

第十一届“恩智浦”杯全国大学生 智能汽车竞赛 技术报告



学 校：电子科技大学

队伍名称：成电追风

参赛队员：何永强

王昊天

李雪松

带队教师：程玉华 陈凯

关于技术报告和研究论文使用授权的说明

本人完全了解第十一届“恩智浦”杯全国大学生智能汽车竞赛保留、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。

参赛队员签名：何永强、王昊天、李雷松

带队教师签名：陈凯 杨元华

日期：2016/8/5

摘 要

本文设计的智能车系统以 MK60DN512ZVLQ10 微控制器为核心控制单元，通过 CMOS 摄像头检测赛道信息，使用模拟比较器对图像进行硬件二值化，提取黑色引导线，用于赛道识别；通过编码器检测模型车的实时速度，使用 PID 控制算法调节驱动电机的转速和转向舵机的角度，实现了对模型车运动速度和运动方向的闭环控制。

关键字： MK60DN512ZVLQ10, CMOS, PID

Abstract

In this paper we will design a smart car system based on MK60DN512ZVLQ10 as the micro-controller unit. We use a CMOS image sensor to obtain lane image information. Then convert the original image into the binary image by the analog comparator circuit in order to extract black guide line for track identification. An infrared sensor is used to measure the car's moving speed. We use PID control method to adjust the rotate speed of driving electromotor and direction of steering electromotor, to achieve the closed-loop control for the speed and direction.

Keywords: MK60DN512ZVLQ10, CMOS, PID

目 录

摘 要.....	II
Abstract.....	III
目 录.....	IV
引 言.....	1
第一章 系统总体设计.....	2
1.1 系统概述.....	2
1.2 整车布局.....	3
第二章 机械系统设计与实现.....	4
2.1 智能车机械参数调节.....	4
2.1.1 前轮调整.....	4
2.1.2 其他部分调整.....	6
2.2 底盘高度的调整.....	7
2.3 编码器的安装.....	7
2.4 舵机转向结构的调整.....	7
2.5 摄像头的安装.....	9
第三章 硬件系统设计与实现.....	11
3.1 MK60DN512ZVLL10 主控模块.....	12
3.2 电源管理模块.....	12
3.3 摄像头模块.....	14
3.4 电机驱动模块.....	15
3.5 测速模块.....	16
3.6 陀螺仪模块.....	16
3.7 辅助调试模块.....	16
第四章 软件系统设计与实现.....	17
4.1 赛道中心线提取及优化处理.....	17

4.1.1 原始图像的特点.....	17
4.1.2 赛道边沿提取.....	18
4.1.3 推算中心	19
4.1.4 路径选择	20
4.2 PID 控制算法介绍.....	21
4.2.1 位置式 PID.....	22
4.2.2 增量式 PID.....	22
4.2.3 PID 参数整定	22
4.3 转向舵机的 PID 控制算法	23
4.4 驱动电机的 PID 控制算法	23
第五章 系统开发及调试工具.....	24
5.1 开发工具	24
5.2 上位机图像调试.....	24
5.3 SD 卡模块.....	24
5.3.1 SD 卡介绍.....	24
5.3.2 SPI 总线介绍.....	25
5.3.3 软件实现	25
第六章 模型车的主要技术参数.....	27
结 论.....	28
参 考 文 献.....	I
附录 A: 电原理图.....	I
附录 B: 程序源代码.....	III

引 言

随着科学技术的不断发展进步，智能控制的应用越来越广泛，几乎渗透到所有领域。智能车技术依托于智能控制，前景广阔且发展迅速。目前，掌握着汽车工业关键技术的发达国家已经开发了许多智能车的实验平台和商品化的车辆辅助驾驶系统。有研究认为智能汽车作为一种全新的汽车概念和汽车产品，在不久的将来会成为汽车生产和汽车市场的主流产品。

面向大学生的智能汽车竞赛最早始于韩国，在国内，全国大学生“恩智浦”杯智能汽车竞赛从 2006 年开始已经举办了十一届，得到了各级领导及各高校师生的高度评价。大赛为智能车领域培养了大量后备人才，为大学生提供了一个充分展示想象力和创造力的舞台，吸引着越来越多来自不同专业的大学生参与其中。

全国大学生“恩智浦”杯智能汽车竞赛包括光电组、摄像头组和电磁组，其中数摄像头组的智能车速度最快，备受关注。

本技术报告主要包括机械系统、硬件系统、软件系统等，详尽地阐述了我们的设计方案，具体表现在硬件电路的创新设计以及控制算法的独特想法。智能车的制作过程包含着我们的辛勤努力，这份报告凝聚了我们智慧，是我们团队共同努力的成果。

在准备比赛的过程中，我们小组成员涉猎控制、模式识别、传感器技术、汽车电子、电气、计算机、机械等多个学科，几个月来的经历，培养了我们电路设计、软件编程、系统调试等方面的能力，锻炼了我们知识融合、实践动手的能力，对今后的学习工作都有着重大的实际意义。

第一章 系统总体设计

1.1 系统概述

智能车系统的总体工作模式为：CMOS 图像传感器拍摄赛道图像，输出 PAL 制式信号，经过信号处理模块进行硬件二值化，二值化图像信号、奇偶场信号、行同步信号输入到 MK60N512ZVLQ10 采用 MK60DN512ZVLQ10 的输入捕捉功能进行脉冲计算获得速度和路程；转向舵机采用 PD 控制；驱动电机采用 PID 控制，通过 PWM 控制驱动电路调整电机的功率；而车速的目标值由默认值、运行安全方案和基于图像处理的优化策略进行综合控制。

根据智能车系统的基本要求，我们设计了系统结构图，如图 1.1 所示。在满足比赛要求的情况下，力求系统简单高效，因而在设计过程中尽量简化硬件结构，减少因硬件而出现的问题。

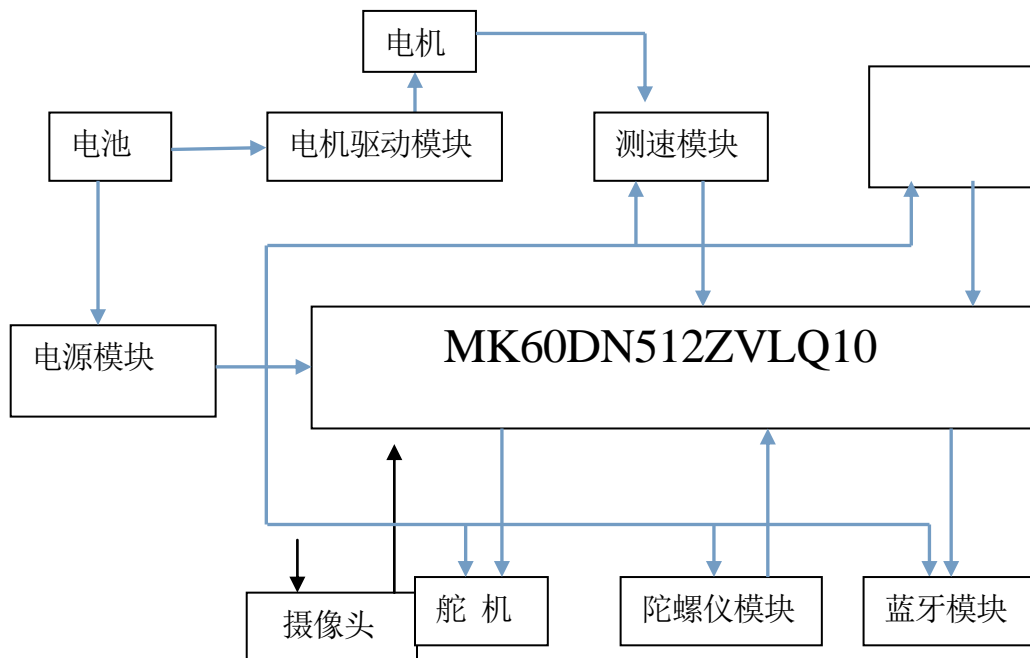


图 1.1 系统结构图

1.2 整车布局

- (1) 车模底盘降低，主板低放，以降低重心。
- (2) 舵机卧式放置，方便控制。
- (3) 用轻便坚固的碳纤杆作为摄像头杆的材料。
- (4) 摄像头安于电池的前方，有利于重心分布和盲区与前瞻的匹配。

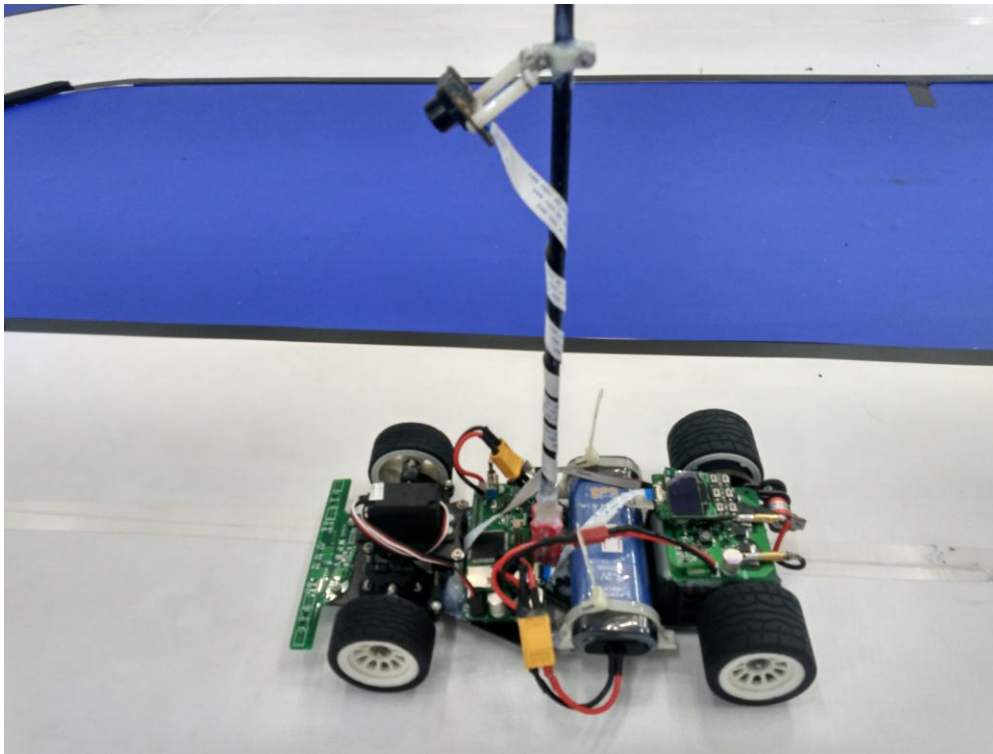


图 1.2 整车布局图

第二章 机械系统设计及实现

根据组委会的相关规定，今年摄像头组比赛车模更换为 B 型车模。针对不同的车模，必然会有不同的调整方式。在比赛备战之初，我们就对该车模进行了详细的系统分析。B 型车模精度不是很高，因此在规则允许范围内尽量改造车模，提高车模整体精度是很必要的。本章将主要介绍智能汽车车模的机械结构及调整方案。

