

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： 中南大学
队伍名称： 比亚迪金牛座 2016
参赛队员： 黄竞辉
 任宏宇
 魏佳雯
带队教师： 王 击

关于技术报告和学术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关保留、使用技术报告和学术论文的规定，即：参赛作品著作权归参赛者本人，比赛组委会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：黄竞辉
魏佳雯
任宏宇

带队教师签名：

日期：2016.08.15

目 录

第一章 引言	- 1 -
1.1 背景介绍	- 1 -
1.2 整车设计思路	- 1 -
1.2.1 控制系统	- 1 -
1.2.2 赛车整体结构设计	- 2 -
1.3 文章结构	错误!未定义书签。
第二章 机械结构调整	3
2.1 总体思路	4
2.2 底盘调整	4
2.3 前轮定位调整	4
2.3.1 主销后倾角	4
2.3.2 主销内倾角	4
2.3.3 车轮外倾角	5
2.3.4 前轮前束	5
2.4 舵机的安装与控制延时解决办法	5
2.5 摄像头支架的设计安装	6
2.6 电路板安装	7
第三章 硬件设计	7
3.1 总体方案	7
3.2 电源模块	9
3.2.1 主电机供电与单片机供电分离电路	9
3.2.2 单片机和摄像头供电电路	9
3.2.3 舵机供电电路	10
3.3 电机驱动模块	10
3.4 3.3V、5V 和 12V 供电电路	11
3.5 传感器模块	13
3.5.1 摄像头的选择	13
3.5.2 编码器测速模块	13
3.6 MCU 主控模块	13
第四章 软件设计	- 14 -
4.1 黑线的提取和图像中心的计算	- 15 -
4.1.1 原始图像的特点及校正	14
4.1.2 黑线的提取和中心的计算	错误!未定义书签。
4.2 方向控制方案	- 15 -
4.2.1 方向控制数据的选取	- 15 -
4.2.2 道路形状精确识别算法	错误!未定义书签。
4.3 分类进行方向控制算法	17
4.3.1 直道的方向控制算法	17

4.3.2 大弯的方向控制算法	17
4.3.3 急弯的方向控制算法	18
4.3.4 “小 S 型”道的方向控制算法	18
4.4 速度控制方案	18
4.4.1 速度控制的重要性	18
4.4.2 速度控制的算法设计	19
4.5 棒—棒控制算法	20
4.5.1 棒—棒控制算法简介	20
4.5.2 棒—棒控制算法用于该赛车系统	20
4.5.3 棒—棒控制与 PID 控制两种方法结合的必要性	21
4.6 PID 控制算法	21
4.6.1 PID 控制算法简介	21
4.6.2 数字 PID 控制算法	22
4.7 经典 PID 算法实现速度的控制	24
4.7.1 经典 PID 算法实现速度控制的计算机仿真	24
4.7.2 经典 PID 算法实现速度控制的具体实现	25
4.7.3 PID 参数调节	25
第五章 仿真与调试	26
5.1 调试	27
5.2.1 系统硬件调试	27
5.2.2 摄像头调试	27
5.2.3 软件系统调试	27
第六章 赛车主要技术参数	28
第七章 总结与展望	- 29 -
7.1 总结	29
7.2 展望	30
参考文献	-
32 -31	
附录 程序代码	I

第一章 引言

1.1 背景介绍

全国大学生智能汽车竞赛是在统一汽车模型平台上，使用恩智浦公司的 16 位、32 位微控制器作为核心控制模块，通过增加道路传感器、设计电机驱动电路、编写相应软件以及装配模型车，制作一个能够自主识别道路模型汽车，按照规定路线行进，以完成时间最短者为优胜。该竞赛是涵盖了控制、模式识别、传感技术、电子、电气、计算机、机械及车辆工程等多个学科的科技创意性比赛。

全国大学生智能汽车竞赛已经成功举办了十届，比赛规模不断扩大、比赛成绩不断提高。通过比赛促进了高等学校素质教育，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。

从 2016 年 1 月开始，我们就开始着手准备这项赛事。历时 7 个月时间，经历了机械构造、硬件方案、算法思路的创新，这些创新体现在设计理念上，也贯穿赛车制作过程的始终。由于这些创新，赛车各方面综合性能得到提升，并且获得了良好的赛场表现。本技术报告将详细介绍我们为第十一届“恩智浦”杯全国大学生智能汽车竞赛而准备的智能车系统方案。

1.2 整车设计思路

1.2.1 控制系统

智能车系统是一个相对复杂的反馈系统(如图 1.1)。CMOS 摄像头采集的赛道图像信息、编码器采集到的车体运行速度，是反馈控制系统的输入量。执行器（直流电机和伺服舵机）以及模型赛车构成反馈系统的装置。恩智浦 Kinetis 系列的 32 位单片机 MK60DN512ZVLQ144 是系统的控制器。在智能车系统的搭建到赛车快速、稳定地按照赛道行驶的整个过程中，反馈原理是我们分析问题和解决问题的基本原理。

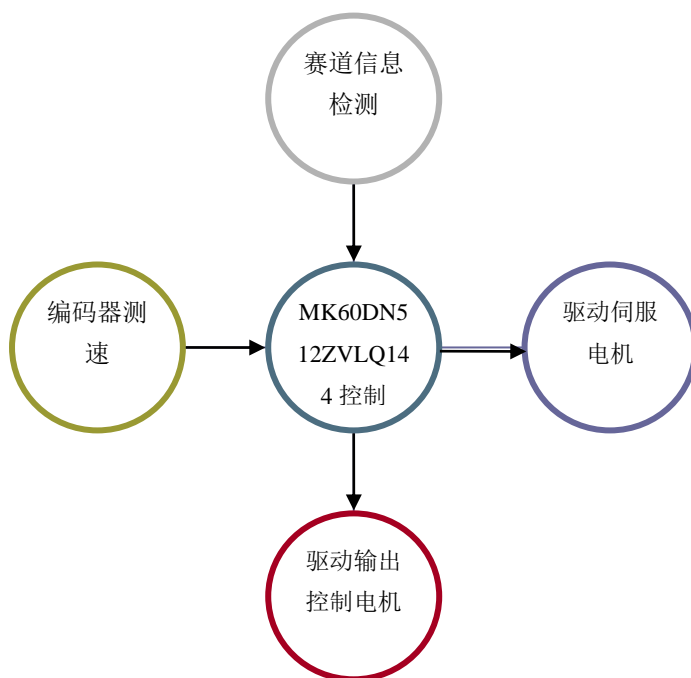


图 1.1，智能车控制系统

1.2.2 赛车整体结构设计

关于赛车的整体结构布局（如图 1.2 所示），我们主要思路是：减轻车体重量，电路板模块化处理，尽量降低并合理调整车体重心。为此，我们将电路板体积做到了最小，在双面板情况下集成度做到了最高；选用轻质单板 CMOS 摄像头；选用轻质碳卷支杆安装摄像头；尽可能的降低了摄像头高度；微控制板与大功率驱动板分开安装。



图 1.2 智能车的整体结构布局

1.3 文章结构

技术报告分七个章节：第一章节主要是对模型车设计制作的主要思路以及实现的技术方案概要说明，提出技术报告的行文框架。第二章介绍了赛车机械改造的总体思路，并详细说明了机械结构调整情况。第三章说明系统的硬件实现。第四章说明方向控制和速度控制算法设计。第五章介绍仿真和调试的方法。第六章是赛车的具体参数。第七章总结了整个制作过程中的创新点和不足之处，提出了下届备赛过程的努力方向。

第二章 机械结构调整

2.1 总体思路

关于赛车机械结构的调整，我们主要从以下几个方面考虑：

① 车体重量：比赛规则规定，智能车的驱动电机和传动机构不允许更改，意味着赛车的最大驱动力是一定的。要想提高赛车的平均速度，必须提高其加速和制动性能，在驱动力一定的情况下，尽量减轻车体的重量是必要的，我们将电路板体积做到了最小，在双面板情况下集成度做到了最高。

②重心安排：弯道的最小半径为 0.5 米，要使得智能车能在弯道上高速通过，必须防止侧滑和侧翻，尤其是侧翻。重心位置是影响侧翻的最关键的因素，所以设计中应尽量降低智能车的重心。为此，选用轻质单板 CMOS 摄像头；选用轻质碳卷支杆安装摄像头；尽可能的降低了摄像头高度；微控制板与大功率驱动板分开安装；

2.2 底盘调整

在独立悬架下摆臂与底板之间可以通过增减垫片来调整底盘离地间隙。垫片有 1mm 和 2mm 两种规格。一片垫片不加，车前部离地间隙为 9mm，故离地间隙的调整范围为 9~12mm。由于采用 CMOS 摄像头式布置结构，为保证车模高速行驶时的稳定性并降低重心，将底盘离地间隙设为最低 9mm。

2.3 前轮定位调整

调试中发现，在赛车过弯时，转向舵机的负载会因为车轮转向角度的增大而增大。前轮定位可保证转向的稳定性，定位是由主销内倾、主销后倾、前轮外倾和前轮前束四个因素决定的。

2.3.1 主销后倾角

主销后倾角是指在纵向平面内主销轴线与地面垂直线之间的夹角。它在车辆转弯时会产生与车轮偏转方向相反的回正力矩，使车轮自动恢复到原来的中间位置上。所以，主销后倾角越大，车速越高，前轮自动回正的能力就越强，但是过大的回正力矩会使车辆转向沉重。通常主销后倾角值设定在 1° 到 3° 。

模型车通过增减黄色垫片的数量来改变主销后倾角的，由于竞赛所用的转向舵机力矩不大，过大的主销后倾角会使转向变得沉重，转弯反应迟滞，所以设置为 0° ，以便增加其转向的灵活性。

2.3.2 主销内倾角

主销内倾角是指在横向平面内主销轴线与地面垂直线之间的夹角，它的作用也是使前轮自动回正。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 $0\sim 8^{\circ}$ 度范围内皆可”。在实际的调整中，只要将角度调整为 5° 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 8 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，由于过大的内倾角也会增大转向阻力，增加轮胎磨损，所以在调整时可以近似调整为 $0^{\circ}\sim 3^{\circ}$ 左右，不宜太大。

