第十二届“恩智浦”杯全国大学生智能汽车竞赛

光电四轮组技术报告

学　　校：　华中科技大学

队伍名称：　华中科技大学光电竞速一队

参赛队员：　谭培鑫

耿铭良

杨子龙

指导老师：　何顶新

池明

关于技术报告和研究论文使用授权的说明

本人完全了解第十届“恩智浦”杯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，比赛组委会和恩智浦半导体公司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。

参赛队员签字：谭培鑫　耿铭良　杨子龙

指导老师签字：　何顶新　　池明

日　　期：　2017 年 8 月 13 日

摘 要

本文以第十二届“恩智浦”杯全国大学生智能车竞赛为背景，介绍了智能赛车控制系统的软硬件结构及开发流程。针对比赛的具体情况，我们设计的智能车系统是以 MK60DN512ZVLQ10 微控制器为核心控制单元，通过数字摄像头 OV7725 来采集赛道信息，并对采集到的图像进行处理，进行赛道元素判断进而控制车模以最佳路线运动；采用双编码器对车模两轮速度进行实时监控，通过 PID 算法对电机进行闭环控制，同时通过 PID 算法对小车舵机进行横向控制。在小车调试过程中，为了提高小车的稳定性，我们增添了 OLED 显示模块，五向按键模块，蓝牙接收模块，以及 SD 卡读写模块，经过大量长时间的调试与修改，我们的系统设计方案是可行的。

关键字：智能车；MK60DN512ZVLQ10；OV7725；PID

目录

第一章 引言

1.1 比赛背景介绍

教育部为了加强大学生实践、创新能力和团队精神的培养，在已举办全国大学生数学建模、电子设计、机械设计、结构设计等四大竞赛的基础上，委托教育部高等学校自动化专业教学指导分委员会主办每年一度的全国大学生智能汽车竞赛。并成立了由教育部、自动化分教指委、清华大学、飞思卡尔半导体公司等单位领导及专家组成的“‘飞思卡尔’杯全国大学生智能汽车竞赛”组委会。该竞赛是为了提高大学生的动手能力和创新能力而举办的，具有重大的现实意义。与其它大赛不同的是，这个大赛的综合性很强是以迅猛发展的汽车电子为背景，涵盖了控制、模式识别、传感、电子、电气、计算机和机械等多个学科交叉的科技创意性比赛，这对进一步深化高等工程教育改革，培养本新意识，培养硕士生从事科学、技术研究能力。

以智能汽车为研究背景的科技创意性制作，是一种具有探索性的工程实践活动，其本质也是人类创造有用人工物的一种训练性实践，其过程属性是综合，而结果属性很可能是创造。通过竞赛，参赛的同学们培养了对已学过的基础与专业理论知识与实验的综合运用的能力；带着背景对象中的各种新问题，学习控制、模式识别、传感技术、电子、电气、计算机、机械等多个学科新知识，包括来自不同学科背景大学生的相互学习，逐渐学会了在学科交叉、集成基础上的综合运用；若是以实用为目的，还必须考虑考虑可靠性、寿命、外观工业设计、集成科学与非科学，在具体约束条件下融合形成整体的综合运用。这样的训练是很有意义的。

在智能车的开发过程中，各参赛队伍需要改装竞赛车模，完成智能巡线小车的制作。在此过程中需要学习嵌入式系统开发环境与在线编程方法、单片机接口电路设计，自行设计并实现识别引导导线的硬件电路、电机的驱动电路、

车速反馈电路、智能车舵机控制电路以及能使小车在不驶出赛道的前提下可能快速行驶的控制策略与软件算法。

智能车的开发与设计涉及到多个专业领域，对于大学生综合素质的培养，知识面的拓展和分析问题解决问题的能力的提高很有意义，并且有利于提高大学生的动手能力、激发创新能力。此外，制作这样一个高性能智能小车的过程，也是需要同组成员相互协作、紧密配合的过程，在此过程中，团队成员的交流与合作也显得尤为重要。

为响应教育部的号召，本校积极组队参加第十二届“恩智浦”杯全国大学生智能汽车竞赛。从 2016 年 12 月开始着手进行准备，历时近 8 个月，经过设计理念的不断进步，制作精度的不断提高，经历几代智能车硬件平台及相关算法的改进，最终设计出一套完整的智能车开发、调试平台。作为光电组的华中科技大学光电竞速一队采用轻质量机械设计和连续化算法处理的基本技术路线，在前瞻距离、噪声抑制、驱动优化、整车布局等方面加强研究创新，在有限计算能力下获得了较高的赛道信息准确率。使智能车能够满足高速运行下的动力性和稳定性需求，获得了良好的综合性能和赛场表现。

本文将对智能车的总体设计和各部分的详细设计进行一一介绍。

1.2 概述

1.2.1 机械设计

机械方面，在保证车子灵活性和稳定性的前提下，尽量降低重心和减轻质量。为获取最佳的机械性能，还需注重各个小细节。比如舵机臂的安装，编码器的选型与固定方式，以及有利于车模进行巡线、提速条件下的电池、电路板和传感器的固定的研究和设计。

1.2.2 电路设计

电路设计主要是用来沟通机械与程序的，一个稳定高效的电路将是智能车稳定高速运行的保障。在经过多次改进之后，我们形成了以 K60 为核心的一整套稳定的电路系统。系统中子模块可分为电源部分、电机驱动、 编码器、调试

接口，数字摄像头等几个部分。

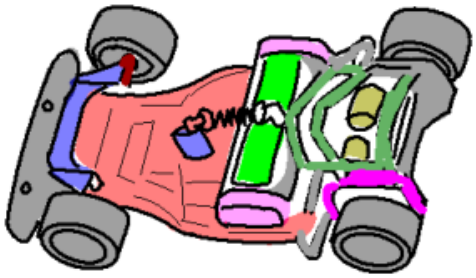
1.2.3 程序设计

程序是小车系统的控制核心，本届光电 C 车除了在方向、速度控制的基础上，同时还需要对斑马线、障碍物、坡道、环路等元素的识别，这其中的难度体现在要识别不同的元素并采取相应的控制策略以及对采回的信号进行处理，并且要使得程序上的处理变得更加灵活。

第二章 机械设计

2.1 车体结构

本届恩智浦光电四轮组比赛采用 C 型车模，组委会要求车模改装完毕后，长度（包括传感器）不超过 40cm，宽度不超过 25cm，高度不超过 40cm。

编号	车模外观和规格	赛题组	供应厂商
C 型车模	 电机 RN-260 ，舵机：FUTABA3010	A.光电四轮 D.电磁节能 E.光电追逐 G.H.创意组	东莞市博思电子数码科技有限公司

智能车竞赛所使用的车模是一款带有两个电机的四轮驱动模型赛车，它由大赛组委会统一提供，其机械结构简单、材料轻便强硬、零件加工精度较高。下面具体介绍 C 型车模的基本参数。

表 2.1C 型车模的基本参数

基本参数	尺寸
轴距	200mm
前轮距	134mm
后轮距	144mm
车轮直径	26mm
传动比	15/122

2.2 摄像头安装

摄像头的安装要求结实稳固， 减少车正常行进过程中的晃动，并且能够经受得住意外的碰撞，我们采用了一根外径 8mm 材质轻巧坚固碳纤维杆作为主杆，底部采用塑料制底座固定在车模主体上，顶部采用自制的铝制支架固定摄像头，同时使用环氧树脂板雕制连接件将摄像头与支架相连接，并将摄像头的 FFC 线夹紧防止在车体跑动过程中，FFC 线脱落。整套装置具有很高的定位精度和刚度，使数字摄像头镜头便于拆卸和维修。数字摄像头固定如下图 2.2 所示。

图 2.2 数字摄像头安装示意图

2.3 电池的固定

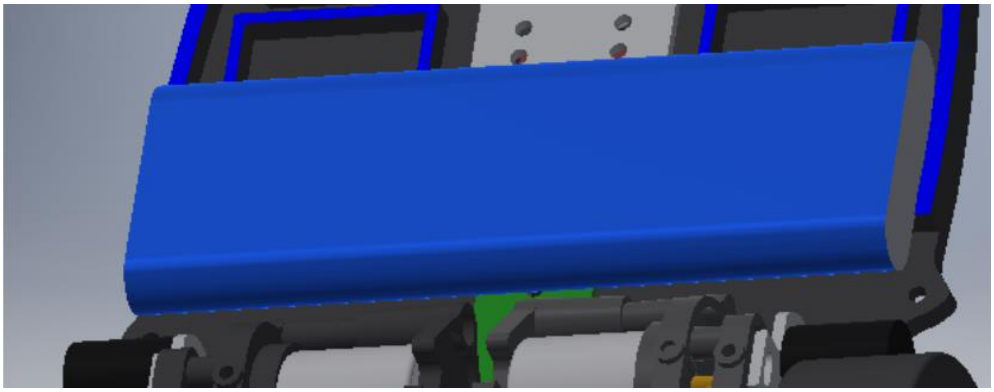


图 3-3 电池固定示意图

由于电池的重量较重，在整车中占有很大的比例，影响着重心的位置，所以电池固定的位置至关重要。我们选择将电池尽量靠后安装，并且保证电池水平，以保证车的稳定性。同时为了防止电池的甩动，我们用扎带进行了固定。

2.4 电路板的安装

为了使小车具有较好的稳定性及转向性能，我们在组装小车时尽量降低重心，因此也将电路板地板前端紧贴底板的位置，从而实现降低重心，提高小车的稳定性。

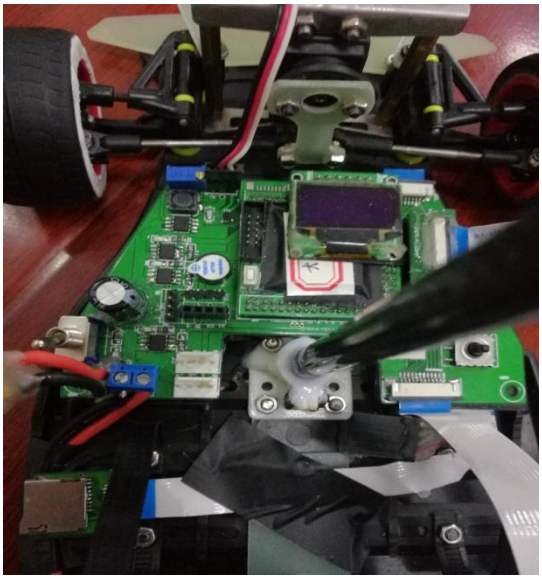


图 2.4 电路板安装示意图

2.5 底盘高度调整

理论上底盘越低，车的重心就越低，小车在行驶过程中就越稳定，但由于坡道和障碍，使底盘不能像以往一样几乎贴着赛道行驶。若底盘过低，则在上坡时会损坏赛道，过障碍时可能会被障碍顶住。只能在保证小车行驶时不碰到坡道和障碍的前提下底盘尽量低。底盘通过前轮支撑架下面加垫片和后轮轴承处调整。

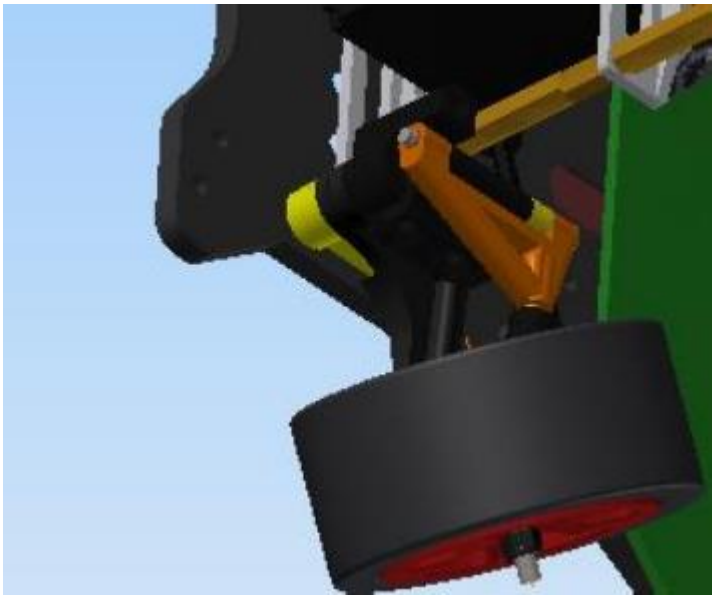


图 2.5 前轮支撑架加垫片示意图

2.6 编码器的固定

为了提高精度，本车使用了龙邱 1024 线编码器。为了便于安装，使用加工件将编码器与电机齿轮进行连接，较好的保证了两啮合轮轴的平行度及传动的平顺性。

图 2.6 编码器固定示意图

2.7 智能车前轮定位的调整

前轮定位对赛车的速度有很大的影响，虽然车模比较小，但是前轮定位作用还是不容忽视的。前轮定位的内容有：主销后倾、主销内倾、主销外倾、前轮前束。

2.7.1 主销

转向轮（前轮）围绕主销-进行旋转，前轴的轴荷通过主销传给转向轮，具备这两点的就叫做主销。主销内倾角和车轮外倾角度主要是由转向节决定。

2.7.2 主销后倾角

主销向后倾斜，主销轴线与地面垂直线在赛车纵向平面内的夹角称为主销后倾角。前轮重心在主销的轴线上，由于主销向后倾斜使前轮的重心不在车轮与地面的接触点上，于是产生了离心力，主销后倾形成的离心力，可以保证汽车直线行驶的稳定性，还可以帮助车轮自动回正。主销后倾角延长线离地面实际接触越远，即主销后倾角越大，车速越高，离心力越大，前轮自动回正的能

力就越强，行驶就更加稳定。

主销后倾角在车轮偏转后形成一回正力矩，阻碍车轮偏转，主销后倾角越大，车速愈高，车轮偏转后自动回正力矩越强。但回正力矩过大，将会引起前轮回正过猛，加速前轮摆振，并使转向沉重，通常后倾角为 1～4 度，对于智能汽车模型，通过改变前后红色垫片的数量可以改变它的主销后倾角。

2.7.3 主销内倾角

主销在横向平面内向内倾斜，主销轴线与地面垂直线在赛车横向断面内的夹角称为主销内倾角。

主销内倾角也有使轮胎自动回正的作用，当汽车转向轮在外力作用下发生偏转时，由于主销内倾，则车轮连同整个汽车的前部将被抬起一定高度，在外力消失后，车轮就会在重力作用下力图恢复到原来的中间位置。角度越大前轮自动回正的作用就越强，但转向时也就越费力，轮胎磨损增大；反之，角度越小前轮自动回正的作用就越弱。通常汽车的主销内倾角不大于 8°，主销内倾的调整应该保持在一个合适的范围，“一般来说 0~8 度范围内皆可”。但主销内倾角不宜过大，否则在转弯时轮胎将与赛道间产生较大的滑动，从而会增加轮胎与路面间的摩擦阻力，使转向变得沉重，同时会加速轮胎的磨损。在实际的调整中，只要将角度调整为 5 度左右就会对于过弯性能有明显的改善。如果赛道比较滑，可以将这个角度再调节的大一些。在实际制作中，这个角度调节为 5 度左右。