2.1 智能车机械参数调节

为保证智能小车直线行驶稳定，转向轻便灵活并尽可能的减少轮胎磨损，需要对小车的四轮定位参数进行调整。四轮定位内容主要有：主销后倾角，主销内倾角，前轮外倾角，前轮前束，外侧车轮二十度时，内外转向轮转角差，后轮外倾角，后轮前束。其中，前轮定位的参数对小车性能有着至关重要的影响，这四个参数反映了前轮、主销和前轴三者之间在车架上的位置关系。本节将对这四个参数做详细阐述

2.1.1 前轮调整

小车在调试过程中，转向轮定位参数是很重要的因素，它通常不易被察觉，但是却有着较大的危害。如果取得不恰当，那么将造成转向不灵活，效率低以及转向轮侧滑等问题，使得小车性能下降，加速轮胎的磨损。

转向轮定位参数包括：主销内倾角、主销后倾角、转向轮外倾角及转向轮前束。这其中最重要的就是转向轮外倾角和转向轮前束。

主销内倾是指主销装在前轴略向内倾斜的角度，它的作用是使前轮自动回正，如图 2.1.1。内倾角度越大前轮自动回正的作用就越强烈，但转向时也越费力，轮胎磨损增大；反之，角度越小前轮自动回正的作用就越弱。

主销后倾（Caster）是指主销装在前轴，上端略向后倾斜的角度，如图 2.1.1。它使车辆转弯时产生的离心力所形成的力矩方向与车轮偏转方向相反，迫使车轮偏转后自动恢复到原来的中间位置上。由此主销后倾角越大，车速越高，前轮稳定性也愈好。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，因此高速时后倾的回正作用大，低速时内倾的回正作用大。

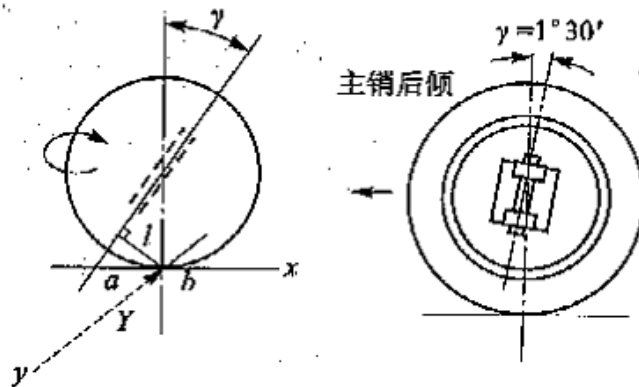


图 2.1.1 主销后倾角

车轮外倾角（Camber）是指从前放看前轴时，轮胎的中心平面不是垂直的，而是上面向外倾斜一个角度，如图 2.1.2。设置转向轮的外倾角是为了平衡和协调因为车重造成的前轮内倾倾向，使轮胎和路面呈垂直接触的最佳状态。

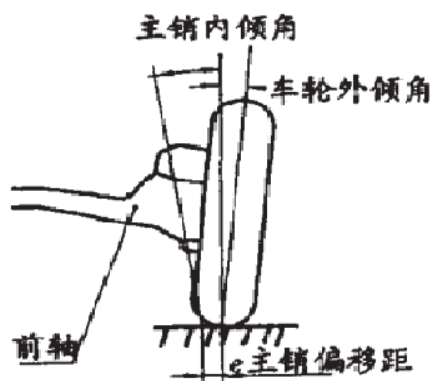


图 2.1.2 主销内倾角和车轮外倾角

转向轮前束(Toe out)是指同一轴两端车轮轮辋内侧轮廓线的水平直径的端点为等腰梯形的顶点，底边为车轮轴线。等腰梯形两底边长度之差为前束。如图 2.1.3 所示，当梯形前底边小于后底边时，前束为正（ $A < B$ ），反之为负。车轮的水平直径与纵向平面之间的夹角为前束角。正的前束角在车轮中心产生向内的侧向力，而

正的外倾角在车轮中心产生向外的侧向力，因此前束角的作用是与外倾角协调，保持车轮做纯滚动和直线行驶，从而减少轮胎磨损，提高汽车的操纵稳定性。

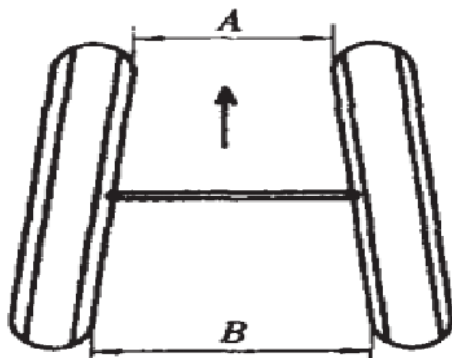


图 2.1.3 前束^[4]

2.1.2 其他部分调整

其他部分调整主要涉及到小车底盘高度、小车重心位置、后轮距、减震器、齿轮咬合、差速器等。具体调整如下：

1、底盘高度调整：底盘高度可以影响重心。适当降低底盘高度可以使小车重心降低，有利于过弯稳定。实际调整可以通过调整前轮高度、后轮轴高度调节块等方式来调节。

2、重心位置：重心位置同样影响小车性能。重心过前，增加转向阻力，引起转向迟滞。另外，如果小车速度很快的情况下，上下坡道的时候会造成前轮首先着地，很可能造成小车意外事故。重心过后，则会使小车前轮抓地不足，造成过弯非常不稳定。实际调整以重心在电池处为准，保持各部分重量均衡。

3、后轮距调整：后轮距可以通过换装后轮宽度调整块来调整。合适的后轮宽度会是小车直线性能和弯道性能更优，利于小车稳定。

4、减震器弹簧强度调整：坡道时候，减震器影响显得尤为突出。小车平面行驶时，无垂直方向速度，遇到坡道时候，小车会瞬间在垂直方向上出现一个速度，此时减震器在耦合小车前部和小车后部时候会出现一个缓冲区，增加坡道稳定性。下坡时候则是垂直方向上的速度瞬间减小为 0，减震器的作用亦如上坡。实际调整中，我们通过加装弹簧调节块来调整。

5、齿轮咬合调整：调整齿轮咬合，以不松动，无卡滞，松紧合适为准。另外还要保证齿轮间咬合有足够的接触面积。

6、差速器调整：合适的差速器调整能够提高小车入弯速度，提高弯道性能。差速器调整可以通过右后轮轮轴上面的螺丝。注意调整过松，会严重影响直道加速性能；调整过紧则会使差速器处于无效状态，影响小车在过弯时的速度，并可

能会照成甩尾。

2.2 底盘高度的调整

在保证顺利通过坡道的前提下，底盘尽量降低，从整体上降低模型车的重心，可使模型车转弯时更加稳定、高速。

2.3 编码器的安装

选用编码器进行速度的测量。根据编码器的形状，我们自制了一个支架，速度传感器用螺钉通过支架固定在后轮支架上，这样固定好之后，就有了较高的稳定性。然后调节编码器齿轮，使其与差速齿轮紧密咬合，增大测速的精确性，但是咬合过紧也增大了摩擦，减小了对电机做功的利用率，影响小车的快速行驶，因此减小摩擦同时增强齿轮间的咬合是我们主要考虑的因素。如图 2.3 所示。

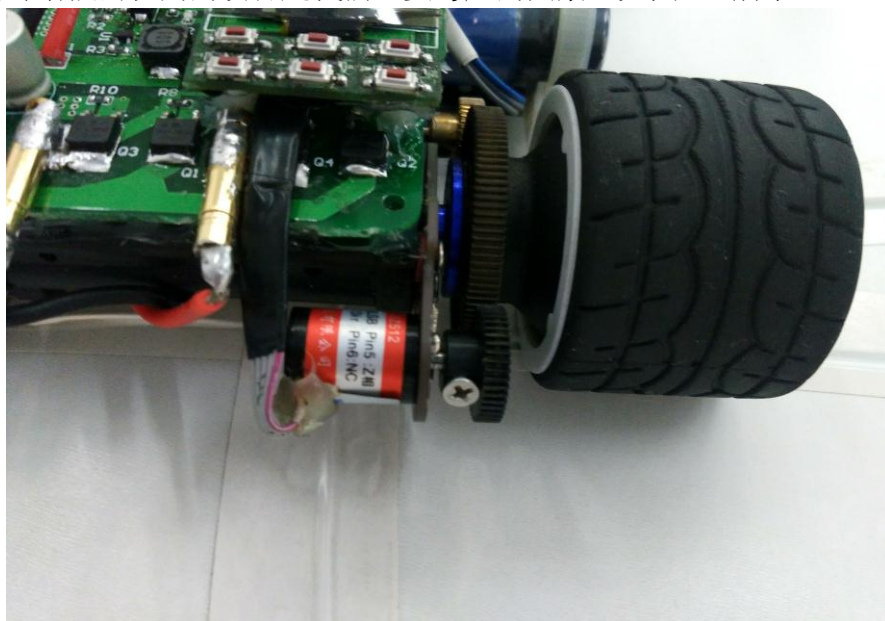


图 2.3 编码器安装

2.4 舵机转向结构的调整

车模转向由前轮舵机经过连杆将转动转变为平行四边形双摇杆结构的转动，最后由平行四边形双摇杆机构带动前轮的左右摆动实现转向。经过分析和测试，发现此转向机构结构相对复杂，效率底下。转向机构原理图 2.4.1。

