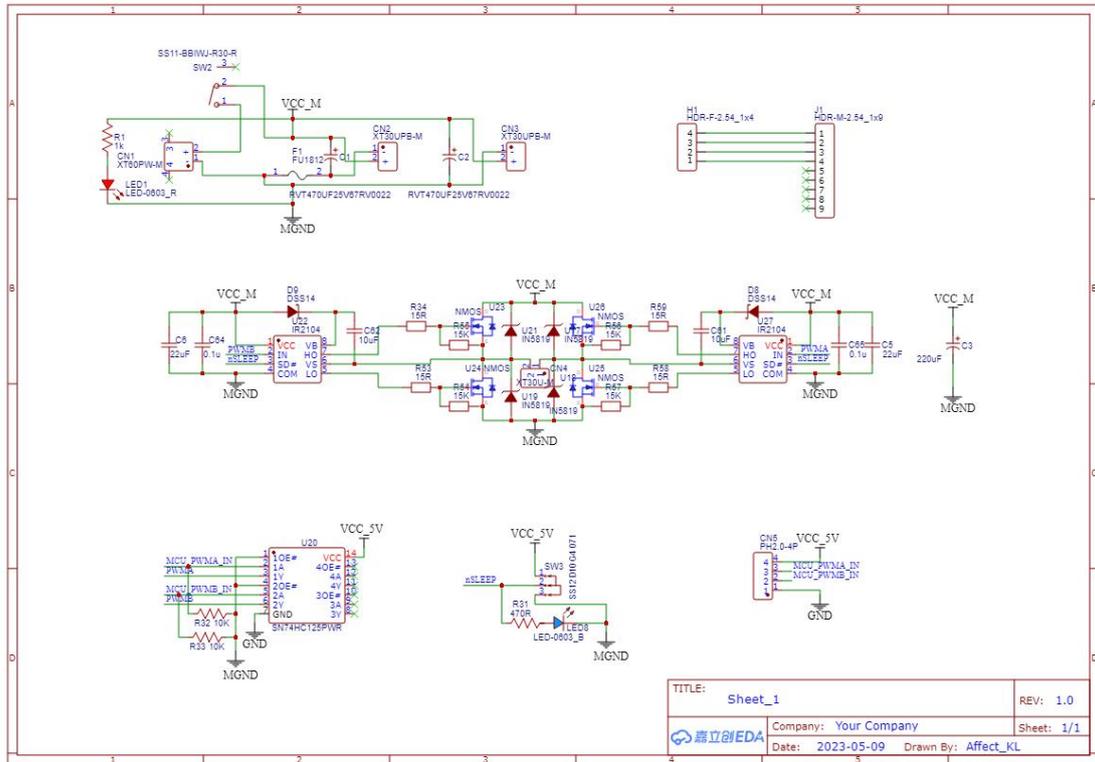


图 3.2 电机驱动电路

### 3.3 下位机方案

车模的控制部分由下位机上 TC264 英飞凌单片机完成，下位机软件设计为前后台系统，并通过串口中断将外设控制与实时性强的舵机控制分离开来，实现下位机稳定运行。

# 第十八届全国大学生智能汽车邀请赛技术报告

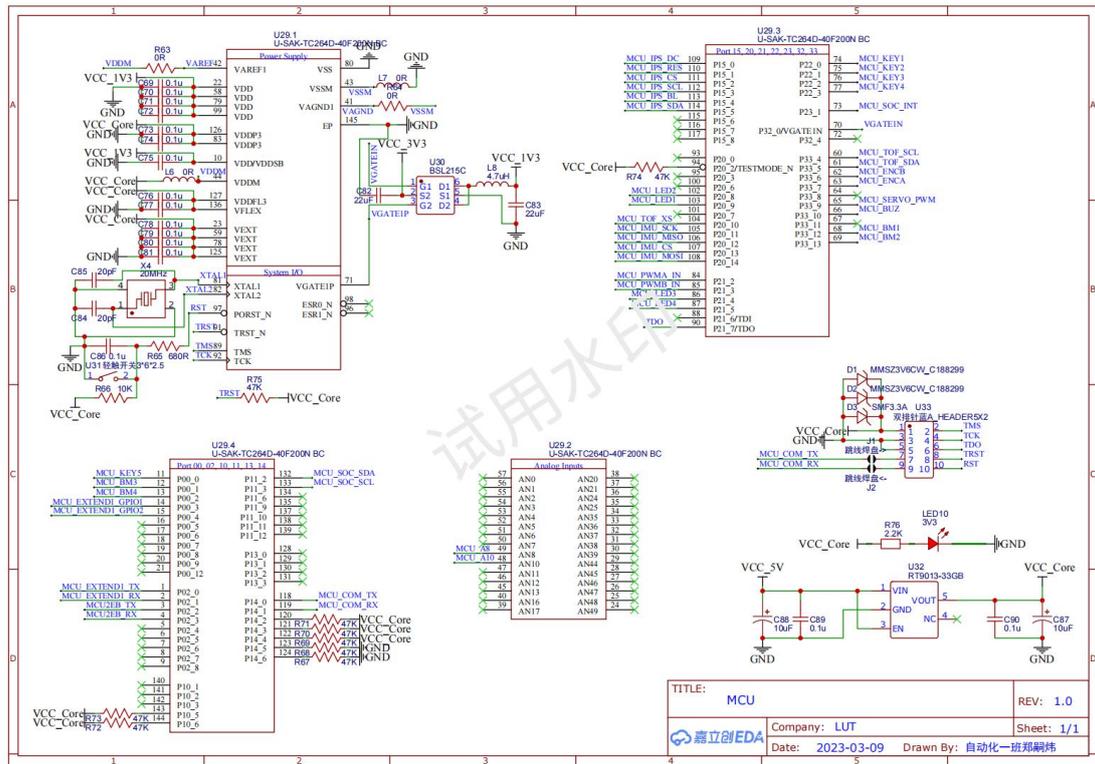


图 3.3 TC264 电路

## 第四章 软件方案

### 4.1 图像处理方案

智能车采用的 170 度 RGB 摄像头。在 Paddlepi 对图像进行采集后对其进行处理，以提取主干赛道中心线。考虑到 Paddlepi 的强大算力，本款程序基于 openCV，通过大津法进行二值化、迷宫法提取边线，较传统巡线算法更稳定提取出赛道边界，之后通过逆透视，再通过左右平移边线的方法来获得中线，并使用逆透视后的图像进行元素识别。

### 4.2 基于 YOLOV3 的模型识别

首先通过前摄像头采集的图像进行图像处理，来判断前方是否存在标志。如果前方发现存在标志，将采集的图片进行剪裁后传给模型，利用模型去区分谷仓，锥桶，拖拉机等。

通过 ai studio 平台训练对比了 ssd 与 yolov3 模型后，我们发现经过裁剪后 yoloV3\_moblient\_v3 在速度，准确率方面均表现优良，因而最终采用该模型进行标志的识别。同时为了加大模型识别的准确率，我们增广了数据集，达到了 8000 张图片。

### 4.3 PID 运动控制算法

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。

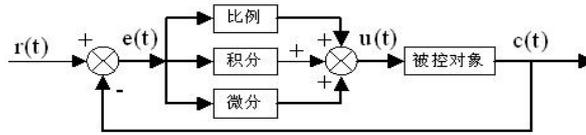


图 4.3 PID 控制器原理框图

在计算机控制系统中，使用的是数字 PID 控制器，控制规律为：

$$e(k) = r(k) - c(k) \quad (\text{公式 5.1})$$

$$u(k) = K_p \left\{ e(k) + \frac{T}{T_i} \sum_{j=0}^k e(j) + \frac{T_D}{T} [e(k) - e(k-1)] \right\} \quad (\text{公式 5.2})$$

式中

- |                                 |                         |
|---------------------------------|-------------------------|
| k——采样序号， $k = 0, 1, 2, \dots$ ; | $r(k)$ ——第 $k$ 次给定值；    |
| $c(k)$ ——第 $k$ 次实际输出值；          | $u(k)$ ——第 $k$ 次输出控制量；  |
| $e(k)$ ——第 $k$ 次偏差；             | $e(k-1)$ ——第 $k-1$ 次偏差； |
| $K_p$ ——比例系数；                   | $T_i$ ——积分时间常数；         |
| $T_D$ ——微分时间常数；                 | $T$ ——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：**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：**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：**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。

### 4.3.1 位置式 PID

位置式 PID 中, 由于计算机输出的  $u(k)$  直接去控制执行机构(如阀门),  $u(k)$  的值和执行机构的位置(如阀门开度)是一一对应的, 所以通常称公式(4.2)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。因而产生了增量式 PID 控制的控制算法, 所谓增量式 PID 是指数字控制器的输出只是控制量的增量  $\Delta u(k)$ 。

### 4.3.2 增量式 PID

当执行机构需要的是控制量的增量(例如: 驱动步进电机)时, 可由式(5.2)