对于模型车，通过调整前桥的螺杆的长度可以改变主销内倾角的大小。主销内倾和主销后倾都有使汽车转向自动回正，保持直线行驶的功能。不同之处是主销内倾的回正与车速无关，主销后倾的回正与车速有关，因此高速时主销后倾的回正作用大，低速时主销内倾的回正作用大。

2.7.4 前轮外倾角

通过车轮中心的汽车横向平面与车轮平面的交线与地面垂线之间的夹角，称为前轮外倾角。前轮外倾角是前轮的上端向外倾斜的角度，对汽车的转向性

能有直接影响，它的作用是提高前轮的转向安全性和转向操纵的轻便性。如果前面两个轮子呈现“V”字形则称正倾角，呈现“八”字则称负倾角。如果车轮垂直地面一旦满载就易产生变形，可能引起车轮上部向内倾侧，导致车轮联接件损坏。前轮外倾可以抵消由于车的重力使车轮向内倾斜的趋势，减少赛车机件的磨损与负重。所以事先将车轮校偏一个正外倾角度，一般这个角度约在 1°左右，以减少承载轴承负荷，增加零件使用寿命，提高汽车的安全性能。

2.7.5 前轮前束

前轮前束是前轮前端向内倾斜的程度，也指前轮中心线与纵向中心线的夹角。当两轮的前端距离小后端距离大时为内八字，前端距离大后端距离小为外八字。由于前轮外倾使轮子滚动时类似于圆锥滚动，从而导致两侧车轮向外滚开。但由于拉杆的作用使车轮不可能向外滚开，车轮会出现边滚变向内划的现象，从而增加了轮胎的磨损。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。前轮外八字与前轮外倾搭配，一方面可以抵消前轮外倾的负作用，另一方面由于赛车前进时车轮由于惯性自然的向内倾斜，外八字可以抵消其向内倾斜的趋势。外八字还可以使转向时靠近弯道内侧的轮胎比靠近弯道外侧的轮胎的转向程度更大，则使内轮胎比外轮胎的转弯半径小，有利与转向。在实际的汽车中，一般前束为 0~12mm 。

前束的调整总是依据主销内倾的调整。只有主销内倾确定后才能确定合适的前轮前束与之配合。前轮前束的调整是方便的。主销内倾的调整由于要拧开螺丝钉，固定件又为塑料，所以频繁的调整容易引发滑丝现象。而前束不会，所以调整前束是最安全、方便的。前束在摩擦大的时候有明显的效果。但是一定不要太大，适当的放开一两圈就够了。

在模型车中，前轮前束是通过调整伺服电机带动的左右横拉杆实现的。主销在垂直方向的位置确定后，改变左右横拉杆的长度即可以改变前轮前束的大

小。在实际的调整过程中，我们发现较小的前束，约束 1mm 可以减小转向阻力，使模型车转向更为轻便，但实际效果不是十分明显。调节合适的前轮前束在转向时有利过弯，还能提高减速性。将前轮前束调节成明显的内八字，运动阻力加大，提高减速性能。由于阻力比不调节前束时增大，所以直线加速会变慢。

2.8 阿克曼转向仿真

2.8.1 阿克曼转向原理

阿克曼原理的基本观点是汽车在行驶线行驶和转弯行驶过程中每个车轮的运动轨迹都必须完全符合它的自然运动轨迹，从而保证轮胎与地面间处于纯滚动而无滑移现象。因此阿克曼转向要求前轮在转向时左右轮转角满足一定关系，如图 2-6 所示。