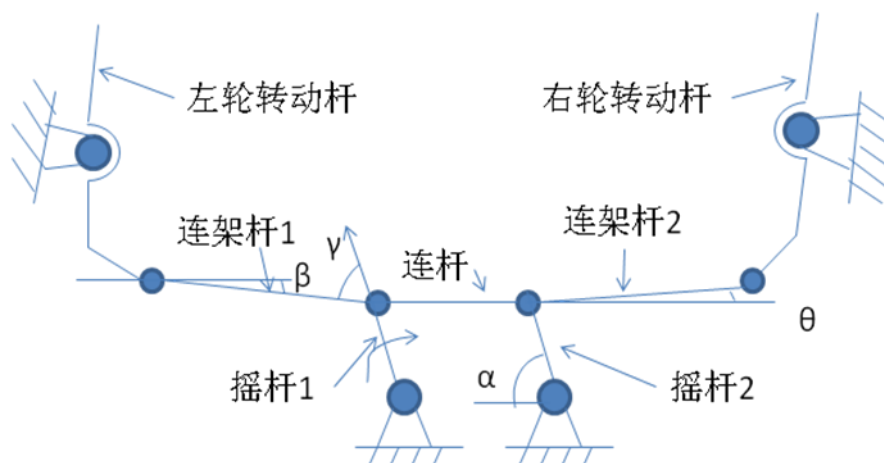


图 2.4.1 转向机构原理图

图 2.4.1 取车体为参考系，各固定铰相对车体静止。舵机连杆输出推动平行四边形双摇杆机构转动进而带动前轮转动。 β 角为压力角， γ 角为传动角，他们共同决定了舵机效率的高低， α 角反映了舵机的转角。

由于对称性，故只讨论 α 在 $0-90$ 度的情况，在舵机处于中心位置时角为 0 度，角为 90 度， α 角为 90 度，此时舵机效率最高。随着 α 角的减小，传动角减小，而压力角则增大，根据上式可以看出，效率将迅速减小。当角到达 0 度，即平行四边形双摇杆机构摇杆1与连接杆1在同一直线上时，此时左轮转动杆转角达到最大值；而当连架杆2和右轮转动杆的夹角等于 180 度时， α 角将不能在减小，此时到达舵机的左极限，同理如果 α 增大也会存在上述两个极限点。由于上述两个极限点并不是同时到达，甚至还会造成一小段时间内左右轮转向不同的问题。

对于C车来说，转向角度越大越好，但是从对平行四边形双摇杆机构的分析来看，随着 α 角的减小， θ 角将增大，当到达极限位置时， θ 角也到达极限，也就是右轮的右极限。而对于现有的安装方式角是比较大的，因此这也就限定了理论上的转角。而车模本身的限制不仅仅是理论上的，由于各个部件尺寸并不是很合适，造成了转向机构的机械限位，因此在不影响车模强度的情况下，修改和更换掉一些部件，将这些机械限位去掉。

为了最大限度的增大转向范围，设计了一种简单且效率较高的方案——长连杆方案，如图 2.4.2。

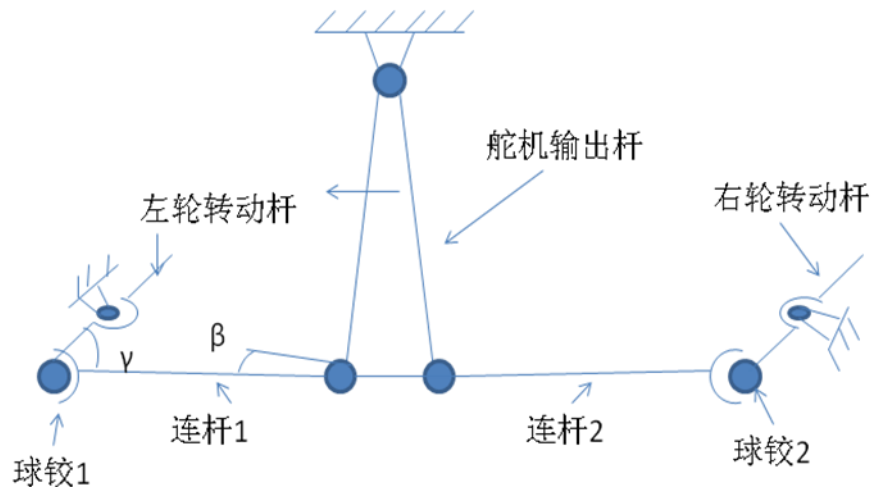


图 2.4.2 长连杆方案原理图

长连杆方案的优点是：

一、可以根据需要选择舵机输出杆的长度，从而获得所需要的灵敏度，但是舵机输出杆的长度也不能太长，因为这会对舵机的输出力矩有较高的要求，太长会烧坏舵机。

二、效率较高，舵机的单边（例如取左边）效率,对于长连杆方案来说，左轮角在增大的同时，右轮角在减小，而且角是在 0-45 度之间，同时的变动也比较小（45 度-135 度），因此长连杆的效率变动较小且效率较高。

三、转角大，由于长连杆方案中舵机输出杆的转动在同一个平面内，当其到达极限位置时，转角比平行四边形方案要大。

四、转向更灵敏，因为放大倍数较平行四边形方案要大。

2.5 摄像头的安装

为了降低整车重心，需要严格控制 CMOS 摄像头的安装位置和重量，我们自行设计了轻巧的铝合金夹持组件并采用了碳纤维管作为安装 CMOS 的主桅，这样可以获得最大的刚度质量比，整套装置具有很高的定位精度和刚度，使摄像头便于拆卸和维修，具有赛场快速保障能力。摄像头的安装如图 2.5 所示。



图 2.5 摄像头安装

第三章 硬件系统设计及实现

赛车共包括八大模块: MK60DN512ZVLL10 主控模块、传感器模块、电源模块、电机驱动模块、速度检测模块、陀螺仪模块和辅助调试模块。

各模块的作用:

MK60DN512ZVLL10 主控模块: 智能车系统以 MK60DN512ZVLL10 为控制核心, 将采集摄像头、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汽车的控制并实现了单片机硬件的最优化设计和单片机资源的合理化使用。