2.3.3 车轮外倾角

前轮外倾角是指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，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。在汽车的横向平面内，轮胎呈“八”字型时称为“负外倾”，而呈现“V”字形张开时称为正外倾。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。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。

模型车提供了专门的外倾角调整配件，近似调节其外倾角。由于竞赛中模型主要用于竞速，所以要求尽量减轻重量，其底盘和前桥上承受的载荷不大，所以外倾角调整为 0° 即可，并且要与前轮前束匹配。

2.3.4 前轮前束

所谓前束是指两轮之间的后距离数值与前距离数值之差，也指前轮中心线与纵向中心线的夹角。前轮前束的合理使用可以减小轮胎内外侧的磨损，同时保证稳定性。像内八字那样前端小后端大的称为“前束”，反之则称为“后束”或“负前束”。在实际的汽车中，一般前束为 $0\sim 12\text{mm}$ 。

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 $0\sim 2\text{mm}$ 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。智能车采用稳定速度策略或者采用在直道高速弯道慢速的策略时，应该调节不同的前束。后一种策略可以适当加大前束。

以上所述的前轮四轮定位的调整都是理论上的方法，受制于车模精度，实际调整时，应该以具体实验效果来进行修正与微调。

2.4 舵机的安装与控制延时解决办法

舵机的控制输出延时由微控制器输出周期延时和舵机机械延时两部分组成。可以说，舵机的响应速度直接影响智能车通过弯道时的最高速度。因此提高舵机的响应速度是提高智能车平均速度的一个关键。根据以上的分析，提出以下几个解决办法：

1. 提高 PWM 波频率。指定舵机可接收的 PWM 信号频率范围为 50 – 300Hz，对应的周期为 20ms – 3.3ms，通过上述分析可知，频率越高，微控制器输出周期延时就越短，可以有效减少延迟时间。

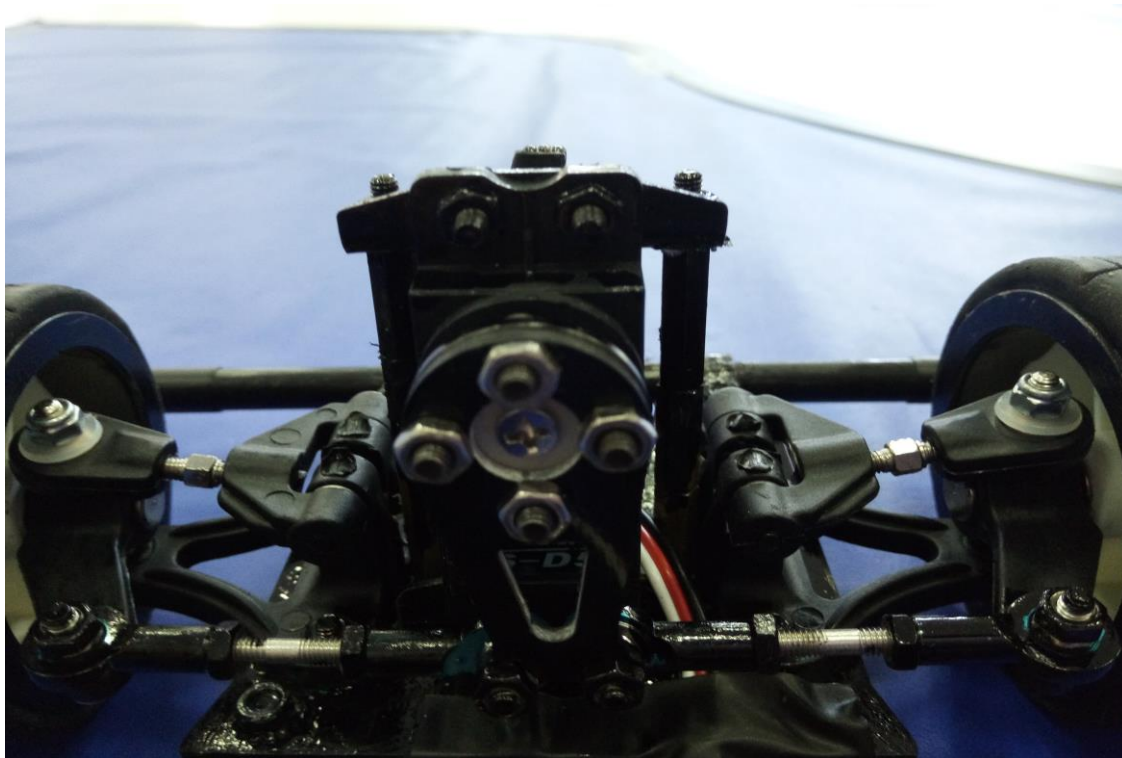


图 2.2 舵机安装方式

2. 提高舵机工作电压。在上述分析中可以看出，工作电压越高，响应速度则越快，我们使用舵机电压为 5V，略高于舵机标定的工作电压范围，但是可以直接舵机供电。如此一来，舵机的响应速度还可以进一步提高，大约为 0.15sec/60 degrees。
3. 安装更长的舵机输出臂。采用杠杆原理，在舵机的输出舵盘上安装一个较长的输出臂，将转向传动杆连接在输出臂末端。这样就可以在舵机输出较小的转角下，取得较大的前轮转角，从而提高了整个车模转向控制的速度。如图 2.2 所示，这种方法是通过机械方式，利用舵机的输出转距余量，将角度进行放大，加快了舵机响应速度。

2.5 摄像头支架的设计安装

赛车 CMOS 图像传感器的架设主要考虑以下几个因素：

- 1、确保 CMOS 图像传感器位置的居中且正对前方。因为当 CMOS 图像传感器不居中，其采集进来的图像也不是居中的，而处理程序对舵机输出量是居中的，这样就会导致智能车在直道上也偏离赛道中央，即使可以通过程序校正，也会导致扫描到的图像面积左右不

对称，会浪费一部分图像信息。

2、CMOS 图像传感器的安装高度要足够高。这样可以使得智能车在 CMOS 图像传感器的安装角度不是很大的情况下就能够前瞻到前方足够远处的路况信息。因为当摄像头的角度过大时候，采集进来的图像形变过大，且图像中的干扰信息增多，对模型车的处理算法十分的不利。而且当摄像头的安装位置较高时，所能扫描到的图像靠近智能车的部分范围较大，当智能车偏离赛道一定距离时，依然可以扫描到黑线，这样会便于图像的算法处理。一般要求该范围为 50cm 宽。

3、CMOS 图像传感器的安装是可调整的。这样以便于 CMOS 图像传感器居中的校正，以及在实际调试中选择最佳探测角度，以及对 CMOS 图像传感器视野范围进行标定。经过多次的实验和总结，我们对 CMOS 图像传感器的远度进行了标定，对 CMOS 图像传感器的采集的图像信息进行了中心位置的校正。将 CMOS 图像传感器的视频信息通过视频采集卡传到计算机中，通过调节 CMOS 图像传感器各个旋转变量使得摄像头的图像位置居中。校正后的摄像头能够采集到小车前方上底为 70cm，下底为 280cm，高为 260cm 的近似等腰梯形图像。

2.6 电路板安装

电路板应安装于智能车的最低的部位，并且固定于底盘。由于电池的安装位于车身前方，考虑到赛车的空间，PCB 板应设计成足够小，使得电池靠近中心并使整体重心位于摄像头支架的正下方。

第三章 硬件设计

3.1 总体方案

系统硬件电路主要由两块 PCB 板构成，主控板和电源电机驱动板，分别集成了整个系统的逻辑电路和驱动电路。系统主控板和驱动板如图 3.1。主控制芯片采用官方推荐的 32 位微控制器 K60，最小系统版使用山外 K60。