左右轮转角应满足下面的关系式：

ctg β-ctg α=K/L

式中

β——汽车前外轮转角

α——汽车前内轮转角

K——两主销中心距

L——轴距

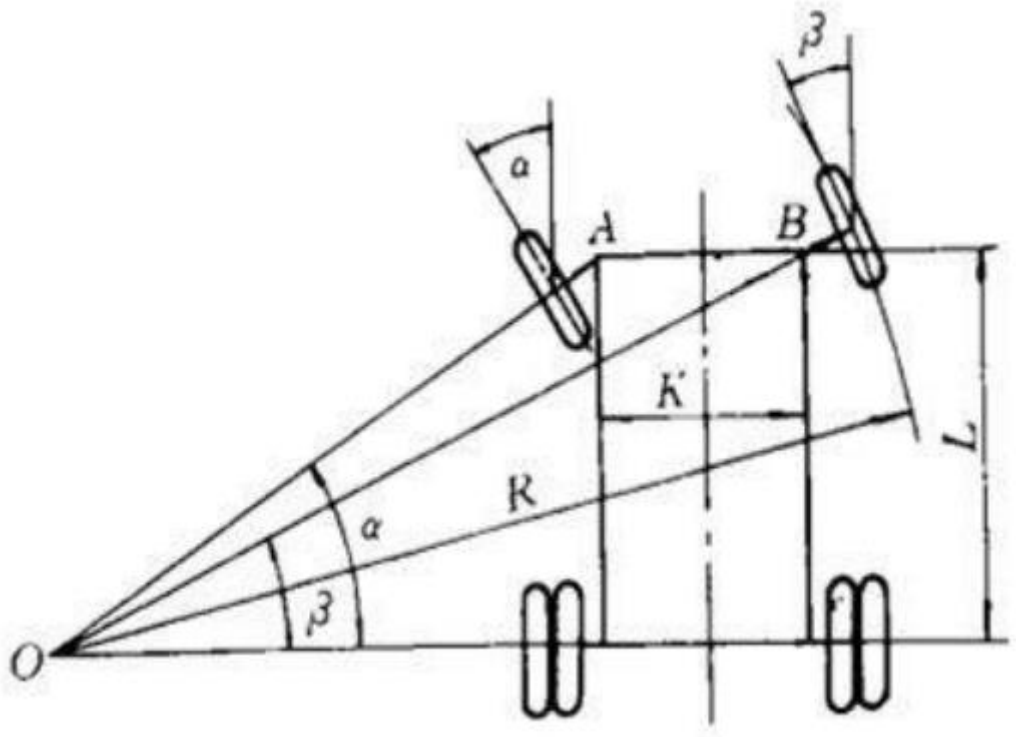


图 2-7 阿克曼转向示意图

在此次仿真过程中我们将转弯半径定义为：

R=ctg α*L+0.5*K= ctg β*L-0.5*K。

2.8.2 舵机安装方式

（1） 水平方向

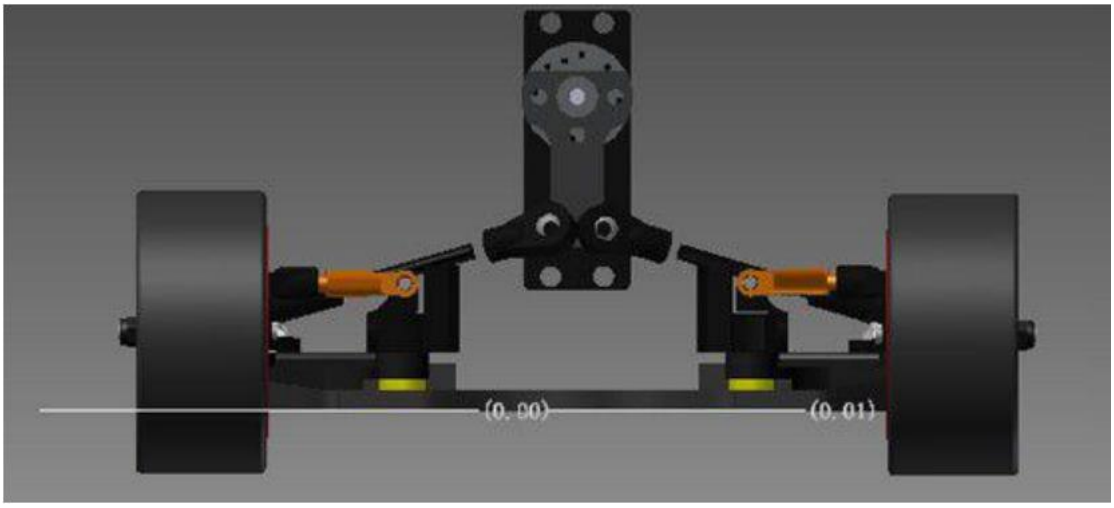


图 2.8 正视八字

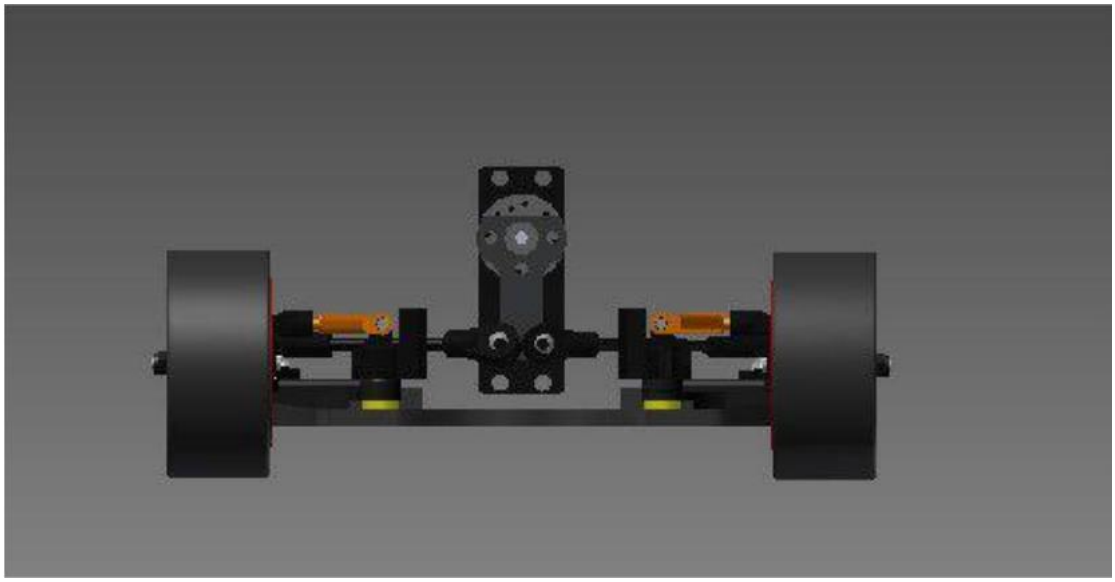


图 2.9 正视一字

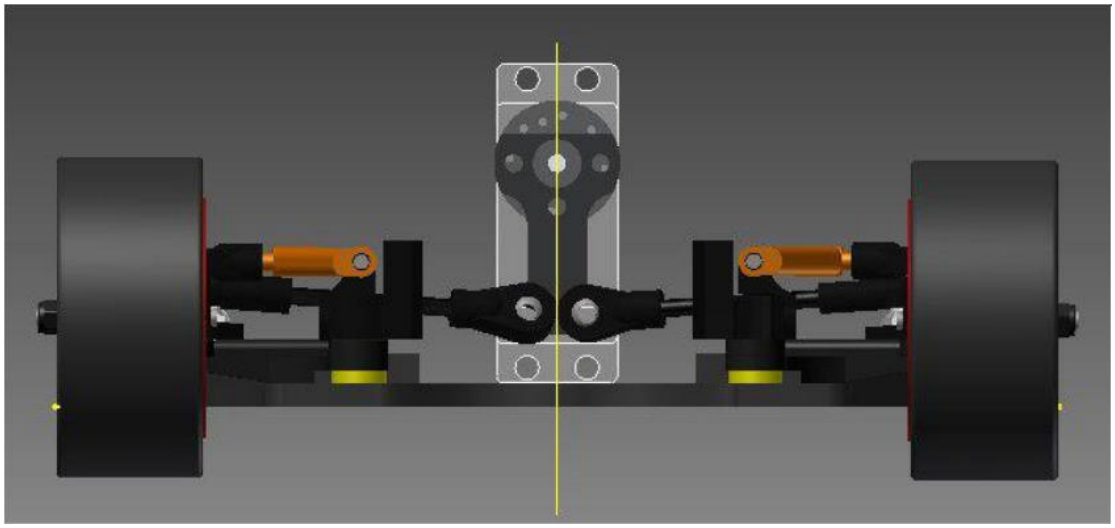


图 2-10 正视 V 字

（2） 垂直方向

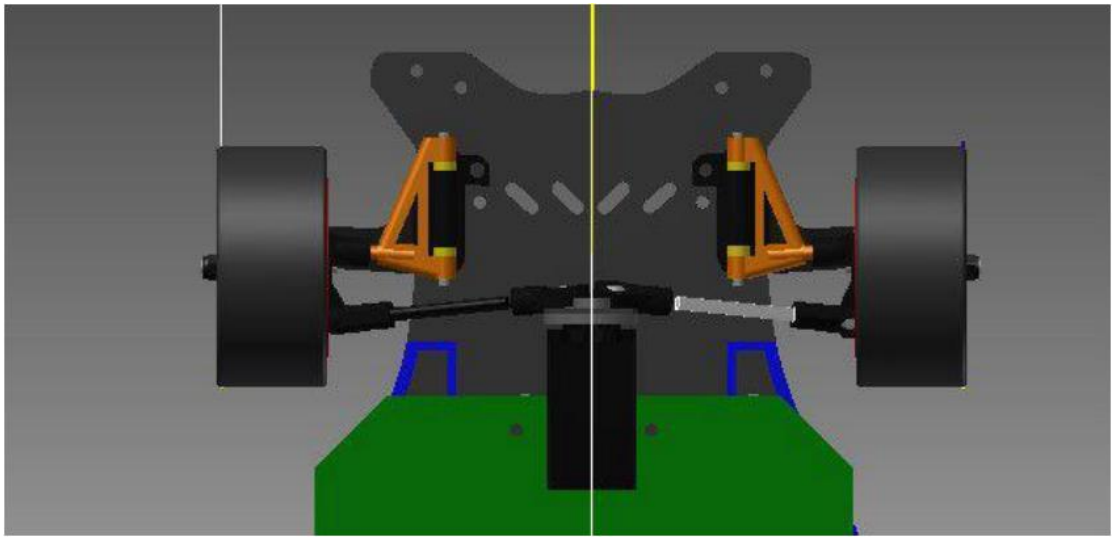


图 2-11 俯视八字

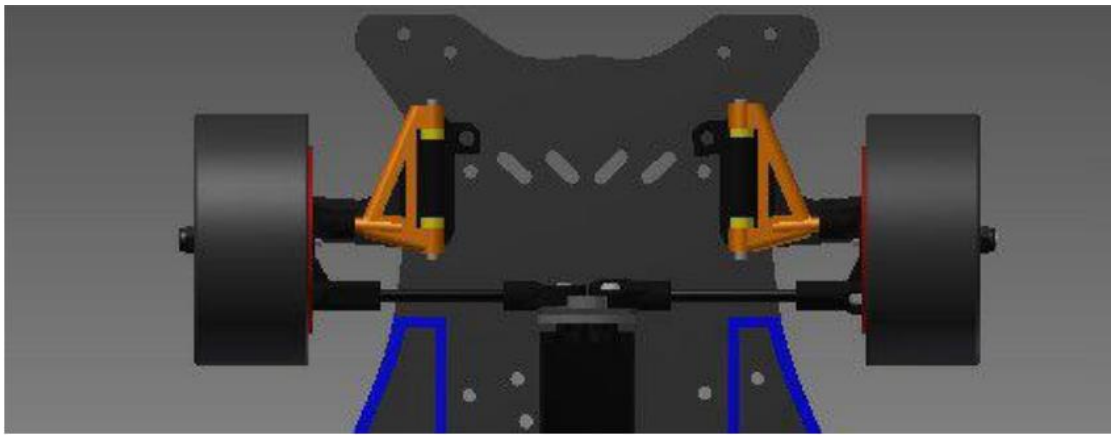


图 2-12 俯视一字

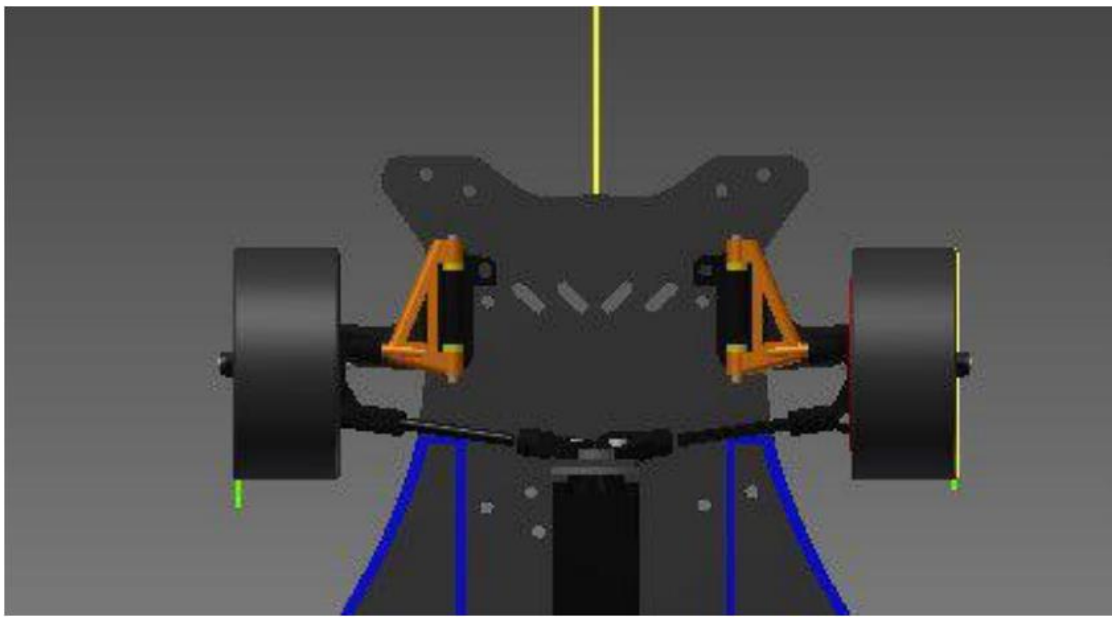


图 2-13 俯视 V 字

2.8.3 仿真结果

通过仿真数据，作图分析，对每种安装方式进行对称性、线性、实际与理论的拟合程度进行分析，最终舵机安装方式选择正视 V 字，俯视一字的安装方式。

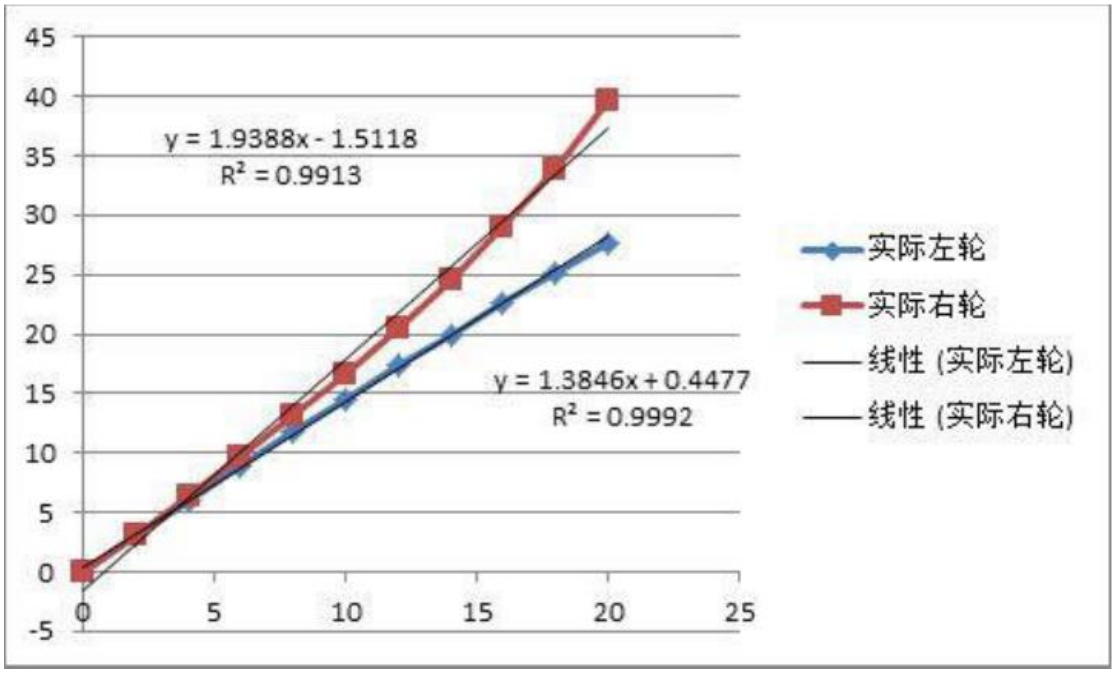


图 2-14 线性示意图

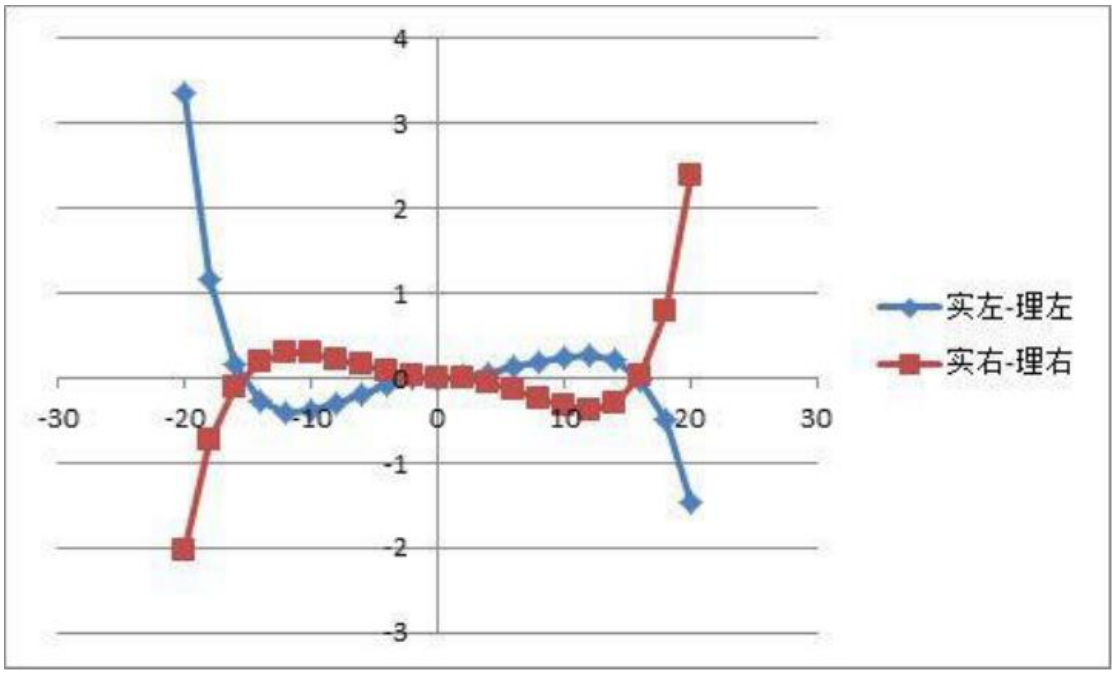


图 2-15 实际与理论拟合程度分析示意图

2.9 后轮差速

由于是双电机，所以采用的是程序差速。由实地测试得到舵机打角与后轮差速的关系如下表。

舵机臂打角/度	内轮转弯半径/mm	外轮转弯半径/mm
20	135.4325	284.505
19	153.999	303.179
18	180.1913	329.371
17	203.543	352.723
16	230.817	379.997
15	264.626	413.806
14	295.254	444.434

13	331.8993	481.0793
12	372.1908	521.3708
11	423.426	572.606
10	479.431	628.611
9	546.5655	695.7455
8	633.6114	782.7914
7	742.9217	892.1017
6	889.2498	1038.43
5	1086.076	1235.256
4	1382.536	1531.716
3	1880.439	2029.619

表 2-2 舵机打角与后轮差速关系表

2.10 齿轮啮合

齿轮传动机构对车模的驱动能力有很大的影响。齿轮传动部分安装位置的不恰当，会大大增加电机驱动后轮的负载，会严重影响最终成绩。调整的原则是：两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，浪费动力；传动部分要轻松、顺畅，不能有迟滞或周期性振动的现象。

判断齿轮传动是否良好的一句是，听电机带动后轮空转的声音，若声音刺耳，则是齿轮撞齿的声音，说明齿轮间隙过大；声音沉闷，则表明电机负载过重，说明齿轮间隙太小。同时应注意齿轮互相平行，并尽量保证表面在同一平面。因为有电机和差速盘，编码器和差速盘两处齿轮啮合，可以先调整电机与差速盘的间隙，调至无明显噪声后在调节编码器与差速盘间的啮合。调节至电机全速转动时齿轮间的噪声很小。齿轮传动机构对车模的驱动能力有很大的影响。由于齿轮是不能改变的，所以我们只通过涂润滑油使传动更加顺畅。

2.11 前后固定方式

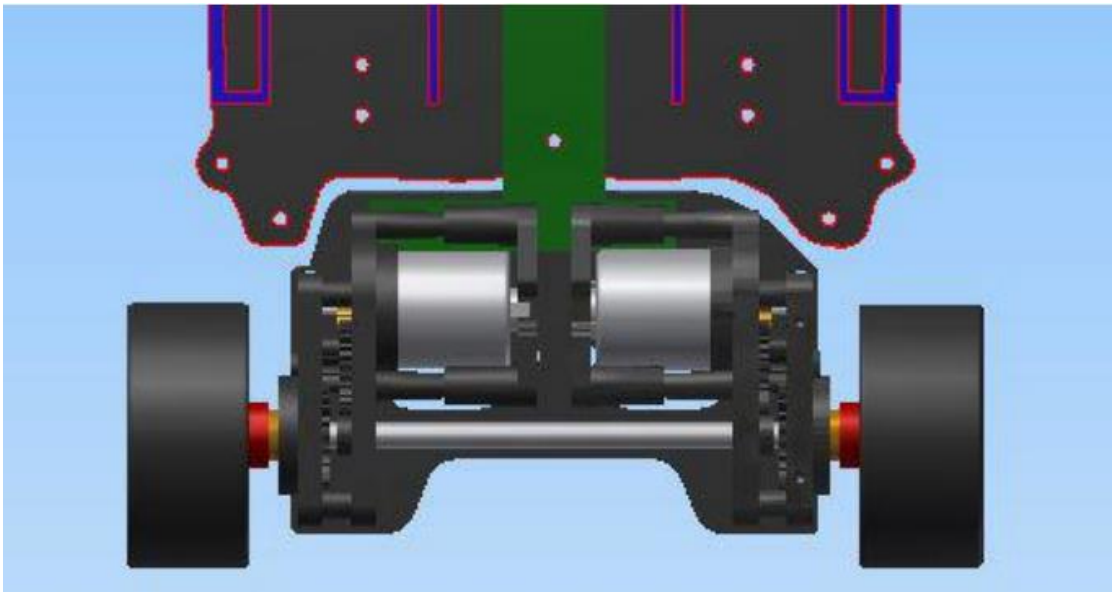


图 2-16 固定板示意图

将车模前后部分的连接块改为强度较大的环氧树脂板，减轻小车高速行驶时的抖动，以增强车的稳定性。

第三章 电路设计

3.1 电路系统框图

3.1.1 电源部分

电源部分主要包括：电池电源保护电路、电源管理、主控制板和传感器电源、 电机驱动、辅助调试电源。电源部分结构框图如下：

图 3-1 电源部分系统框图

3.1.2 最小系统部分

最小系统为小车控制中枢，包括：MCU 复位，通讯，人机交互，外围硬件接口，结构框图如下：

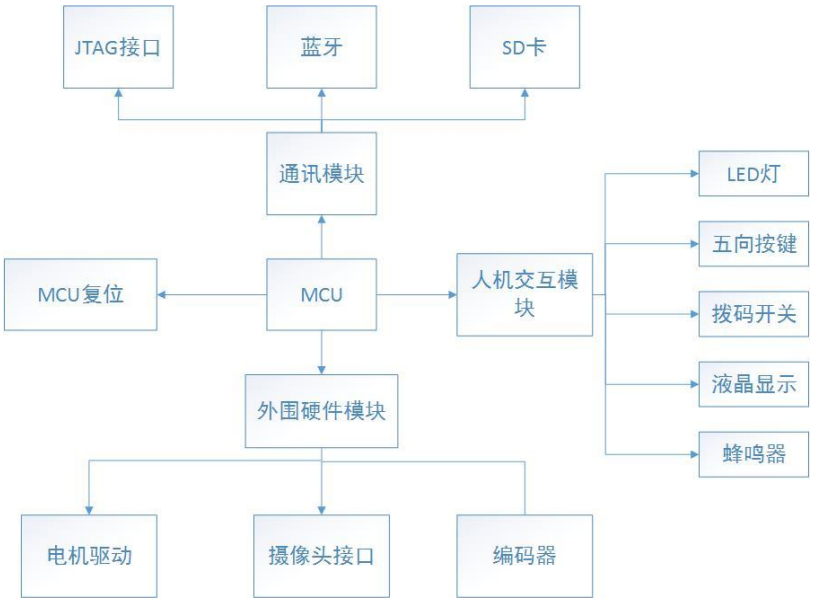


图 3-2 最小系统部分框图

3.2 电源部分

小车的电源部分是小车硬件电路设计中十分关键的一环，因为芯片的正常稳定工作是需要一个正确稳定的供电电压的，对于电压波动较大的电路，其对器件的损害是不言而喻的，轻则器件不能工作，重则烧毁芯片，所以，在设计电源部分时，我们应当根据我们所用器件对供电电压的需求而进行设计，同时要考虑不同电压之间的干扰问题。

3.2.1 电源开关指示保护

由于智能车使用的是直流电源，故一旦电池的正负极接反将造成很严重的后果，很有可能烧坏主控制板，故考虑在电源的最开始部分设计仿插反的结构和插反后的保护电路。一般来说，二极管拥有良好的单向导电性，故可考虑用二极管作为反接保护器件。