推导出提供增量的 PID 控制算式。由式(5.2)可以推出式(5.3), 式(5.2)减去式(5.3)可得式(5.4)。

$$u(k-1) = K_p \left\{ e(k-1) + \frac{T}{T_i} \sum_{j=0}^{k-1} e(j) + \frac{T_D}{T} [e(k-1) - e(k-2)] \right\} \dots\dots\dots 5.3$$

$$\begin{aligned} \Delta u(k) &= K_p \left\{ [e(k) - e(k-1)] + \frac{T}{T_i} e(k) + \frac{T_D}{T} [e(k) - 2e(k-1) + e(k-2)] \right\} \dots\dots\dots 5.4 \\ &= K_p \Delta e(k) + K_I e(k) + K_D [\Delta e(k) - \Delta e(k-1)] \end{aligned}$$

$$\text{式中 } \Delta e(k) = e(k) - e(k-1); \quad K_I = K_p \frac{T}{T_i}; \quad K_D = K_p \frac{T_D}{T}$$

公式(5.4)称为增量式 PID 控制算法, 可以看出由于一般计算机控制系统采用恒定的采样周期  $T$ , 一旦确定了  $K_P$ 、 $K_I$ 、 $K_D$ , 只要使用前后三次测量值的偏差即可由式(5.4)求出控制增量。

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，我们选择带死区、积分分离等改进 PID 控制算法。

### 4.3.3 PID 参数整定

运用 PID 控制的关键是调整 KP、KI、KD 三个参数，即参数整定。PID 参数的整定方法有两大类：一是理论计算整定法。它主要是依据系统的数学模型，经过理论计算确定控制器参数；二是工程整定方法，它主要依赖工程经验，直接在控制系统的试验中进行，且方法简单、易于掌握，在工程实际中被广泛采用。由于智能车系统是机电高耦合的分布式系统，并且要考虑赛道的具体环境，要建立精确的智能车运动控制数学模型有一定难度，而且我们对车身机械结构经常进行修正，模型参数变化较为频繁，理论计算整定法可操作性不强，最终我们采用了工程整定的方法。

## 4.4 舵机保护与控制

为了保护舵机，第一步要做的就是确定舵机左右极限值，从软件上进行限制，防止舵机堵转烧毁舵机，随后初略设置舵机中值，车辆放置于直道行进，加大 KP 参数，根据小车路线调整，确定中值。在行驶中，通过模糊 pid 将巡线偏差通过 usb 协议发送给下位机控制舵机。模糊 pid 控制效果好，故在舵机控制上选用模糊 pid

## 4.5 电机 PI 控制

由于使用了 2104 方案，所以需要两路 PWM 信号的占空比来控制电机的转速，通过编码器采样得到电机的实际转速，随后通过上位机软件收到的编码器数据来进行调整，实际可以略微超调一点调的硬一点，我们的方法是若车模能以 0.1 米的速度上坡则能够满足我们的需求。

## 第五章 上位机与下位机开发环境

在系统的设计制作和调试的过程中,不管是软件的开发还是硬件电路的仿真和电路板的制作都离不开 PC 机。所以对于 PC 机上的各种辅助设计软件必须要有一定的熟悉程度。这样可以提高开发的效率。

### 5.1 EdgeBoard 的开发环境

我们主要采用了 Xming 以及 Visual Studio Code 来进行 Edgeboard 的调试。

Xming 是一个免费的 X Window 服务器软件,用于在 Windows 操作系统上运行 Linux/Unix 的图形应用程序。它允许你在 Windows 中远程连接到 Linux/Unix 服务器,并在本地窗口中显示和操作图形界面应用程序。Xming 的目的是提供一个轻量级、高性能和易于使用的 X 服务器解决方案,使用户能够在 Windows 系统上运行基于 X Window 系统的软件。

Visual Studio Code 是一款由微软开发的免费的源代码编辑器。它可运行在 Windows、macOS 和 Linux 平台上,并且提供了丰富的功能,旨在提高开发人员的生产力和编写代码的体验。

VS Code 可以用于编辑各种编程语言,包括但不限于 JavaScript、TypeScript、Python、C#、Java 等,并且支持代码高亮、智能代码补全、调试、版本控制等功能。它还可以通过扩展进行功能扩展,使其支持更多的语言和工具。

由于其轻量级、快速启动和响应的特点,以及强大的功能和可扩展性,VS Code 在开发人员社区中非常受欢迎,并被广泛使用。此外,我们还应用了外接显示屏显图,可以用来调试图像。

### 5.2. 下位机开发环境

运用 AURIX Development Studio 作为软件开发工具。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、编译及调用代码没有时间及代码大的限制。调试界面如图所示:

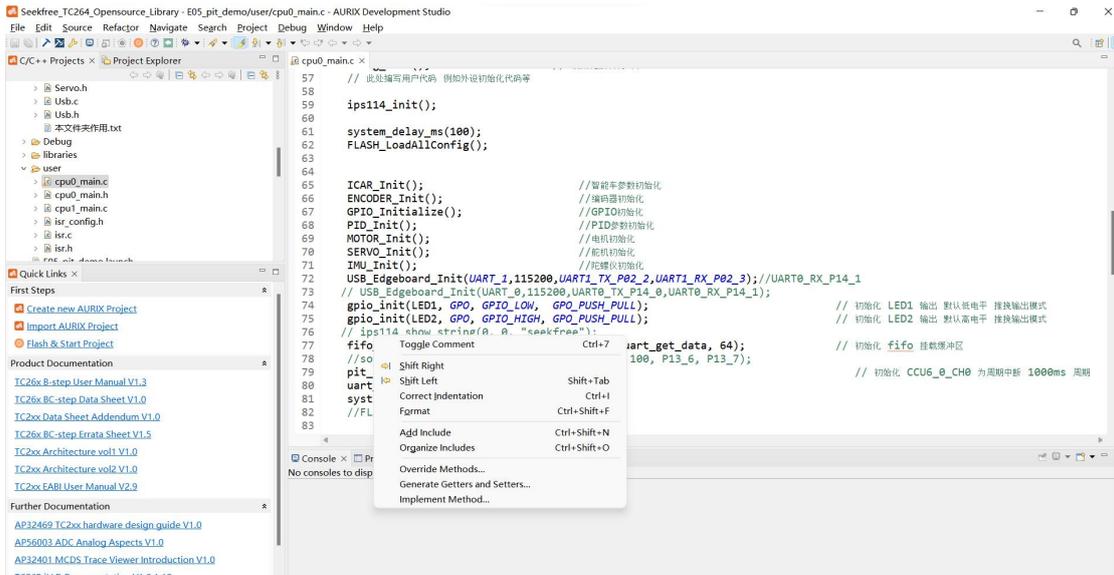


图 5.2 调试界面

## 第六章 总结

### 6.1 车模总结

#### 6.1.1 控制方面

采用模糊 PID 进行速度和转向控制，利用摄像头进行寻线以及模型识别。

#### 6.1.2 结构方面

尽力做到了结构的合理性与简洁性，将车模的重量和重心尽可能合理化。

电路板均为自己设计与焊接，完成了供电，采集，控制等各项任务。

#### 6.1.3 不足之处

- I 车模的结构具有巨大的优化空间，底盘太轻极易翻车。
- 差速器公差略大，啮合效果不好，有异响。
- 舵机烧毁问题，经过拆解，赛曙 cs-3120 舵机烧毁的原因大多是舵机控制板上的 pmos apm4953 烧毁，查阅芯片手册发现该 mos 过流只有 5.1A，在高速连续打脚下极易烧毁。希望厂商进行改进。

### 6.2 工作综述

在方案选择阶段，我们研究了不同的策略和技术，评估了其优缺点并进行了充分的讨论。然后，我们进行了方案的改进和重新设计，考虑了性能、可行性和可靠性等因素。在实施方案的过程中，我们进行了参数的微调和优化，以确保系

统的稳定性和精确性。

同时，我们还进行了机械和硬件制作工作。这涵盖了底盘结构的设计与制作、电路板的设计与焊接、传感器的安装与调试等。我们深入了解了机械和电子器件之间的相互配合，学习了如何进行装配和调整，以使整个系统能够协同工作。

另外，我们进行了软件编写工作。我们使用了多种编程语言和开发工具，针对不同的功能和模块进行了软件设计与开发。涉及到的任务包括图像处理、路径规划、控制算法实现等。通过编写代码，我们加深了对编程原理和软件开发的理解，并学会了解决实际问题的能力。

通过参加智能车大赛，我们获得了宝贵的经验和技能。我们学会了将不同学科的知识融合应用到实际项目中，提升了问题解决和创新能力。此外，我们的团队合作和沟通能力得到了提升，我们也更加深刻地认识到每个细节的重要性的影响。

我们衷心感谢智能车大赛组委会为我们提供了这样一个锻炼和展示自我的平台。同时，我们也感谢老师们的指导和支持。我们将继续努力，不断提升自己，在今后的工作和学习中取得更进一步的进展。

## 致谢

在备赛第十八届全国大学生智能汽车大赛的过程中，首先，我们要向我们的指导老师林浩老师和张文旭老师表达最深的敬意和感谢。我们能有机会站在国赛的舞台上离不开老师们在背后的奉献。

