

第二届“飞思卡尔”杯 全国大学生智能汽车邀请赛

技 术 报 告



学 校： 上海交通大学

队伍名称： 速度之星队

参赛队员： 方兴

郭思明

马明煌

带队教师： 王 冰

摘要

当今世界各国的大城市无不存在着交通拥挤问题，以上海为例，外环线以内大部分车辆平均时速在 20 公里以下。在现有道路有限的情况下，发展智能交通系统（ITS）成为提高交通资源利用率和交通安全的一种有效手段。智能车作为 ITS 的组成部分，能够靠自己的智能在道路上行驶。本文以第二届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和开发流程。该比赛采用组委会规定的标准车模，以 Freescale 半导体公司生产的 16 位单片机 MC9S12DG128 为核心控制器，在 CodeWarrior IDE 开发环境中进行软件开发，要求赛车在未知道路上完成快速寻线。整个系统涉及车模机械结构调整、传感器电路设计及信号处理、控制算法和策略优化等多个方面。为了提高智能赛车的行驶速度和可靠性，对比了不同方案（如摄像头与光电管检测方案）的优缺点，并结合 Labview 仿真平台进行了大量底层和上层测试，最终确定了现有的系统结构和各项控制参数。它采用摄像头对赛道进行检测，通过边缘提取获得黑线位置，用 PID 方式对舵机进行反馈控制。同时通过速度传感器获取当前速度，采用优化后的 Bang-Bang 控制实现速度闭环，根据预判信息和记忆信息对速度进行合理分配。实验结果表明，目前系统在机械结构、驱动电机控制、赛道信息提取和利用上都比第一届比赛有所提高。

关键词：智能车，快速寻线，摄像头，速度传感器，PID，Bang-Bang

ABSTRACT

Nowadays hardly a metropolis in the world does not suffers traffic jam, take Shanghai for example, in the outer ring road most vehicles' average speed are below 20km/h. With the current road limited, it is an effective method to improve the utilization of the traffic resources and the traffic safety to develop the Intelligent Transportation System (ITS). As the composition of the ITS, intelligent vehicle is able to drive on the road with its own intelligence. In the background of the 2nd National Intelligent Car Contest for College Students, this article introduces the soft hardware structures and the development flow of the vehicle control system. This contest adopting the standard model car prescribed by the contest organization committee, using the 16-bit MCU MC9S12DG128 produced by Freescale Semiconductor Company as the core controller, developing under the CodeWarrior IDE, requires the car track the line fast on the road. The whole system includes the aspects of the mechanism structure adjustment, the sensor circuit design and signal process, control algorithm and strategy optimization etc. In order to increase the speed and the reliability of the car, the advantage and disadvantage of the different schemes (such as the camera and photoelectric cell scheme) are compared, and a great number of the bottom layer and the upper layer tests are carried on combined with the Labview simulation platform. At last, the current system structure and each control parameters are determined. It captures the road information through a camera, then abstracts the black line position by edge-detection method. After that, PID feedback control is used on the steering. At the same time, the system obtains the current speed using a speed sensor, so that it can realize the feedback control of the speed by an optimized Bang-Bang control method. According to the pre-judge inform and the memorized inform, it allocates the speed properly. The result of the experiment demonstrates that the current system has superior features in the mechanism structure, the driving motor control, the road information extraction and utilization than last year.

Key words: intelligent vehicle, fast line track, camera, speed sensor, PID, Bang-Bang

目录

摘要.....	2
目录.....	4
第一章 绪论.....	6
1.1 背景介绍.....	6
1.2 赛车研究内容.....	7
1.2.1 赛道信息识别.....	7
1.2.2 车体控制.....	8
1.3 比赛规则.....	9
第二章 赛车整体结构设计.....	12
2.1 赛车主要技术参数.....	12
2.2 赛车硬件设计概述.....	12
2.3 赛车软件设计概述.....	13
第三章 赛车机械结构介绍和调整.....	15
3.1 前轮调整和舵机安装.....	15
3.1.1 前轮调整.....	15
3.1.2 舵机安装.....	17
3.2 后轮差速调整.....	17
3.3 车体重心调整.....	18
3.4 其他部分调整.....	19
第四章、赛车系统硬件设计.....	20
4.1、电路板设计.....	20
4.1.1 驱动电路板.....	20
4.1.2 核心电路板.....	21
4.2 电机.....	23
4.2.1 33886 使用概述.....	23
4.2.2 反向制动介绍及其使用.....	27
4.3 速度传感器.....	28
4.4 加速度传感器.....	32
4.4.1 工作方式:.....	32
4.4.2 干扰因素.....	33
4.4.3 安装方式分析.....	33
4.4.4 加速度传感器测试.....	34
4.5 计时器.....	36
第五章 赛车系统软件设计.....	40
5.1 s12 系统片内资源.....	40