摄像头模块: 可以通过一定的前瞻性, 提前感知前方的赛道信息, 为智能汽车做出决策提供必要的依据和充足的反应时间。

电源管理模块: 为整个系统提供合适而又稳定的电源。

电机驱动模块: 驱动直流电机和伺服电机完成智能汽车的加减速控制和转向控制。

速度检测模块: 检测反馈智能汽车轮的转速, 用于速度的闭环控制。

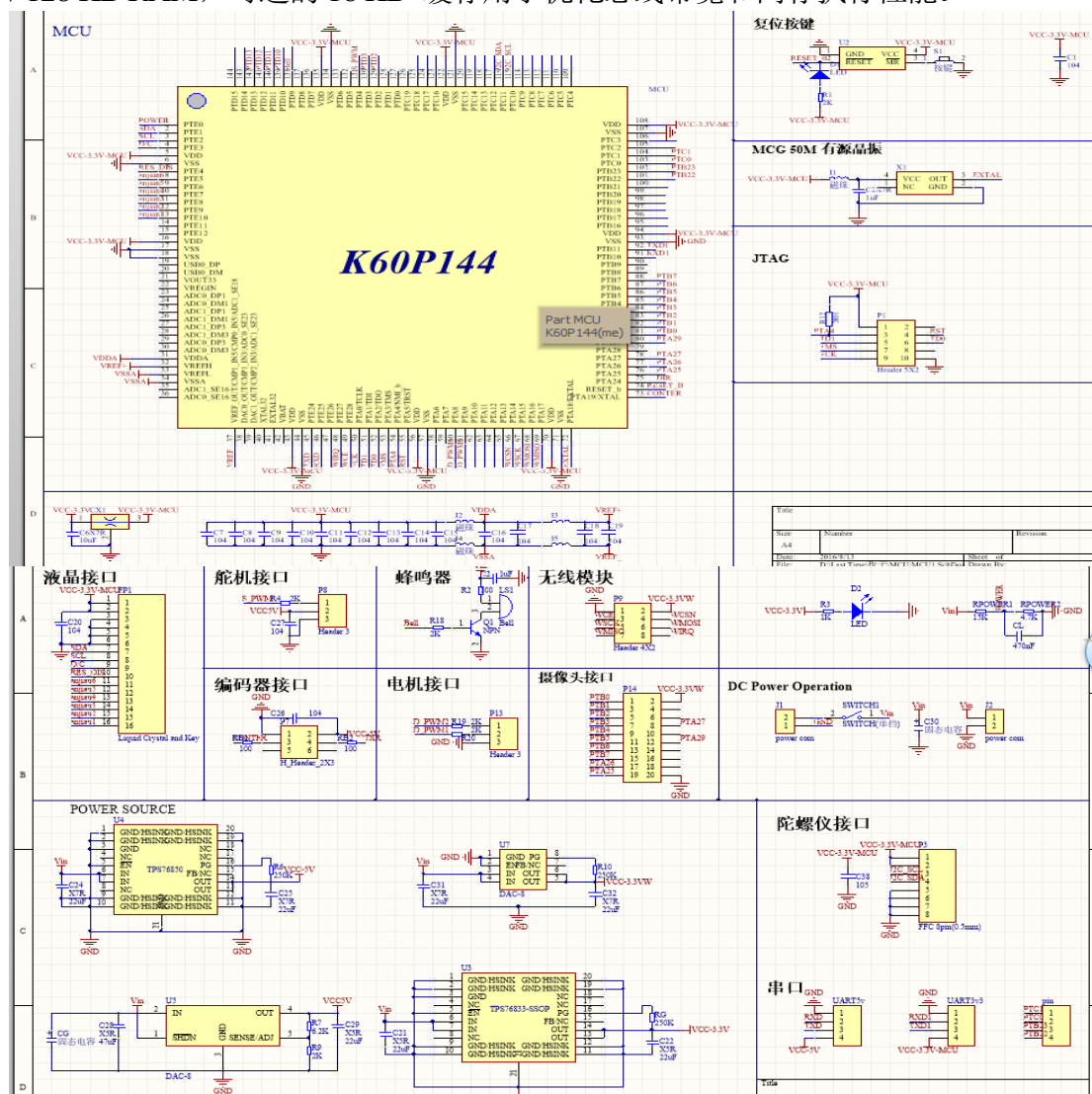
陀螺仪模块: 用于坡道检测, 辅助智能车过坡。

灯塔检测模块: 检测灯塔发出的方波信号, 完成智能车的发车与停车。

辅助调试模块: 主要用于智能汽车系统的功能调试、赛车状态监控

3.1 MK60DN512ZVLQ10 主控模块

MK60N512ZVLQ10 是 K60 系列 MCU。Kinetis 系列微控制器是 Cortex-M4 系列的内核芯片。K60 内存空间可扩展，从 32 KB 闪存/8 KB RAM 到 1 MB 闪存/128 KB RAM，可选的 16 KB 缓存用于优化总线带宽和闪存执行性能。



3.2 电源管理模块

本系统中电源稳压电路分别需要有+5V, +3.3V, +6V, +12V 供电。+3.3V 给单片机, 摄像头, 陀螺仪供电; +5V 为编码器模块, 蓝牙模块供电; +6V 给舵机供电; +12V 给电机驱动电路中 **HIP4082** 供电。

由于整个系统中+5V 电路功耗较小,为了降低电源纹波,我们考虑使用线性稳压电路。另外,后轮驱动电机工作时,电池电压压降较大,为提高系统工作稳定性,必须使用低压降电源稳压芯片,我们选用了 TPS76850。TPS76850 是微功耗低压差线性电源芯片,具有完善的保护电路,包括过流,过压,电压反接保护。使用这个芯片只需要极少的外围元件就能构成高效稳压电路。

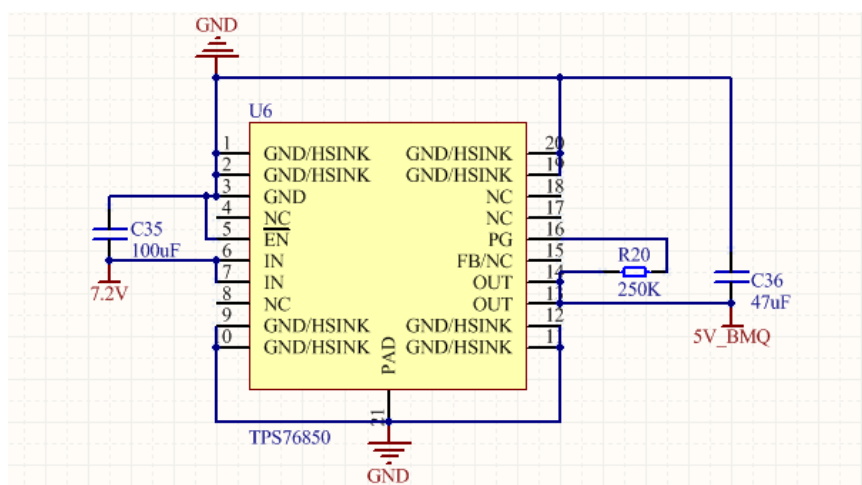


图 3.2.1 TPS76850 原理图

TPS76833 是超低噪,高电源电压抑制比,高速射频,线性调节的低压稳压芯片。

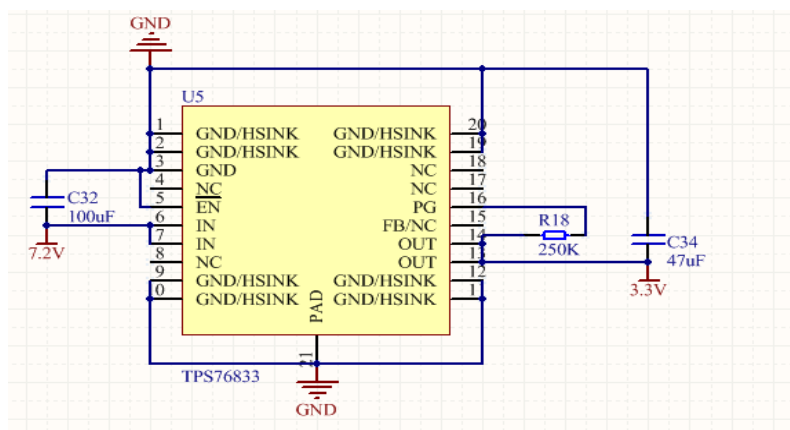


图 3.2.2 TPS76833 原理图

舵机用 LT1764-ADJ 进行供电，通过调整，使用 6V 的电压给舵机供电。较高的电压可以提高舵机的响应速度，但过高电压容易导致舵机工作不稳定。舵机电源的稳压电路原理图如图 3.2.3 所示。

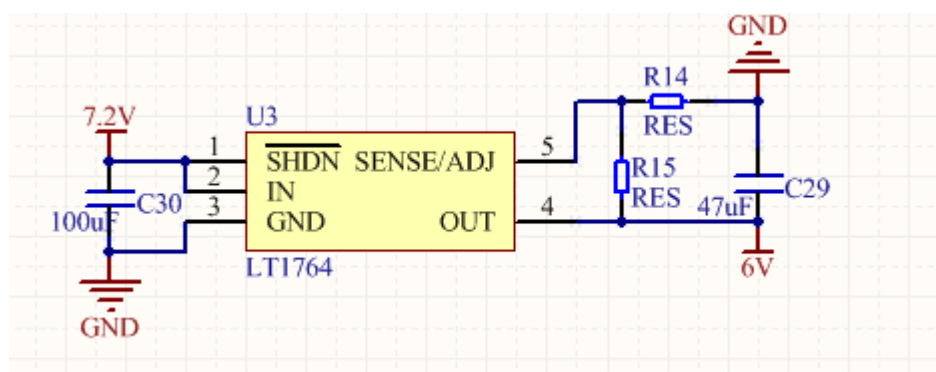


图 3.2.3 LT1764 原理图

3.3 摄像头模块

摄像头简介

摄像头分黑白和彩色两种，为达到寻线目的，只需提取画面的灰度信息，而不必提取其彩色信息，所以我们所使用的 CMOS 摄像头输出的信号为黑白视频信号。我们所选用的摄像头芯片为 OV7725，1/3 英寸数字式 CMOS 图像传感器 OV7725，总有效像素单元为 664(水平方向)×492(垂直方向)像素；内置 10 位双通道 A/D 转换器，输出 8 位图像数据；具有自动增益和自动白平衡控制，能进行亮度、对比度、饱和度、 γ 校正等多种调节功能；其视频时序产生电路可产生行同步、场同步、混合视频同步等多种同步信号和像素时钟等多种时序信号；5V 电源供电，工作时功耗<120mW，待机时功耗<10 μ W。可应用于数码相机、电脑摄像头、可视电话、第三代网络摄像机、手机、智能型安全系统、汽车倒车雷达、玩具，以及工业、医疗等多种用途。OV7620 是 1/3" CMOS 彩色 / 黑白图像传感器。它支持连续和隔行两种扫描方式，VGA 与 QVGA 两种图像格式；最高像素为 664×492，帧速率为 30fps；数据格式包括 YUV、YCrCb、RGB 三种，能够满足一般图像采集系统的要求。

我们采用集成摄像头模块，其优良的性能能很好地满足要求，如图 3.3.1



图 3.3.1

3.4 电机驱动模块

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。

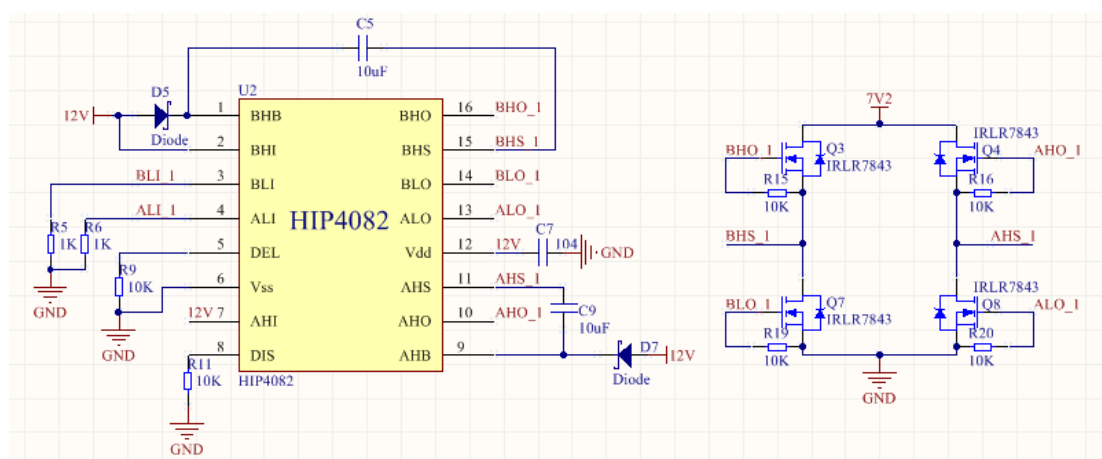


图 3.4 电机驱动模块原理图

3.5 测速模块

本小车使用广州科创电子的 mini 型编码器进行小车的测速，并由 TPS76850 为其提供 5V 工作电压。处理器通过读取编码器脉冲数来实现小车速度的检测，通过读取编码器旋转方向脚的高低电平来检测电机的正反转。

