第十七届全国大学生
智能汽车竞赛

技 术 报 告

学　　校：长江大学

队伍名称：惟楚有才车队

参赛队员：蒋思健 钱可嘉

金权韬 王梦阳

李岚

带队教师：郑恭明 胡林

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：**蒋思健 钱可嘉**

金权韬 王梦阳

李岚

带队教师签名：**郑恭明 胡林**

日　　期：**2022 年 8 月 20 日**

摘　要

本文以第十七届全国大学生智能汽车竞赛为背景，在“立足培养，重在参与，鼓励探索，追求卓越”的指导思想下，自主设计并制作出基于完全模型的智能小车，而且在此基础上研究了一套切实可行的竞速方案。

本队参见完全模型竞速组的智能车的设计与实现方案。按照官方要求与推荐采用EdgeBoard板作为赛道和任务模型推理的主板，英飞凌单片机作为赛车的主控制器，并选用推荐的I型车模作为赛车主体。完成了从图像采集、模型建立与优化、双核心板通信与闭环PID控制小车完成循迹、任务识别与实现以及尽量提高速度的算法实现与优化。

关键字：智能车、AI 模型、图像识别、舵机控制、电机 PID 控制

目录

摘　要 III
第一章 引言 1
1.1 智能车大赛简介 1
1.2 智能汽车制作情况概述 1
第二章 系统整体设计 2
2.1 系统设计框架介绍 2
2.2 整体车模布局 3
第三章 智能车机械结构调整与优化 3
3.1 智能车整体参数调整 3
3.2 前轮定位调整 4
3.2.1 主销后倾角 4
3.2.2 主销内倾角 4
3.2.3 前轮外倾 4
3.2.4 前轮前束 5
3.3 舵机的安装 5
3.4 编码器的安装 5
第四章 硬件电路设计 6
4.1 上位机主控板 6
4.2 下位机单片机最小系统 7
4.3 电源模块 7
4.4 电机驱动模块 8
4.5 传感器模块 9
第五章 软件开发设计 11
5.1 软件功能与框架 11
5.2 经典 PID 算法控制策略 12
5.2.1 经典 PID 算法介绍 12
5.2.2 位置式 PID 算法 13
5.2.3 增量式 PID 算法 13
5.3 模糊 PID 算法控制策略 13
5.3.1 模糊算法的原理 13
5.3.2 模糊 PID 算法的设计 14
5.4 巡航模型 17
5.4.1 巡航数据采集 17
5.4.2 图片数据预处理 18
5.4.3 深度模型构建以及训练 19
第六章 开发工具及调试 20

6.1 开发工具 20
6.2 蓝牙调试模块及串口调试 20
第七章 模型车主要技术参数说明 20
结论 22
参考文献 23
附录：程序源代码 23

第一章 引言

1.1 智能车大赛简介

全国大学生智能汽车竞赛是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）主办的一项具有导向性、示范性和群众性的全国竞赛活动。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。全国大学生智能汽车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。全国大学生智能汽车竞赛以“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，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。

在本文中，我们详细介绍了基于完全模型的智能车系统。详细介绍车体机械结构的调整，传感器电路的设计，舵机控制算法，电机控制，巡航模型与任务识别模型训练与推理。

算法。在做车的整个过程中，培养了我们团队合作能力，动手的能力，创新的能力，对我们今后的学习产生积极的影响。

1.2 智能汽车制作情况概述

在本次比赛中，本组使用大赛组委会推荐的 I 车模竞赛车，控制采用前轮转向后轮驱动的方案，使用 EdgeBoard 开发板为图像处理单元和英飞凌 TC212 单片机核心板为核心控制单元，自主构思控制方案及系统设计，融合摄像头图像采集处理、电机驱动输出，最终实现一套能够自主识别路线、进行比赛的完备系统。

在制作小车的过程中，我们对小车的整体构架进行了深入的研究，分别在

机械结构、硬件和软件上都进行过改进，硬件上主要是对电路板进行多次优化及部分机械结构。

在这份报告中，我们主要通过对整体方案、机械、硬件、算法等方面的介绍，详细阐述我队在此次智能汽车竞赛中的思想和创新。具体表现在电路的创新设计、算法以及辅助调试模块等方面的创新。我队参与开发者均为电子信息专业，在准备比赛的过程中，队员查阅了大量的专业资料，日夜反复地调试汽车模型的各项参数。为此次智能汽车竞赛付出了艰苦的劳动。

第二章 系统整体设计

2.1 系统设计框架介绍

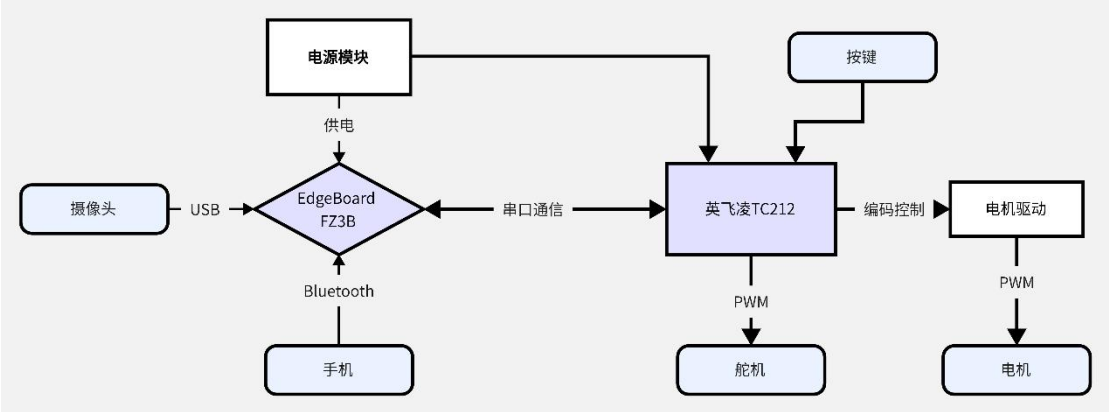


图 2.1 系统整体框图

本系统是以赛道和任务图标的图像采集、识别为基础，再通过英飞凌单片机控制小车的方向和速度，完成巡航和任务。系统电路部分需要包括单片机控制单元、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、舵机控制转向、直流电机驱动车体。综上所述，本智能车系统包含了以下几个模块：根据以上系统方案设计，智能车共包括八大模块：EdgeBoard 图像处理模块、英飞凌 TC212 核心板运动控制模块、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、EdgeBoard 图像处理模块：将深度神经网络模型部署在 EdgeBoard 板卡上，输入摄像头采集的图像，此时网络模型会输出小车的转向角度和标志物等信息；然后，根据网络模型输出的信息得出相应的处理流程；最后，EdgeBoard 板卡通过串口通信将处理数据发送给英飞凌开发板，让其去实时控制小车的运动。

2．英飞凌 TC212 核心板运动控制模块：作为整个智能车的运动控制中心，接收 EdgeBoard 图像处理的数据和编码器等传感器的信号，根据控制算法做出控制决策，驱动直流电机和舵机完成对智能车的控制。