5.2	系统初始化.....	41
5.2.1	摄像头初始化.....	41
5.2.2	PWM初始化.....	41
5.2.3	速度传感器初始化.....	42
5.2.4	串口初始化.....	42
5.2.5	超频设置.....	43
5.2.6	参数设置模块初始化.....	43
5.3	视频采集和黑线提取.....	44
5.3.1	视频信号时序.....	44
5.3.2	视频信号采集.....	45
5.3.3	黑线位置提取.....	46
5.4	速度闭环控制.....	47
5.4.1	速度控制介绍.....	47
5.4.2	实验依据.....	48
5.5	EEPROM.....	49
第六章 系统开发与调试.....		51
6.1	软件调试平台.....	51
6.1.1	Codewarrior IDE 功能介绍.....	51
6.1.2	Codewarrior IDE 基本使用方法.....	52
6.2	辅助调试工具.....	56
6.2.1	无线通讯模块.....	56
6.2.2	数码开关.....	62
6.3	仿真平台LABVIEW.....	64
6.3.1	虚拟仪器Labview和串口通讯.....	64
6.3.2	Labview测试程序.....	65
6.3.3	智能车仿真平台.....	67
第七章 结论.....		71
谢辞.....		72
参考文献.....		73
附录.....		74

第一章 绪论

现在半导体在汽车中的应用原来越普及,汽车的电子化已成为行业发展的必然趋势。它包括了汽车电子控制装置,即通过电子装置控制汽车发动机、底盘、车身、制动防抱死及动力转向系统等,到车载汽车电子装置,即汽车信息娱乐系统、导航系统、汽车音响及车载通信系统等等,几乎涵盖了汽车的所有系统。汽车电子的迅猛发展必将满足人们逐步增长的对于安全、节能、环保以及智能化和信息化的需求。

作为全球最大的汽车电子半导体供应商,飞思卡尔一直致力于为汽车电子系统提供全范围应用的单片机、模拟器件和传感器等器件产品和解决方案。飞思卡尔在汽车电子的半导体器件市场拥有领先的地位并不断赢得客户的认可和信任。其中在 8 位、16 位及 32 位汽车微控制器的市场占有率居于全球第一。飞思卡尔的 S12 是一个非常成功的芯片系列,在全球以及中国范围内被广泛应用于各种汽车电子应用中。例如引擎管理、安全气囊、车身电子、汽车网络和资讯娱乐等。

1.1 背景介绍

受教育部高等教育司委托(教高司函[2005]201 号文),高等学校自动化专业教学指导分委员会主办的“飞思卡尔”杯全国大学生智能汽车邀请赛是在飞思卡尔公司资助下以HCS12单片机为主控微控制器芯片的模型车体的基础上进行设计,制作出具有自主道路识别能力的智能汽车,是教育部主办的全国大学生五大竞赛之一。

全国大学生智能汽车竞赛与已举办的全国数学建模、电子设计、机械设计、结构设计等。

4 大专业竞赛不同,是以迅猛发展的汽车电子为背景,涵盖了控制、模式识别、传感技术、电子、电气、计算机、机械等多个学科交叉的科技创意性比赛。第一届“飞思卡尔”杯全国大学生智能汽车邀请赛于2006 年8 月在清华大学举行。来自全国共57 所大学的112 支参赛队的同学们在9 个月的时间里,历经报名、资格评审、赛车制作和调试,以及技术报告的评审等各个阶段的挑战,共有 25 支参赛队参加了决赛,清华大学2 队、上海交大速度之星队及清华大学1 队