同时，我们要向电信学院的领导们以及组委会的老师表示感谢，他们的支持和鼓励使我们能够充满信心地完成这份报告。他们为我们提供了宝贵的资源和帮助，让我们能够在学术上不断进步和成长。

此外，我们要向我们的队内队友表示最深的感谢。在过去的五个月里，他们陪伴我们一起度过了无数个日日夜夜，一起经历了挑战和困难，一起分享了成功和喜悦。他们的友谊和支持是我们最大的动力和支撑，他们的鼓励和支持使我们能够克服困难。这段在 203 的日子将成为我们每个人一段珍贵的回忆，我们会永远珍惜这份情谊。

感谢一路走来所有人的支持和帮助。

## 参考文献

- [1] 卓晴.第十七届全国大学智能汽车竞赛竞速比赛规则[EB/OL].  
<https://zhuoqing.blog.csdn.net/article/details/121187014>
- [2] 王盼宝. 智能车制作: 从元器件、机电系统、控制算法到完整的智能车设计[M]. 北京: 清华大学出版社, 2017.
- [3] 王淑娟, 蔡惟铮, 齐明. 模拟电子技术基础 [M]. 北京: 高等教育出版社. 2009
- [4] 卓晴, 黄开胜, 邵贝贝. 学做智能车 [M]. 北京: 北京航空航天大学出版社. 2007.
- [5] 哈尔滨工业大学紫丁香十队第十七届全国大学生智能车大赛技术报告

## 附录

### 主要程序:

```

#include "isr_config.h"
#include "zf_common_headfile.h"
#pragma section all "cpu0_dsram"
#define PIT_NUM (CCU60_CH0 )
// 使用的周期中断编号
#define LED1 (P20_9)
#define LED2 (P20_8)
uint8 pit_state = 0;
extern uint8 uart_get_data[64];
// 串口接收数据缓冲区
extern uint8 fifo_get_data[64];
// fifo 输出读出缓冲区
extern uint32 fifo_data_count;
extern uint8 get_data;
// 接收数据变量
extern uint32 fifo_data_count0;
// fifo 数据个数
extern fifo_struct uart_data_fifo;
int core0_main(void)
{
    clock_init(); // 获取时钟频率<务必保留>
    debug_init(); // 初始化默认调试串口
    ips114_init();
    system_delay_ms(100);
    FLASH_LoadAllConfig();
    ICAR_Init(); //智能车参数初始化
    ENCODER_Init(); //编码器初始化
    GPIO_Initialize(); //GPIO 初始化
    PID_Init(); //PID 参数初始化
    MOTOR_Init(); //电机初始化
    SERVO_Init(); //舵机初始化
    IMU_Init(); //陀螺仪初始化

    USB_Edgeboard_Init(UART_1,115200,UART1_TX_P02_2,UART1_RX_P02_3)
;
    gpio_init(LED1, GPO, GPIO_LOW, GPO_PUSH_PULL);
// 初始化 LED1 输出 默认低电平 推挽输出模式
    gpio_init(LED2, GPO, GPIO_HIGH, GPO_PUSH_PULL);
// 初始化 LED2 输出 默认高电平 推挽输出模式

```

```

    fifo_init(&uart_data_fifo, FIFO_DATA_8BIT, uart_get_data,
64);           // 初始化 fifo 挂载缓冲区
    pit_us_init(CCU60_CH0, 1000);
// 初始化 CCU6_0_CH0 为周期中断 1000ms 周期
    uart_rx_interrupt(UART_1,1);
    system_delay_ms(100);
    cpu_wait_event_ready();           // 等待所有核心初始化完毕
    GPIO_BuzzerEnable(BuzzerOk); //开机音效
    while (TRUE)
    {
        USB_Edgeboard_Handle();           //USB 通信控制
        GPIO_Handle();                   //GPIO 控制: LED/蜂鸣
        ICAR_Handle();                   //智能车控制
        FLASH_Handle();                   //Flash 存储
        IMU_Handle();
    }
}

IFX_INTERRUPT(cc60_pit_ch0_isr, 0, CCU6_0_CH0_ISR_PRIORITY)
{
    interrupt_global_enable();           // 开启中断嵌
套
    pit_clear_flag(CCU60_CH0);
    USB_Edgeboard_Timr(); //USB 通信线程
    GPIO_Timer();         //GPIO 外设线程
    MOTOR_Timer();       //电机控制线程
    IMU_Timer();
    ICAR_Timer();        //智能车综合处理线程计数器
    pit_state = 1;
}

#pragma section all restore