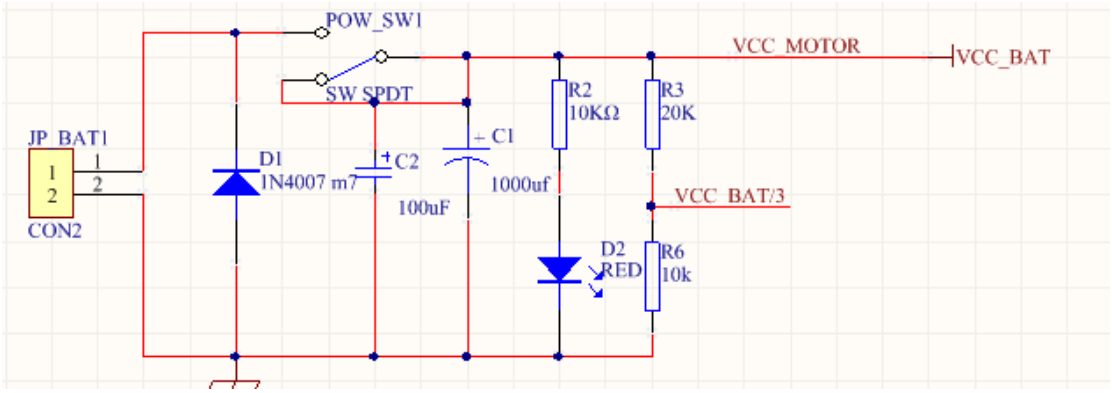


图 3-3 电源开关指示保护

3. 2. 2 最小系统供电

由于我们采用的主控芯片是 K60，为 3.3V 供电，故需要将电池电压分压至 3.3V。在市场上有很多 3.3V 的稳压芯片，依据类型可以分为开关稳压和线性稳压，其中开关稳压的转化效率高，但是纹波大；线性稳压的转化效率低但是纹波小，而 K60 对于供电电压比较敏感，电压波动大，很可能造成芯片低压复位，所以我们选择使用线性稳压芯片。在线性稳压芯片中含有 LM，TPS 系列的，为了保证最小系统板的稳定工作，我们最终采用了 TPS7333 作为最小系统的供电芯片，其优于 LM1117 的是其纹波小，只有几十毫伏，并且在热保护以及防静电方面就要优于 LM1117。如下为 TPS7333 的电路图：

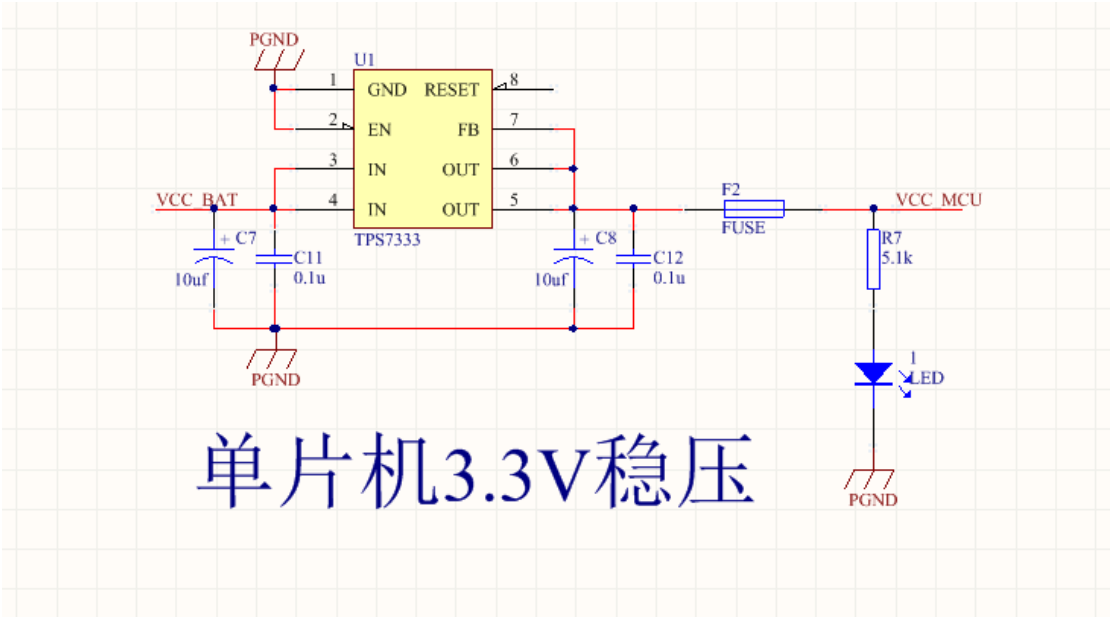


图 3-4 最小系统稳压

3.2.3 外设 3.3V 与 5V 供电

在该部分中主要的器件是数字摄像头，以及一些辅助调试器件：OLED，五向按键，拨码开关，蓝牙，SD 卡，蜂鸣器。之所以要将这部分供电与最小系统板的供电分隔开，主要是怕这些外设器件会影响最小系统板供电，一旦某个外设出现故障就可能导致供电不稳，则将影响最小系统板正常工作，所以为了整个硬件电路更稳定，将其分隔供电要好很多。同时我们采用的 3.3V 稳压芯片为 TPS76833，该芯片与 7333 的区别为其最大的电流为 1A 而 7333 的最大电流为 500mA；5V 稳压芯片为 TPS76850。如下为外设 3.3V，5V 稳压电路图：

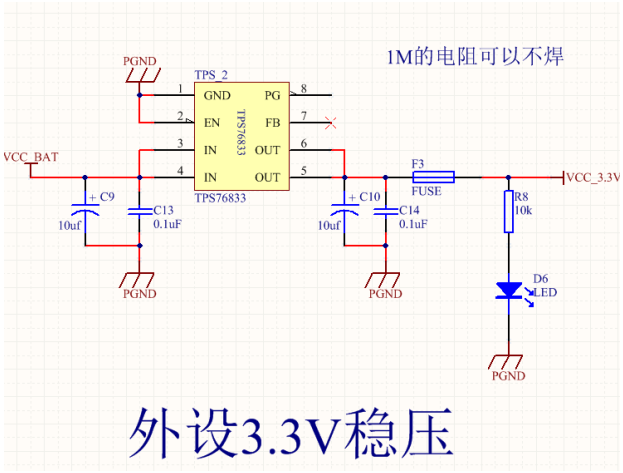


图 3-5 外设 3.3V 稳压

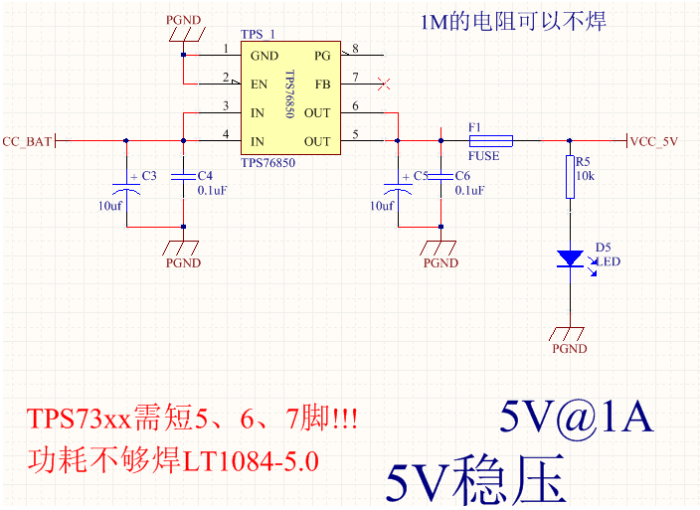


图 3-6 外设 5V 稳压

3.2.4 舵机电源

舵机的使用手册中推荐使用 5V 或 6V 给舵机供电，5V 下使用舵机能有效延长舵机的使用寿命，但由于比赛的性质，更高的电压可以带来更快的响应，因此没有使用 5V 供电，考虑到电源和压降，设计时选择了 5.2V 稳压，使用性能良好，同时延长舵机的使用寿命，保护舵机。舵机供电，我们没有实际测量出舵机的工作电流最大值，但是据网上找到的资料显示舵机的峰值电流能达到 3A，所以必须选择能够提供大电流的稳压芯片，同时我们也考虑过给更高的电压，所以就将舵机稳压设计为可调的，这样就可以很便捷的调节舵机电压，同时了解那个电压段能更好的提高舵机的反应速度，所以我们选用了 AS1015 作为舵机的稳压芯片，因为 AS1015 的最大电流能达到 5A。如下为舵机稳压电路图：

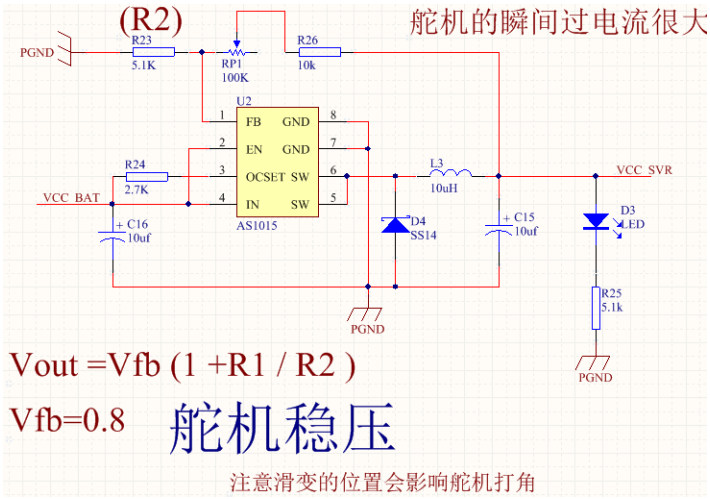


图 3-7 舵机稳压电路图

3.3 电机驱动模块

作为竞速组，速度快为第一要素，而小车的速度实现与电机的驱动密不可分。因为小车的电机为大赛组委会唯一指定，我们禁止自行改装，所以我们能

想办法提升的就是小车的驱动电路。

我们组采用的驱动电路为经典的H桥控制电路，该电路为通过控制四个MOS管的开通和关断来控制电机的运转，其电路图如下：

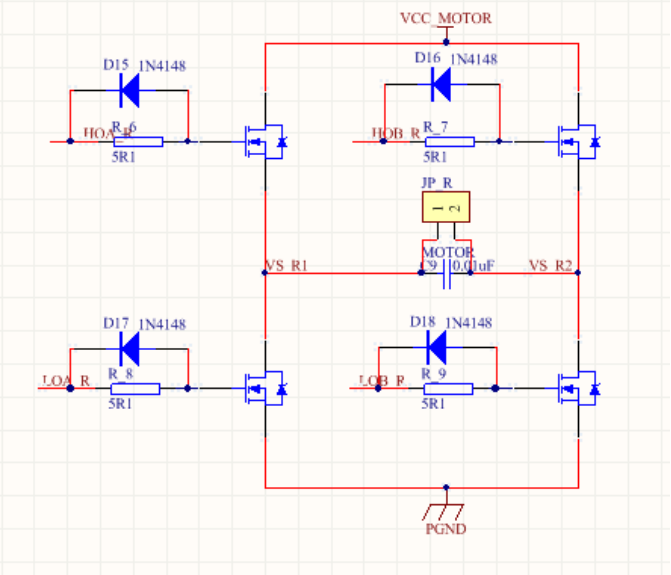


图 3-8H 桥电路图

从上图我们可以看出四个 MOS 管与电机直接相连，即在电机通电过程中MOS 管是串在电机上的，则 MOS 管必将分去一部分电压，而我们希望电机是全压工作，这样电机的转速才更快。所以选择一款好的 MOS 管是提升驱动能力的一个途径，我们组采用的 MOS 管为 IRLR7843，该款 MOS 的 Rds=3.3mΩ，最大的通过的电流能达到 Id=161A，同时栅极电荷为 Qg=34nC，栅极电荷会影响 MOS 的开关速率，一般小一些，开关速率要快。

在选好 MOS 管后，还有一款芯片也是很重要的，那就是电机的驱动芯片。Mos 管的开通和关断是通过驱动芯片的控制来完成的，当驱动信号从最小系统板中出来以后，不能直接接在 MOS 管上，因为最小系统板不能提供这么大的驱动能力，所以要经过驱动芯片来驱动 MOS 管，在驱动芯片上，我们选择的是 IR 公司的 IR2104， IR2104 型半桥驱动芯片可以驱动高端和低端两个 N 沟道 MOSFET，能提供较大的栅极驱动电流，并具有硬件死区、硬件防同臂导通等功能。使用两片 IR2184 型半桥驱动芯片可以组成完整的直流电机 H 桥式驱动电

路。由于其功能完善，价格低廉容易采购，所以我们选择它进行设计。如下为2104的驱动示意图：

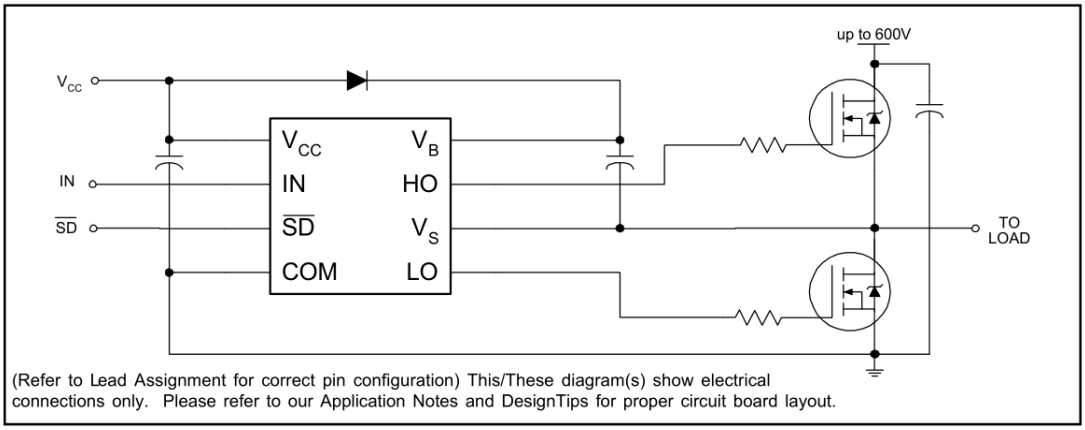


图 3-9 IR2104 驱动示意图

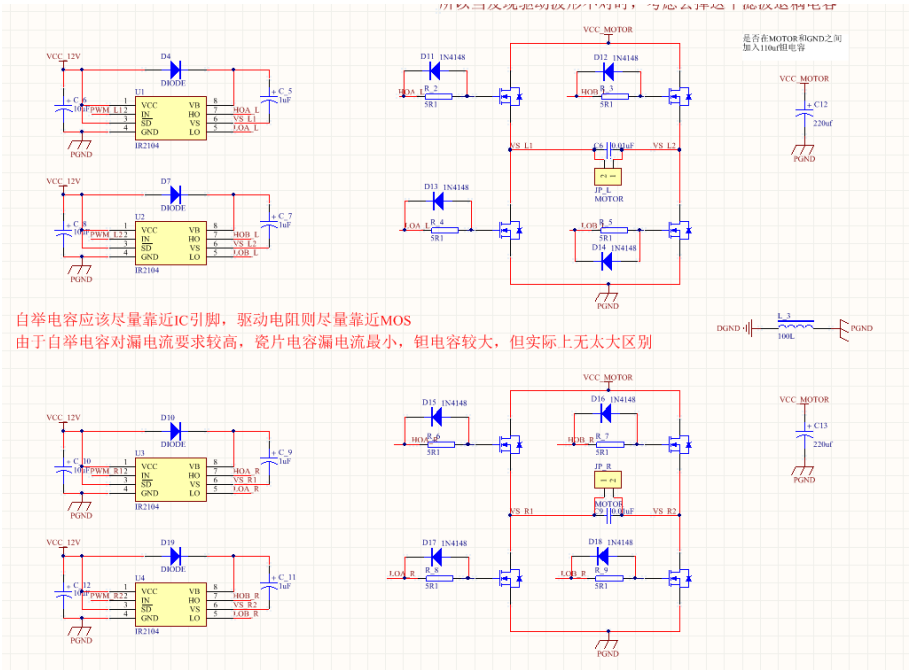


图 3-10 电机驱动原理图

3.4 编码器

为了进行精确的速度控制，必须及时的知道小车的时时刻刻的速度，就需用于测速的编码器。在编码器的选型上，我们选择的是 1024 线 mini 型编码器，输出两相正交方波。相比较传统光电码盘式编码器，1024 线光电编码器精度更