分列成绩前三名。。

今年已是第二届比赛，在本次比赛中，参赛选手仍须使用大赛组委会统一提供的竞赛车模，以 Freescale 公司生产的 16 位微控制器 MC9S12DG128B 作为核心控制单元，自主构思控制方案及系统设计，包括传感器信号采集处理、控制算法及执行、动力电机驱动、转向舵机控制等，最终实现一套能够自主识别路线，并且可以实时输出车体状态的智能车控制软硬件系统。各参赛队完成智能车工程制作及调试后，于指定日期与地点参加比赛。参赛队伍之名次以赛车现场成功完成赛道比赛时间为主，技术方案及工程制作质量为辅来决定。但与去年不同的是，今年将先分五个赛区进行预赛选拔，然后再进行决赛，决赛地点定于我校。同时，今年的赛道还将增加坡道，难度有所加大。本文所述的内容即是为本届比赛而准备的技术方案。

1.2 赛车研究内容

按照以往各队比赛经验，识别赛道信息以及对赛车速度、转向角进行优化控制，是确保赛车在不冲出赛道的前提下尽可能快速行驶的关键。在这些方面，研究了一些已有的设计方案，并阅读了各种参考文献，总结如下：

1.2.1 赛道信息识别

赛道信息的识别是车辆控制的基础，为了提高行驶稳定性，很多设计都把重点放在了使传感器感知更远的距离上。现有赛道信息检测方案总体上有两大类：光电传感器方案和摄像头方案。前者电路设计简单、信息检测频率高，但检测范围、精度有限且能耗较大；后者获取的赛道信息丰富，但电路设计和软件处理较复杂，且信息更新速度较慢。关于光电管方案，采用“线型检测阵列”的方法，以4个发光二极管和1个光敏二极管构成一个检测点，将多个检测点成直线排列。共排成两排，一排安置在机器人底盘前端，一排安置在后端。然后采用巡回检测方式来寻找白色导引线。“线型检测阵列”寻线装置，采用了大规模可编程器件来实现对光电管检测信号的处理、分析，并实现对步进电机控制脉冲的产生。采用可编程器件，大大减少了系统体积，也方便了检测信号的处理和电机控制。虽然采用光电传感器构成“线型检测阵列”的方案简单易行，但是将其应用于智

能车竞赛中仍有一定局限性。例如，它对赛道的检测精度低，即使采用多个光电管，通常也只能确定10-20 个状态，这对赛车转向的平滑控制是非常不利的。此外，线型检测阵列的探测距离较短，如果简单地将检测阵列元件向车模前方伸出安装，又将增加舵机的转向负载，另外光电管的耗电量也非常显著。

基于前述对“线型检测阵列”寻线能力局限性的考虑，选择采用摄像头作为寻线传感器。一方面，摄像头所能检测的赛道信息远多于“线型检测阵列”所能检测的信息，有利于区分各种道路类型；另一方面，摄像头检测范围调整灵活，可以提供足够远的预判距离。实际上，通过“超频”和提高代码效率，并选择合适的图像处理算法，使用比赛规定的单片机完全可以对低线数黑白摄像头的视频信号进行采样和处理，有效识别出导引线的位置和相关几何信息。

1.2.2 车体控制

智能车应根据前方道路类型和当前车体位置误差及速度，对舵机转角和驱动电机PWM波的占空比进行及时调整。关于这方面的控制方案有PID 控制、Bang-Bang 控制和模糊控制等。去年我校CyberSmart 参赛队的技术报告中对舵机PID 控制进行了深入研究，文中列出了不同P、D 参数下赛车的平均速度和调整程度，并指出当P 参数适中、D 参数较小时，性能最佳。此外，该文还介绍了S12 单片机模糊控制指令的应用。虽然理论上模糊控制在非线性场合存在优势，但实际测试表明PD 控制已能满足需求，所以舵机控制仍采用PD 或者P控制。电机速度闭环控制是实现速度合理分配的基本前提，去年大部分队伍使用的调速方案是PID 控制，我校速度之星队使用的是Bang-Bang 控制。前者速度稳态误差小但响应较慢，后者响应迅速但稳态误差较大。针对这一特点，将PID 和Bang-Bang 控制结合可以得到改进的Bang-Bang 控制。