3.6 陀螺仪模块

采用 L3G4200D 角速度传感器和 MMA8451 加速度传感器融合检测坡道。

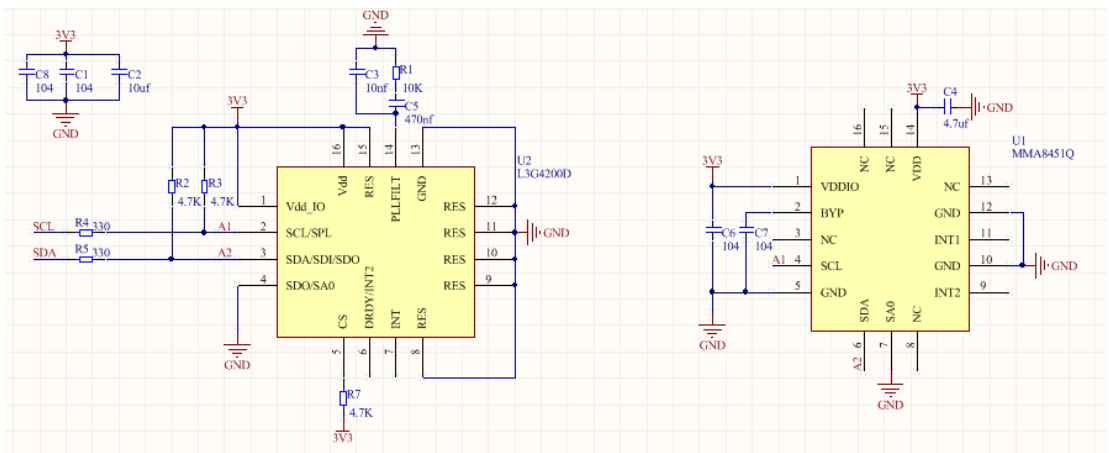


图 3.6.1 陀螺仪模块原理图

3.7 辅助调试模块

为了方便调试，本车有蓝牙模块，有效进行运行参数之间的传送，除此之外，还设置了键盘、oled 液晶显示屏，以方便控制参数的修改，便捷的智能车的调试。

第四章 软件系统设计及实现

高效的软件程序是智能车高速平稳自动寻线的基础。我们设计的智能车系统采用 CMOS 摄像头进行赛道识别，图像采集及校正处理就成了整个软件的核心内容。在智能车的转向和速度控制方面，我们使用了鲁棒性很好的经典 PID 控制算法，配合使用理论计算和实际参数补偿的办法，使智能车能够稳定快速寻线。

4.1 赛道中心线提取及优化处理

4.1.1 原始图像的特点

在单片机采集图像信号后需要对其进行处理以提取主要的赛道信息，由于赛道赛道由于畸变及干扰等，图像效果会下降，因此软件必须进行一些必要的处理，以便正确识别赛道中心，及特殊赛道类型，比如十字交叉，小 S 等。

在图像信号处理中我们提取的赛道信息主要包括：赛道两侧边沿点位置、赛道中心位置，赛道类型判别。摄像头第一行对应最远处，大约 180cm，图像的最底部一行对应最近处，大约 5cm。远处的图像小，近处的图像大，黑线为梯形状。

单片机通过比较器电路将每一行的黑白跳变点（跳变点按从右到左的顺序）记录下来，保存到两个二维数组里（分别表示上升沿、下降沿）。通过遍历上升沿和下降沿可以完成赛道边沿的提取。

摄像头采集到几种典型赛道图像如图 4.1～图 4.4 所示。



图 4.1 普通弯道



图 4.2 十字交叉图像

4.1.2 赛道边沿提取

边沿提取算法的基本思想如下：

- (1) 直接逐行扫描原始图像，根据设定的阈值提取黑白跳变点；
- (2) 赛道宽度有一个范围，在确定的赛道宽度范围内提取有效赛道边沿，这样可以滤除不在宽度范围内的干扰；
- (3) 利用赛道的连续性，根据上一行白块的位置和边沿的位置来确定本行的边沿点；
- (4) 求边沿点时，因为近处的图像稳定，远处图像不稳定，所以采用由近及远的办法；
- (5) 进出十字的时候，通过校正计算出边沿角度可较好的滤除十字并补线；
- (6) 直角一边丢线，另一边为直道，根据这个特征提取边界
- (7) 黑提示线依据赛道连续性进行补线，得出边界

边沿提取算法的程序流程如图 4.4 所示。

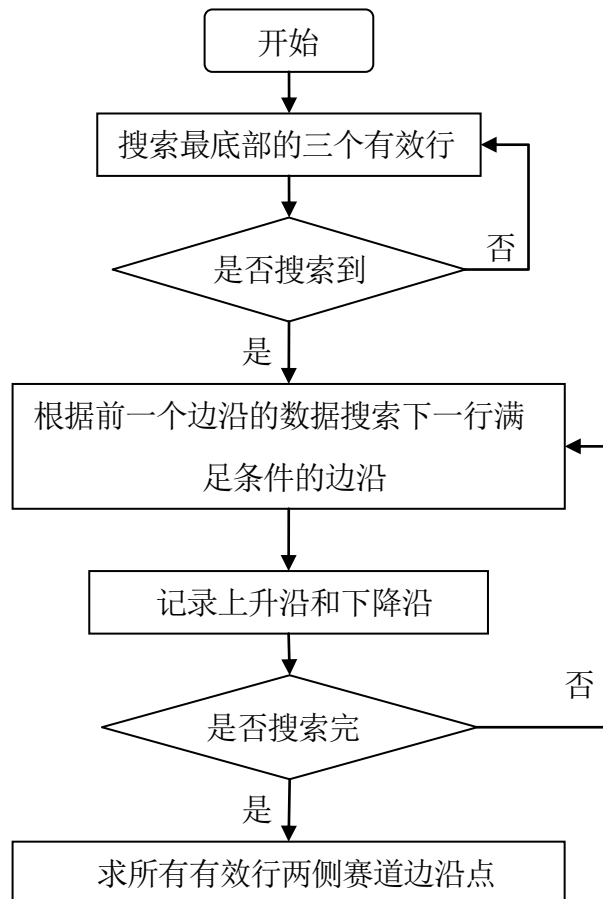


图 4.4 算法流程图

处理后得到的黑线中心如图 4.5~4.7 图所示。

4.1.3 推算中心

通过之前提取的赛道边沿数据推算中心：若只有单边有边沿点数据，则通过校正对单边数据按法线平移赛道宽度一半的距离；当能找到与一边能匹配上的另一边沿点时则直接求其中心作为中心点。判到直角直接给最大偏差。判到黑提示线根据两边补线求出中心线。计算出的中心点效果如下：

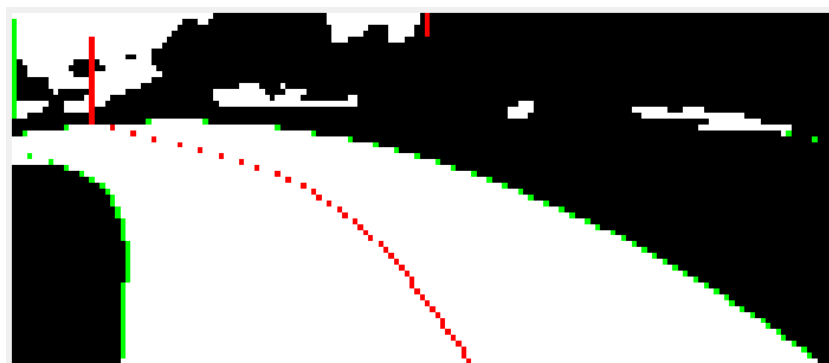


图 4.11 弯道处理后图像

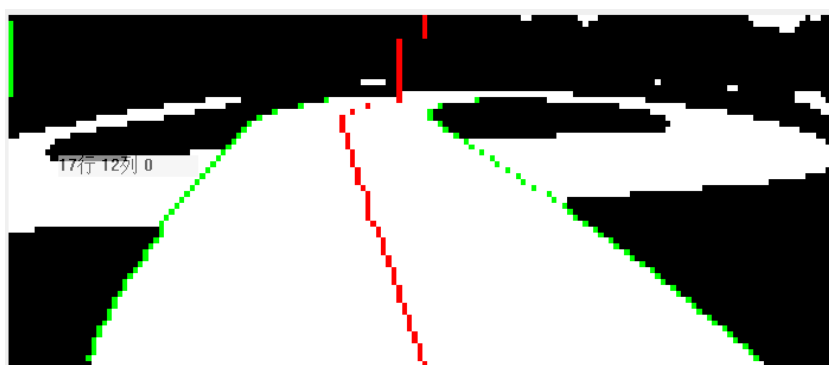


图 4.12 十字处理后图像

4.1.4 路径选择

对于竞速赛，小车能否以一个合理的路径跑完全程，能否合理地加减速，是提高小车速度的关键因素。在优化路径方面，在两个方面采取措施：

1) 增加视场的长度和宽度

如果摄像头的视场太过狭窄，则近处容易出现全白行，且有些赛道没法覆盖完整，会对图像的处理提出更高的要求。若视场太短，前瞻很可能不足，对摄像头的进一步加速会产生限制。因此，尽量增大视场的长度和宽度就很有必要了。