高，稳定性也更好。

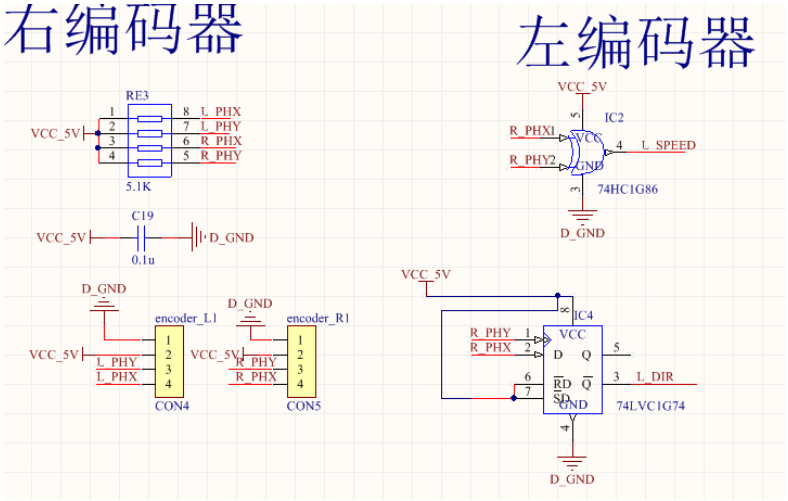


图 3-11 编码器接口原理图

上拉电阻 RE3 把编码器输出信号拉高是为了保证编码器不工作的时不产生随机信号；D 触发器实现鉴相功能；后一级电路的异或门把输出的两个信号进行倍频，可以提高分辨率（例如把原脉冲数为 1000，倍频后脉冲数为 2000，这样 就可以把分辨率从 0.36 度提高到 0.18 度）。

3.5 数字摄像头

作为光电组，光电传感器的选择是十分重要的，在市场上主要有 CCD 与 CMOS 摄像头，数字与模拟摄像头。CCD 摄像头具有对比度高、动态特性好的优点，但需要工作在 12V 电压下，对于整个系统来说过于耗电,而且 CCD 体积大，质量重，会抬高车体的重心，这对于高速情况下小车的行驶非常不利。我们组之前使用的就是 SONY 960H CCD 摄像头，该摄像头在工作时，发热大，功耗大，同时体积也大，车体在运动过程中，摄像头杆晃动厉害，有时就造成误判，并且该摄像头为模拟摄像头，传输信号为模拟信号，需要进行硬件二值化，同时对于干扰要求较高，如果电路设计不合理，将导致图像失真严重，雪花点很多。

与之相比，COMS 摄像头具有体积小、质量轻、功耗低，图像动态特性好等优点，因为小车对图像的清晰度，分辨率要求并不高，所以我们组选用的是野火鹰眼 OV7725 数字摄像头。该款摄像头较我们之前使用的模拟摄像头，优点

在于：OV7725 的帧率要高，即每秒能得到的图像数据多，同时体积小，输出的数据为数字信号，抗干扰强，硬件实现简单。

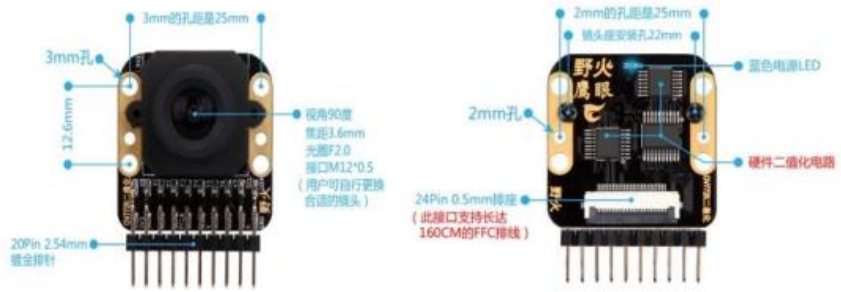
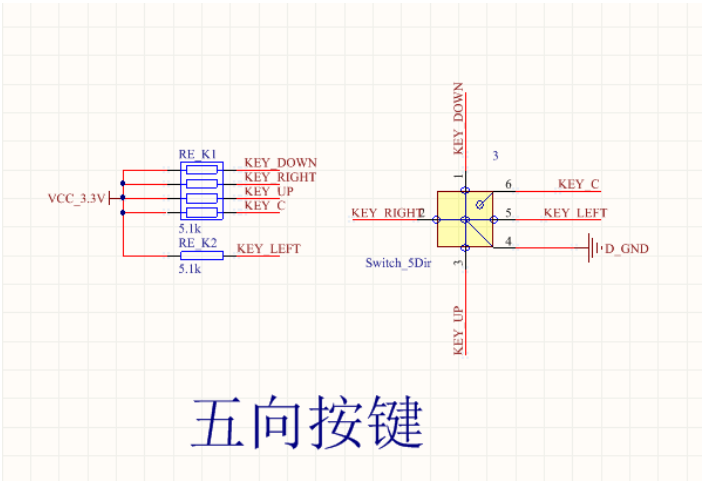


图 3-12　　OV7725 实物图

3.6 辅助调试接口

在该部分中，主要是人机交互与通讯模块，其中包括：OLED，五向按键，拨码开关，蓝牙模块，LED 指示灯，蜂鸣器。OLED 是用来显示小车重要参数的，通过五向按键，可以对这些参数进行更改，从而避免多次烧写程序；拨码开关与五向按键作用类似，用来决定一些比较重要的选择条件，或是选择小车的速度档位；蓝牙模块则是用来将一些信息实时发送给调试者，便于调试者实时掌握小车的运行情况；LED 灯则作为一些指示灯，用来表明程序运行的情况；蜂鸣器的作用则是用来通过改变声调的变化来显示各个特殊元素。

如下为辅助接口的电路图：



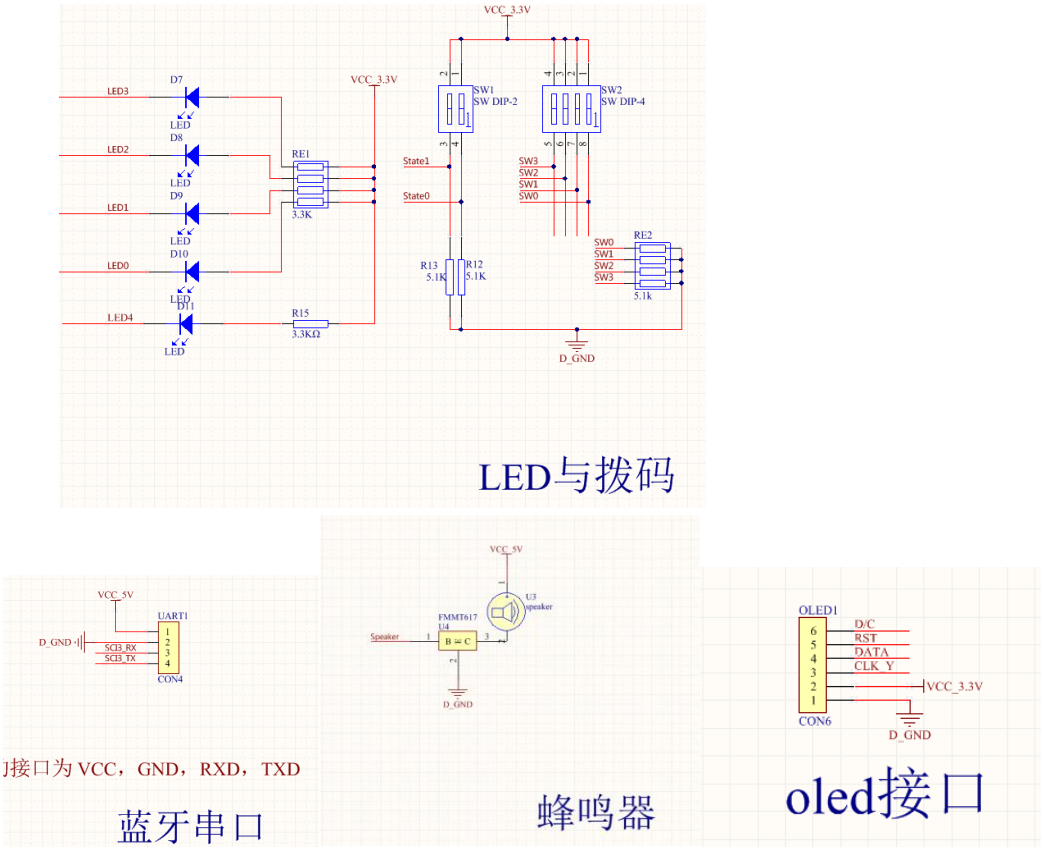


图 3-13 辅助接口电路图

第四章 软件部分

4.1 图像采集与分析

图像的采集，通过 DMA 的方式将数字摄像头得到的整幅图像数据存入内存，通过数组的下标便可以知道点的坐标。然后对图像进行压缩， 得到每行边缘点，通过搜索算法求得两边路，最终提取出比较真实的道路，主要流程如下：

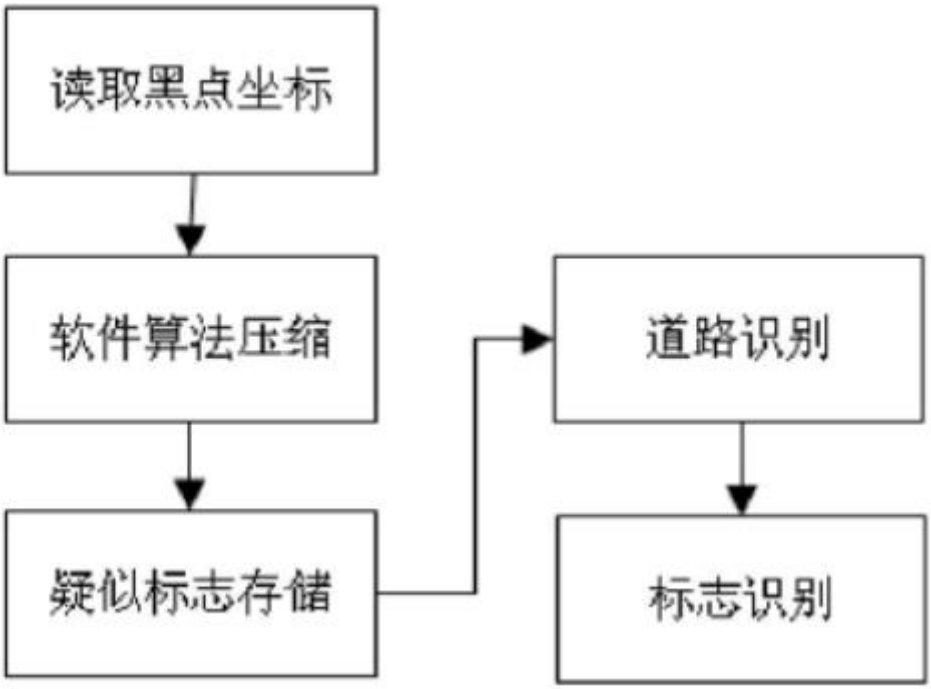


图 4.1 图像采集算法流程

4.1.1 图像的读取

由于 K60 内部存储空间足够，所以我们将摄像头获得的一幅图像通过上位机显示如下：

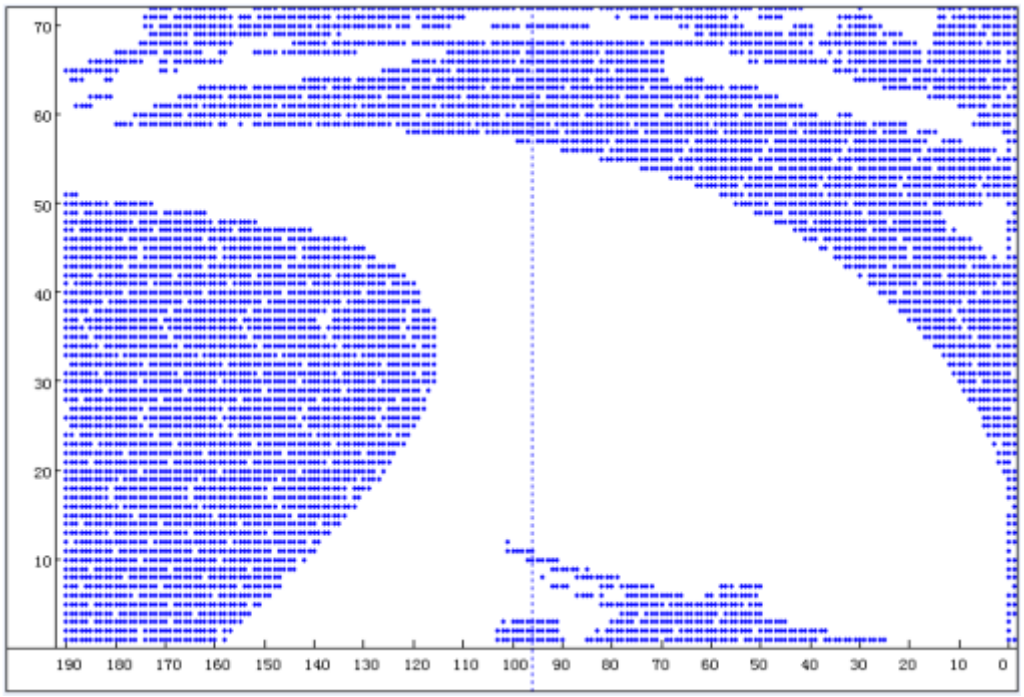


图 4.2 完整图像

4.1.2 图像压缩

由于完整图像在的分辨率为 160x 80，如果直接对于图像矩阵进行处理，十分消耗时间和空间。因此，我们在算法中，对于完整的图像进行滤波处理，最终将左右两边的道路分别存储为长度是 80 的数组，其下标作为行数，内容为本行跑道的坐标。如图:

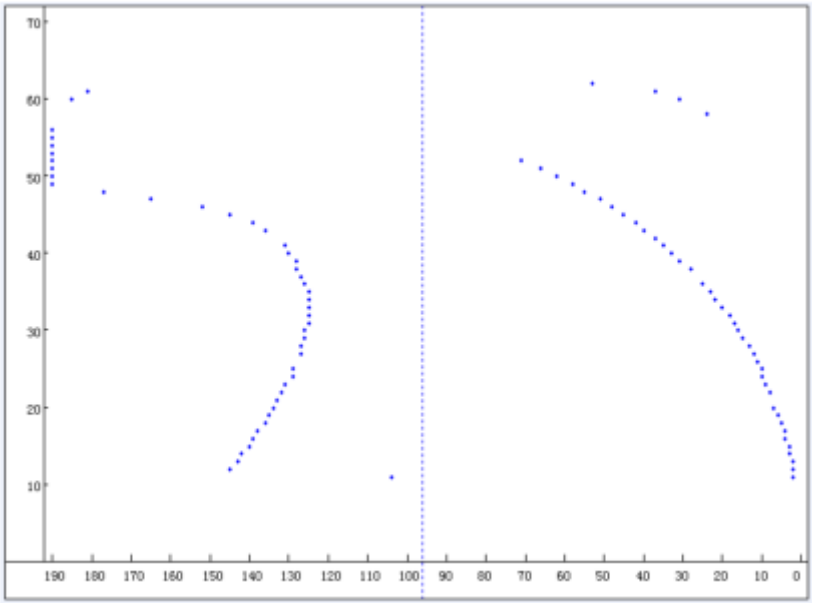


图 4.3 压缩后的图像

通过两边的路最后得到中间的一条路径如下图：

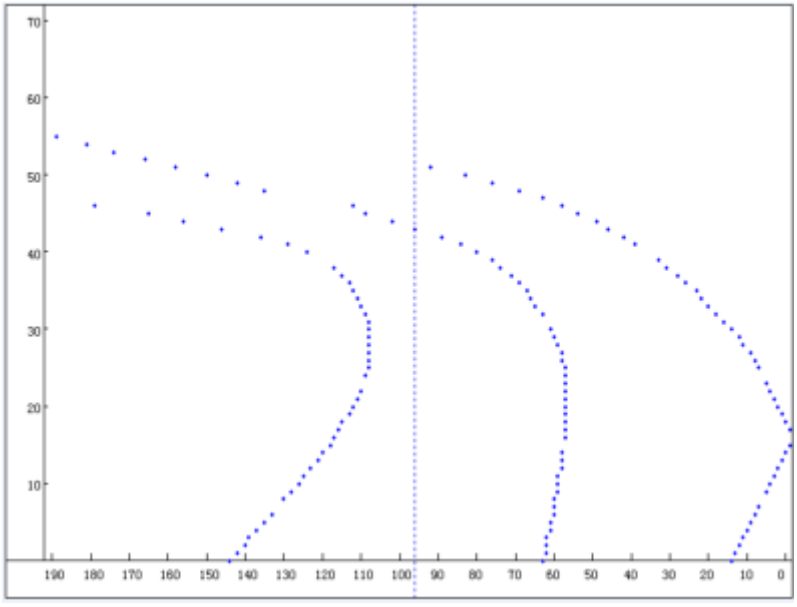


图 4.4 得到中间路径

4.1.3 赛道元素识别

摄像头传感器信息十分丰富，能不能充分挖掘其优势，就取决于图像分析算法的优劣，摄像头的图像具有以下特点：视野较大，盲区较小，信息丰富，

前瞻较远，可能会有噪点，动态时信息可能会有损失。针对以上特点，制定合适的分析算法如下：

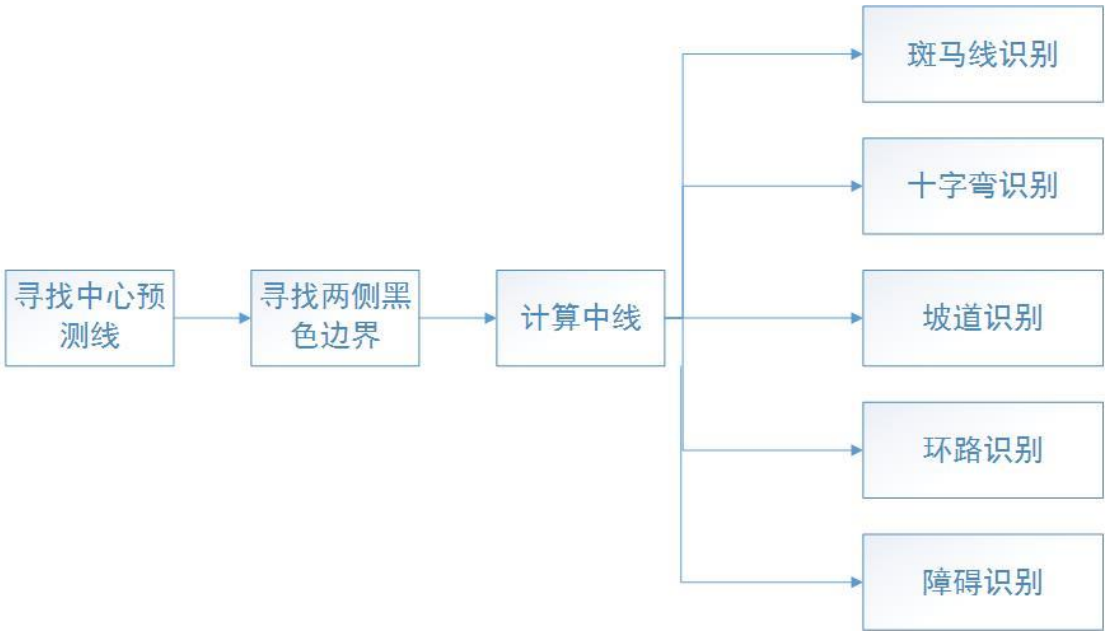


图 4-5 赛道元素识别

寻找中线预测线的基本思想是，预测一条以第一行中点为起点的直线为中线，在众多这样的直线中找到一条有效信息最多，也即有效行最多的直线，预测其为和中线拟合成都最高的直线，并以此确定结束行。

寻找两侧黑线的基本思想为，从预测中线开始，每行向两侧寻找，直至找到黑色边界，从第一行到结束行依次进行该操作。计算中线直接由边界去平均。

十字弯识别基本思想为，计算出的边界中有连续的两边都丢线的情况出现，连续多少行依情况而定。

中心引导线识别基本思想为，视场中出现由白到黑并由黑到白，且宽度满足一定阈值的线段时认为是中心引导线。

障碍识别的基本思想是，当前偏差较小，边界特征前瞻较远，两侧都没丢线，一侧边界连续，一侧向中间跳变，结束行附近两侧边界不收拢。

坡道识别的基本思想是，边界特征是，前瞻较远，当前偏差很小，两侧

都不丢线，都没有明显的跳变，近处赛道较宽，结束行附近边界很窄。

环路识别的基本思想是，图像视野变宽，即白色道路变宽，并且在白色区域中间出现黑色区域。

4.1.4 模式识别

由于特殊元素的加入，普通寻迹已经不能让小车完成比赛，必须加入相应的模式识别。根据赛道中的元素，需要设置以下特殊模式：十字弯，中心引导线，障碍，坡道，斑马线，环路等六种特殊模式。并根据时间和距离等适时跳出特殊模式。每种模式的处理方式和跳出条件如下表：

模式	处理方式	跳出条件
十字弯	修改参考中线	图像分析给出的标志位
障碍	修改参考中线	距离
中心引导线	根据中心引导线打角	图像分析给出的标志位
坡道	给出合理的期望速度	距离
斑马线	修改参考中线	距离
环路	修改参考中线	图像分析给出的标志位

表 4-1 模式识别

4.2 控制策略

4.2.1 角度控制与模糊控制

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。

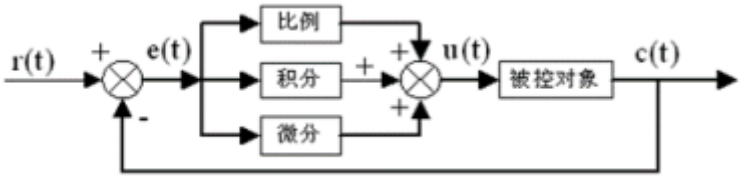


图 4-6PID 控制器原理框图

在计算机控制系统中，使用的是数字 PID 控制器，控制规律为：

$$e(k) = r(k) - c(k) \quad \text{(公式 4.1)}$$

$$u(k) = K_P\{e(k) + \frac{T}{T_I}\sum_{j=0}^{k} e(j) + \frac{T_D}{T}[e(k) - e(k-1)]\} \quad \text{(公式 4.2)}$$

式中

k——采样序号，k = 0，1，2…； 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；

KP——比例系数； TI——积分时间常数；

TD——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。

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△u(k)。

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(4.2)推导出提供增量的 PID 控制算式。由式(4.2)可以推出式(4.3)，式(4.2)减去式(4.3)可得式(4.4)。

$$u(k-1)=K_P\{e(k-1)+\frac{T}{T_I}\sum_{j=0}^{k-1}e(j)+\frac{T_D}{T}[e(k-1)-e(k-2)]\} \quad \text{(公式 4.3)}$$

$$\Delta u(k)=K_P\{[e(k)-e(k-1)]+\frac{T}{T_I}e(k)+\frac{T_D}{T}[e(k)-2e(k-1)+e(k-2)]\}$$

$$=K_P\Delta e(k)+K_I e(k)+K_D[\Delta e(k)-\Delta e(k-1)]\} \quad \text{(公式 4.4)}$$

式中 $\Delta e(k)=e(k)-e(k-1)$；$K_I=K_P\frac{T}{T_I}$；$K_D=K_P\frac{T_D}{T}$。

公式(4.4)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 KP、TI 、TD，只要使用前后三次测量值的偏差，即可由式(4.4)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△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，常选择带死区、积分分离等改进 PID 控制算法。比例 P ：比例项部分其实就是对预设值和反馈值差值放大的倍数。举个例子，假如原来电机两端的电压为 U0，比例 P 为 0.2，输入值是 800，而反馈值是 1000，那么输出到电机两端的电压应变为 U0+0.2*（800-1000）。从而达到了调节速度的目的。显然，比例 P 越大时，电机转速回归到输入值的速度将越快，调节灵敏度也越高。从而，加大 P 值，可以减少从非稳态到稳态的时间，但是同时也可能造成电机转速在预设值附近振荡的情形，所以又引入积分 I 解决此问题。

积分 I：顾名思义，积分项部分其实就是对预设值和反馈值之间的差值在时间上进行累加。当差值不是很大时，为了不引起振荡。可以先让电机按原转速继续运行。当时要将这个差值用积分项累加。当这个和累加到一定值时，再一次性进行处理。从而避免了振荡现象的发生。可见，积分项的调节存在明显的滞后。而且 I 值越大，滞后效果越明显。

微分 D：微分项部分其实就是求电机转速的变化率。也就是前后两次差值的差而已。也就是说，微分项是根据差值变化的速率，提前给出一个相应的调节动作。可见微分项的调节是超前的。并且 D 值越大，超前作用越明显。可以在一定程度上缓冲振荡。比例项的作用仅是放大误差的幅值，而目前需要增加的是“微分项”，它能预测误差变化的趋势，这样，具有比例+ 微分的控制器，就能够提前使抑制误差的控制作用等于零，甚至为负值，从而避免了被控量的严重超调。

由各个参数的控制规律可知，比例 P 使反应变快，微分 D 使反应提前，积分 I 使反应滞后。在一定范围内，P、D 值越大，调节的效果越好。各个参数

的调节原则是：在输出不振荡时，增大比例增益 P ；在输出不振荡时，减小积分时间常数 Ti 。输出不振荡时，增大微分时间常数 Td。

模糊控制系统是在控制原理的基础上以模糊集合论、模糊语言形式的知识表示和模糊逻辑推理为理论基础，模拟人的模糊思维方式，对复杂过程进行控制的一种智能控制系统。它是将操作者的控制经验，用一系列含有语言变量值的条件语句规则，恰当的表达成具有模糊性的语言变量和条件语句，或者说把控制经验用模糊条件语句表示，再用模糊集合理论对语言变量进行量化，然后通过模糊推理对系统的实时输入状态进行处理，对难以建立精确数学模型的对象实施的一种控制。

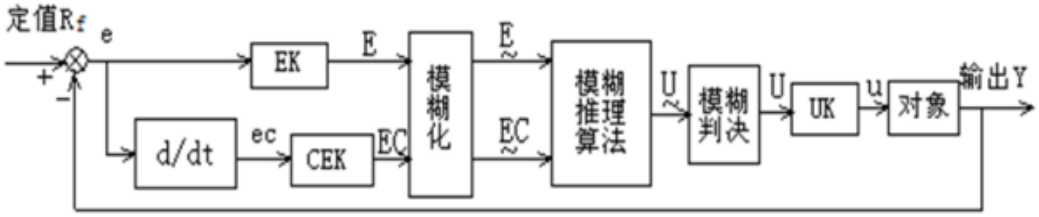


图 4-7 模糊控制过程

模糊 PD 控制器是在一般 PD 控制系统的基础上，加上一个模糊控制规则环节，利用模糊控制规则在线对 PD 参数进行修改的一种自适应控制系统。它以误差 e 和误差变化 ec 作为输入，可以满足不同时刻的 e 和 ec 对参数自整定的要求。它将模糊控制和 PD 控制器两者结合起来，扬长避短，既具有模糊控制灵活而适应性强的优点，又具有 PD 控制精度高的特点，对复杂控制系统和高精度伺服系统具有良好的控制效果。

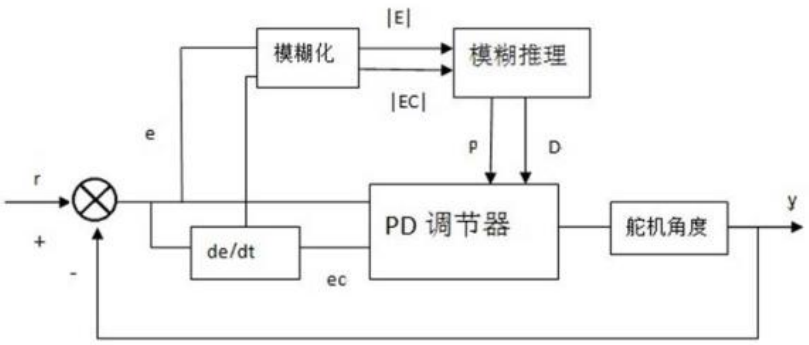


图 4-8 模糊控制流程图

实现过程：事先规划好模糊区间划分，建立模糊控制规则表，程序运行时将信号处理得到的偏差以及偏差的变化率进行模糊化，根据模糊区间得到隶属度，最后根据隶属度用重心法反模糊，得到一个确定的 P 和确定的 D，然后用来计算舵机角度。

4.2.2 速度控制

总体上用角度决定速度的算法，舵机打角越大速度越慢。另外设置长直道模式，此时设置高度，以适应长直道和曲率半径很大的弯道。电机 PWM 的确定使用 PI 控制，根据当前速度和期望速度偏差以及偏差的积累来确定 PI 参数，从而确定电机输出 PWM。