今年，较之去年有了一些新的特色。

(1) 机械安装

把摄像头支架安装在车中间，并且把电路板均匀分布在车体上，这样可以使小车重心放低，小车在转弯时会更稳定。

电池横向安装在车体中间，同样起到稳定重心的作用。

(2) 驱动电路

用单片机产生的 PWM 脉宽调制信号控制 33886 驱动电机，具有调速简单、可靠的特点。把 33886 的多个输出端并联在一起，可以增大驱动电流，提升小车的加减速性能。

而在驱动电路中加入反向制动电路来控制电机，可以极大地提高小车的减速性能。

(3) 控制策略

控制策略方面，使用经典控制方法里的 PID 控制。由于 PID 算法具有对模型参数要求不高、参数整定方便的优点，比较适合用来做小车寻迹。

在 PID 算法的基础上，整合加入模糊控制算法，有利于对小车系统的非线性特性因素的控制。

1.3 比赛规则

参赛选手须使用大赛组委会统一提供的竞赛车模，采用飞思卡尔 16 位微控制器 MC9S12DG128 作为核心控制单元，自主构思控制方案及系统设计，包括传感器信号采集处理、控制算法及执行、动力电机驱动、转向舵机控制等，完成智能车工程制作及调试，于指定日期与地点参加场地比赛。参赛队伍之名次（成绩）由赛车现场成功完成赛道比赛时间为主，技术方案及制作工程质量评分为辅来决定。组委会制定如下竞赛规则，在实际可操作性基础上力求公正与公平参与。组委会将邀请独立公证人监督现场赛事及评判过程。

比赛过程中，如果赛车碰到赛道两边的立柱并使之倾倒或移动，裁判员将判为赛车冲出跑道。赛车前两次冲出跑道，选手可以申请恢复比赛，即将冲出跑到赛车重新放置在裁判指定的赛车冲出跑道的位置，恢复比赛。整个恢复比赛过程中计时不间断。选手也可以在赛车冲出跑道后放弃比赛。

比赛过程中如果出现有如下一种情况，判为比赛失败：

- 1) 裁判点名后，2 分钟之内，参赛队没有能够进入比赛场地并做好比赛准备；
- 2) 比赛开始后，赛车在 30 秒之内没有离开出发区；
- 3) 赛车在离开出发区之后 10 分钟之内没有跑完两圈；
- 4) 赛车冲出跑道的次数超过两次；

- 5) 比赛开始后未经裁判允许, 选手接触赛车;
- 6) 决赛前, 赛车没有通过技术检验。

如果比赛失败, 则不计成绩。

比赛中禁止:

- 1) 不允许在赛道周围安装辅助照明设备及其它辅助传感器等;
- 2) 选手进入赛场后, 不允许进行任何硬件和软件的修改;
- 3) 比赛场地内, 除了裁判与 1 名队员之外, 不允许任何其他人员进入场地;
- 4) 不允许其它影响赛车运动的行为。

对于智能竞赛车模

- 1) 禁止改动车底盘结构、轮距、轮径及轮胎;
- 2) 禁止改动驱动电机的型号及传动比;
- 3) 禁止改造滚珠轴承;
- 4) 禁止改动舵机;
- 5) 禁止改动驱动电机以及电池, 车模主要前进动力来源于车模本身直流电机及电池;
- 6) 为了车模的行驶可以安装电路、传感器等, 允许在底盘上打孔或安装辅助支架等。

电路器件及控制驱动电路限制

- 1) 核心控制模块可以采用组委会提供的 HCS12 模块, 也可以采用 MC9SDG128 自制控制电路板, 除了 DG128MCU 之外不得使用辅助处理器以及其它可编程器件;
- 2) 伺服电机数量不超过 3 个;
- 3) 传感器数量不超过 16 个 (红外传感器的每对发射与接受单元计为 1 个传感器, CCD 传感器记为 1 个传感器);
- 4) 直流电源使用大赛提供的电池;
- 5) 禁止使用 DC-DC 升压电路为驱动电机以及舵机提供动力;
- 6) 全部电容容量和不得超过 2000 微法; 电容最高充电电压不得超过 25 伏。

可以选择参数:

- 1) 开发软件可以选择 CodeWarrior 3.1, 也可以另行选择;

- 2) 开发调试硬件可以选择 BDM（清华大学制作）工具，也可以另行选择；
- 3) 电路所使用元器件（传感器、各种信号调理芯片、接口芯片、功率器件等）种类与数量都可以自行设计选择。

赛道基本参数（不包括拐弯点数目、位置以及整体布局）

- 1) 赛道路面用纸制作，跑道所占面积不大于 $5000\text{mm} \times 7000\text{mm}$ ，跑道宽度不小于 600mm ；
- 2) 跑道表面为白色，中心有连续黑线作为引导线，黑线宽 25mm ；
- 3) 跑道最小曲率半径不小于 500mm ；
- 4) 跑道可以交叉，交叉角为 90° ；
- 5) 赛道为二维水平平面；
- 6) 赛道有一个长为 1000mm 的出发区，如下图所示，计时起始点两边分别有一个长度 100mm 黑色计时起始线，赛车前端通过起始线作为比赛计时开始或者与结束时刻。

第二章 赛车整体结构设计

2.1 赛车主要技术参数

此次比赛选用的赛车车模采用 1/10 的 Matiz 仿真车模。赛车机械结构只使用竞赛提供车模的底盘部分及转向和驱动部分。控制采用前轮转向，后轮驱动方案。具体车模数据如表 2-1:

表 2-1 赛车主要技术参数

项目	参数
车模几何尺寸（长、宽、高）（毫米）	288、165、335
车模轴距/轮距（毫米）	195
车模平均电流（匀速行驶）（毫安）	500
电路电容总量（微法）	1622
传感器种类及个数	摄像头，1 个
新增加伺服电机个数	0
赛道信息检测空间精度（毫米）	1.8
赛道信息检测频率（次/秒）	50
除 MC9S12DG128 之外其它主要芯片	LM1881, MC33886
车模重量（带有电池）（千克）	1.3

2.2 赛车硬件设计概述

根据摄像头方案设计，赛车共包括六大模块：控制处理芯片 MC9S12DG128，图像采样模块，速度传感模块，舵机驱动模块，电机驱动模块和辅助调试模块。其中 S12 单片机是系统的核心部分。它负责接收赛道图像数据、