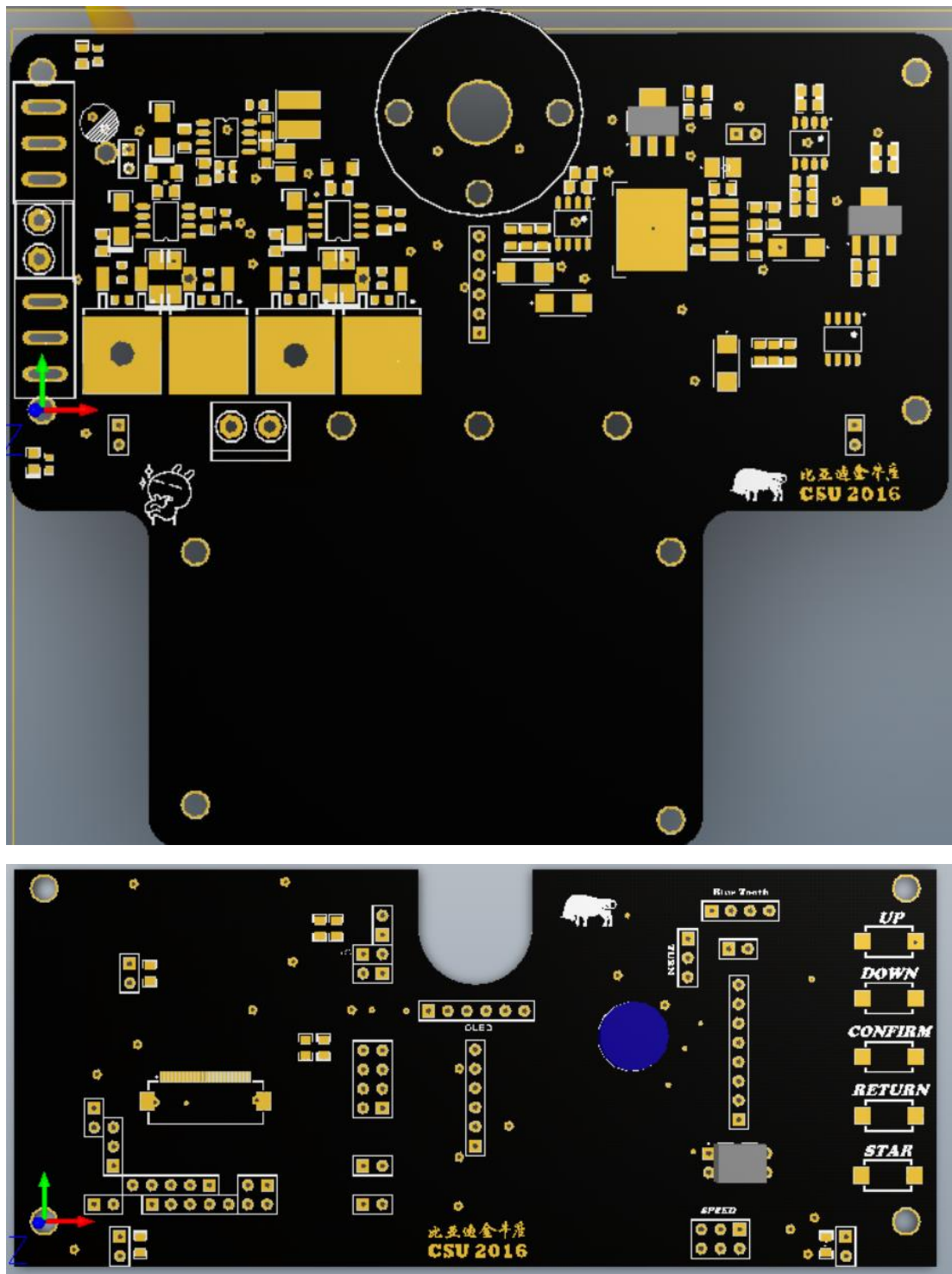


图 3.1 主控板与电源驱动板

主电机驱动电路采用功率 MOSFET 管搭建的 H 桥电路，该驱动电路导通内阻极小，相比之下赛车可获得更大的加速度和直道上限速度。

3.2 电源模块

由于电源对高频干扰具有较强的抑制作用。同时由于其低功耗特点，在进行电路板设计时，可以减少散热片的体积和 PCB 板的面积，有时甚至不需要加装散热片，方便了电路设计与使用，提高了稳定性能。

3.2.1 主电机供电与单片机供电分离电路

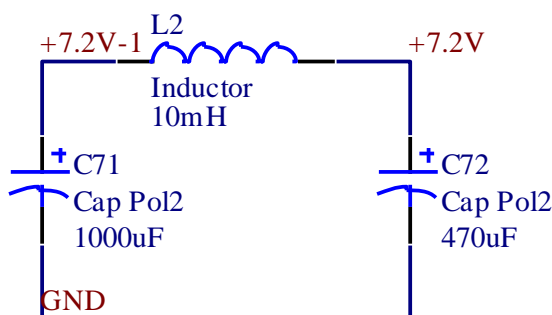


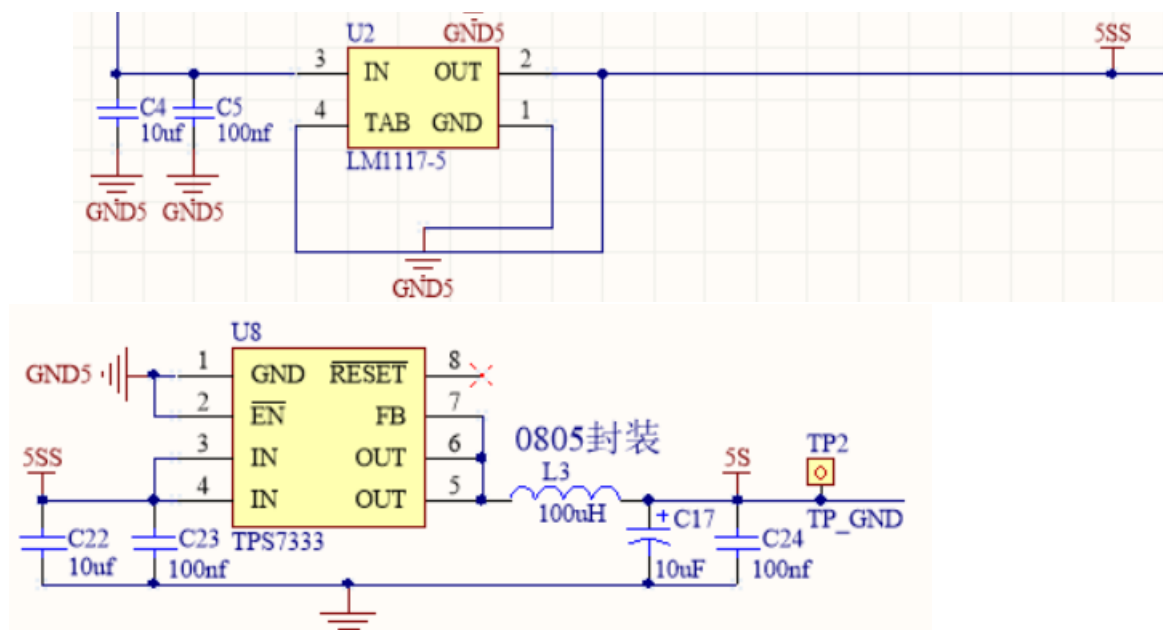
图 3.2 单片机电源和电机电源分离

主电机在启动的瞬间和反转的瞬间会产生高达 20A 的冲击电流。由于对赛车速度的控制采用的是 bang—bang 控制，因此在赛车的行驶的过程中始终存在较大的电流波动。而主电机和单片机都是由单独电池供电，若在主电机电源和单片机电源系统之间不加任何隔离措施，有可能导致在赛车行驶的过程中单片机复位，这是绝对不允许的。所以在主电机电源和单片机电源系统之间加了一极 π 滤波器。电路图如图 3.2 所示。

3.2.2 摄像头供电电路

摄像头的工作电压都是 3.3V,我们选用的芯片是 TPS7333 和 LM1117-5.0,具有纹波小、电路结构简单的优点，其输入的电压为电源电压，输出 3.3v。

其典型应用电路图如图



3.3 摄像头电源电路

3.2.3 舵机供电电路

舵机采用的官方规定的 SD-5，舵机电压为 5V，系统板中采用了 TI 公司生产的 LM2941 芯片，该芯片具有输出电压可调，输出稳定，而且可以在网上免费申请试用。电路图如图 3.4

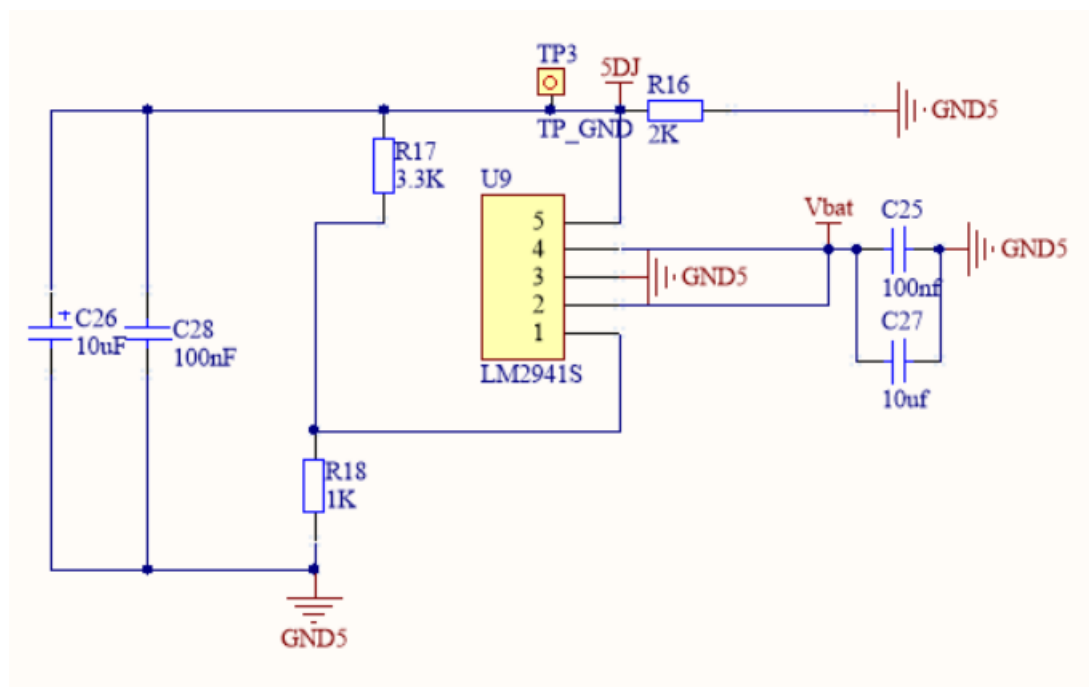


图 3.4 舵机稳压电路

3.3 电机驱动模块

驱动单元是控制系统的重要组成部分，采用低导通内阻的 MOSFET 功率管搭建 H 桥结合逻辑电路驱动，赛车的加速和制动性能以及上限速度得到了很大程度的提高。MOSFET 选型为 IRLR7843。