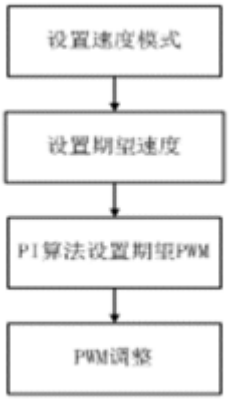


图 4-9 速度控制流程图

4.2.3 赛道优化

整个比赛，光电组组最大的优势在于其较远的前瞻，使得赛车能够判断前方赛道的信息，并能够做出相应的判断措施，尽早对赛车做出调整。赛车对于赛道的优化，主要是对于小 S 赛道的优化和普通弯道的优化，对于超越赛车直线行走的弯道，可以按照普通弯道处理。

(1) 小 S 形赛道

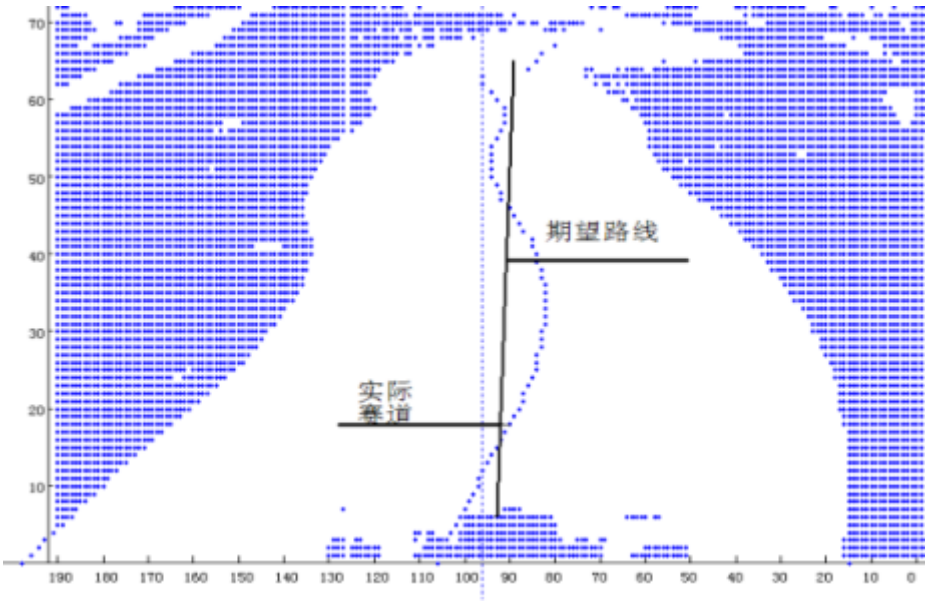


图 4.10 小 S 赛道处理

对于这种赛道的处理，我们直接利用了摄像头的“畸变”效应，由于摄像头的畸变效应，远处的像素点单位距离所代表的世界距离会很大，在图像中看到远处赛道与中心的偏差很小，所以在控制的思路在于取较远的参考行，计算偏差进行处理。这样也与上述的对于赛车的整体控制理论相符合。

(2) 对于普通弯道赛道

处理时只要将选取的参考行向远的方向移动，即可调节。如图：

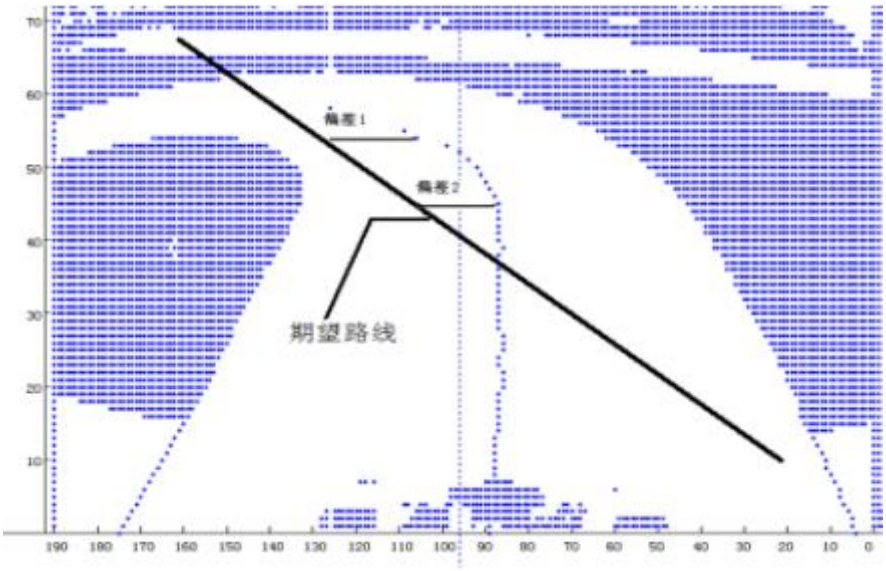


图 4.11 普通弯道处理

这样，选择不同的参考行作为控制行可以实现入弯姿态的调整。

(3) 对于大型波浪弯道的处理

大型波浪弯的处理直接作为普通弯道，即可实现优化，这里不再赘述。

4.3 单片机总体控制

单片机对于整个车体的控制，在于实现各个模块的整合。在整个系统中，不仅仅实现的是对于图像、速度、角度的调节，更重要的是实现了一个能够“稳定”工作的系统，在总体控制中，我们还加入了调试工具的单片机通讯代码，使得调试过程简洁，方便。

总体而言，单片机的信号输入主要有以下几个来源：摄像头、编码器、按键，输出主要为电机、舵机的控制信号，以及各种状态指示灯。单片机的程序分设计到的一些控制参数，可以通过上位机输入并进行控制，这在调试的过程中是极为方便的！整个系统工作模块图如下：

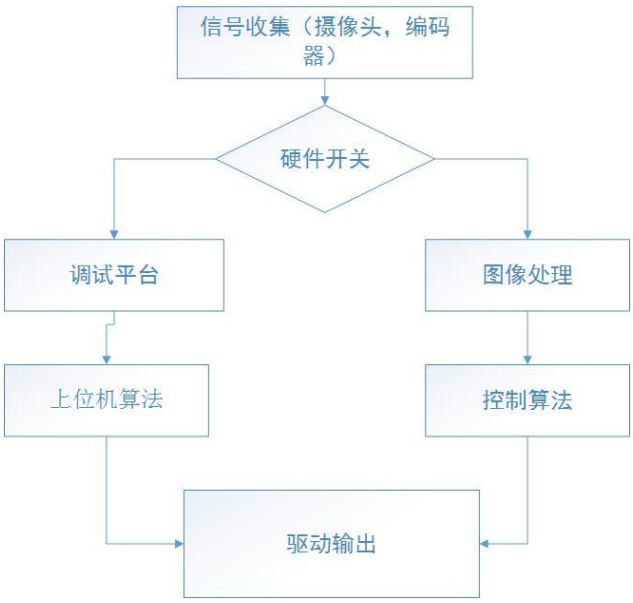


图 4-12 系统工作模块

在赛车的整个运行过程中，人为的输入主要是模式开关的选择，实现了整个赛车在不同模式状态下运行，尽量将赛车的速度发挥到极致。

第五章 调试

程序的基本框架建立起来以后，主要要做的就是进行 PID 参数的整定，具体的整定方法可以参见《过程控制》与《微机控制技术》，以下是我们的调试概况：

5.1 图像调试

调整信号处理的参数，考虑好赛道干扰和滤波问题，整定之后将测得的偏差通过无线串口发到电脑上，用上位机 接收数据。看接收到的图像正不正常，有无畸变，如有畸变则将图像进行预处理。

5.2 巡线调试

我们在程序中先通过判断 Err 和 ErrD 来判断出当前小车的状态然后在不同的状态下结合公式给出 P 分量和 D 分量。具体原理如下：

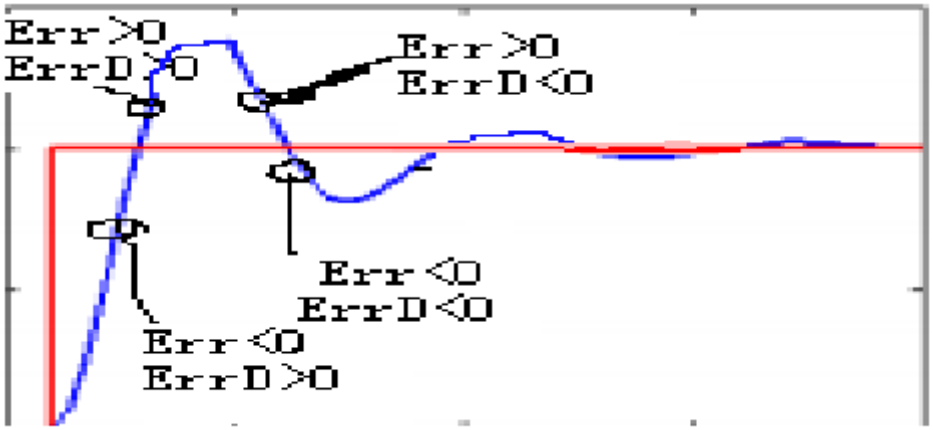


图 5.1 参数整定原理图

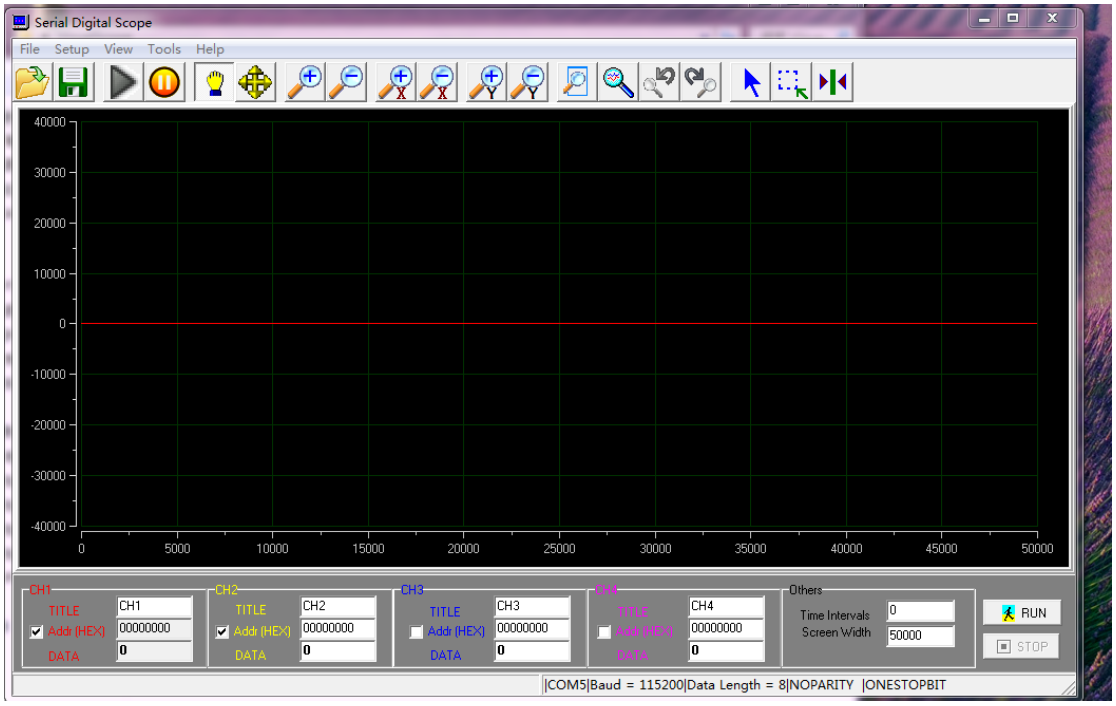
如上图，当 Err>0 且 ErrD>0 和 Err<0 且 ErrD<0 时，正处于远离预期状态； 当 Err>0 且 ErrD<0 和 Err>0 且 ErrD<0 时,正处于靠近预期的状态。那么我们在 控制过程中就要注意，当小车远离中心线时小车要及时的往中间打角，当小车 正在靠近中心线时，要适当的减小角度，避免由于角度过大，来不急及时调整 导致超调，从而产生振荡。

5.3 速度调试

通过速度的 PID 控制，将速度控制在一定的范围内，使速度尽可能的平滑。对于四轮车来说，维持速度的高速稳定是很有必要的，所以选择合适的 PID 参数是很有必要的。采用 4.5.2 的速度控制中的 PID 参数调节法可以做到速度的平滑控制。

5.4 调试平台

我们光电组采用的是通过 蓝牙 传输数据到虚拟示波器，看看速度的 PID 参数以及左右轮的速度，同时进行参数的修改。



第六章 全文总结

6.1 智能车主要技术参数

表 6.1 智能车主要技术参数

项目	参数
路径检测方法（赛题组）	摄像头
车模几何尺寸（长/宽/高）（mm）	280*160*300
车模轴距/轮距（mm）	轴距/205 前轮/141 后轮/143
车模平均电流（匀速行驶）(mA)	2100
电路电容总量（uF）	1910
传感器种类及个数	摄像头一个，编码器两个
新增加伺服电机个数	无
赛道信息监测频率（次/秒）	500
赛道信息检测空间精度（毫米）	5
主要集成电路种类/数量	摄像头/1
车模重量（带有电池）（kg）	1.3

6.2 不足与改进

机械改装和支架设计中，其强度和稳定性需要更加注意，以免造成支架容易撞断和运行不稳定的情况，同时注意平时对车模的保护，车模底板的变形将造成小车的左右不对称；还有在组装新 C 车时遇到小车左右转不对称，这个问题一直困扰；

硬件设计的兼容性和易扩展性考虑，在电路设计的过程中，端口分配是一个很重要的步骤，设计时需要预留一些可能用到的资源，以备后面增加是方便扩展，同时注意走线的间距及走向，注意信号的流向，也要和机械相配合好；

在程序设计和调试的过程中，不能太模糊化，需要进行实际的建模和仿真；对智能车的运行一定要进行定量的分析和计算，对智能车运行中出现的各种情况都能做到理解—解决，站在理论支持的角度进行分析和调试。

在备赛过程中，对于学校内部应加强不同组别和不同传感器方案队间的交流，同时应加强与其他兄弟院校的交流，使智能车往更好的方向发展。

6.3 致谢与总结

经过近一年多的设计、制作和调试，在队员们的分工和配合下，最终顺利完成了智能车电路、机械、程序等方面的工作。在这个过程中，大家学到了很多，积累了很多，能够一步一步走下来，靠的是整个团队的协作和团结。总体而言，大家学到了很多，提高了很多，这将对以后的学习和工作都会起到很多有意义的作用。

作为一个融合多学科交叉的的复杂系统，完成整智能车的设计、制作和调试是一个非常庞大的工程，仅靠几名队员是很难完成的，它需要一个高效运作、规范管理的团队的紧密合作才能完成。本文所论述的内容不仅仅是作者个人工作的结晶，同时也蕴含了他人的劳动、汗水和智慧。在本文即将结束之际，特向整个智能车团队的成员以及其它提供过帮助、建议和意见的人们表示衷心的感谢。

感谢指导老师的指导、支持和帮助。没有他的指导和帮助，整个智能车队就很难高效稳定运行下去，设计的方向也很难把握得这么准确。感谢华中科技大学启明学院和自动化学院在调试场地，物资上的支持和帮助，正是有了学校的大力支持，才使我们能专心做事，仔细研究。还要感谢教育部、自动化分教指委会给大家提供的这个锻炼团队合作和创新能力的竞赛平台，感谢各个赛区承办学校，感谢各个院校为大家提供优良的竞赛环境，使大家能够在竞赛中发挥出高水平。

参考文献

[1] 卓晴 黄开胜 邵贝贝 等.学做智能车. 北京：北京航空航天出版社，2007 年 3 月第 1 版

[2] 第 12 届全国大学生“飞思卡尔”杯智能汽车竞赛规则，2016/11

[3] International Rectifier，IRLR7843Pbf Datasheet，2008/04

[4] International Rectifier，IR2104s Datasheet，2004/02

[5] 杨杰 吴凡.粗糙表面可见光散射特性的实验研究. 中国测试，2009，02 期

[6] 邵贝贝.单片机嵌入式应用的在线开发方法. 北京：清华大学出版社，2004 年 10 月第 1 版.