赛车速度等反馈信息，并对这些信息进行恰当的处理，形成合适的控制量来对舵机与驱动电机进行控制。

图像采样模块由S12 的AD 模块，外围芯片（LM1881）和电路，与摄像头组成。其功能是获取前方赛道的图像数据，以供S12 作进一步分析处理。

速度传感器模块由小型编码器和ect脉冲捕捉功能构成

舵机模块和驱动模块分别用于实现赛车转向和驱动。

辅助调试模块主要用于赛车系统的程序烧写，功能调试和测试，赛车状态监控，赛车系统参数和运行策略设置等方面。

本赛车系统的结构示意图如图2-1

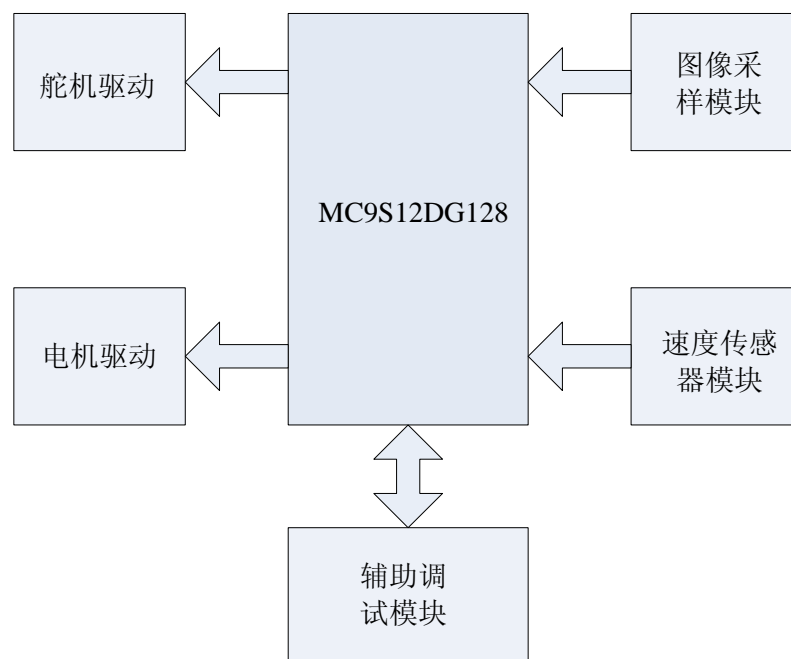


图 2-1 系统硬件结构

2.3 赛车软件设计概述

如果说系统硬件对于赛车来说是它的骨架和躯体，那么软件算法就是它的思想。有了健壮、灵敏的躯体还需要有聪明、智慧的大脑。所以软件系统对于赛车来说至关重要。首先，赛车系统通过图像采样模块获取前方赛道的图像数据，同时通过速度传感器模块实时获取赛车的速度。然后S12 利用边缘检测方法从图像数据中提取赛道黑线，求得赛车于黑线位置的偏差，接着采用PID 方法对舵机进行反馈控制。最终赛车根据检测到的速度，结合我们的速度控制策略，对赛车速

度不断进行恰当的控制调整，使赛车在符合比赛规则情况下沿赛道快速前进。设计赛车系统的软件结构如图2-2所示。

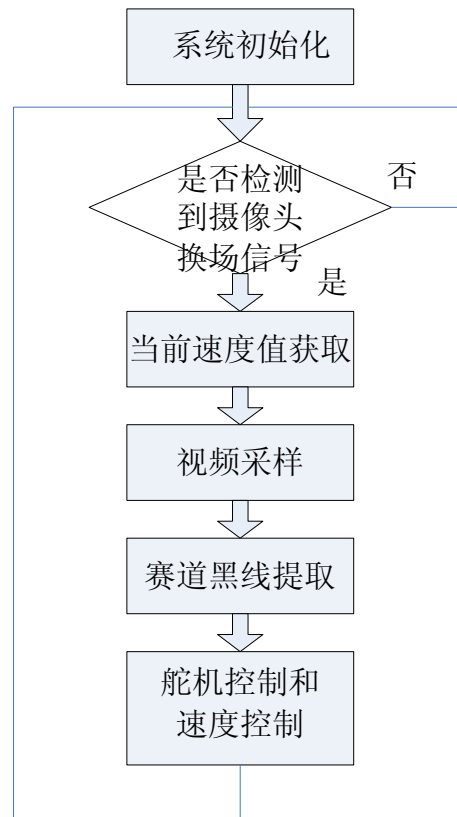


图 2-2 系统软件结构

只有将上述硬件、软件部分进行有效的融合，进行充分的实验、测试和标定工作，才能使赛车具有一个相对良好的整体驾驶性能。这也就是在接下来的各个分模块中需要做的工作。

第三章 赛车机械结构介绍和调整

3.1 前轮调整 and 舵机安装

3.1.1 前轮调整

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。为了尽可能降低转向舵机负载，对前轮的安装角度，即前轮定位进行了调整。前轮定位的作用是保障汽车直线行驶的稳定性，使转向轻便和减少轮胎的磨损。前轮是转向轮，它的安装位置由主销内倾、主销后倾、前轮外倾和前轮前束 4 个项目决定，反映了转向轮、主销和前轴等三者在车架上的位置关系

表 3-1 车模基本尺寸参数

基本参数	尺寸
轴距	200cm
前轮距	126cm
后轮距	135cm
车轮直径	54cm
车长	288mm
车宽	165m
传动比	18/76



图 3-1 车模示意图

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。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。

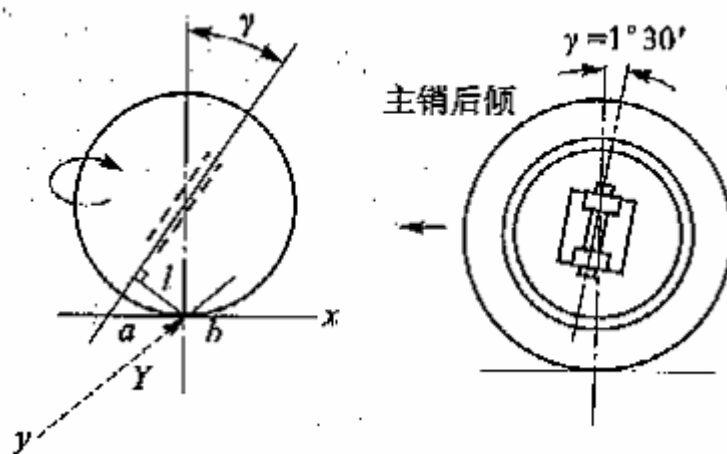


图 3-2 主销后倾纠正车轮偏转原理图

如图3-2所示，主销后倾（Caster）是指主销装在前轴，上端略向后倾斜的角度。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。