3．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。

4．电源模块：将电源电压转化为各模块所需要的电压。

5．电机驱动模块：驱动直流电机完成智能车的速度控制。

6．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。

7．辅助调试模块：主要用于智能车系统的功能调试、智能车状态监控等方面。

2.2 整体车模布局

（1）车模底盘降低，主板低放，以降低重心。

（2）舵机竖直放置，增长力臂，利于转角，方便控制。

（3）用轻便坚固的碳纤杆作为摄像头支架。

（4）电池靠前放置，有利于重心分布。

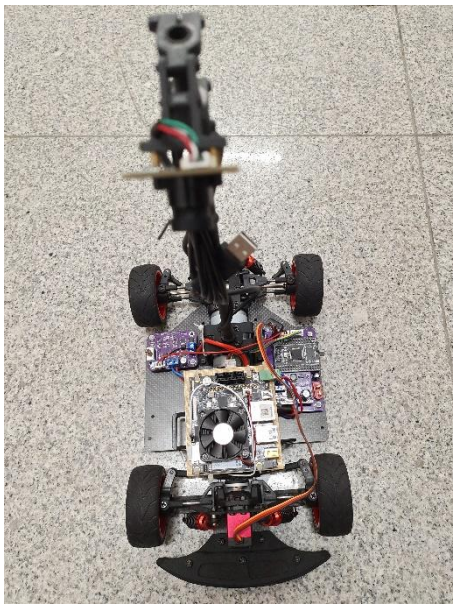


图 2.2 智能车实物图

第三章 智能车机械结构调整与优化

3.1 智能车整体参数调整

智能车的整体参数，包括车体重心、舵机的固定方式、传感器的布局方式等，

这些都对整个智能车系统的稳定运行起着至关重要的作用。

因此，对智能车机械系统的调节，有助于小车更快更稳定的运行。小车的布局以重心低，前后轮的承重分布均匀为主。

3.2 前轮定位调整

智能车出现直线走偏、转弯费力、轮胎磨损快等情况时大多与轮胎安装角度有关，涉及到一个非常重要的转向轮位置角度定位问题，叫做“前轮定位”。它的作用是保障智能车直线运行时的稳定性，使其转向轻便并减少轮胎的磨损。前轮是转向轮，它的安装位置由主销内倾、主销后倾、前轮外倾和前轮前束等四个项目决定，反映了转向轮、主销和前轴等三者在车架上的位置关系。

3.2.1 主销后倾角

从侧面看车轮，转向主销(车轮转向时的旋转中心)向后倾倒，称为主销后倾角。设置主销后倾角后，主销中心线的接地点与车轮中心的地面投影点之间产生距离（称作主销纵倾移距，与自行车的前轮叉梁向后倾斜的原理相同)，使车轮的接地点位于转向主销延长线的后端，车轮就靠行驶中的滚动阻力被向后拉，使车轮的方向自然朝向行驶方向。设定很大的主销后倾角可提高直线行驶性能，同时主销纵倾移距也增大。但主销纵倾移距过大，会使转向沉重，而且由于路面干扰而加剧车轮的前后颠簸。

3.2.2 主销内倾角

从车前后方向看轮胎时，主销轴向车身内侧倾斜，该角度称为主销内倾角。当车轮以主销为中心回转时，车轮的最低点将陷入路面以下，但实际上车轮下边缘不可能陷入路面以下，而是将转向车轮连同整个车模前部向上抬起一个相应的高度，这样车模本身的重力有使转向车轮回复到原来中间位置的效应，因而车模容易回正。

此外，主销内倾角还使得主销轴线与路面交点到车轮中心平面与地面交线的距离减小，从而减小转向时舵机上的力，使转向更加轻便，同时也可减少从转向轮传到舵机上的冲击力。

3.2.3 前轮外倾

从前后方向看车轮时，轮胎并非垂直安装，而是稍微倾倒呈现“八”字形张开，称为负外倾，而朝反方向张开时称正外倾。车模一般将外倾角设定得很小，接近垂直。若设定大外倾角会使轮胎磨偏，降低轮胎摩擦力。

3.2.4 前轮前束

四轮定位前束值脚尖向内，所谓“内八字脚”的意思，指的是左右前轮分别向内。采用这种结构目的是修正上述前轮外倾角引起的车轮向外侧转动。

3.3 舵机的安装

舵机安装方式有立式安装和卧式安装两种，比较两种安装方式，可以发现力臂较短的连接方式优点是能够输出更大力矩，调节精度更高，但是不足的是反应速度不够快，而对于长的连接方式优点是反应速度快，但是调节精度低，输出力矩不足，所以综合考虑 CS-3120 舵机输出力矩较大，速度较慢的特性，我们决定立式安装舵机，最大限度地增加舵机的灵敏度。

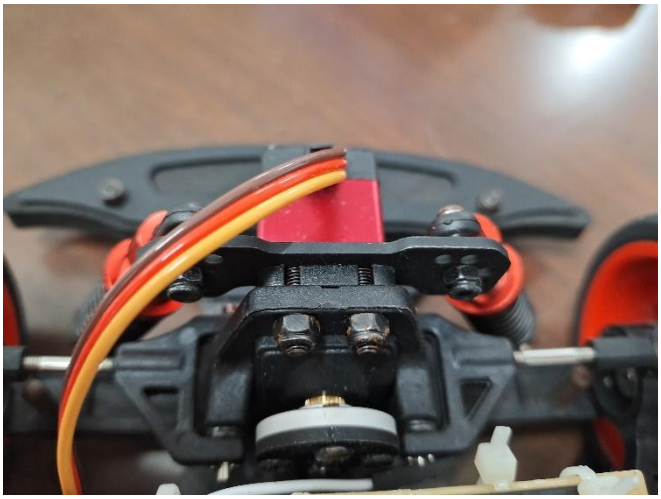


图 3.3 舵机实物安装

3.4 编码器的安装

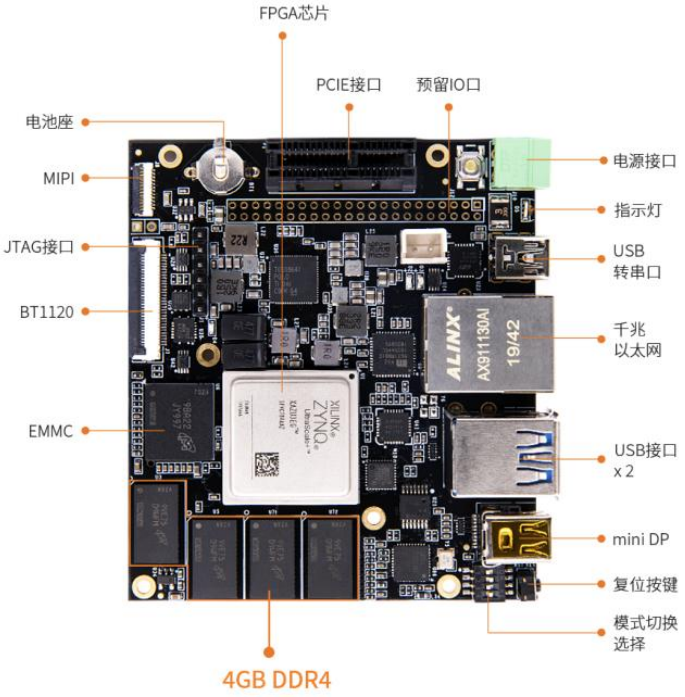
电机转速检测用的是 CSPE5-500 光电编码器，该编码器线数为 500 线，精度高，满足较高速度的 PID 闭环控制。在安装编码器时要保证有合适的齿轮咬合。咬合完美的原则是：两个传动齿轮轴保持平行，齿轮间的配合间隙要合适，过松容易打坏齿轮，过紧又会增加传动阻力；传动部分要轻松、顺畅，容易转动。判断齿轮传动是否调整好的一个依据是，听一下电机带动后轮空转时的声音。声音刺耳响亮，说明齿轮间的配合间隙过大，传动中有撞齿现象；声音沉闷而且有迟滞，则说明齿轮间的配合间隙过小，咬合过紧，或者两齿轮轴不平行，电机负载加大。调整好的齿轮传动噪音小，并且不会有碰撞类的杂音。

第四章 硬件电路设计

硬件电路的可靠运行是整个系统正常工作的基础，经过查阅相关芯片的资料和长期智能车调试试验，确定了整个系统的硬件设计，本章将重点介绍各个模块（包括上位机、最小系统、主板电路、电机驱动等）的电路原理和设计方案。