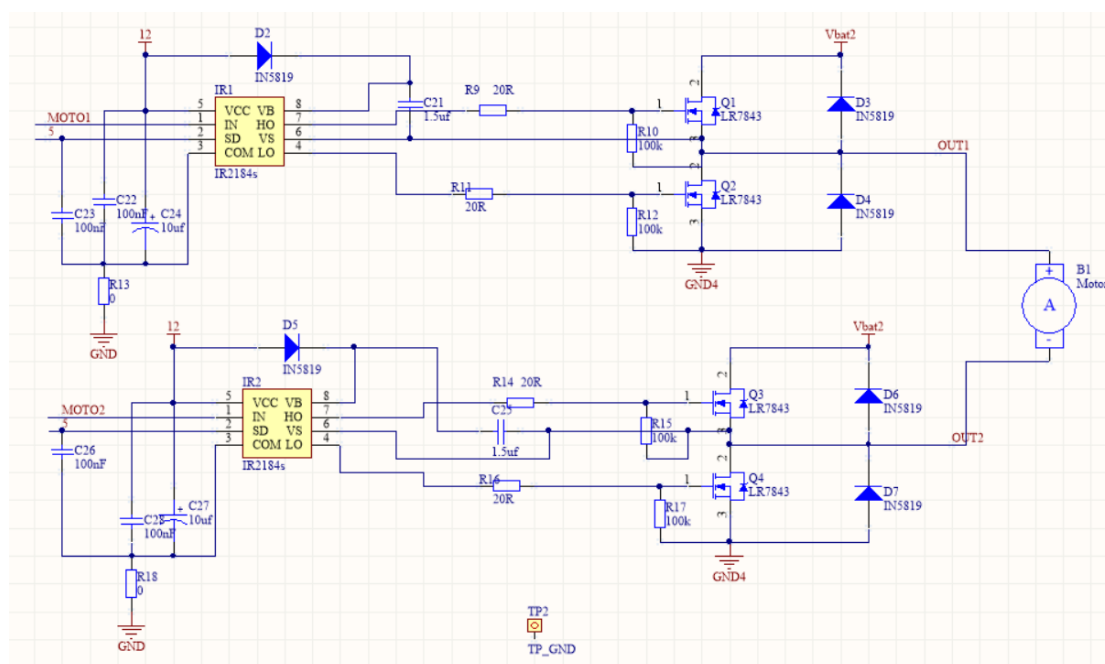


图 3.5 电机驱动电路

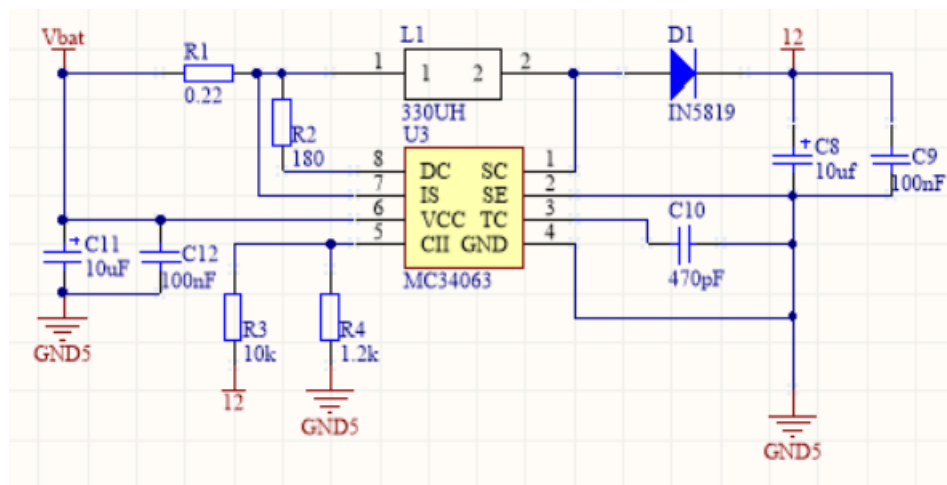
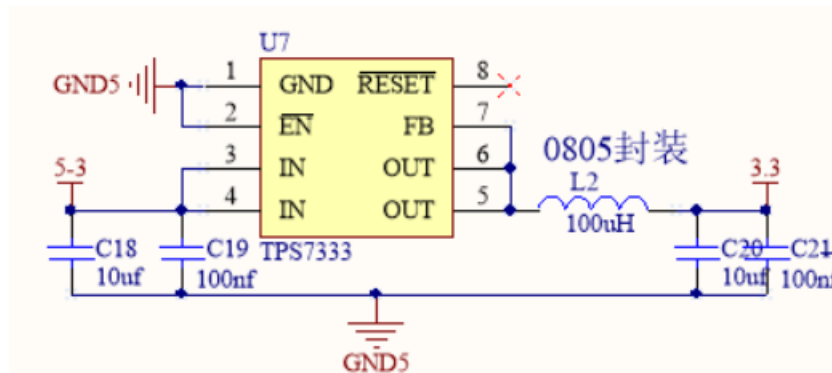
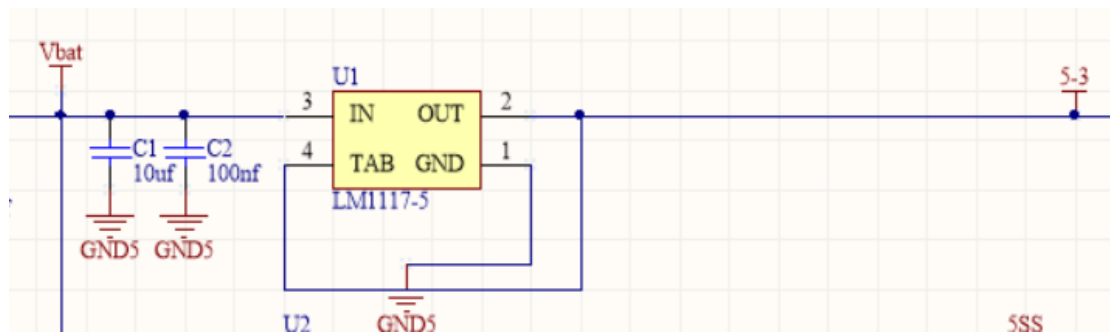
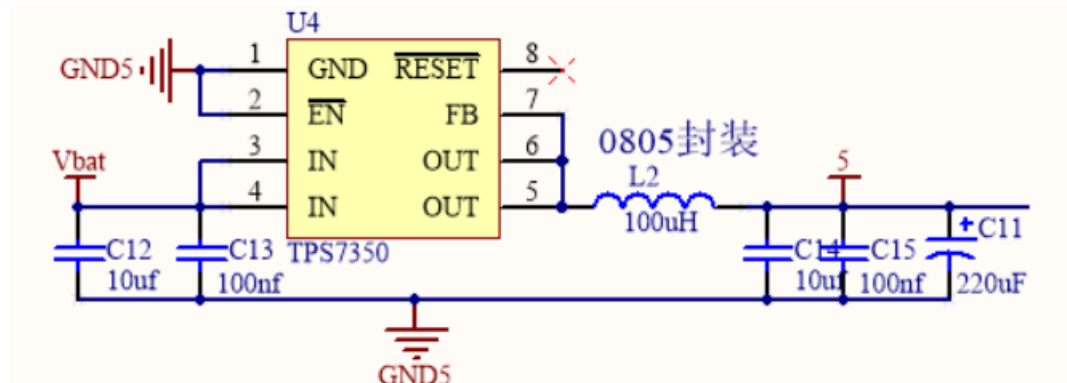
3.4 3.3V、5V 和 12V 供电电路

单片机、蜂鸣器和蓝牙等的工作电压都是 3.3V，我们选用的芯片是 TPS7333 和 LM1117-5.0，具有纹波小、电路结构简单的优点，其输入的电压为电源电压，输出 3.3V。

编码器等工作电压是 5V，我们选用的是 TPS7350，具有纹波小、电路结构简单的优点，其输入的电压为电源电压，输出 5V。

电机驱动芯片使用的 12V 电压，我们采用的是 MC34063 进行升压。

其典型应用电路图如图



3.3 电源电路

3.5 传感器模块

3.5.1 摄像头的选择

目前市面上常见的摄像头主要有CCD和CMOS两种：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，且图像稳定性不高；CMOS摄像头体积小，耗电量小，图像稳定性较高。因此，经过实验论证之后我们决定采用CMOS摄像头。

对于CMOS摄像头分为数字和模拟两种。其中数字摄像头OV7725可以直接输出8路数字图像信号，使主板硬件电路的简化成为可能，且能够达到所需要80的帧速率，只需要对其内部寄存器进行适当设置，因此，最终我们选择了CMOS数字图像传感器的方案。

3.5.2 编码器测速模块

光电编码器是一种通过光电转换将输出轴上的机械几何位移量转换成脉冲或数字量的传感器，这也是目前应用最多的测速传感器之一。其获取信息准确、精度高、应用简单，缺点是体积较大。

采用 512 线 MINI 编码器，其供电电压为 5V，输出为小幅值的正弦信号。为了将此信号放大整形，设计了信号调理电路，其基本原理是使用一个运放做成比较器电路，调节参考电压，使输出变为 0/5V 的方波信号，送入单片机进行运算。

3.6 MCU 主控模块

最小系统板选择山外 K60，MCU 控制板包括滤波电路，晶振模块，主控芯片 MK60DN512ZVLQ144。

Kinetis 32 位微控制器系列针对一系列成本敏感型汽车车身电子应用进行了优化。Kinetis 产品满足了用户对设计灵活性和平台兼容性的需求，并在一系列汽车电子平台上实现了可升级性、硬件和软件可重用性、以及兼容性。紧凑的封装使得这些器件适于空间受限应用，如小型执行器、传感器模块和转向柱集成模块。

第四章 软件设计

系统硬件位于底层，是整个系统的基础，系统软件结构则根据硬件和控制需求来制定。系统的基本软件流程为：首先，对各功能模块和控制参数进行初始化。然后，通过图像采集模块获取前方赛道的图像数据，同时通过速度传感器模块获取赛车的速度。根据采集到的赛道信息，采用 PD 对舵机进行反馈控制。另外根据检测到的速度，结合速度控制策略，对赛车速度不断进行适当调整，使赛车在符合比赛规则的前提下，沿赛道快速行驶。

4.1 中心线的提取和图像中心的计算

4.1.1 原始图像的特点及校正

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，同时，由于起点线的存在，光线、杂点、赛道连接处以及赛道外杂物的干扰，图像效果会大打折扣。因此，在软件上必须排除干扰因素，对赛道进行有效识别，并提供尽可能多的赛道信息供决策使用。

因为 CMOS 图像传感器所扫描到的图像是一幅发射式的图像，产生了严重的畸变，如果不进行校正，那么扫描到的数据的正确性将无法保证，那么对于道路形状的判定将会不准确，造成赛车判断失误，走的不是最优路径，严重时甚至冲出轨道。因此，在进行赛车方向控制时，进行坐标的校正显得至关重要。下面对坐标校正的方法进行具体的介绍。

由于 CMOS 图像传感器所扫描到的图像总是向外扩张，扫描到的图像并非为一个标准的长方形，在不考虑纵向畸变的情况下可以近似认为是一个梯形，严重影响到了数据采集的准确性，所以我们必须对扫描到的数据进行校正。我们根据实验发现，越是靠近 CMOS 图像传感器的地方，扫描到的范围就越窄，离 CMOS 图像传感器距离较远的地方，扫描到的范围就越宽，因此，在相同的情况下，如果 CMOS 图像传感器在最近处和最远处采集到