我们通过合理地调整摄像头的高度和角度，使摄像头的视场的长度比较合理，经过试验，使得摄像头视场最近为距车 5cm 左右，最远达到了 180cm 左右。为增加摄像头视场的广度，我们采用了广角镜头，经过测试，发现可以完整地覆盖赛道，对于直角等的判断也十分有利。

2) 优化加权算法

对整场有效行的中心求加权平均值的算法，在低速情况下可以有效地优化赛

车路径，但在赛车速度提高到一定程度之后由于过弯时的侧滑，路径不是很好。而由于图像分布不均，近处的行占据绝大部分，在小车速度比较高时适应性比较差。

为了解决这个问题，我们对于参与加权计算的图像行数及权重进行了处理，减小了车体前部 50cm 范围内的图像参与加权的行数和权重，同时增大视场前部图像的权重。

4.2 PID 控制算法介绍

PID 控制是工业过程控制中历史最悠久，生命力最强的控制方式。这主要是因为这种控制方式具有直观、实现简单和鲁棒性能好等一系列的优点。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（Steady-state error）。

为了消除稳态误差，引入积分控制。积分项对误差取决于时间的积分，随着时间的增加，积分项会增大。这样，即便误差很小，积分项也会随着时间的增加而加大，它推动控制器的输出增大使稳态误差进一步减小，直到等于零。

为了预测误差变化的趋势，引入微分的控制器，这样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。PID 控制框图如图 4.2.1 所示。

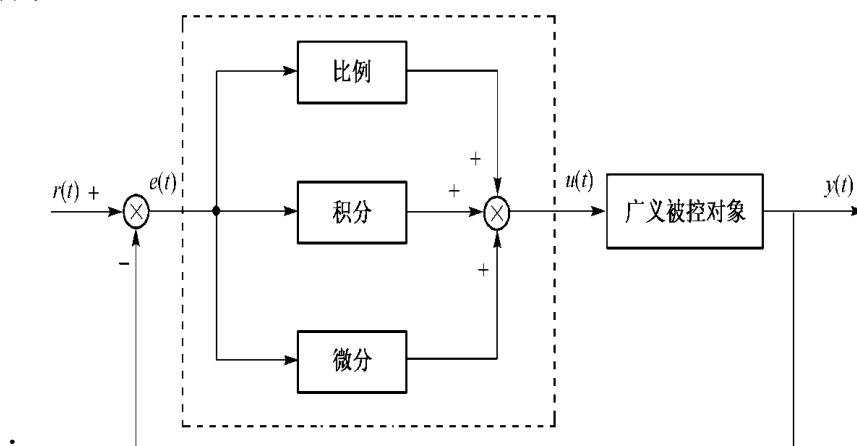


图 4.2.1 PID 控制框图

对应的误差传递函数为：

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i} + T_d \frac{d}{ds} \right) \quad (\text{公式一})$$

式中， K_p 为比例增益； T_i 为积分时间常数； T_d 为微分时间常数； $U(s)$ 为控制量； $E(s)$ 为被控量与设定值 $R(s)$ 的偏差。

时域表达式为：

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right] \quad (\text{公式二})$$

在单片机中，我们仅能对数字信号处理，即数字 PID 控制。将上式离散化，得
错误!未找到引用源。 （公式二）

4.2.1 位置式 PID

直接利用上述离散化公式计算，框图如右图所示。由于积分项（Pi）是将所有采集值偏差相加，在一段时间后会很浪费单片机资源。对其稍加改进，得到增量型 PID 算法。

4.2.2 增量式 PID

根据式二得第 $k-1$ 个采样周期的控制量为

$$\text{错误!未找到引用源。} \quad (\text{公式三})$$

式二减式三得

$$\text{错误!未找到引用源。} \quad (\text{公式四})$$

由此，第 k 个采样时刻实际控制量为错误!未找到引用源。，为方便书写，写为

$$\text{错误!未找到引用源。} \quad (\text{公式五})$$

其中，错误!未找到引用源。， $q_1 = -K_p \left(1 + \frac{2T_d}{T} \right)$ ， $q_2 = K_p \frac{T_d}{T}$ 。

由上可知，利用三个历史数据，递推使用，即可完成 PID 控制量。

4.2.3 PID 参数整定

运用 PID 控制的关键是调整 K_P 、 K_I 、 K_D 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

了工程整定方法。此外，我们先后实验了几种动态改变 PID 参数的控制方法。

4.3 转向舵机的 PID 控制算法

舵机采用位置式 PID 控制算法，将每场图像所选控制行的偏差平均值作为自变量，构建一次函数，经过一段时间的调试发现，只用 PD 两个参数便可以达到较稳定控制效果，分析后发现确实可以将 I 项置零，因为对于舵机，需要的是快速的反应，迅速的较准确的打角，即便存在一些静态误差也没有大的问题，通过调整 PD 两个参数，小车在直道也不会震荡。经过试验，我们没有采用动态 PD 算法，而是均使用定值，采用最保守的策略，确保稳定性。另外因为 D 项会影响车子打角的预判性，和偏差的变化率联系在一起，在 P 项的调整达到一定的程度后，D 项的调整也会影响车子的稳定性和路径，需要细细调试。这样的算法可以适应大部分赛道，但大小 S 弯并不能最小抖动的通过，为了应对这两种赛道，加入了偏差加权平均算法，单独处理，基本可以保证直线通过，最大程度减少耗时。

4.4 驱动电机的 PID 控制算法

速度 PID 采用增量式控制算法，事实上，对于速度控制 PID 算法，尝试了位置式和增量式两种，在 C 车上的表现并没有很大的差别，还尝试了 PID 和 bang-bang 结合的控制方法，bang-bang 控制的优点就是最大程度加快系统的调节速度，PID 控制进行精细调节，因为调试时间较短，调节死区和参数的选择没有达到最佳，最终选用了增量式 PID 算法，无论选用哪一种，都需要特殊处理，均不能简单的采用经典的 PID 控制，需要加入积分限幅和积分分离处理，最终用的增量式 PID 还用到了前馈处理，为了提高电机的反应速度。

第五章 系统开发及调试工具

5.1 开发工具

程序开放在 IAR Embedded Workbench IDE 下进行， Embedded Workbench for ARM 是 IAR Systems 公司为 ARM 微处理器开发的一个集成开发环境(下面简称 IAR EWARM)。比较其他的 ARM 开发环境，IAR EWARM 具有入门容易、使用方便和代码紧凑等特点。

EWARM 中包含一个全软件的模拟程序(simulator)。用户不需要任何硬件支持就可以模拟各种 ARM 内核、外部设备甚至中断的软件运行环境。从中可以了解和评估 IAR EWARM 的功能和使用方法。

5.2 上位机图像调试

实际调试过程中为了获得更多的底层数据、信号信息，有必要开发、使用一些辅助的调试工具和方法。我们使用了 VS 软件编程语言编写了上位机工具，进行各种模块的调试。

通过上位机，我们可以实时采集到传感数据、偏差数据、脉冲数据、PWM 占空比数据等各种小车数据，也可以实现实时的参数调节，提高了工作的效率。

5.3 SD 卡模块

5.3.1 SD 卡介绍

SD 卡(Secure Digital Memory Card)是一种基于半导体快闪存的新一代记忆设备。由日本松下、东芝及美国 SanDisk 公司于 1999 年 8 月共同开发研制，其大小犹如一张邮票，重量只有 2 克，却拥有高记忆容量、快速数据传输率、极大的移动灵活性以及很好的安全性。SD 卡的数据存储管理可以类似于硬盘的磁盘管理系统，以 FAT 格式来存储数据。SD 卡的接口支持 SD 模式和 SPI 模式，主机系统可以选择其中任一模式。SPI 模式允许简单通用的 SPI 通道接口，这种模式相对于 SD 模式的不足之处是降低了速度。由于飞思卡尔系列单片机拥有 SPI 接口，所以我们使用了 SD 卡的 SPI 模式。

5.3.2 SPI 总线介绍

SPI (Serial Peripheral Interface, 串行外围设备接口总线) 总线技术是 MOTOROLA 公司推出的一种同步串行总线接口, 它是目前单片机应用系统中最常用的几种串行扩展接口之一。SPI 总线主要通过三根线进行数据传输: 同步时钟线 SCK, 主机输入/从机输出数据线 MISO、主机输出/从机输入数据线 MOSI, 另外还有一条低电平有效的从机片选线 CS。SPI 系统的片选信号以及同步时钟脉冲由主机提供。SPI 总线模式的数据是以字节为单位进行传输的, 每字节为 8 位, 每个命令或者数据块都以字节对齐的(8 个时钟的整数倍)。主机与 SD 卡的各种通信都由主机控制, 主机在对 SD 卡进行任何操作前都必须先要拉低 SD 卡的片选信号 CS (card select), 然后由主机向 SD 卡发送命令, SD 卡对主机发送的任何命令都要进行响应, 不同的命令会有不同的响应格式(1 个字节或 2 个字节响应)。SD 卡除了对命令响应外, 在执行写操作时, 还要对主机发送的每个数据块进行响应(向主机发送一个特殊的数据响应标志)。

5.3.3 软件实现

首先需要将 SPI 模块设置为主机模式, 并设置相关的寄存器使 SPI 模块有高速和低速之分。SD 卡的软件设计主要包括两部分内容: SD 卡的上电初始化过程和对 SD 卡的读写操作。对 SD 卡初始化程序流程如图 5.11 所示。