4.1 上位机主控板

EdgeBoard 计算卡（FZ3B 赛事定制版）4GB DDR4,64bit,速率 2400Mbps。EdgeBoard 计算卡是百度面向嵌入式场景打造的 A1 计算卡,采用 FPGA 芯片架构，可支持最前沿算法、百度算法、客户自定义算法部署，内置可视化管理系统，极大方便了模型部署与二次开发。小体积高性价比，丰富的 IO 口，自带主控系统，实测算力可达 1.2TOPS 算力,轻松玩转 AI 应用。



类目	可支持内容
百度大脑开放能力	人体检测、人体属性识别、人体关键点、静态/动态人流统计、车牌识别、车辆检测、车型识别、车辆属性识别、车流统计、安全帽佩戴合规检测、烟火检测、电子围栏等 （如需带算法版本，请在本页面咨询另购）
开发平台	零门槛AI开发平台EasyDL 全功能AI开发平台BML AI学习与实训社区AIStudio
框架	原生支持PaddlePaddle 支持TensorFlow/Caffe转PaddlePaddle
网络模型	MobilenetV1/MobilenetV2/ ResNet18/ResNet34/ResNet50/ResNet101/ResNet152/ ResNeXt50/ResNeXt101/ SE_ResNet18/SE_ResNet34/SE_ResNet50/SE_ResNeXt50/ SE_ResNeXt101/Res2Net50/ InceptionV3/InceptionV4/Xception41/Xception65/ DenseNet121/DPN68/HRNet_W18_C/ ssd_mobilenet_v1/ssd_vgg16/ yolov3_darknet/yolov3_r34/yolov3_mobilenet_v1/ deeplabv3/HRnet 更多算法持续适配中...
可视化管理系统	摄像头管理：配置摄像头、自动解码抽帧、查看检测结果 模型管理：加载EasyDL/Paddle模型、百度大脑开放能力 其他：数据采集、HTTP接口配置、二次开发
操作系统	Linux Ubuntu

图 4.1 EdgeBoard 主控板

4.2 下位机单片机最小系统

此次我们选择英飞凌 TC212 核心板作为电机运动控制单元。TC212 采用单核 TriCore 架构，原生主频为 133MHz，芯片 ROM512KB,芯片 RAM56KB。TriCore 是 Infineon 自己研发的 32bit 微控制器架构，主要面向汽车类应用，在安全性上高于通用的 ARM Cortex-M 架构。

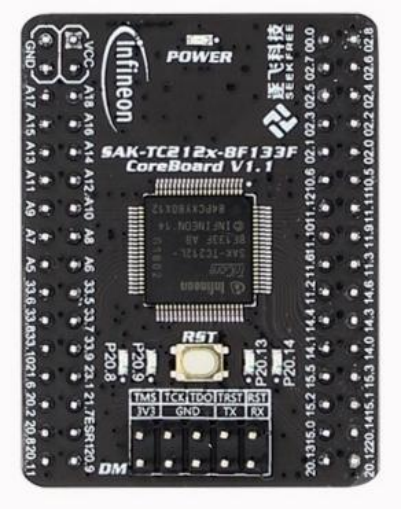


图 4.2 TC212 单片机最小系统板

4.3 电源模块

系统要稳定工作，则电源需要稳定。本系统中采用了多种幅值的直流供电：12V、6V、5V、3.3V。12V 电池电压直接接到电机驱动模块，6V 为转向舵机供电，5V 为信号采集模块，编码器模块等供电，3.3V 为英飞凌 TC212 核心板供电。

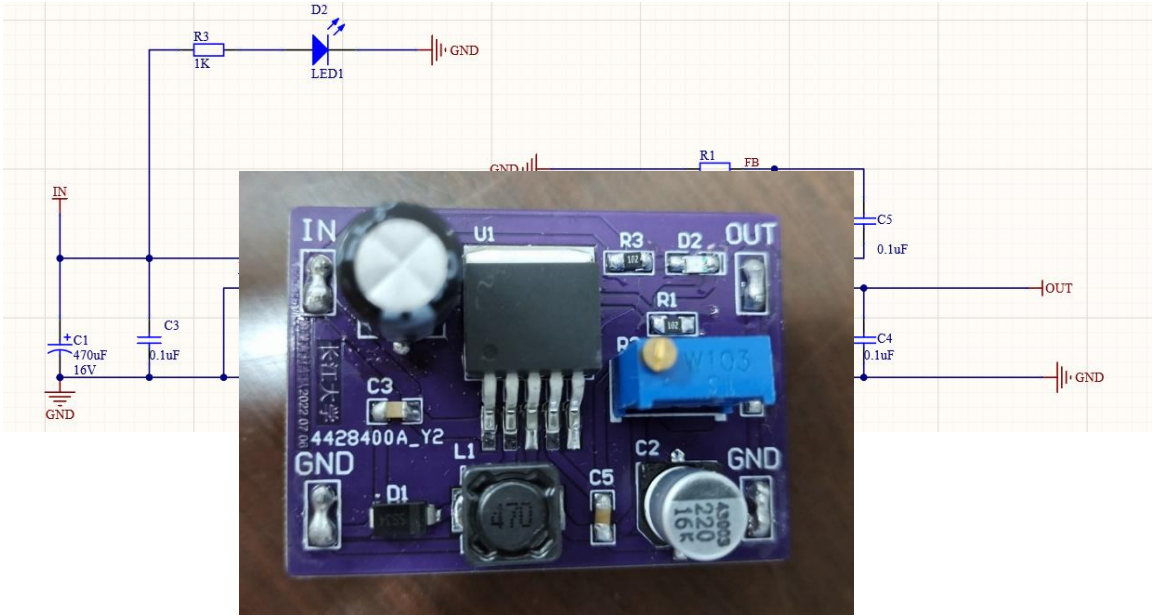


图 4.2 降压模块电路

4.4 电机驱动模块

I 车的重量和布局都对电路的设计提出了一定的要求。IR2104+IR7843 驱动和 BTN 驱动我们都用过，感觉都不是很理想，IR2104 的方案驱动能力有点弱，而且还需要在主板上单独提供设计一个 BOOST 电路，提供大于 10V 的电压，BTN 方案则是体积太大了。因此便萌生了寻求新的驱动方案的想法。新的驱动方案的主要优化方向也就是这两点，第一是尽可能缩小整个电路所用的体积，二是在提升或者保证性能的同时简化电路设计。

1、无需升压电路，简化设计，应用更加广泛。在以往的智能车电机驱动方案 中,使用了栅极驱动芯片 IR2104 + IR7843,但由于栅极驱动芯片需要 10V 以上 电压供电，因此需要在驱动电路上加一个 BOOST 升压电路，而在使用中发现升压 电路在电磁组中可能对电磁信号采集产生一定干扰。而此方案中的门极驱动芯片 DRV8701E 采用了内部自举升压，而不需要外部升压电路，可以适用于更加广泛的 输入电源（5.9V-45V）。

2、带过流保护功能，保护电机，保护智能车，使用更加安全。门极驱动芯片 DRV8701E 内部自带 OCP（Overcurrent Protection）功能,当芯片内部检测到触 发过流事件。此时芯片内部将禁止使能，达到控制输出保护驱动板以及电机的效果。

3、驱动性能更加强劲，24V 大电机也能搞定。驱动性能强劲主要由以下几面 体现：

（1） 首先 N-MOS 管 TPH1R403NL 的 DS 导通内阻更小，只有 1.7 毫欧，而之 前的 IR7843 的为 3.3 毫欧。且 TPH1R403NL 的瞬间峰值电流可达 200A，持续电流 可达 60A，均高于 IR7843 的对应参数。