赛道边线的数字信号距离中心位置均是 30 的话，事实上最远处偏离中心位置的距离可能比最近处偏离中心位置的距离大很多。按照赛车的安装方法，在扫描到的数字量相同的情况下，在第 30 行处偏离中心位置的距离超过了第 0 行偏离中心位置的 2 倍还要多，故必须进行校正，校正的方法就是在不同的行乘以一个不同的系数，系数大小的确定是先进行理论粗略的计算，然后再在事物上进行校正。

4.1.2 中心线的提取和中心的计算

中心线提取算法的基本思想如下：

- (1) 直接逐行扫描原始图像，根据硬件二值化结果直接提取出边界坐标；
- (2) 图像数据量大，全部扫描一遍会浪费很多时间，利用前面已经求出的中线位置判断出中线的趋势，从而推断出下一行的中线大概位置，确定出扫描范围，避免整行逐点扫描，节省时间。求中线时，因为近处的中线稳定，远处中线不稳定，所以采用由近及远的办法；根据上一行的左右两边的坐标，确定本行的边线扫描范围。在确定的赛道宽度范围内提取有效赛道，这样可以滤除不在宽度范围内的干扰；
- (3) 根据采集到的赛道宽度，判断是否采集为真正赛道，滤除噪点造成的影响；
- (4) 图像是远处小近处大，所以赛道宽度范围和前后行赛道中心的位置差别都要动态调整；
- (5) 比较左右赛道边界坐标的大小，判断赛道的有效性；

4.2 方向控制方案

4.2.1 方向控制数据的选取

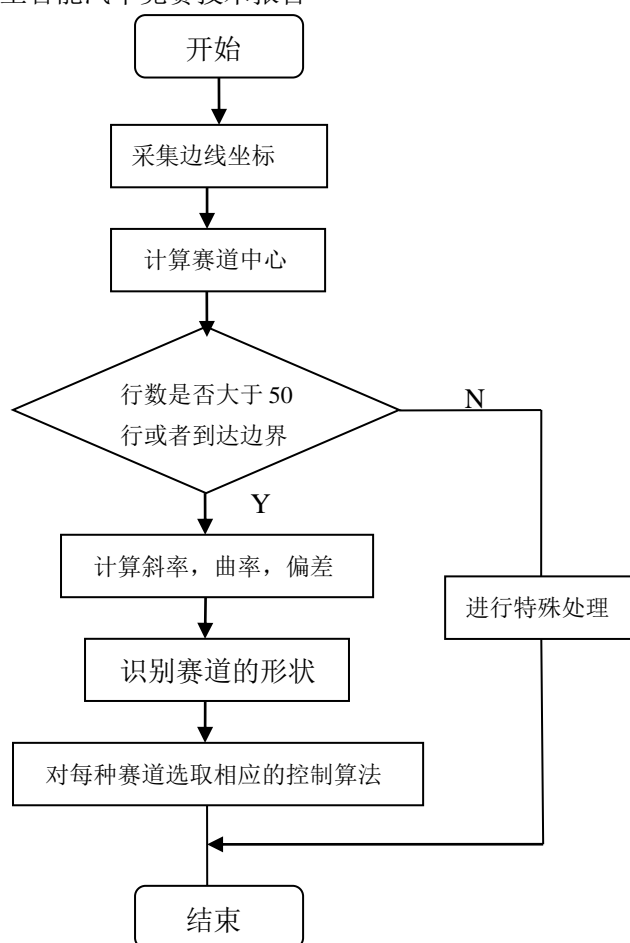


图 4.1 方向控制流程图

根据前面介绍的 CMOS 图像传感器采集处理得到的只是 50 行中每一行的中线坐标。并没有得出赛道的具体形状，到底是直道还是弯道，是大弯还是急弯，是偏向一个方向的弯道还是“S 型”弯道，是“大 S 型”弯道还是“小 S 型”弯道。这就可以通过曲率的计算来得出。下面给出曲率的具体计算方法。

首先，将扫描到的 50 行中每 9 行分为一组，而且彼此交叉，也就是说，在能扫描到的情况下，划分为从第 0 行到第 8 行，从第 1 行到第 9 行，从第 2 行到第 10 行，依此类推，最后从 21 行到第 29 行，一共可以划分为 22 段，分别计算每一段的曲率，以第一段为例，计算的方法为：将第 i 行的坐标定义为 $coordinate[i]$ ，将 CMOS 图像传感器的中心坐标定义为 $middle$ ，那么每一行相对于中心坐标的偏差就为

$$ek[i]=coordinate[i]-middle \quad (4.1)$$

再乘以每一行各自的校正值 $emendation[i]$ ，计算公式为

$$ek[i]=ek[i]*emendation[i] \quad (4.2)$$

就得到了各行实际偏离中心位置的距离。然后用第 0 行的偏差加上第 8 行的偏差之和除以 2 再减去第 4 行的偏差，得出一个相对偏差 $ek_comparatively$ 。 $ek_comparatively$ 的计算公式为

$$ek_comparatively = (ek[8] + ek[0]) / 2 - ek[4] \quad (4.3)$$

然后计算出该段曲线对应的直道，即将第 0 行的黑点与第 8 行的黑点直接相连的直线的长度计算出来，计算公式为

$$distance_beeline = \sqrt{(ek[8] - ek[0])^2 + 48^2} \quad (4.4)$$

最后将公式(4.3)除以公式(4.4)，得到最后曲线的弯曲度为 $curve_degree$ 。计算公式为

$$curve_degree = ek_comparatively / distance_beeline \quad (4.5)$$

就得出了曲线的弯曲程度，然后根据每一段的弯曲程度，最终得出所扫描到的全部道路的形状。

4.3 分类进行方向控制算法

4.3.1 直道的方向控制算法

对于赛道中的直道，是方向控制中最好处理的一种情况，因为只要小车没有偏离赛道，就可以不进行偏转，而当智能车偏离赛道时也只需要一个较小的偏转，让智能车能缓慢回归赛道就可以了，具体的方法是计算扫描到图像中线偏离赛道中线的平均值和中线的斜率，再将这两个数分别乘上各自的比例系数，加上舵机偏向中心位置时需要给出的高电平的值，作为 PWM 波的高电平送入给舵机，就可以实现直道上的智能车方向控制了。

4.3.2 大弯的方向控制算法

在道路中遇到大弯时需要转弯，转向角只受弯道的弯曲程度影响，但是距离大弯多远就开始转弯却需要受到当前速度的影响，因为舵机在转向时有一定的延时，舵机转 60 度大约需要 140 毫秒，而智能车的最快速度可以达到 10 米/秒，如果智能车当时运行的速度就是 10 米/秒，那么当舵机转过 60 度时智能车已经走过 1 米多了，所以不能等到了弯道才开始转弯，而是需要提前转弯，而且速度越快，越是要提前得多，所以智能车转弯提前多少就由速度控制，用比例控制就能收到很好的效果，而转弯时转多少却仅仅受弯道的弯曲程度控制，弯曲程度越大，转向角也就越大，弯曲程度越小，转向角也就越小。具体的对应关系通过实验获得，在此没有列举出具体的数据。

4.3.3 急弯的方向控制算法

急弯的方向控制算法与大弯的方向控制算法相似，也是需要提前转弯，而且提前多少转弯也要受当前速度的控制，速度越大，越要提前转弯，转向角也是用弯道的弯曲程度来决定的，急弯比大弯的弯曲程度更加大，因此需要的转向角也要相应增大。急弯的方向控制比大弯更难，很可能出线，因此在急弯的方向控制时更应多加注意。

4.3.4 “小 S 型”道的方向控制算法

“小 S 型”道一般是指“S 型”道弯曲角度小于等于 60 度形成来回转向，智能车过这种弯道时，基本可以从中间直接通过，而不需要转弯和减速。因此，找准“小 S 型”道的正中心显得至关重要，因为只有找准了“小 S 型”道的正中心，智能车才能不转弯不减速地通过“小 S 型”道，如果“小 S 型”道的正中心找得不准，那么智能车就很容易冲出赛道。找“小 S 型”道的正中心的方法为：先找出“小 S 型”道最左边和最右边的坐标值，然后将最左边和最右边的坐标值相加之后再除以 2，就得到了“小 S 型”道的正中心坐标，智能车就可以沿着这个方向前进，而不需要转弯。但是，由于 CMOS 图像传感器扫描到的图像可能出线误差，那么智能车就有可能产生误判，而冲出赛道。因此为了保险起见，必须当“小 S 型”道足够小时，才能采用直接冲的办法。

4.4 速度控制方案

4.4.1 速度控制的重要性

要想智能车以最短的时间跑完整个赛道，不但要求优越的方向控制算法，让智能车尽量走更近的路，转更少的弯。而且对于速度控制算法也有很高的要求，智能车在行驶过程中，不能够完全匀速，在直道上应该提速，而在弯道上应该减速。

在智能车沿着既定轨迹行驶的过程中，速度不能够恒定不变，而应该根据道路的具体形状进行适当的调整。当然，最好的方法是让智能车始终以最快的速度通过整个赛道，但是如果速度始终调得太高，当小车遇到急弯或者是圆道时必然无法转过来，就会冲出赛道。因此不能够始终给最快的速度。然而，如果速度调得始终过低，就无法以最短的时间通过赛道。因此，速度应该要随着检测到的道路形状变化而变化。例如当前方道路为直道或类似直道时，应该将

PWM 波提得最高，让通过主电机的电流达到最大，从而让智能车以最快的速度行驶。当进入弯道时，应该将 PWM 波降到最低，让通过主电机的反向电流达到最大，从而让智能车以最短的时间将速度降到理想状态。

4.4.2 速度控制的算法设计

智能车速度控制的总体思路与智能车方向控制的思路完全相同，也是将道路的形状进行分类，对于不同的类给以不同的速度。而在某一时刻给出智能车的理想运行速度以后，采用棒—棒控制与 PID 控制相结合的算法实现对智能车的速度调节，以最短的时间使智能车达到这一理想速度。从而实现希望智能车的速度完全由道路情况所控制。