```

## § 03 附录

## 一、车模技术检查表

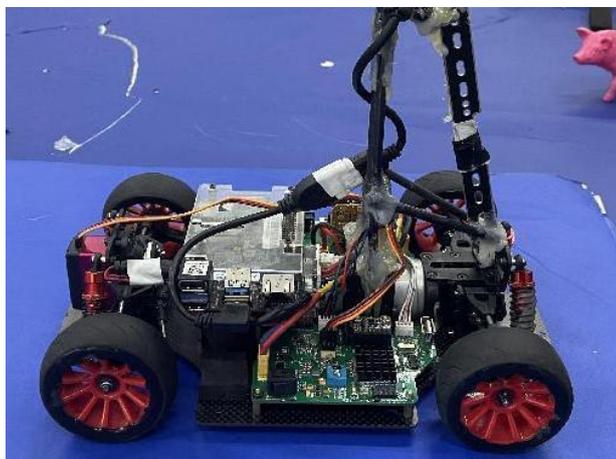
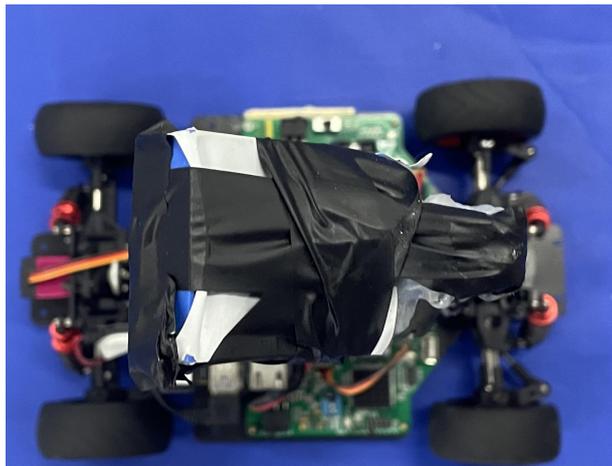
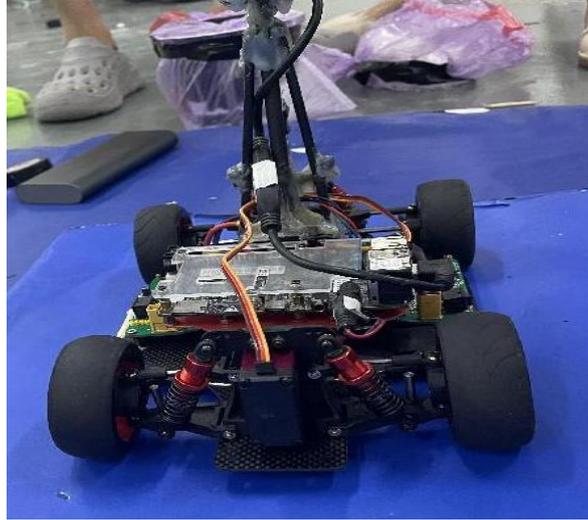
队伍名称	红柳狗熊岭			
参赛学校	兰州理工大学			
赛题组组别	完全模型组			
检查项目	规格 (选手自行填写)	符合 (√)	不符合 (×)	备注
1. 车模类型是什么?	I 车模			如果是自制车模, 请标明自制。
车模整体尺寸: 1. (包括传感器在内) 长, 宽, 高(mm) 2. 摄像头组标明镜头距离地面高度。	长 33cm 宽 19cm 高 38cm 摄像头高度 38cm			在填写是, 请将所在组别规则对于车模尺寸限制同时进行填写。
1. 传感器种类、规格(型号)数量。 2. 是否使用自带MCU的成品传感器模块? 型号是什么?	1. icm20602 六轴陀螺仪 2. I 车模编码器  否			
1. 控制转向舵机型号是否自行改装舵机? 2. 防伪易损标签是否完整?	CS-3120 否  是			
1. 是否增加伺服电机? 2. 如果有那么种类、个数和作用?	否			

1. 电路中微处理器型号和个数?	英飞凌 TC264*1 个			
1. 是否具有其它可编程器件, 个数与作用?	Edgeboard 计算卡			
1. 是否有无线通讯装置? 2. 如果有, 那么种类和个数?	是  无线网卡 1 个			
1. 电池的种类、规格和数量?	3s 2300mah			
1. 是否使用 GPS 导航? 2. 是否没有使用 RTK?	否			
1. 后轮驱动电机是否是原车模电机? 2. 是否具有防伪易损标签?	是  是			
1. 车模轮胎是否原有的纹理可辨析? 2. 轮胎表面是否具有粘性物质? 3. 对于麦克纳姆轮是否更换过小轮胶皮?	是  否			
1. 车模底盘是否是原车模底盘? 2. 是否有大面积切割?	是  否			
1. 车轮轴距、轮距是否改装? 2. 改装参数是什么?	否			
1. 车模驱动轮传动机构是否改装?	否			