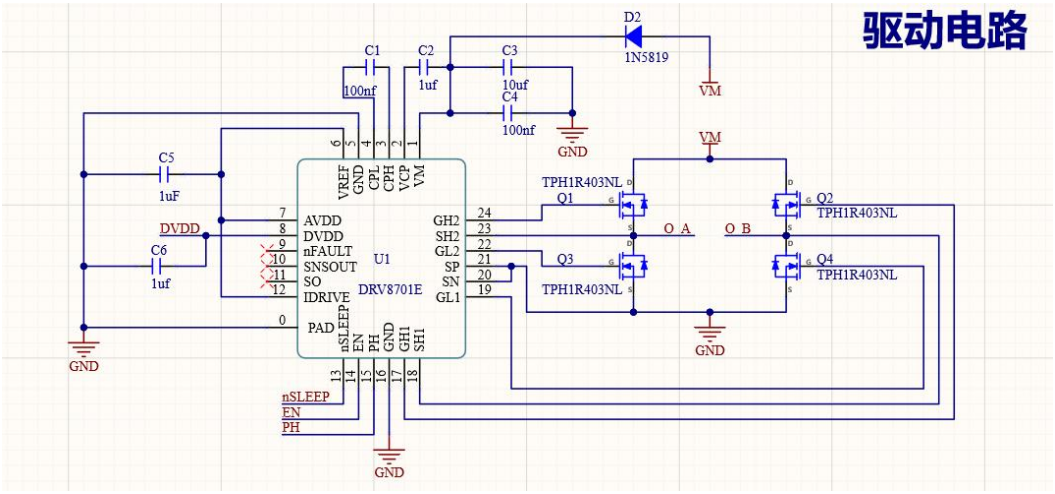
（2） 其次 DRV8701E 芯片的理论输入电压范围为 5.9V-45V，这就可使用更高 电压的输入电源，进而轻松驱动 12V-24V 的电机。

（3） 内阻小减小了内部损耗，驱动板长时间驱动高速电机，大电流使用也 不发烫，不影响性能。

4、驱动信号更加简单，解放 PWM 资源。在以往的 IR2104+IR7843 方案中，若控制两个电机的正反转，则需要 4 路 PWM 信号来作为输入信号。而在此 DRV8701E+TPH1R403NL 方案中，只需要 2 路 PWM 信号+2 路普通 IO 口引脚，

即可 进行控制。即控制 1 个电机只需要输入一个 PWM 信号控制转速，通过一个普通 IO 口输出高低电平来控制转向。

5、体积更小，适用于各组别智能车搭建。



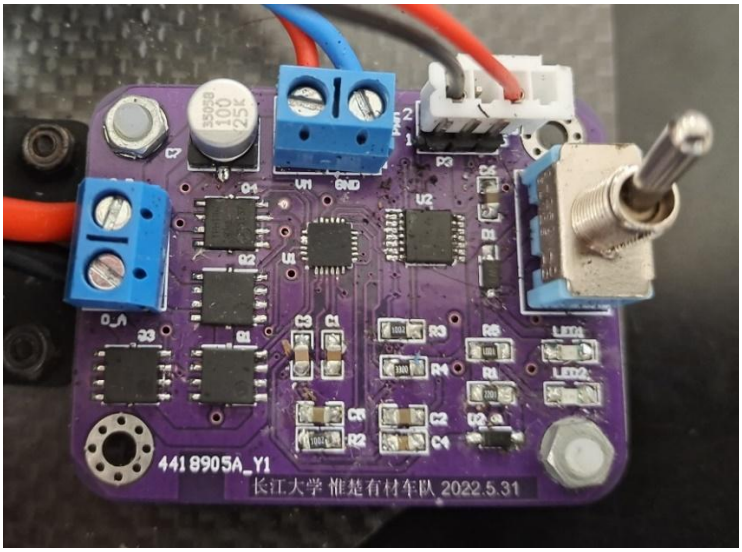


图 4.3 电机驱动电路

4.5 传感器模块

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。

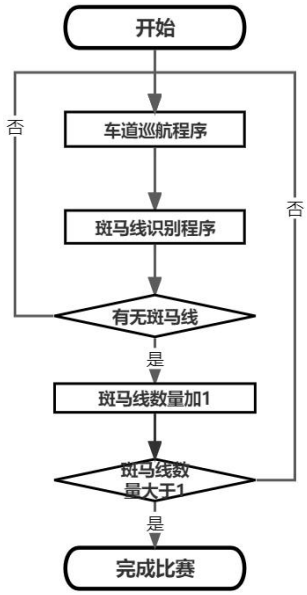
CCD 摄像头图像对比度高、动态特性好，但供电电压比较高，需要 12V 的工作电压。在智能车的实际运行中，电机加减速时会产生很大的冲击电流，会对

12V 的升压模块造成冲击。同时 CCD 摄像头的耗电也比较严重，这会使拍摄的图像稳定性不高。

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。但为了保证系统的稳定性，最终决定选用 CMOS 摄像头。竞赛专用彩色 USB 摄像头，支持 Edgeboard 免驱使用。镜头 2.8mm/110°；水平视角 62°；垂直视角 46°。

图 4.4 摄像头实物

第五章 软件开发设计



5.1 软件功能与框架

1、软件的主要具体功能包括有：

（1）各个外设的初始化；

（2）各传感器信号的采集、处理；

（3）车模运行控制：方向控制、速度控制、特殊路段识别；

（4）车模信息显示与参数设定：状态显示、上位机监控、参数设定等。

2、程序上电运行后，便进行单片机的初始化。

在程序中使用定时器，产生 5ms 的周期中断。中断服务程序的任务被分配在 5ms 的中断片段中。这些任务包括：

（1）电机测速脉冲计数器读取与清除。累积电机速度，为后面车模速度控制提供平均数；

（2）车模方向控制。根据前面读取的数值，根据模糊控制算法输出 PWM 控制信号。

小车的控制包括舵机的 PD 控制，电机的 PID 控制，传感器数值的处理，路径优化处理。根据传感器采集的数据进行速度，以及打角的控制，实现小车快速，稳定的运行。

5.2 经典 PID 算法控制策略

5.2.1 经典 PID 算法介绍

PID 控制是工程实际中应用最为广泛的调节器控制规律。问世至今 70 多年来，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。单位反馈的 PID 控制原理框图如图：