智能车速度控制的程序流程图如图 4.6 所示。

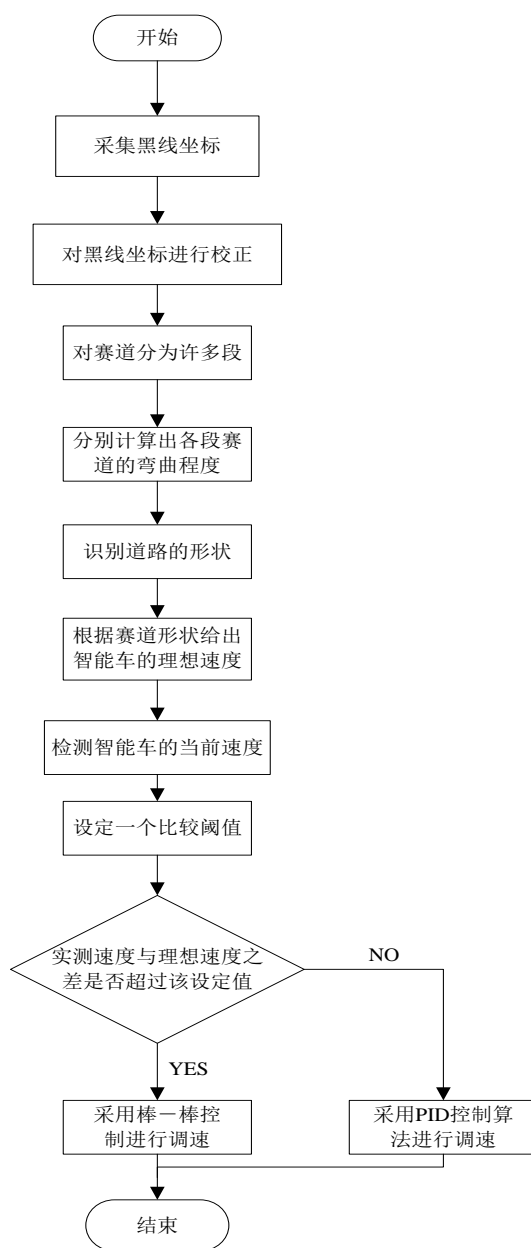


图 4.6 智能车速度控制总体流程图

4.5 棒-棒控制算法

4.5.1 棒-棒控制算法简介

棒-棒控制的基本思路是当检测到的输出量大于理想输入量时，就急剧将控制量增加到最大，让输出量以最快的速度减少，当检测到的输出量小于理想的输入量时，就急剧将控制量降低到最小，让输出量以最快的速度减少。

4.5.2 棒—棒控制算法用于该赛车系统

本赛车系统的速度控制采用了棒—棒控制与PID控制相结合的方式。其基本思路为：智能车的理想速度由当前的道路信息决定。当编码器检测到的速度与理想速度相差较远时，采用棒—棒控制；当编码器检测到的速度与智能车的理想速度逐渐接近以后，就采用PID控制。对主电机的控制是通过一个H桥，将H桥的两个口分别接单片机的两个管脚，在本系统中，H桥的两端分别接单片机口中的PTP3和PTP1，当PTP1为高电平，PTP3为高电平时，主电机全速正转，当PTP1为低电平，PTP3为低电平时，主电机全速反转。采用棒—棒控制进行速度控制的原理就是检测智能车的当前速度，如果当前速度低于理想的速度，就PTP3置为高电平，从PTP1口输出一路占空比很高的PWM波，电机就能以最大的加速度升速。反之，就把PTP3置为低电平，从PTP1口输出一路占空比很大的PWM波，从而让电机以最大的加速度降速。

4.5.3 棒—棒控制与 PID 控制两种方法结合的必要性

棒—棒控制算法由于其快速性非常好，而本赛车系统的速度调节也正好要求很好的快速性，因而在本智能车的调速过程中用到了棒—棒控制算法也就理所当然，但是，棒—棒控制算法在用于本赛车调速时也存在着一一定的不足，由于主电机的数学模型为一个惯性环节，存在着一定时间的延时，因此如果仅仅使用棒—棒控制一种算法可能导致超调，也不可能达到稳定，且在任何时刻通过主电机的电流都始终为最大，只是在不同的时刻可能会出现一正一反。正是出现一正一反的这种情况，对主电机以及主电机驱动电路的冲击最大，既造成整个智能车功耗的增加，又可能导致主电机及主电机驱动芯片因电流冲击过大而烧坏。因而在智能车的实际检测速度与理想速度接近时，可以采用 PID 控制算法，这样就既能实现调速的快速性，又能实现速度的平稳性，降低智能车的整体功耗，达到很好的调速效果。

4.6 PID 控制算法

4.6.1 PID 控制算法简介

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 4.7 所示。

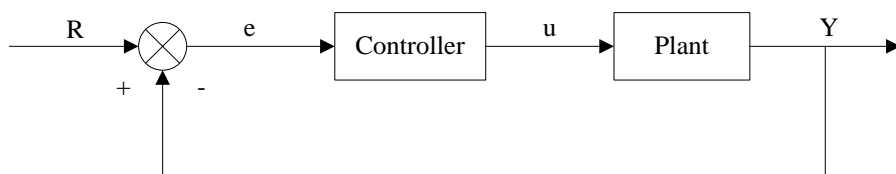


图 4.7 单位反馈的 PID 控制原理图

单位反馈 e 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 。

$$u(t) = k_p[e(t) + \frac{1}{T_i} \int_0^t e(t)dt + T_D \frac{de}{dt}] \quad (4.6)$$

对应的模拟 PID 调节器的传递函数为

$$D(s) = \frac{U(s)}{E(s)} = K_p(1 + \frac{1}{T_i s} + T_D s) \quad (4.7)$$

其中， K_p 为比例增益， K_p 与比例带 δ 成倒数关系即 $K_p = \frac{1}{\delta}$ ， T_i 为积分时间常数， T_D 为微分时间常数， $u(t)$ 为控制量， $e(t)$ 为偏差。

4.6.2 数字 PID 控制算法

1. 数字PID位置型控制算法

在单片机控制系统中，PID 控制规律的实现必须用数值逼近的方法。当采样周期相当短时，用求和代替积分，用后向差分代替微分，使模拟 PID 离散化变为差分方程。

为了便于单片机实现，必须把公式(4.6)变换成为差分方程，为此可作如下近似

$$\int_0^t e(t)dt \approx \sum_{i=0}^k T e(i) \quad (4.8)$$

$$\frac{de(t)}{dt} \approx \frac{e(k) - e(k-1)}{T} \quad (4.9)$$

式中， T 为采样周期， k 为采样序号。

由公式(4.6)、公式(4.7)、公式(4.8)可得数字 PID 位置型控制算式为

$$u(k) = K_p[e(k) + \frac{T}{T_i} \sum_{i=0}^k e(i) + T_d \frac{e(k) - e(k-1)}{T}] \quad (4.10)$$

公式(4.10)表示的控制算法提供了执行机构的位置 $u(k)$ ，所以被称为数字 PID 位置型控制算式。

2. 数字PID增量型控制算法

由公式(4.10)可以看出，位置型控制算法不够方便，这是因为要累加偏差 $e(i)$ ，不仅要占用较多的存储单元，而且不便于编写程序，为此可对公式(4.5)进行改进。

根据公式(4.10)不难写出 $u(k-1)$ 的表达式^[8]，即

$$u(k-1) = K_p[e(k-1) + \frac{T}{T_i} \sum_{i=0}^{k-1} e(i) + T_d \frac{e(k-1) - e(k-2)}{T}] \quad (4.11)$$

将公式(5.10)和公式(5.11)相减，即得数字 PID 增量型控制算法为

$$\begin{aligned} \Delta u(k) &= u(k) - u(k-1) \\ &= K_p[e(k) - e(k-1)] + K_i e(k) + K_d[e(k-1) - 2e(k-1) + e(k-2)] \end{aligned} \quad (4.12)$$

其中 $K_p = \frac{1}{\delta}$ 称为比例增益

$K_D = K_p \frac{T}{T_i}$ 称为积分系数

$K_D = K_p \frac{T_D}{T}$ 称为微分系数

我们在实际代码实现时，处理成

```
error[0]=0;
```

```
error[1]=0;
```

```
error[2]=0;
```

```
PreU=0;
```

```
for(;; )
```

```
{error[0]=error[1];
```

```
error[1]=error[2];  
error[2]=r-y;  
PreU+=  $K_p$  *(error[2]-error[1])+  $K_I$  *error[2];  
PreU+=  $K_D$  * (error[2]-2*error[1]+error[0]);  
.....  
}
```

其中，error[2]为本次的偏差，error[1]为上一次的偏差，error[0]为上上次的偏差，PreU 为输出的控制量。

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，其实如果对控制有更高的要求或者干扰因素较多，我们可以对 PID 算法做各种改进，比如用梯形法做数值积分以提高精度，将差分改成一阶数字滤波等等，在实际调车的过程中，我们确实遇到过由于光电编码器采样得到的脉冲上升下降沿不够陡，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^[8]。

4.7 经典 PID 算法实现速度的控制

4.7.1 经典 PID 算法实现速度控制的计算机仿真

智能车的主电机可以看着是一个惯性单元，应用 PID 控制算法可以达到较好的控制效果，该智能车在 MATLAB 上进行仿真得到的单位阶跃响应如图 4.12 所示，因为在本赛车速度控制系统中，其中一个最重要的指标就是快速性，而

对于超调量和过渡过程时间要求并不严格。特别是过渡过程时间，该智能车的速度时时刻刻都需要发生变化，所以还等不到速度稳定便又要变化，由仿真结果可见，该智能车速度调节的快速性很好，能够完全满足调速的要求。