2. 改装方式是什么?				
1. 车模差速器是否改装? 2. 改装方式是什么?	否			
1. 是否更换过原装车模中的机械元器件? 更换后的规格 是什么?	否			。
1. 车模电路板个数及功能。 2. 其中是否有购买成品、哪一些?	3 个 驱动板, 控制板, 与 Edgeboard 计算卡 Edgeboard 计算卡			
1. 自制电路板是否标记有学校名称、队伍名称、制作日期等信息? 2. 标示信息在 PCB 的哪一层?	是  正面覆铜层			请在表格中注明电路板队伍信息的内容。
其它待说明内容				
检查人员签名:	检查意见:			

## 二、车模照片

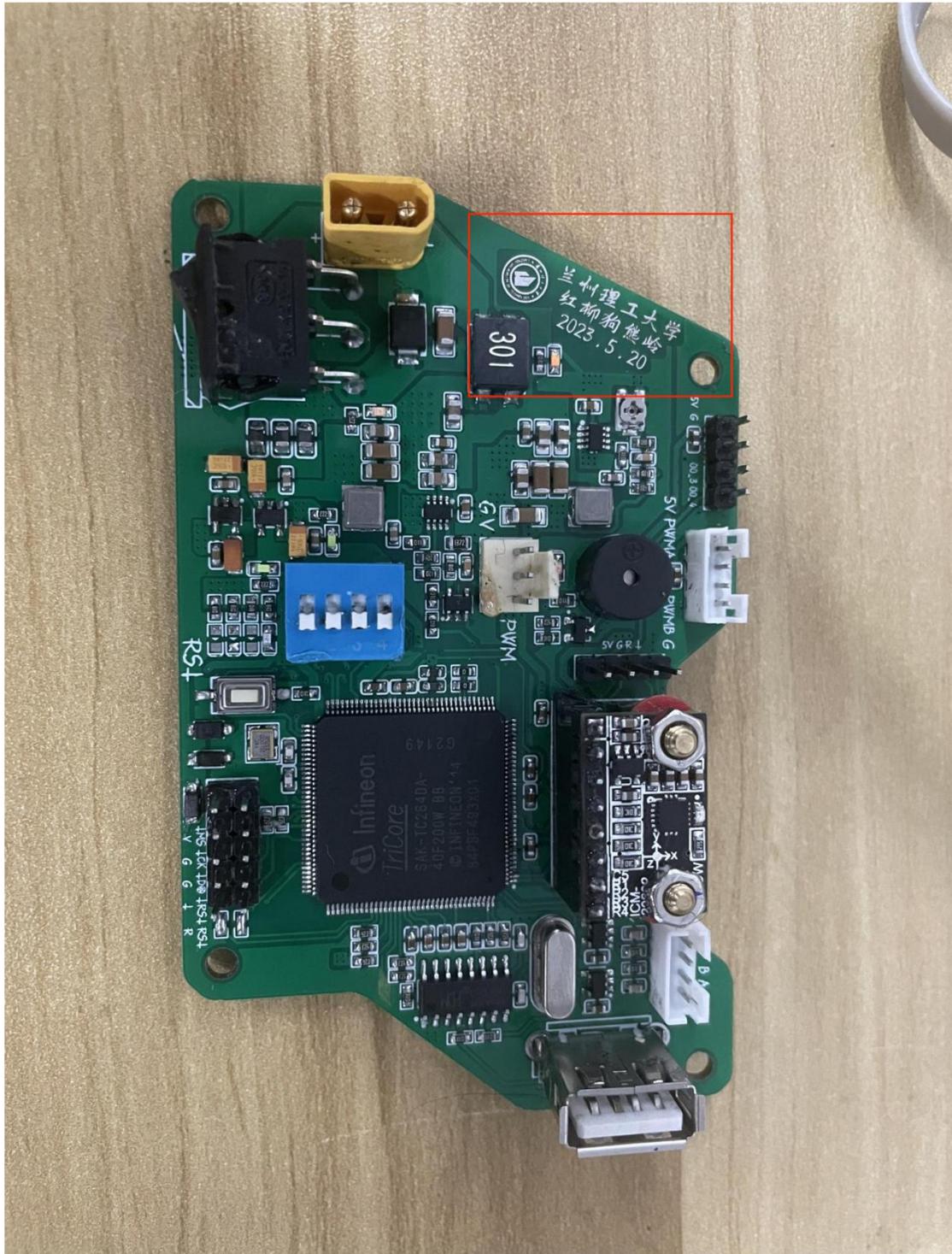
### 1、车模外观照片



## 2、电路板 PCB 图

车模中所有电路板正反面照片。对于自制电路板，需要使用红色方框标出队伍独自 LOGO 所在处。

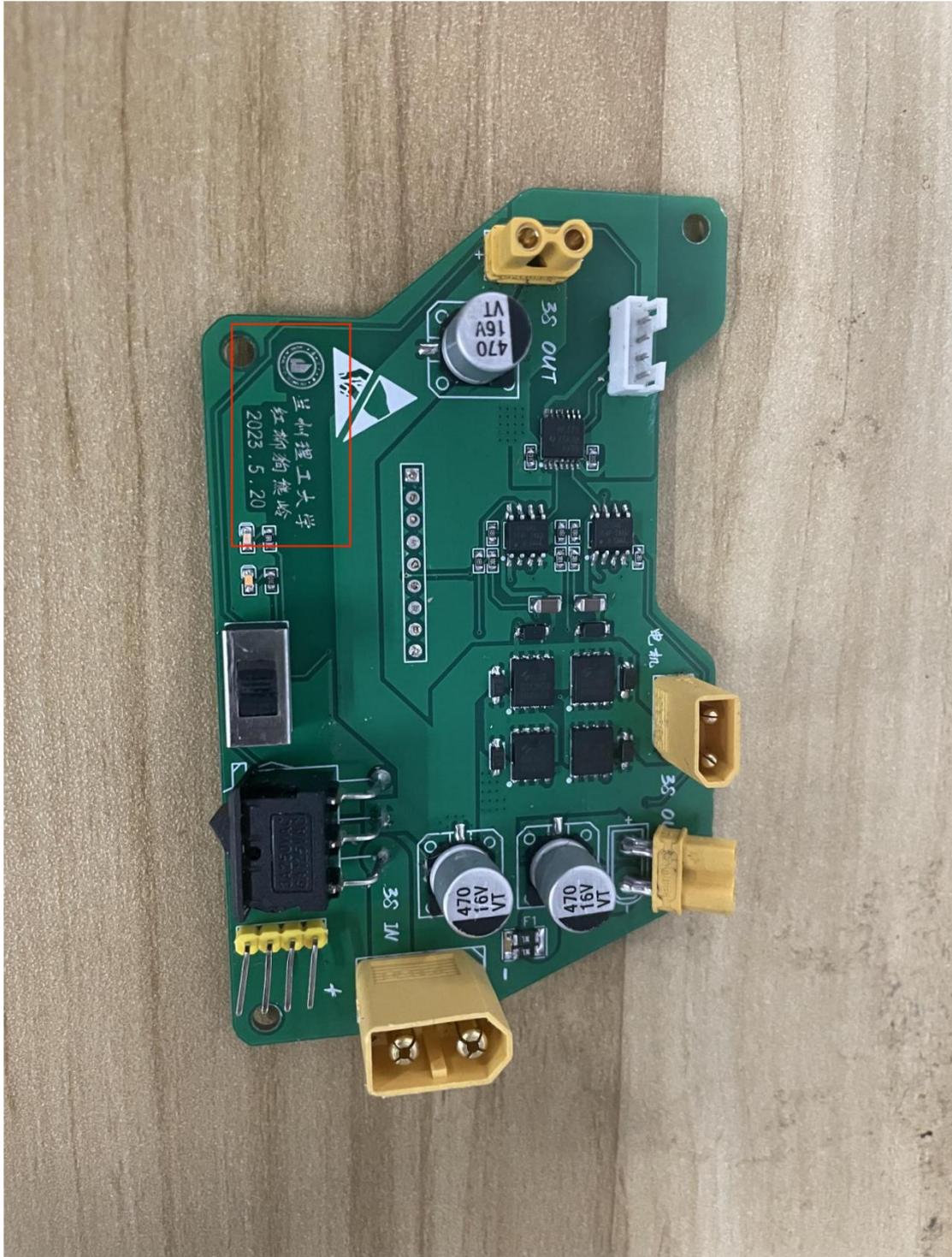
▲ 图 1.2 电路板



照片示例



在图中标记处电路板本身的 LOGO 信息





如果在覆铜层的参赛队伍信息被黑色的阻焊层遮挡, 请将相应位置的阻焊层使用刀片刮开, 以备检查查看。

▲ 图 1.3 深色阻焊层电路板无法显示覆铜层队伍信息

### 3、电路板原理图