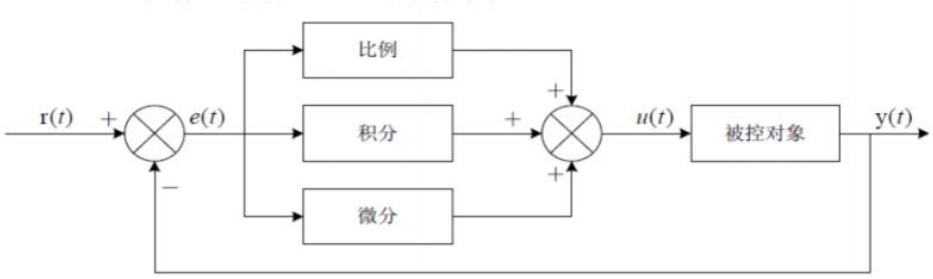


图 5.2

e(t) 代表理想输入与实际输出的误差，这个误差信号被送到控制器，控制器算出误差信号的积分值和微分值，并将它们与原误差信号进行线性组合，得到输出量 u。

$$u = k_p e + k_i \int e dt + k_d \frac{de}{dt}$$

（公式 1）

其中，kp、kd、ki 分别为比例系数、积分系数、微分系数。接着，u 被送到了执行机构，然后获得新的输出信号 u,这个新的输出信号 u 再次被送到感应器，得到新的误差信号 e(t)，这个过程就这样周而复始地进行。PID 各个参数作用基本介绍：比例调节（P）作用：是按比例反应系统的偏差，系统一旦出现了偏差，比例调节立即产生调节作用，以减少偏差。比例作用大，可以加快调节，减少误差，但是过大的比例，使系统的稳定性下降，甚至造成系统的不稳定。积分调节（I）作用：是使系统消除稳态误差，提高无差度。因为有误差，积分调节就进行，直至无差，积分调节停止，积分调节输出一常值。积分作用的强弱取决与积分时间常数 Ti，Ti 越小（注：Ki=1/Ti），积分作用就越强。反之 Ti 大则积分作用弱，加入积分调节可使系统稳定性下降，动态响应变慢。积分作用常与另两种调节规律结合，组成 PI 调节器或 PID 调节器。微分调节（D）作用：微分作用反映系统偏差信号的变化率，具有预见性，能预见偏差变化的趋势，因此能产生超前的控制作用，在偏差还没有形成之前，已被微分调节作用消除。因此，可以改善系统的动态性能。在微分时间选择合适情况下，可以减少超调，减少调节时间。微分作用对噪声干扰有放大作用，因此过强的加微分调节，对系统抗干扰不利。

此外，微分反应的是变化率，而当输入没有变化时，微分作用输出为零。微分作用不能单独使用，需要与另外两种调节规律相结合，组成 PD 或 PID 控制器。

5.2.2 位置式 PID 算法

矩形数值积分代替上式中的积分项，对导数项用后向差分逼近，得到数字 PID 控制器的基本算式（位置算式）

$$u_k = Kp[e_k + \frac{T}{Ti}\sum_{j=0}^{k} e_j + Td\frac{e_k - e_{k-1}}{T}] \qquad \text{（公式 2）}$$

5.2.3 增量式 PID 算法

对位置式加以变换，可以得到 PID 算法的另一种实现形式（增量式）

$$\Delta u_k = u_k - u_{k-1} = Kp(e_k - e_{k-1} + \frac{T}{Ti}e_k + Td\frac{e_k - 2e_{k-1} + e_{k-2}}{T})$$

（公式 3）

这种算法用来控制步进电机特别方便，对直流电机也可以采用。在实际应用中，由于码盘采样得到的脉冲上升下降沿不够明显，使得速度采样出现不稳定和失真，但由于这些附加处理比较耗费代码的运行时间，出于代码效率和实际效果的比较，我们没有采用这些改进的方案，另外可以考虑加反向器来整波形得到较为理想的方波。运用 PID 控制的关键是调整三个比例系数，即参数整定。

PID 整定的方法有两大类：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。由于智能车整个系统是机电高耦合的分布参数系统，并且要考虑赛道具体环境，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不断修正，模型参数变化较频繁，可操作性不强；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，我们采用了这种方法，同时，我们先后实验了几种动态改变 PID 参数的控制方法。

5.3 模糊 PID 算法控制策略

模糊 PID 控制器是将模糊算法与 PID 控制参数的自整定相结合的一种控制算法。可以说是模糊算法在 PID 参数整定上的应用。

5.3.1 模糊算法的原理

模糊算法是一种基于智能推理的算法，虽然称之为模糊算法其实并不模糊，实际上是一种逐步求精的思想。一个模糊控制器主要是由模糊化，模糊推理机和精确化三个功能模块和知识库（包括数据库和规则库）构成的。

1.输入量的量化

输入数据都是精确的，要实现模糊算法需要现对其实现量化。所谓量化就是通过量化函数将输入量投射到一定的数字级别，一般都是相对 0 对称的数字区间。具体投射到怎样的区间根据实际情况而定，因为这会直接影响到计算的精度。

2.模糊化

模糊化是模糊算法非常重要的一步，首先确定对应各语言变量的模糊子集，然后根据量化的结果，我们就可以判断该输入所属的集合并计算出对应的隶属度。计算隶属度的方法有很多，最常用的是使用三角形隶属度函数或梯形隶属度函数等来计算获得。

3.规则库

规则库是基于控制量的模糊化而的味道的，是实现模糊推理的基础，很大程度上依赖于经验来完成。规则库的表现形式可以有多种，具体实现的形式根据我们实现的方便。

4.推理机

推理决策才是模糊控制的核心，它利用知识库中的信息和模糊运算方式，模拟人的推理决策的思想方法，在一定的输入条件下激活相应的控制规则给出适当的模糊控制输出。

5.精确化

我们通过模糊推理，得到一系列的模糊表达，需要进行解模糊操作才能得到紧缺的数据。常用的解模糊方法有：

a.最大隶属度法——计算简单，适用于控制要求不高场合。

b.重心法——输出更平滑，但计算难度大。

c.加权平均法——一般在工业上应用最广泛。

5.3.2 模糊 PID 算法的设计

所谓模糊 PID 控制是以偏差 e 及偏差的变化 ec 为输入，利用模糊控制规则在线对 PID 参数进行调整，以满足不同的偏差 e 和偏差的增量 ec 对 PID 参数的不同要求。其结构图如下：

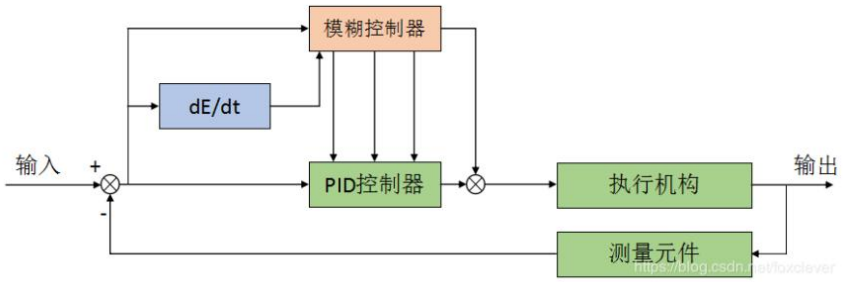


图 5.3

1.输入值的模糊化

输入值的模糊化就是将用于计算的输入对应到标准化的数值区间，并根据量化结果和模糊化子集得到该输入对子集的隶属度。我们在使用偏差 e 和偏差增量 ec 作为输入实现控制参数调整则需要对 e 和 ec 进行模糊化。

首先，我们确定 e 和 ec 的模糊子集，对于 PID 控制我们选则：负大[NB]、负中[NM]、负小[NS]、零[ZO]、正小[PS]、正中[PM]、正大[PB]等 7 个语言变量就能够有足够精度表达其模糊子集。所以我们定义 e 和 ec 的模糊子集均为{NB，NM，NS，ZO，PS，PM，PB}。

确定了模糊子集，还需要引入量化函数。要确定量化函数，我们先引入 e 和 ec 模糊集对应的论域，定义为{-6，-5，-4，-3，-2，-1，0，1，2，3，4，5，6}。对于任何一个物理量测量信号都有一个量程范围，我们记为 Vmax 和 Vmin，和自然在 PID 调节时设定值的范围预期相同，所以偏差 e 的范围就是 Vmin-Vmax 到 Vmax-Vmin 的范围内，而偏差的增量范围则是其两倍。这里我们采用线性方式量化，则其函数关系为：

$$f(e)=\frac{6*e}{V\max-V\min}\qquad f(ec)=\frac{6*ec}{2(V\max-V\min)}$$

（公式 4）

利用上述的量化函数就可以将 e 和 ec 量化，我们可以采用如 4 舍 5 入的方式获取确定的模糊子集。最后我们确定 e 和 ec 在模糊子集上的隶属度。隶属度是一个介于 0 和 1 之间的值，用以描述对应一个输入属于某一个模糊自己的程度。一般我们描述成隶属度函数，可采用的隶属度函数很多，我们在此采用线性的隶属度函数，或者称为三角隶属度函数。