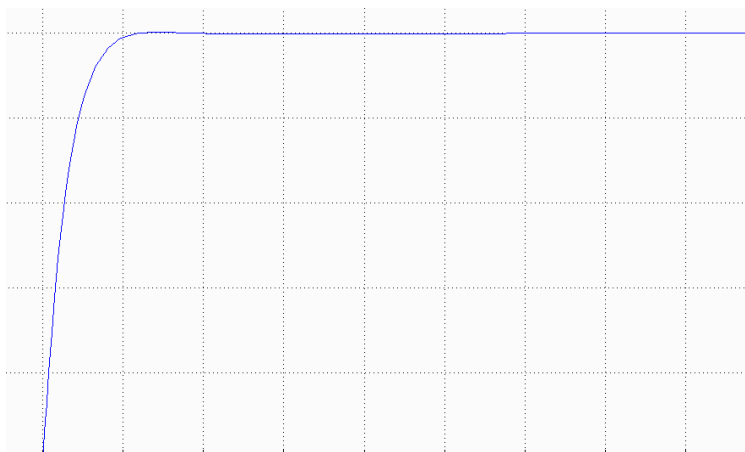


图 4.8 PID 速度控制算法的 MATLAB 仿真

4.7.2 经典 PID 算法实现速度控制的具体实现

我们对速度的控制采用了增量式 PID 算法。该算法的基本策略是急弯降速，直道和小弯提速，由于我们的摄像头扫描到的范围较宽，所以在过急弯道时可以提前减速，先把扫描到的图像进行处理，然后得出经过反复实验，将图像经过算法处理后得到的黑线位置和对应的速度 PID 参照速度处理成二次曲线的关系。在实际测试中，发现小车直道和弯道相互过度时加减速比较灵敏，与舵机转向控制配合较好。

但是，存在的局限一方面是车在弯道进直道后的加速和直道入弯道的速度控制并达不到最好的控制效果，弯道入直道减速不够快速，直道入弯道加速得时机也不够及时。我们做了进一步的改进，根据入弯时中线位置的特点动态改变二次曲线中最高点（直道的最高速度）和最低点（弯道的最低速度）的大小，使得控制效果更合理。

另一方面是没有考虑到实际比赛中长直道急速冲刺的情况，赛前在程序中人为设定直线速度不够灵活合理，所以在程序中根据赛道状态动态提高直线速度，使得我们可以在长直道的赛场充分发挥车的潜能。

4.7.3 PID 参数调节

比例控制能迅速反应误差，从而减少误差，但是比例控制不能消除静态误差， K_p 的加

大，会引起系统的不稳定；积分控制的作用是，只要系统存在误差，积分控制作用就不断地积累，输出控制量以消除误差，因而，只要有足够的时间，积分控制将能完全消除误差，积分作用太强会使系统超调加大，甚至使系统出现振荡现象；微分控制可以减小超调量，克服振荡，使系统的稳定性提高，同时加快系统的动态响应速度，减少调整时间，从而改善系统的动态性能。而在该智能车系统调速的过程中，由于快速性是最重要的指标之一，因而应主要考虑其快速性，在调试的过程中，把 K_p 参数调得较大，这是为了增大速度调节的快速性，将 K_D 参数调得较为适中，因为除了快速性以外，最重要的就是要减少系统的超调量，而将 K_I 参数调得较小，因为在该速度调节中，并不需要达到静态，由于理想速度就在时时刻刻发生变化，所以也不可能达到静态。实验结果表明， K_p 参数取 260， K_D 参数取为 0，而 K_I 参数取得尽可能小，大约为 1.5 时，可以取得较好的效果。

第五章 仿真与调试

要想将所设计的算法运用到智能小车上，并很好的控制智能小车的各个机构协调工作，并不是一蹴而就的事情，必须经过初期的设计、仿真、制作和长期的调试工作。而这一系

列工作也是反复循环进行的，在调试过程中对设计方案进行修改，然后重新仿真制作后再进行新一轮的调试。

5.1 调试阶段

5.1.1 摄像头调试

CCD/CMOS 摄像头是智能小车最重要的传感器，知道安装在智能小车上的摄像头看到的路面图像是什么样的，这一点在调试过程中是很重要的。由于我们所使用的摄像头可以实现硬件二值化，因此再加上 NRF 无线发送装置后，将会使实时图像传输变得更为简单。

5.1.2 软件系统调试

本文之前已经详细的讨论了智能小车的硬件结构和图像处理算法、控制算法等，但是这些算法都需要用软件即使用程序语言来实现。我们使用 ARM 内核芯片通常使用的 IAR 软件进行程序的编写以及调试。

第六章 赛车主要技术参数

表 6.1 赛车技术参数统计

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）（毫米）	320*160*310
车模轴距/轮距（毫米）	200（轴距），160（前轮距）， 170（后轮距）
车模平均电流（匀速行驶）（毫安）	2000
电路电容总量（微法）	1000
传感器种类及个数	OV7725 摄像头 1 个，迷你 512 线编码器 1 个
新增加伺服电机个数	0 个
赛道信息检测空间精度（毫米）	10
赛道信息检测频率（次/秒）	75
主要集成电路种类/数量	数字 74 系列集成电路
车模重量（带有电池）（千克）	1.250

第七章 总结与展望

7.1 总结

要实现对高速行驶汽车的自主智能控制并不是一个简单的自动控制问题，它涵盖了控制、模式识别、力学、光学、电磁学、传感技术、电子、电气、计算机、机械及车辆工程等多个学科。本文采用的控制核心是一款恩智浦公司生产的 32 位微控制器——MK60DN512ZVLQ144，利用了微控制器的强大功能实现了智能小车对路径的自主寻迹，以及在未知环境下，结合一定的算法，实现了对智能小车的高速导航控制，从最终测试结果来看，本系统具有较好的控制性能与对未知环境的适应能力。

对智能小车系统的自主控制，可以分为三大部分：首先是对路径信息的提取与识别，即智能小车的寻迹；然后是对前轮舵机方向的控制，即智能小车的导航控制；最后是对后轮车速的控制，即智能小车的驱动控制。

智能小车的导航控制，主要就是对智能小车的舵机系统进行控制，但是，前轮舵机系统作为一个机械执行机构，舵机系统中存在着饱和、摩擦和间隙等非线性因素，再加上 MK60 微控制器自身产生 PWM 波形机制的问题，使舵机系统具有机械延时和输出周期延时两部分，造成了比较大的延时。本文在详细论述了延时原因的基础上，提出了几点解决办法，分别从软件和硬件两方面下手，减少了舵机的延时，使对舵机控制的实时性更好。智能小车的舵机转向控制是一个复杂的随动控制问题，很难求得其准确的模型，此时 PID 的控制效果将难于达到预期的目标。因此我们根据人驾驶车辆的经验，采用了分类控制算法。

在搭建好智能小车的软硬件平台后，我们进行了一系列的仿真和调试，实验证明，我们的寻迹方案与导航控制策略是稳定、有效的。完成了对智能小车的自动控制，并在制作的赛道上实现了智能小车的高速、自主行驶。在调试过程中，我们也发现了一些问题，由于对汽车工程理论与机械结构设计的不熟悉，使我们对智能小车结构的改造和对摄像头安装支架的设计存在着缺陷和不足，这就约束了智能小车的提速，例如，摄像头支架过重使智能小车重心偏高，导致在高速行驶转弯时会发生翻车的情况。此外，我们对图像信息的

处理算法和导航控制算法也有改进和提升的空间，例如，可以尝试在导航控制中使用机器学习等先进的算法。这些都是我们今后努力的目标。

7.2 展望

智能车系统的研究十分复杂，需要解决的问题很多，任务非常艰巨，不是一蹴而就的，必须经过长期的理论研究和实践探索才能够取得突破和进展。展望未来，对于我们所研究的这类智能小车，我们认为今后还可在以下几个方面做进一步的研究和提升。

1)、采用优质的数字摄像头，数字摄像头可以非常方便的调整其参数，对环境适应能力更强。

2)、通过反光镜成像原理更加减小摄像头的高度，可以使整车摄像头支架高度降低一倍，这样重心更低更有利于弯道提速。

3)、在舵机控制策略上，可以考虑使用更先进的控制算法。

4)、硬件传感器方面。可以尝试采用 CMOS 摄像头装于摄像头云台方案。

5)、小车机械性能方面。需要学习一些汽车工程方面的知识，以便对智能小车的车体改造有理论指导和依据。此外可以在考虑外增加制动装置在入弯时进行减速。这样直道速度可以提高，平均速度更快。但如何选择合适的制动装置并对其进行控制，还需要进一步深。

此外，我们还有在车模上安装主动悬挂装置的方案，鉴于时间不充裕，于是在此次比赛中取消了该方案，今后可能会考虑加入。

谢 辞

今年能够参加第十一届恩智浦杯全国大学生智能汽车赛我们首先要感谢信息科学与工程学院王击老师的悉心指导，指导老师多次询问研究进程并为我们指点迷津，甚至和我们一起奋战至凌晨三四点。其严谨的治学态度以及对我们的关怀使我们获益甚多，感动颇深。

在准备比赛期间我们得到了以前参加过比赛的学长的许多非常实用的帮助和建议，在这里要特别感谢他们。另外也感谢一起准备比赛的队员，没有他们的帮助我们不可能完成工作。

另外我们还要感谢德州仪器公司以及恩智浦公司为我们提供的芯片支持，没有一个大范围比较是不可能选出最优最合适元器件出来的，我们的小车能够顺利的参赛要感谢这两大公司的帮助。