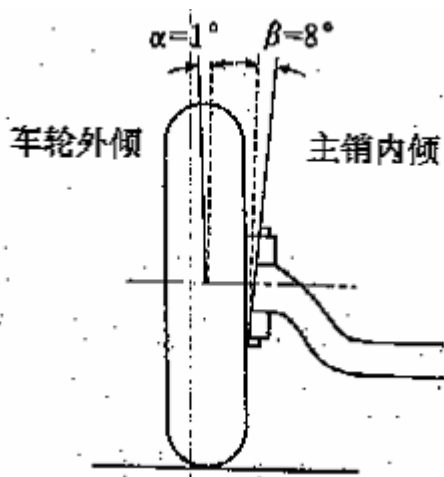


图3-3 前轮外倾角示意图

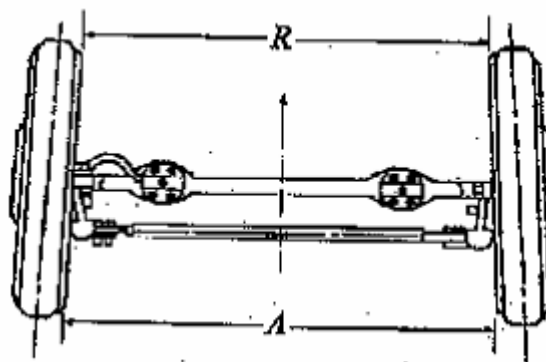


图3-4 前轮约束示意图

如图3-3，前轮外倾角（Camber）对赛车的转弯性能有直接影响，它的作用是提高前轮的转向安全性和转向操纵的轻便性。前轮外倾角俗称“外八字”，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外八字角度，这个角度约在 1° 左右。

如图3-4，所谓前束（Toe-out）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，其惯性力会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。经过反复试验，对Camber、Toe-out和 Caster 三个角度进行了适当调整，最后前轮调整的结果如表3-2 所示。

表 3-2 前轮角度调整结果

安装角度	调整结果
Camber	-1 度
Toe-out	1 度
Caster	垫片前 3 片后 1 片

3.1.2 舵机安装

舵机转向是整个控制系统中延迟较大的一个环节，为了减小此时间常数，通过改变舵机的安装位置，而并非改变舵机本身结构的方法可以提高舵机的响应速度。分析舵机控制转向轮转向的原理可以发现，在相同的舵机转向条件下，转向连杆在舵机一端的连接点离舵机轴心距离越远，转向轮转向变化越快。这相当于增大力臂长度，提高线速度。针对上述特性，改变了去年舵机的安装方式（如图3-5），将舵机安装在了相对靠后的位置（如图3-6）。这样安装的优点是：1）改变了舵机的力臂，使转向更灵敏2）舵机安装在了正中央，使左右转向基本一致 3）重心更靠后，减轻了舵机负载。

3.2 后轮差速调整

差速机构的作用是在车模转弯的时候，降低后轮与地面之间的滑动，并且还可以保证在轮胎抱死的情况下不会损害到电机。



图3-5 去年舵机安装方式



图3-6 今年舵机安装方式

当赛车在正常过弯时（假设无转向不足亦无转向过度），4 个轮子的转速皆不相同，依序为：外侧前轮>外侧后轮>内侧前轮>内侧后轮。此次所使用车模配备的是后轮差速机构。差速器的特性是：阻力越大的一侧，驱动齿轮的转速越低；而阻力越小的一侧，驱动齿轮的转速越高。以此次使用的后轮差速器为例，在过弯时，因外侧后轮轮胎所遇的阻力较小，轮速便较高；而内侧后轮轮胎所遇的阻力较大，轮速便较低。