如果我们量化后的结果是 1，那么属于 ZO 的隶属度为 0.5，同样属于 PS 的隶属度也是 0.5。至此，模糊化全部完成。

2.建立模糊规则表

a.Kp 模糊规则设计

在 PID 控制器中，Kp 值的选取决定于系统的响应速度。调节初期应适当取较大的 Kp 值以提高响应速度，而在调节中期，Kp 则取较小值，以使系统具有较小的超调并保证一定的响应速度；而在调节过程后期再将 Kp 值调到较大值来减小静差，提高控制精度。基于上述描述我们定义 Kp 的模糊规则如下：

表 5.1

EC

E / ΔKp	NB负大	NM负中	NS负小	ZO零	PS正小	PM正中	PB正大
NB 负大	PB正大	PB正大	PM正中	PM正中	PS正小	ZO零	ZO零
NM 负中	PB正大	PB正大	PM正中	PS正小	PS正小	ZO零	NS 负小
NS 负小	PM正中	PM正中	PM正中	PS正小	ZO零	NS 负小	NS 负小
ZO 零	PM正中	PM正中	PS正小	ZO零	NS 负小	NM 负中	NM 负中
PS 正小	PS正小	PS正小	ZO零	NS 负小	NS 负小	NM 负中	NM 负中
PM 正中	PS正小	ZO零	NS 负小	NM 负中	NM 负中	NM 负中	NB 负大
PB 正大	ZO零	ZO零	NM 负中	NM 负中	NM 负中	NB 负大	NB 负大

b.Kd 模糊规则设计

微分环节的调整主要是针对大惯性过程引入的，微分环节系数的作用在于改变系统的动态特性。在调节初期，应加大微分作用，这样可得到较小甚至避免超调；而在中期，由于调节特性对 Kd 值的变化比较敏感，因此，Kd 值应适当小一些并应保持固定不变；然后在调节后期，Kd 值应减小，以减小被控过程的制动作用，进而补偿在调节过程初期由于 Kd 值较大所造成的调节过程的时间延长。依据以上分析，我们制定 Kd 的模糊规则如下：

表 5.2

EC

E / ΔKd	NB负大	NM负中	NS负小	ZO零	PS正小	PM正中	PB正大
NB 负大	PS正小	NS 负小	NB 负大	NB 负大	NB 负大	NM 负中	PS正小
NM 负中	PS正小	NS 负小	NB 负大	NM 负中	NM 负中	NS 负小	ZO零
NS 负小	ZO零	NS 负小	NM 负中	NM 负中	NS 负小	NS 负小	ZO零
ZO 零	ZO零	NS 负小	NS 负小	NS 负小	NS 负小	NS 负小	ZO零
PS 正小	ZO零	ZO零	ZO零	ZO零	ZO零	ZO零	ZO零
PM 正中	PB正大	NS 负小	PS正小	PS正小	PS正小	PS正小	PB正大
PB 正大	PB正大	PM正中	PM正中	PM正中	PS正小	PS正小	PB正大

接下来，根据偏差 E 和偏差增量 EC 模糊化的结果以及规则库推理出ΔKp、ΔKd 对应的模糊子集。由于前面我们设计的是采用隶属度函数来定义输入输出量在模糊子集的隶属度，所以推理出来的ΔKp、ΔKd 的模糊子集通常是一个由模糊

变量组成的矩阵。而输入量 E 和 EC 则是一个由模糊变量组成的向量。最后，我们需要明确不同的模糊变量所对应的量化数据。

3.解模糊处理

我们前面设计了三角隶属度函数，并采用相同的量化目标即论域{-6，6}，所以在某一时刻，输入输出所处的模糊变量的隶属度是相同的，基于这一基础，我们采用重心法计算各输出量的量化值。其公式如下：

$$Vo=\frac{\sum_{i=0}^{n}M_i*F_i}{\sum_{i=0}^{n}M_i}$$ 其中M为隶属度，F为模糊量化值 （公式 5）

每一个对象的计算实际上就是矩阵操作，公式如下：

$$K=\begin{bmatrix}M_{e1} & M_{e2}\end{bmatrix}\begin{bmatrix}F_a & F_b\\ F_c & F_d\end{bmatrix}\begin{bmatrix}M_{ec1} & M_{ec2}\end{bmatrix}^T$$ （公式 6）

如果使用的是量化值，则还需要转为实际值，关于这一点直接使用物理量值也是没问题的，怎么处理根据实际需要确定。得到增量后，我们也可以引入系数来放大和缩小 Kp 和 Kd 变化量，具体实现公式如下：

$$K(n)=K(n-1)+\Delta K*\partial$$ （公式 7）

其中ΔK 为我们所计算得到的值，而 α 为系数，设定增量对最终值的影响。

5.4 巡航模型

对于实现小车巡航模型这一部分，主要从最开始的数据采集过程，图片数据的预处理过程，以及最后深度学习网络模型的构建与训练过程这几部分来进行描述。

5.4.1 巡航数据采集

对于实现巡航图片的采集，主要通过小车的摄像头进行图片的读取，如图 5.4 所示，小车主要以 15 的速度进行向前运行，通过操作手机，来使小车完成巡航过程，得到训练的图片数据集。

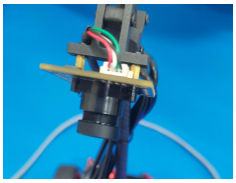


图 5.4 摄像头

通过手机蓝牙调试器连接小车蓝牙，蓝牙调试器界面可自行设计，如图 5.5 所示，通过运行 collect.py 程序，点击“BOOL”按钮开始遥控小车行驶，右下角的角度控制摇杆能够控制小车的转弯，车速可以自行调控，当巡航数据搜集完毕时，点击“BOOL”按钮小车停止运动，相应的数据采集也结束。上述过程会将采集所得图片和对应的遥控数据 json 角度文件存放到 train 文件夹中。

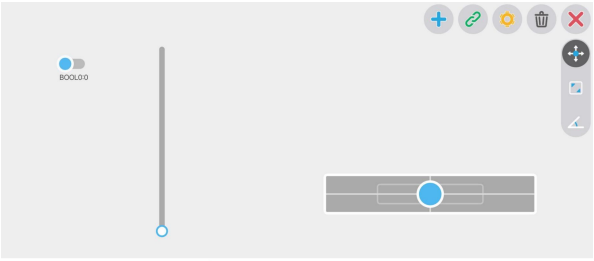


图 5.5 蓝牙调试器

5. 4. 2 图片数据预处理

本文对采集到的数据进行预处理操作，把采集到图片的大小统一成 128*128 的图片大小格式，接着对图片数据进行了随机增强主要分为四种方式：

1.应用色调方式

主要将图片参数 low, high, prob 设置为 [-18, 18, 0.5]，来进行对图片数据的色调进行随机处理。

2.应用饱和度方式

主要将图片参数 low, high, prob 设置为 [0.5, 1.5, 0.5]，来进行对图片数据的饱和度进行随机处理。

3.应用对比度方式

主要将图片参数 low, high, prob 设置为 [0.5, 1.5, 0.5]，来进行对图片数据的对比度进行随机处理。

4.应用亮度方式

主要将图片参数 low, high, prob 设置为 [0.5, 1.5, 0.5]，来进行对图片数据的亮度进行随机处理。

在本文采取的每一种随机增强图片数据概率设置为 0.25，随机数据增强方式内每一种的概率大小为 0.5，从而对图片数据进行随机增强。最后把图片格式设置为 RGB 格式投入到深度学习网络中进行训练。