参考文献

- [1] 教育部高等学校自动化专业教学指导分委员会. 关于举办第十一届全国大学生“恩智浦”杯智能汽车竞赛的通知[EB/OL]. 2015-11-01.
- [2] 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.
- [3] 方兴, 郭思明, 马明煌. 上海交通大学速度之星队技术报告.
- [4] 卓晴, 黄开胜, 邵贝贝. 学做智能车[M]. 北京: 北京航空航天大学出版社, 2007. 24~27.
- [5] Freescale Semiconductor Inc. S12PWM8B8CV1/D Block User Guide V01.17[Z]. 2004. 1~51.
- [6] National Semiconductor. LM1881 Video Sync Separator General Description[Z]. June 2003
- [7] 杨明, 宋雪峰, 王宏, 张钺. 面向智能交通系统的图像处理[J]. 计算机工程与应用. 2001 年 09 期.
- [8] 余灿键, 程东成, 李伟强. PID 算法在智能汽车设计上的应用.——《学做智能车——挑战“飞思卡尔”杯》[C]. 北京: 北京航空航天大学出版社, 2007. 133~135.
- [9] Todd D. Motton. 嵌入式微控制器 Embedded Microcontroller[M]. 北京: 科学出版社. 2003 年 8 月第 1 版.
- [10] 吴志辉, 陈水保, 陈彬. 中南大学比亚迪金牛座 2012 队技术报告

```

#include "common.h"
#include "include.h"
#include "ALEX_camera.h"
#include "ALEX_motor.h"
#include "ALEX_steer.h"
#include "ALEX_key.h"
#define LPTMR0COUNT          0xFFFF
#define Centermeterpoint      71
#define BEEP_ON                gpio_set(PTE10,1)    //蜂鸣器
#define BEEP_OFF               gpio_set(PTE10,0)
#define led1_on                gpio_set(PTB20,0)
#define led1_off               gpio_set(PTB20,1)
#define led2_on                gpio_set(PTB21,0)
#define led2_off               gpio_set(PTB21,1)
#define led3_on                gpio_set(PTB22,0)
#define led3_off               gpio_set(PTB22,1)
#define led4_on                gpio_set(PTB23,0)
#define led4_off               gpio_set(PTB23,1)
uint8    imgbuff[CAMERA_SIZE];
uint8    img[CAMERA_H][CAMERA_W];
uint8    OLED_img[60][80]={ 0 };
int8     vcan_ware[2]={ 0 };
float    speedA=0;
float    speedB=0;
float    speedC=0;
int16    speed_count=0;
int16    exspeed=0;
int16    speed[4]={ 105,97,105,140 };
float    mkp[7]={ 250,260,270,280,290,300,320 };
float    mki=2.3;
float    motorkp=0,motorki=0,motorkd=0;
uint8    motor_on=0;
int      speedbuff=0;
int      length_centermeter=0;
int      length_meter=0;
uint8    exlength=0;
float    length=0;
float    obstaclekaishi=0;
int16    speederr=0;
uint8    Timeflag5ms=0,Timeflag10ms=0,Timeflag15ms=0,Timeflag100ms=0;
uint8    ENDFLAG=0;

```

第十一届全国大学生智能汽车竞赛技术报告

```
uint8    ENDCOUNT=0;
uint8    ABSdelay=0;
uint8    BUZZdelay=0;
uint8    Dangerflag=0;
uint8    Stop=0;
uint8    dangerstop=0;
uint8    obstacleflag=0;
uint8    obstaclesignal=0;
uint8    Startline=0;
float    Timenow=0;
uint8    functionchoose[8]={0,0,0,0,0,0,0,0};
uint8    key=0;
uint8    typechoose=0;
float    obstacleyanchi=0.3;
float    QPjiance=26;
float    obstaclejiance1=10;
float    obstaclejiance2=20;
float    obstaclejiance3=30;
float    rampjiance=60;
uint8    ZAnum=0;
uint8    ZA=0;
extern uint8    line[8];
extern int      Rmax,Lmax;
extern unsigned char    obstaclecount;
extern unsigned char    lastobstaclecount;
extern unsigned char    obstacleline;
extern float      Leftobstacle,Rightobstacle;
extern unsigned char    Leftobstacleline,Rightobstacleline;
extern float      rskp100[5],rskp105[5],rskp110[5],rskp115[5],rskp120[5];
extern float      lskp100[5],lskp105[5],lskp110[5],lskp115[5],lskp120[5];
void PORTA_IRQHandler();
void DMA0_IRQHandler();
void PIT0_IRQHandler();
void LPTMR_IRQHandler();
void Hardware_init();
void menu();
void function();
void oledprintimage();
void motorset();
void Steerset();
void BUZZ();
```

```
void cameraset();
void steerRP70();
void steerLP70();
void Steertest();
void main(void)
{
    DisableInterrupts;
    Hardware_init();
    EnableInterrupts;
    key=0;
    typechoose=0;
    motor_on=0;
    exlength=200;
    obstacleflag=0;
    obstaclesignal=0;
    speederr=0;
    motorkp=mkp[6];
    speedbuff=0;
    length_centermeter=0;
    length_meter=0;
    exspeed=speed[0];
    ENDFLAG=0;
    ENDCOUNT=0;
    Startline=0;

    typechoose=typekey();
    if(typechoose==0)
    {
        speed[0]=115;
        speed[1]=110;
        speed[2]=125;
        speed[3]=170;
        obstaclejiance1=0;
        obstaclejiance2=0;
        obstaclejiance3=0;
        ZAnum=0;
        functionchoose[2]=1;
        functionchoose[3]=1;
        functionchoose[5]=1;
    }
}
```

第十一届全国大学生智能汽车竞赛技术报告

```
else if(typechoose==1)
{
    obstaclejiance1=0;
    obstaclejiance2=0;
    obstaclejiance3=0;
    ZAnum=0;
    speed[0]=110;
    speed[1]=110;
    speed[2]=120;
    speed[3]=170;
    functionchoose[2]=1;
    functionchoose[6]=1;
}
else if(typechoose==2)
{
    obstaclejiance1=0;
    obstaclejiance2=0;
    obstaclejiance3=0;
    ZAnum=0;
    speed[0]=108;
    speed[1]=106;
    speed[2]=116;
    speed[3]=150;
    functionchoose[2]=1;
    functionchoose[6]=1;
}
else if(typechoose==3)
{
    obstaclejiance1=0;
    obstaclejiance2=0;
    obstaclejiance3=0;
    ZAnum=0;
    speed[0]=105;
    speed[1]=105;
    speed[2]=110;
    speed[3]=150;
    functionchoose[2]=1;
    functionchoose[6]=1;
}
```



```

menu();
if(functionchoose[2]==1)
{
    speedA=(speed[2]-speed[1])/3600;
    speedB=(speed[1]-speed[2])*14/360;
    speedC=(49*speed[2]-13*speed[1])/36;
}
DELAY_MS(500);
BEEP_ON;
DELAY_MS(200);
BEEP_OFF;

pit_init_ms(PIT0,1);
set_vector_handler(PIT0_VECTORn,PIT0_IRQHandler);
enable_irq(PIT0_IRQn);
lptmr_pulse_init(LPT0_ALT2,LPTMR0COUNT,LPT_Rising);
Timenow=0;
while(1)
{
    length=(float)length_centermeter*0.01+(float)length_meter;
    if(functionchoose[3]==1 || functionchoose[5]==1 || functionchoose[6]==1)
    {
        if(length_meter>=exlength || ENDCOUNT>=6 || dangerstop>=4)
            motor_on=0;
        else
            motor_on=1;
    }
    else
    {
        motor_on=1;
    }
    camera_get_img();
    img_extract(img, imgbuff, CAMERA_SIZE);
    cameradeal();
    if(functionchoose[5]==1)
    {
        if(top_line>=50 && roadtype!=12 && roadtype!=11 && roadtype!=4)dangerstop++;
    }
    if(functionchoose[2]==1)

```

第十一届全国大学生智能汽车竞赛技术报告

```
{
    if((roadtype==1)&& top_line==10 && Center[11]<45 && Center[11]>35 && Center[20]<45 &&
Center[20]>35 && Center[30]<45 && Center[30]>35 && Center[40]<45 && Center[40]>35)exspeed=speed[3];
    else if(obstacleflag==1)
    {
        exspeed=0;
    }
    else
    {
        if(roadtype==3)exspeed=speed[0];
        else if(roadtype==11 || roadtype==12 || roadtype==4)exspeed=speed[1];
        else
        {
            if(abs(kkk)<=10) exspeed=speed[2];
            else if(abs(kkk)>=70) exspeed=speed[1];
            else exspeed=(int16)(speedA*abs(kkk)*abs(kkk)+speedB*abs(kkk)+speedC);
        }
    }
}

if(functionchoose[6]==1)
{
    if(Timenow>=QPjiance)
    {
        for(int ii=58;ii>=45;ii--)
        {
            if(Wide[ii]<=25)
                Startline++;
        }
        if(Startline>=2)
            ENDFLAG=1;
    }
}

if(functionchoose[7]==1)
{
    if(ZA<ZAnum)
    {
        if(obstaclecount>=2 && ((ZA==0 && length>=obstaclejiance1) || (ZA==1 && length>=obstaclejiance2)
|| (ZA==2 && length>=obstaclejiance3)));
        {
            obstacleflag=1;
        }
    }
}
```

```

    ZA++;
    obstaclecount=0;
    obstaclekaishi=length;
}
}
if(obstaclecount==1 && lastobstaclecount==1)obstaclecount=0;

if(obstacleflag==1)
{
    if((length-obstaclekaishi)>=obstacleyanchi)obstacleflag=0;
}
if(obstacleflag==1)BEEP_ON;
else BEEP_OFF;

}

LED_Print_Float(20,1,length,2);
LED_Print_int(20,3,speed_count);
LED_Print_Float(20,5,Timenow,1);
if(length>=obstaclejiance1)    LCD_P6x8Str(80,1,"ZA_ON");
    else    LCD_P6x8Str(80,1,"ZA_OFF");
if(Timenow>=QPjiance)    LCD_P6x8Str(80,3,"QP_ON");
    else    LCD_P6x8Str(80,3,"QP_OFF");
if(length>=rampjiance)    LCD_P6x8Str(80,5,"PD_ON");
    else    LCD_P6x8Str(80,5,"PD_OFF");

if(functionchoose[0]==1)
{
    vcan_ware[0]=speed_count;
    vcan_ware[1]=exspeed;
    vcan_sendware(vcan_ware,2);
}
if(functionchoose[1]==1)
{
    uart_putchar(UART0,0xFF);
    uart_putchar(UART0,abs(kkk));
    uart_putchar(UART0,roadtype);
    uart_putchar(UART0,speed_count);
    uart_putchar(UART0,0xFF);
}
}

```

第十一届全国大学生智能汽车竞赛技术报告

```
}
```

```
void PORTA_IRQHandler()
```

```
{
```

```
    uint8  n;
```

```
    uint32 flag;
```

```
    while(!PORTA_ISFR);
```

```
    flag = PORTA_ISFR;
```

```
    PORTA_ISFR  = ~0;
```

```
    n = 29;
```

```
    if(flag & (1 << n))
```

```
    {
```

```
        camera_vsync();
```

```
    }
```

```
}
```

```
void DMA0_IRQHandler()
```

```
{
```

```
    camera_dma();
```

```
}
```