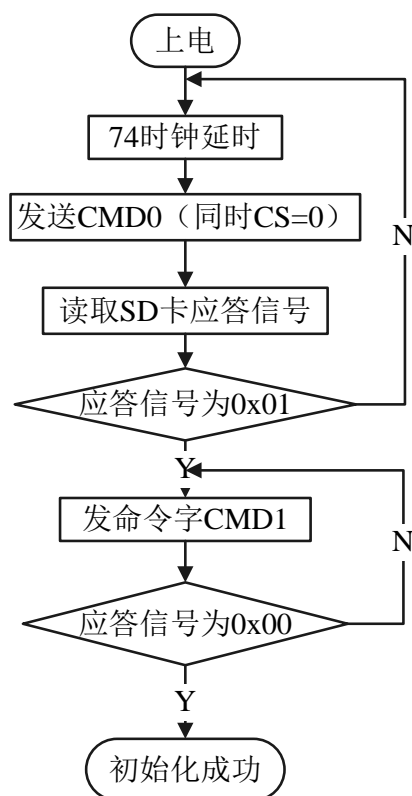


图 5.11 SD 卡初始化步骤

SD 卡上电后，主机必须先向 SD 卡发送 74 个时钟周期，以完成 SD 卡上电过程。SD 卡上电后会自动进入 SD 总线模式，并在 SD 总线模式下向 SD 卡发送复位命令(CMD0)，若此时片选信号 CS 处于低电平态，则 SD 卡进入 SPI 总线模式，否则 SD 卡工作在 SD 总线模式。SD 卡进入 SPI 工作模式会发出应答信号，若主机读到的应答信号为 01，即表明 SD 卡已进入 SPI 模式，此时主机即可不断地向 SD 卡发送命令字(CMD1) 并读取 SD 卡的应答信号，直到应答信号为 00，以表明 SD 卡已完成初始化过程，准备好接受下一命令。

此后，系统便可读取 SD 卡的各寄存器，并进行读写等操作，每次读写数据都是按照扇区操作的，每次操作 512 字节。

第六章 模型车的主要技术参数

赛车基本参数	长	28cm
	宽	18cm
	高	34cm
车重		1199g
功耗	空载	8W
	带载	大于 10W
电容总容量		1700uF
传感器	编码器	1 个
	CMOS 摄像头	1 个
	陀螺仪	0 个
除了车模原有的驱动电机、舵机之外伺服电机个数		0
赛道信息检测	视野范围（近瞻/远瞻）	5/170cm
	精度(近/远)	2/12.5mm
	频率	100Hz

结 论

自报名参加“飞思卡尔”杯智能汽车竞赛以来，我们小组成员从查找资料、设计机构、组装车模、编写程序一步一步的进行，最后终于完成了最初目标，定下了现在这个设计方案。

在此份技术报告中，我们主要介绍了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，前轮束角和主销倾角的调整以及在其他细节方面的优化。在电路方面，我们以模块形式分类，在最小系统、主板、电机驱动等模块分别设计，经过不断实验，最后决定了我们最终的电路图。在程序方面，我们使用 C 语言编程，利用比赛推荐的开发工具调试程序，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，做到直道加速、弯道减速，保证在最短时间内跑完全程。

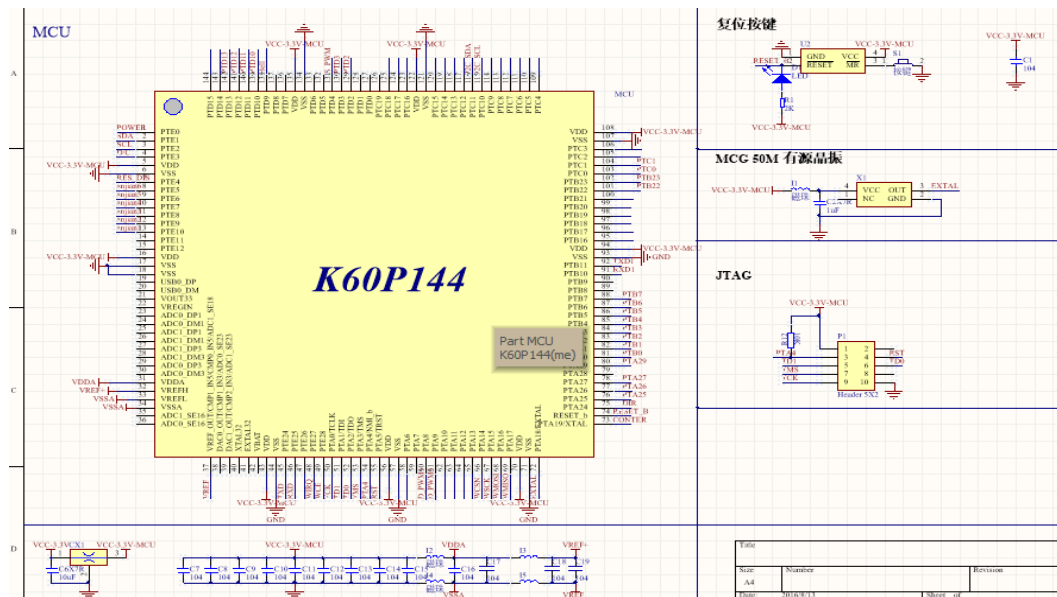
在这几个月的备战过程中，场地和经费方面都得到了学校和学院的大力支持，在此特别感谢一直支持和关注智能车比赛的学校和学院领导以及各位指导老师、指导学长，同时也感谢比赛组委会能组织这样一项有意义的比赛。

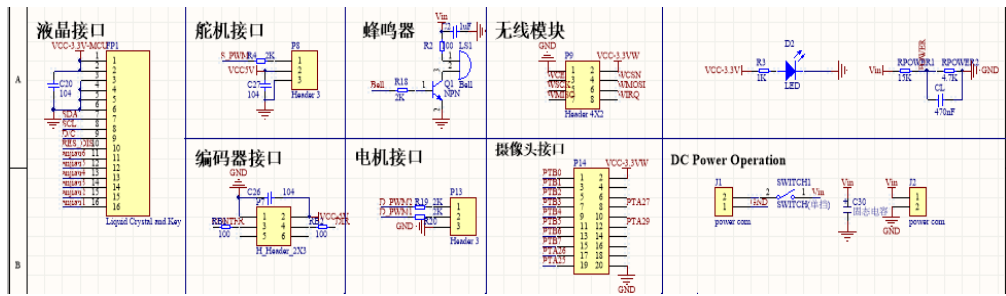
现在，面对即将到来的大赛，在历时近十个月的充分准备以及西部赛的考验之后，我们有信心在全国比赛中取得优异成绩。也许我们的知识还不够丰富，考虑问题也不够全面，但是这份技术报告作为我们小组辛勤汗水的结晶，凝聚着我们小组每个人的心血和智慧，随着它的诞生，这份经验将永伴我们一生，成为我们最珍贵的回忆。

参考文献

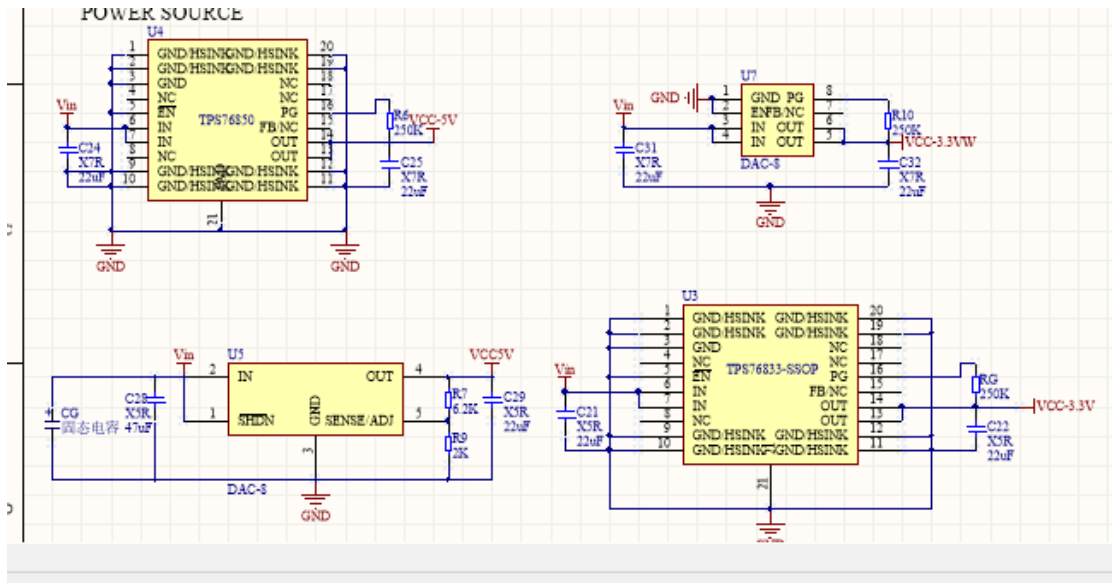
- (1) 田书林等. 电子测量技术. 北京. 机械工业出版社. 2012.
- (2) Mark I. Montrose. 电磁兼容的印制电路板设计. 北京. 机械工业出版社. 2012.
- (3) 余志生. 汽车理论. 北京. 机械工业出版社. 2012.
- (4) 童诗白, 华成英. 模拟电子技术基础 [M]. 北京: 高等教育出版社, 2001.
- (5) 阎石. 数字电子技术基础 [M]. 北京: 高等教育出版社, 2000.
- (6) 谭浩强著. C 程序设计. 北京: 清华大学出版社, 2003.
- (7) Park K.H , Bien Z, Hwang D.H. A study on the robustness of a PID - type iterative learning controller against initial state error [J]. Int. J. Syst. Sci. 1999, 30(1) , 102~135.
- (8) 夏克俭. 数据结构及算法 [M]. 北京: 国防工业出版社, 2001.
- (9) 李太福. 基于在线参数自整定的模糊 PID 伺服控制系统[J]. 交流伺服系统, 2005, 4: 203~215.
- (10) 仲志丹, 张洛平, 张青霞. PID 调节器参数自寻优控制在运动伺服中的应用[J]. 洛阳工学院学报, 2000, 21 (1): 57~60.