5.4.3 深度模型构建以及训练

在本为采取 paddle 深度学习框架对小车巡航数据进行构建，遵循的原则是实时性，轻巧性，是由于在开发板上运行大的网络框架可能会造成一定的延迟，及其容易让小车跑出赛道，所以轻巧的小网络模型是本文需要注意的。采取如下图 5.6 所示轻巧的小网络。对于 conv 层主要是针对图片中心位置进行着重处理的，相反，本次小车是注重边缘部分，由于地图中间是白色部分，两边是黑色赛道线，因此关注边缘部分，利用全连接层及其容易处理这些边缘问题。最后采用 tan 函数作为输出，是由于小车数据在-1 到 1 之间变化，正切函数刚好可以准确的描述出这些输出结果。

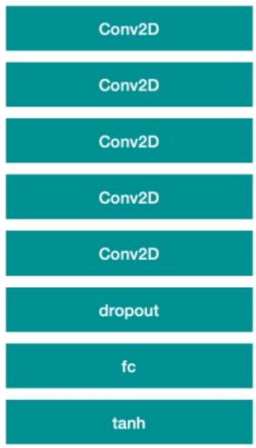


图 5.6 深度学习网络

对于提升巡航模型的效果，主要的方法有如下几种：

1. 提高数据采集的平滑度

在智能车采集巡航图片时，较可能的使转弯时摇杆平滑的变化，控制摇杆，最后再通过微型的处理结果值，平滑处理跳变幅度较大的值，来进行操作。

2. 增加一些自定义的数据增强手段

除了采取四种图片数据随机增强的方法外，在训练前对采集的数据进行模糊处理，增加数据的多样性。

3. 优化网络

在基础的网络模型上增加了两层，通过测试得到效果相对较好。

最后通过调节学习率 lr = fluid.layers.piecewise_decay(boundaries=[700, 1100], values=[0.0001, 0.00001, 0.000001])，本文由于增加了数据集大小，也对最终的学习率进行更大的迭代操作设置。

第六章 开发工具及调试

6.1 开发工具

程序开发在Pycharm下进行,PyCharm是由JetBrains打造的一款Python IDE（Integrated Development Environment，集成开发环境），带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如调试、语法高亮、项目管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该IDE提供了一些高级功能，以用于支持Django框架下的专业Web开发。

6.2 蓝牙调试模块及串口调试

智能车的调试过程，实际上是一个分析实验数据、改进算法的过程，而这需要以大量的实验数据作为基础。蓝牙信号的收发采用蓝牙模块实现，具有片内数字无线处理器DRP、数控振荡器，片内射频收发开关切换，内置ARM7嵌入式处理器等。接收信号时，收发开关置为收状态，射频信号从天线接收后，经过蓝牙收发器直接传输到基带信号处理器。该模块主要用于短距离的数据无线传输领域。可以方便的和PC机的蓝牙设备相连，也可以两个模块之间的数据互通。

第七章 模型车主要技术参数说明

队伍名称	惟楚有才队		
参赛学校	长江大学		
赛题组组别	四轮电磁	四轮摄像头	多车编队
	平衡单车	无线充电	平衡信标
	智能视觉	极速越野	√ 完全模型
1.车模类型是什么？	Ⅰ模型车		
车模整体尺寸：			

1.（包括传感器在内）长，宽，高(mm) 2. 对于无线充电组：显示电能的 LED 板尺寸	300×180×375
1. 传感器种类、规格(型号)数量。	编码器、摄像头、陀螺仪
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？	1.CS-3120 舵机 2.完整
1. 是否增加伺服电机？ 2. 如果有那么种类、个数和作用？	无
1. 电池的种类、规格和数量？	1.锂电池、2200mah 30c、1 个
1. 是否有升压电路驱动舵机和后轮电机？	1、无
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？	1. 是 2. 有
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	1. 是 2. 无 3. 无麦伦

否更换过小轮胶皮？	
1. 车模底盘是否是原车模底盘？ 2. 是否有大面积切割？	1. 是 2. 无
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？	1、已标记 2、top layer、top solder

结论

自报名参加第十七届全国大学生智能车竞赛以来，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，我们主要介绍了准备比赛时的基本思路，包括机械、电路以及最重要的控制算法的创新思想。在机械结构方面，我们分析了舵机转向系统的改进办法，对前轮倾角进行一系列的改动。在电路方面，我们以模块形式分类，在最小系统、主板、电机驱动、电池使用等模块分别设计，经过不断实验，最后确定了最终的电路图。在程序方面，我们使用 python 语言编程，利用开发工具调试程序，经过小组成员不断讨论、改进，终于设计出一套比较通用稳定的程序。在这套算法中，我们结合路况调整车速，做到直道走直、弯道走稳，保证在最短时间内跑完全程。尽管做了很多尝试和努力，但终究还是有缺陷。智能车的机械结构还有待进一步完善，速度 PID 还需要进一步优化，智能车的整体运行稳定性还要改进。

在这几个月的备战过程中，场地和经费方面都得到了学校和学院的大力支持，在此特别感谢一直支持和关注智能车比赛的学校和学院领导以及各位指导老师、指导学长，同时也感谢比赛组委会能组织这样一项有意义的比赛。

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] 卓晴，黄开胜，邵贝贝.学做智能车.北京-北京航空航天大学出版社.2007
[3] 王晓明.电动机的单片机控制[M]. 北京. 北京航空航天大学出版社. 2002
[4] 臧杰，阎岩.汽车构造[M]. 北京. 机械工业出版社. 2005
[5] 童诗白，华成英. 模拟电子技术基础[M]. 北京.高等教育出版社.
[6] 谭浩强.C++程序设计.北京-清华大学出版社.2004
[7] 钱江一号.第六届“飞思卡尔”杯全国大学生智能车大赛技术报告.杭州.电子科技大学.2011
[8]黄开胜，陈宋.汽车理论与智能模型车机械结构调整方法[D].清华大学汽车安全与节能国家重点实验室，2006.

附录：程序源代码

cart 函数

```
#蓝牙数据包格式
#0xA5+数据位：DIR_bool+Angle_byte+Speed_byte;+校验和+0x5A

class Cart:
    def __init__(self):
        self.Angle = 90 #角度
        self.Speed = 20 #速
        self.Direction = 0 #前进  后退
        portx = "/dev/ttyPS1"
        bps = 115200
        self.serial = serial.Serial(portx, int(bps), timeout=1, parity=serial.PARITY_NONE, stopbits=1)
        # self.serial = serial_connection
    def angle(self,angle):
        self.Angle = angle
```

```
    def speed(self,speed):
        self.Speed = speed

    def direction(self,direction):
        self.Direction = direction

    def Control(self,data):
        self.serial.write(data)
        if len(data) != 0:
            data = data.hex()
            self.Direction = data[2:4]
            self.Angle = data[4:6]
            self.Speed = data[6:8]

    def save(self,data):
        #self.serial.write(data)
        if len(data) != 0:
            data = data.hex()
            self.Direction = data[2:4]
            self.Angle = data[4:6]
            self.Speed = data[6:8]

    def Move(self,steer):
        steer = int(steer*70)
        check = self.Direction + self.Speed + steer
        self.serial.write(bytes.fromhex('A5')+
            self.Direction.to_bytes(1,'big',signed = True)+
            steer.to_bytes(1,'big',signed = True)+
            self.Speed.to_bytes(1,'big',signed = True)+
            check.to_bytes(1,'big',signed = True)+
            bytes.fromhex('5A'))
```