[7] 邵贝贝.UCOS-II 嵌入式实时操作系统. 北京：清华大学出版社，2002 年 10 月第 1 版

附录：关键代码

由于篇幅限制在此只附上摄像头初始化，获取图像以及电机舵机控制代码：

摄像头初始化及获取图像代码

```
void Camera_GPIO_Init(void)
{
    //OV 数据口初始化
    data_init.GPIO_PTx = DATA_PORT;
    data_init.GPIO_Dir = DIR_INPUT;
    data_init.GPIO_Pins = DATA_PORT_PIN;
    data_init.GPIO_PinControl = IRQC_DIS | INPUT_PULL_DOWN | INPUT_PF_EN;
    LPLD_GPIO_Init(data_init);

    //OV 行信号接口初始化
    h_init.GPIO_PTx = H_GPIO_PORT;
    h_init.GPIO_Dir = DIR_INPUT;
    h_init.GPIO_Pins = H_PIN;
    h_init.GPIO_PinControl = IRQC_RI | INPUT_PULL_DOWN | INPUT_PF_EN;
    h_init.GPIO_Isr = Camera_H_isr;//Camera_H_isr;
    LPLD_GPIO_Init(h_init);

    //OV 场信号接口初始化：
    v_init.GPIO_PTx = V_GPIO_PORT;
    v_init.GPIO_Dir = DIR_INPUT;
    v_init.GPIO_Pins = V_PIN;
    v_init.GPIO_PinControl = IRQC_RI | INPUT_PULL_DOWN | INPUT_PF_EN;
```

```
v_init.GPIO_Isr = Camera_V_isr;
LPLD_GPIO_Init(v_init);

//OV PCLK 信号接口初始化：
pclk_init.GPIO_PTx = PCLK_PORT;
pclk_init.GPIO_Pins = PCLK_PIN;
pclk_init.GPIO_Dir = DIR_INPUT;
pclk_init.GPIO_PinControl = IRQC_DMAFA | INPUT_PULL_DOWN | INPUT_PF_EN;
LPLD_GPIO_Init(pclk_init);

//图像处理中断（优先级应该不高于 1ms 中断）触发源引脚：
isr_process_source_init.GPIO_PTx = ISR_PROC_SOURCE_GPIO_PORT;
isr_process_source_init.GPIO_Pins = ISR_PROC_SOURCE_PIN;
isr_process_source_init.GPIO_Dir = DIR_OUTPUT;
isr_process_source_init.GPIO_Output = OUTPUT_L;
isr_process_source_init.GPIO_PinControl = IRQC_DIS;
LPLD_GPIO_Init(isr_process_source_init);

//图像处理中断（优先级应该不高于 1ms 中断）被触发引脚：
isr_process_init.GPIO_PTx = ISR_PROC_GPIO_PORT;
isr_process_init.GPIO_Dir = DIR_INPUT;
isr_process_init.GPIO_Pins = ISR_PROC_PIN;
isr_process_init.GPIO_PinControl = IRQC_RI | INPUT_PULL_UP;
isr_process_init.GPIO_Isr = Camera_Process_isr;
LPLD_GPIO_Init(isr_process_init);
```

```
}

void Camera_DMA_Init(void)
{
    static DMA_InitTypeDef dma_init_struct;

    //DMA 参数配置
    dma_init_struct.DMA_CHx = CAMERA_DMA_CH;       //CH0 通道
    dma_init_struct.DMA_Req = DMA_REQ;             //请求源
    dma_init_struct.DMA_MajorLoopCnt = CAMERA_ROW_DMA_NUM; //主循环计数值：行采集点数，宽度
    dma_init_struct.DMA_MinorByteCnt = 1; //次循环字节计数：每次读入 1 字节
    dma_init_struct.DMA_SourceAddr = (uint32)&DMA_SOURCE_ADDRESS;          //源地址
    dma_init_struct.DMA_SourceDataSize = DMA_SRC_8BIT;
    dma_init_struct.DMA_SourceAddrOffset = 0;                  //源地址每次操作后地址增加量为 0
    dma_init_struct.DMA_DestDataSize = DMA_DST_8BIT;           //目的数据大小 8bit
    dma_init_struct.DMA_DestAddr = (uint32)compressImage0;  //目的地址：存放图像的数组
    dma_init_struct.DMA_DestAddrOffset = 1;          //目的地址偏移：每次读入增加 1 或减小 1
    dma_init_struct.DMA_AutoDisableReq = TRUE;      //自动禁用请求
    dma_init_struct.DMA_PeriodicTriggerEnable = FALSE;
    //dma_init_struct.DMA_MajorCompleteIntEnable = TRUE;
    //dma_init_struct.DMA_Isr = Camera_DMA_isr;
```

```
    //初始化 DMA
    LPLD_DMA_Init(dma_init_struct);

    LPLD_DMA_DisableReq(CAMERA_DMA_CH);

}

void Camera_Para_Init()
{
    uint32 i;
    int16 i16StraightWidth_0 = 33, i16StraightWidth_63 = 75;

    float fk;

//      for (i=0; i<IMAGE_X_MAX; i++)
//      {
//          g_row[i] = CAMERA_H / 64.0 * i;
//      }


    for (i=0; i<IMAGE_Y_MAX; i++)
    {
        g_column[i] = (uint16)(CAMERA_W/128.0 * i); //初始化取列的列数
    }

    fk = (i16StraightWidth_63-i16StraightWidth_0) / 63.0;
```

```
        for (i=0; i<IMAGE_X_MAX; i++)
        {
            g_carInfo.CameraInfo.u8StraightRoadWidth[i] = (uint8)(i16StraightWidth_0 +
i*fk);

            g_carInfo.CameraInfo.i16StraightRoadLeft[i] = (int16)(63.5 -
g_carInfo.CameraInfo.u8StraightRoadWidth[i]/2.0);
            g_carInfo.CameraInfo.i16StraightRoadRight[i] = (int16)(63.5 +
g_carInfo.CameraInfo.u8StraightRoadWidth[i]/2.0);
        }
    }

    void Camera_V_isr(void)
    {
        if(LPLD_GPIO_IsPinxExt(V_PORT, V_PIN))
        {
            V_PORT->ISFR |= 1 << V_NUM;     //清除场中断标志位

            disable_irq(V_PORT_IRQn);

            //检测到场开始信号，加载目的地址
            if(u8Useful_v == 0)
            {
                LPLD_DMA_LoadDstAddr(CAMERA_DMA_CH, (uint32)compressImage0);
                u8Useful_v = 1;
            }
```

```
            else
            {
                LPLD_DMA_LoadDstAddr(CAMERA_DMA_CH, (uint32)compressImage1);
                u8Useful_v = 0;
            }

            gs_u16RowCnt = 0;

            g_carInfo.CameraInfo.Is_Image_Ok = 1;//可以显示图像
            LPLD_GPIO_Output_b(ISR_PROC_SOURCE_GPIO_PORT, ISR_PROC_SOURCE_NUM,
OUTPUT_H); //触发信号处理

            g_carInfo.u32StartTime[0] = g_carInfo.time_100us;

            ChangeLEDBit(LED_3);
            enable_irq(V_PORT_IRQn);

        }
    }



    void Camera_H_isr(void)
    {
        if(LPLD_GPIO_IsPinxExt(H_PORT, H_PIN))
        {
```

```
            if (gs_u16RowCnt<240)
            {
                LPLD_DMA_EnableReq(CAMERA_DMA_CH);                    //使能通道
CHn 硬件请求
            }

            //if (gs_u16RowCnt == 239)
            {
//                  g_carInfo.CameraInfo.Is_Image_Ok = 1;//可以显示图像
//                                      LPLD_GPIO_Output_b(ISR_PROC_SOURCE_GPIO_PORT,
ISR_PROC_SOURCE_NUM, OUTPUT_H); //触发信号处理
            }

            gs_u16RowCnt++;
            ChangeLEDBit(LED_4);
        }
    }

    void Camera_Process_isr(void)
    {
        //static uint8 s_u8Cnt;

        if(LPLD_GPIO_IsPinxExt(ISR_PROC_PORT, ISR_PROC_PIN))
        {
```

```
            //更新图像
            if (g_carInfo.CameraInfo.Is_Image_Ok == 1)
            {
                Image_Original_Get();

                Car_Error();

                g_carInfo.CameraInfo.Is_Image_Ok = 2;
                //g_carInfo.u8SD_SendFlag = 1;
            }

            LPLD_GPIO_Output_b(ISR_PROC_SOURCE_GPIO_PORT, ISR_PROC_SOURCE_NUM,
OUTPUT_L);
        }
    }

    void Image_Original_Get()
    {

        if (u8Useful_v == 0)
        //if (u8Useful_v == 1)
        {
            //解压
            //img_extract(extractedImage, compressImage0, CAMERA_DMA_NUM);
            //img_extract_for_64x(&extractedImage_64x[64 * CAMERA_W - 1], (uint8
(*)[CAMERA_ROW_DMA_NUM])compressImage0);
```

```
            img_extract_for_64x(extractedImage_64x,                    (uint8
(*)[CAMERA_ROW_DMA_NUM])compressImage0);
        }
        else
        {
            //img_extract(extractedImage, compressImage1, CAMERA_DMA_NUM);
            //img_extract_for_64x(&extractedImage_64x[64 * CAMERA_W - 1], (uint8
(*)[CAMERA_ROW_DMA_NUM])compressImage1);

            img_extract_for_64x(extractedImage_64x,                    (uint8
(*)[CAMERA_ROW_DMA_NUM])compressImage1);
        }



        //g_carInfo.CameraInfo.u8ImageSource = (uint8 (*)[CAMERA_W])extractedImage;
        g_carInfo.CameraInfo.u8ImageSource = (uint8 (*)[CAMERA_W])extractedImage_64x;
    }
```

电机舵机控制代码:

```
    void Motor_PWM_Init(void)
    {
        // 配置电机 PWM 参数
        g_ftm_motor_struct.FTM_Ftmx = PWM_MOTOR_FTM;
        g_ftm_motor_struct.FTM_Mode = FTM_MODE_PWM;          //PWM 模式
        g_ftm_motor_struct.FTM_PwmFreq = PWM_Motor_PER_1S;   //频率 20000Hz
```

```
//    g_ftm_motor_struct.FTM_PwmDeadtimeCfg = PWM_MOTOR_DEADTIME;          //
通道 0 和 1 插入死区

        // 初始化 FTM
        LPLD_FTM_Init(g_ftm_motor_struct);

        // 左轮电机 PWM 使能
        LPLD_FTM_PWM_Enable(PWM_MOTOR_FTM,  PWM_LEFT_MOTOR_CH,  HALF_DUTY,
PWM_LEFT_MOTOR_PIN, ALIGN_LEFT);//单极性 PWM 要为 0

    #ifdef MOTOR_BIPOLAR
        LPLD_FTM_PWM_Enable(PWM_MOTOR_FTM,  PWM_LEFT_MOTOR_CH1,  HALF_DUTY,
PWM_LEFT_MOTOR_PIN1, ALIGN_LEFT);
    #endif

        // 右轮电机 PWM 使能
        LPLD_FTM_PWM_Enable(PWM_MOTOR_FTM, PWM_RIGHT_MOTOR_CH, HALF_DUTY,
PWM_RIGHT_MOTOR_PIN, ALIGN_LEFT);//单极性 PWM 要为 0

    #ifdef MOTOR_BIPOLAR
        LPLD_FTM_PWM_Enable(PWM_MOTOR_FTM,          PWM_RIGHT_MOTOR_CH1,
HALF_DUTY, PWM_RIGHT_MOTOR_PIN1, ALIGN_LEFT);
    #endif

    }
```

```
void Servo_PWM_Init(void)
{
    //舵机初始化
    g_ftm_servo_struct.FTM_Ftmx = PWM_SERVO_FTM;
    g_ftm_servo_struct.FTM_Mode = FTM_MODE_PWM;          //PWM 模式
    g_ftm_servo_struct.FTM_PwmFreq = PWM_Servo_PER_1S;          //频率 20000Hz

    // 初始化 FTM
    LPLD_FTM_Init(g_ftm_servo_struct);

    //舵机 PWM 使能
    LPLD_FTM_PWM_Enable(PWM_SERVO_FTM, PWM_SERVO_CH, 0, PWM_SERVO_PIN,
ALIGN_LEFT);
}

void CHANGE_PWM_SERVO(uint32 DUTY)
{
    SERVO_FTM_PWM_ChangeDuty(PWM_SERVO_FTM, PWM_SERVO_CH, DUTY);
}

// 修改左轮占空比
void CHANGE_PWM_MOTOR_LEFT(uint32 DUTY)
{
    uint32 left_pwm_duty = 0;
```

```
        left_pwm_duty = DUTY;

        // 占空比限幅
        left_pwm_duty = (uint32)Limiter(left_pwm_duty, MIN_DUTY, MAX_DUTY);

        // 修改左轮占空比
        LPLD_FTM_PWM_ChangeDuty(PWM_MOTOR_FTM,  PWM_LEFT_MOTOR_CH,
left_pwm_duty);

    #ifdef MOTOR_BIPOLAR
        LPLD_FTM_PWM_ChangeDuty(PWM_MOTOR_FTM,  PWM_LEFT_MOTOR_CH1,  10000
- left_pwm_duty);
    #endif

    }


    // 修改右轮占空比
    void CHANGE_PWM_MOTOR_RIGHT(uint32 DUTY)
    {
        uint32 right_pwm_duty = 0;

        right_pwm_duty = DUTY;

        // 占空比限幅
        right_pwm_duty = (uint32)Limiter(right_pwm_duty, MIN_DUTY, MAX_DUTY);
```

```
        // 修改右轮占空比
        LPLD_FTM_PWM_ChangeDuty(PWM_MOTOR_FTM,  PWM_RIGHT_MOTOR_CH,
right_pwm_duty);

    #ifdef MOTOR_BIPOLAR
        LPLD_FTM_PWM_ChangeDuty(PWM_MOTOR_FTM,  PWM_RIGHT_MOTOR_CH1,
10000 - right_pwm_duty);
    #endif

    }
```