附录 A：电原理图

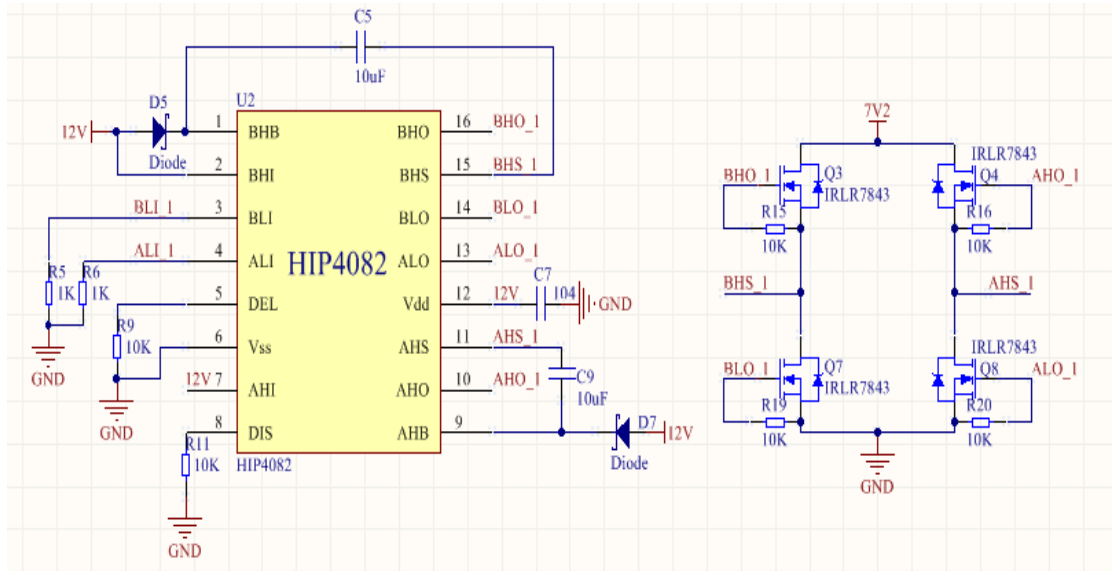




主控板原理图



电源模块原理图



电机驱动原理图


```
motor_out+=(int)(motor_P*10.0*(motor_err-motor_err_last)+motor_I*10.0*motor_err);
```

```
if(motor_out>950)//限幅
```

```
    motor_out=950;
```

```
if(motor_out<-950)
```

```
    motor_out=-950;
```

```
if(motor_out>=0)
```

```
{
```

```
    motor_pwm1=motor_out;
```

```
    motor_pwm2=0;
```

```
}
```

```
else
```

```
{
```

```
    motor_pwm1=0;
```

```
    motor_pwm2=0-motor_out;
```

```
}
```

```
motor_err_last=motor_err;//更新信息
```

```
}
```

```
void pid_servos() //舵机 PD 算法      位置式
```

```
{
```

```
    static int servos_pwm_last=0;
```

```
    static int near_err=0;
```

```
    float RP=servos_P[0],RD=servos_D[0],LP=servos_P[1],LD=servos_D[1];
```

```
    ////////////三输出叠加////////////////////////////////////
```

```

// if(leftinflectpoint!=0||rightinflectpoint!=0)
// {
//     RD/=3;
//     LD/=3;
// }
// if(XSP>15)//near_track_mode==5
// {
//     RP*=ZK;
//     LP*=ZK;
//     RD*=WK;
//     LD*=WK;
// }
far_err=(float)(IMG_COLS/2-far.col);
if(far_err>0)
    far_out=LP*far_err+LD*(far_err-far_err_last);
else
    far_out=RP*far_err+RD*(far_err-far_err_last);

near_err=IMG_COLS/2-near.col;
near_out=WK*near_err;

// servos_pwm=(int)(far_out+near_out*near.angle);

// far_err=(float)(IMG_COLS/2-far.col);
// if(far_err>0)
//     far_out=LP*far_err+LD*(far_err-far_err_last);
// else
//     far_out=RP*far_err+RD*(far_err-far_err_last);
//
// near_err=IMG_COLS/2-near.col;
// near_out=0.15*near_err;

```

```
//
// if(XSP<=15)
//   near.angle=0;
//   servos_pwm=(int)(far_out+near_out*near.angle);
//   servos_pwm=(int)far_out;

if(servos_pwm>35)//限幅 100HZ
    servos_pwm=35;
if(servos_pwm<-32)
    servos_pwm=-32;

// if(servos_pwm>70)//限幅 200HZ
//   servos_pwm=70;
//   if(servos_pwm<-70)
//     servos_pwm=-70;

far_err_last=far_err;//更新信息
servos_pwm_last=servos_pwm;
}
void get_controlline(unsigned char pData[IMG_ROWS][IMG_COLS])
{
    int i,count=0;
    static int cl_speed[10]={0,0,0,0,0,0,0,0,0,0};
    static int col_err[10]={0,0,0,0,0,0,0,0,0,0};
    control_line1=control_line;
    control_line2=speed_control_line;

    // for(i=8;i>=0;i--)
    //   cl_speed[i+1]=cl_speed[i];
    //   cl_speed[0]=actual_speed;
    //
```

```

// for(i=0;i<10;i++)
//     ave_speed+=cl_speed[i];
// ave_speed/=10;
//
//
// for(i=8;i>=0;i--)
//     col_err[i+1]=col_err[i];
// col_err[0]=far_err;
//
// if(ave_speed>=100)
//     control_line1+=(ave_speed-100)/7;
// if(control_line1>control_line+8)
//     control_line1=control_line+8;

//////////求近处控制行//////////
if((near_track_mode!=5)&&(near_track_mode!=6))
{
    if(center[0].row>=IMG_ROWS-5)
    {
        for(i=0;i<centernum;i++)
        {

            if(center[i].row==IMG_ROWS-5)
            {
                near.angle=1;
                near.row=IMG_ROWS-5;
                near.col=center[i].col;
                break;
            }
        }
    }
}

```

```
}

if(far.angle==0)
{
    far.angle=1;

    far.row=center[centernum-2].row;
    far.col=center[centernum-2].col;
    ////////////求远处控制行//////////
    for(i=1;i<centernum;i++)
    {
        if((center[i].col>IMG_COLS+10)||(center[i].col<0-10))
        {
            far.angle=1;
            far.row=center[i-1].row;
            far.col=center[i-1].col;
            break;
        }
        if(center[i].row<IMG_ROWS-control_line1)
        {
            if(center[0].row<IMG_ROWS-20&& i<8)
                i=8;
            far.angle=1;
            far.row=center[i-1].row;
            far.col=center[i-1].col;
            break;
        }
    }
    if(far.row<IMG_ROWS-control_line1)
        far.row=IMG_ROWS-control_line1;
}
```

```

speed_con.row=center[centernum-2].row;
speed_con.col=center[centernum-2].col;
//////////求远处控制行//////////
for(i=1;i<centernum;i++)
{
    if((center[i].col>IMG_COLS+10)||(center[i].col<0-10))
    {
        speed_con.angle=1;
        speed_con.row=center[i-1].row;
        speed_con.col=center[i-1].col;
        break;
    }
    if(center[i].row<IMG_ROWS-control_line2)
    {
        if(center[0].row<IMG_ROWS-20&& i<8)
            i=8;
        speed_con.angle=1;
        speed_con.row=center[i-1].row;
        speed_con.col=center[i-1].col;
        break;
    }
}
if(speed_con.row<IMG_ROWS-control_line2)
    speed_con.row=IMG_ROWS-control_line2;

}

void set_speed(uint8 pData[IMG_ROWS][IMG_COLS])
{
    int speed_out;
    int speed_out2;

```

```
static int speed_con_err[4]={0,0,0,0};
static int speed[10]={0,0,0,0,0,0,0,0,0,0};
int sP_temp=P_speed,sD_temp=D_speed,speed_max=SZ;
if(ave_speed>90)
{
    sP_temp-=(ave_speed-90)/2;
    sD_temp-=(ave_speed-90)/2;
}
if(XSP>5)
{
    speed_max=LZ;
}
else if(ZP>1)
{
    speed_max=LZ;
}

speed_out=speed_max-((speed_max-sP_temp)*my_abs(IMG_COLS/2-far.col)/(IMG_COLS/2)+
(speed_max-sD_temp)*my_abs((int)(IMG_COLS/2-far.col-far_err_last))/(IMG_COLS/8)
);

speed_out2=speed_max-((speed_max-sP_temp)*my_abs(IMG_COLS/2-speed_con.col)/(
IMG_COLS/2)+
(speed_max-sD_temp)*my_abs((int)(IMG_COLS/2-speed_con.col-speed_con_err[3]))/(I
MG_COLS/10));

speed_con_err[3]=speed_con_err[2];
```

```
speed_con_err[2]=speed_con_err[1];
speed_con_err[1]=speed_con_err[0];
speed_con_err[0]=IMG_COLS/2-speed_con.col;

if(speed_out>speed_max)
    speed_out=speed_max;

if(speed_out2>speed_max)
    speed_out2=speed_max;
speed_out=(speed_out*4+speed_out2*6)/10;

if(speed_out>speed_max)
    speed_out=speed_max;

if(IMG_COLS/2-far.col<0)
{
    if(speed_out<RW)
        speed_out=RW;
}
else
{
    if(speed_out<LW)
        speed_out=LW;
}

motor_speed=speed_out;
if(near_track_mode==6)
    motor_speed=PD;
if(near_track_mode==7)
    motor_speed=0;
}
```