Camera 函数

```
class Camera:
    def __init__(self, src=0,shape=[480,320]):
        self.src = src
        self.stream = cv2.VideoCapture(src)
        # self.stream.set(cv2.CAP_PROP_FRAME_WIDTH, shape[0])
        # self.stream.set(cv2.CAP_PROP_FRAME_HEIGHT, shape[1])
        self.stream.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
        self.stream.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
        # self.stream.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter::fourcc('M', 'J', 'P', 'G'));
```

```
            # self.stream.set(cv2.CAP_PROP_FOURCC, cv2.VideoWriter_fourcc('Y', 'U', 'Y', 'V'));
            print("sss")
            self.stopped = True
            for _ in range(10): #warm up the camera
                (self.grabbed, self.frame) = self.stream.read()


    def start(self):
            self.stopped = False
            threading.Thread(target=self.update, args=()).start()

    def update(self):
            count = 0
            while True:
                if self.stopped:
                    return
                (self.grabbed, self.frame) = self.stream.read()
                # time.sleep(1)
                # if self.src == 0:
                #         path = "images/{}.png".format(count);
                #         count = count + 1;
                #         cv2.imwrite(path, self.frame);


    def read(self):
            return self.frame

    def stop(self):
            self.stopped = True
```

driver 函数

```
class Driver:

  def __init__(self):
    self.cart = Cart()
    self.cruiser = Cruiser()
    self.angle = 0.0

  def stop(self):
    self.cart.speed(0)
    self.cart.Move(0.5)

  def go(self, frame):
```

```
        self.angle = self.cruiser.cruise(frame)
        self.cart.Move(self.angle)


# d = Driver();
SLOW_DOWN_RATE = 0.6
if __name__ == &apos;__main__&apos;:
    d = Driver()
    print("hoinonogei")
    # d.change_posture(20)
    d.change_posture_cm(5)
```

cruiser 函数

```
cnn_args = {
        "shape": [1, 3, 128, 128],
        "ms": [125.5, 0.00392157]
}

#cruise_model = config.cruise["model"]

# CNN 网络的图片预处理
def process_image(frame, size, ms):
        #print(frame)
        frame = cv2.resize(frame, (size, size))
        img = frame.astype(np.float32)
        img = img - ms[0]
        img = img * ms[1]
        img = np.expand_dims(img, axis=0)
        return img

# CNN 网络预处理  src
def cnn_preprocess(args, img, buf):
        shape = args["shape"]
        img = process_image(img, shape[2], args["ms"])
        hwc_shape = list(shape)
        hwc_shape[3], hwc_shape[1] = hwc_shape[1], hwc_shape[3]
        data = buf
        img = img.reshape(hwc_shape)
        # print("hwc_shape:{}".format(hwc_shape))
        data[0:, 0:hwc_shape[1], 0:hwc_shape[2], 0:hwc_shape[3]] = img
        data = data.reshape(shape)
        return data
```

```
# CNN 网络预测
def infer_cnn(predictor, buf, image):
    data = cnn_preprocess(cnn_args, image, buf)
    predictor.set_input(data, 0)
    predictor.run()
    out = predictor.get_output(0)
    return np.array(out)[0][0]

class Cruiser:
    def __init__(self):
        hwc_shape = list(cnn_args["shape"])
        hwc_shape[3], hwc_shape[1] = hwc_shape[1], hwc_shape[3]
        self.buf = np.zeros(hwc_shape).astype('float32')
        self.predictor = predictor_wrapper.PaddleLitePredictor()
        self.cruise_model = config.cruise1["model"]
        self.predictor.load(self.cruise_model)

    def cruise(self, frame):
        res = infer_cnn(self.predictor,self.buf, frame)
        # print(res)
        return res

    def cruise_models(self,cruiser_number):
        if cruiser_number == 0:
            self.cruise_model = config.cruise1["model"]
        elif cruiser_number == 1:
            self.cruise_model = config.cruise2["model"]
        print("cru_number")
        print(self.cruise_model)
        self.predictor = predictor_wrapper.PaddleLitePredictor()
        self.predictor.load(self.cruise_model)

if __name__ == "__main__":
    c = Cruiser()
```

run collect 函数

```
#本文件用作无人驾驶车道数据动态采集
driver = Driver()
cruiser = Cruiser()
front_cam = 0
cart = Cart()
class Logger:
```

```
    def __init__(self):
        self.camera = cv2.VideoCapture(front_cam)
        self.camera.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
        self.camera.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
        self.started = False
        self.stopped_ = False
        self.counter = 0
        self.image_data = 0
        self.save_flag = 0
        self.angle = 0
        self.map = {}
        self.result_dir = "train"

    def start(self):
        #print("start")
        self.started = True
        pass

    def stop(self):
        print("stop")
        if self.stopped_:
            return
        self.stopped_ = True
        path = "{}/result.json".format(self.result_dir)
        with open(path, 'w') as fp:
            json.dump(str(self.map.copy()), fp)
        pass

    def log(self, axis):
        if self.started:
            # print("axis:".format(axis))
            return_value, image = self.camera.read()
            path = "{}/{}.jpg".format(self.result_dir, self.counter)
            self.map[self.counter] = axis
            cv2.imwrite(path, image)
            self.counter = self.counter + 1

    def stopped(self):
        return self.stopped_

logger = Logger()
Blue = Bluetooth()
```

```
def blue_thread():
    print("into Bluetooth_thread thread")
    while not logger.stopped():
        if cart.Speed != "00":
            logger.start()
        if cart.Direction == "01":
            logger.stop()
        Blue.Transmit()
        cart.save(Blue.rec)

def main():
    print("into mian")
    t2 = threading.Thread(target=blue_thread, args=())
    t2.start()
    while not logger.stopped():
        if logger.started:
            ret,logger.image_data = logger.camera.read()
            if ret == True:
                logger.angle = driver.go(logger.image_data)
                logger.log(logger.angle)
    t2.join()
    Blue.serial.close()
    logger.camera.release()
    print("end")

if __name__ == "__main__":
    try:
        main()
    except:
        cart.serial.close()
        logger.camera.release()
```

Start 函数

```
#蓝牙数据包格式
#0xA5+数据位：DIR_bool+Angle_byte+Speed_byte;+校验和+0x5A

class Start:
    def __init__(self):
        portx = "/dev/ttyUSB0"
        bps = 115200
        self.rec = None
        self.serial = serial.Serial(portx, int(bps), timeout=0, parity=serial.PARITY_NONE,
stopbits=1)
```

```
            self.state = 0

        def begin(self):
            head = self.serial.read(1)
            if head.hex() == "53":
                self.state = 1
                return True

        def end(self):
            head = self.serial.read(1)
            if head.hex() == "45":
                self.state = 0
                return True
coll = Start()
def main():
        while True:
            if coll.begin():
                print("begin")
            if coll.end():
                print("end")

if __name__ == "__main__":
        try:
            main()
        except:
            print("error")
            coll.serial.close()
```

Run2 函数

```
driver = Driver()
cruiser = Cruiser()
begin = Start()
front_cam = 0

camera = cv2.VideoCapture(front_cam)
camera.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
camera.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
print("camera success")

V = int(sys.argv[1])

def main():
        cruiser_number = 0
```

```
    cnt = 0
    driver.cart.speed(V)
    print(V)
    time.sleep(1)
    begin.state = 1
    while True:
        if begin.begin():
            while True:
                ret,front_image = camera.read()
                if ret == True:
                    #print("curiser")
                    driver.go(front_image)
                    cnt += 1
                    if cnt == 100000:
                        cruiser_number = cruiser_number + 1
                        print('Switch the', cruiser_number, ' model')
                        driver.cruiser.cruise_models(cruiser_number)
                if begin.end():
                    driver.stop()
                    break
        if begin.state == 0 and cnt > 0:
            print(cnt)
            break
    camera.release()
    begin.serial.close()
    print("end")
        #print(front_image)


if __name__ == "__main__":
    # try:
    main()
    # except:
    #     print("err")
    #     camera.release()
    #     begin.serial.close()
```