差速器的调整中要注意滚珠轮盘间的间隙，过松过紧都会使差速器性能降低，转弯时阻力小的车轮会打滑，从而影响车模的过弯性能。好的差速机构，应该在电机不转的情况下，右轮向前转过的角度与左轮向后转过的角度近似相等，不会有迟滞或者过转动的情况发生。

3.3 车体重心调整

车体重心位置对赛车加减速性能、转向性能和稳定性都有较大影响。重心调整主要包括重心高度和前后位置的调整。理论上，赛车重心越低稳定性越好。因此除了摄像头装得稍高以外，其他各个部件的安装高度都很低。除此之外，车辆重心前后方向的调整，对赛车行驶性能也有很大的影响。根据车辆运动学理论，车身重心前移，会增加转向，但降低转向的灵敏度（因为大部分重量压在前轮，转向负载增大），同时降低后轮的抓地力；重心后移，会减少转向，但增大转向灵敏度，后轮抓地力也会增加。因此，确定合适的车体重心，让车模更加适应比赛赛道是很关键的。经过实际调试，将摄像头安装在车体中部，使车体重心位于中间偏后一些的地方，可以使得赛车转向灵敏度、稳定性以及加减速性能都达到

较佳水平。

3.4 其他部分调整

除了以上部分的调整外，还对主悬架弹簧松紧、底盘高度和齿轮传动机构进行了适当调整。通过增加避震弹簧的刚性、降低底盘高度、调整齿轮间隙，改善了赛车的行驶表现。

第四章、赛车系统硬件设计

4.1、电路板设计

系统硬件电路由驱动电路板和核心控制板两部分组成，本章将先对这两个电路进行总体介绍，然后分模块详细说明各部分的功能。

4.1.1 驱动电路板

驱动电路板（如图4-1 所示）安装在车体后部，安装时主要考虑的是稳定性。它主要负责系统各部分电压的分配以及电机的驱动。由于这部分电路功耗较大，单独设计一块电路板后可使散热性能更好。

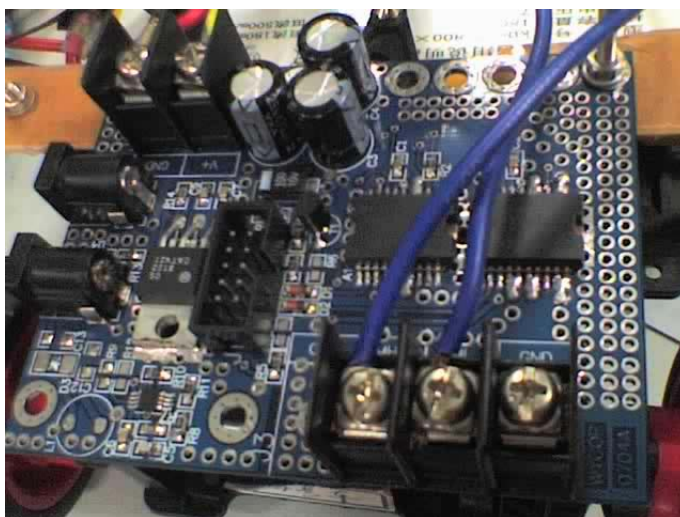


图 4-1 驱动电路板安装位置

驱动电路板的原理图见附录A 所示，其各个端口的分配情况如下：

- 1) J1的1-3口提供VCC；4口为使能信号，控制电机运转；6口状态位查询，检查电机运转情况；9-10口提供PWM，控制电机运转方向和运转速度。
- 2) J2, J5 电池电源输入，J5为并联备用。
- 3) J3、J4 通过比较2路PWM输入控制电机正反转。
- 4) J7 给摄像头提供工作电源，由于选用的摄像头额定工作电压为8V，在电池电压高于7.2V 的范围内能可靠工作。
- 5) A1、A2 33886运行电机，2路并联输入，4路并联输出，可以提高输入和输出的电流值，提高系统稳定性。

目前电路板上将两片33886并联，PWM信号一路直接输入到A1，一路经过反向后输入到A2]这样如果PWM波的占空比高于50%时，电机朝一个方向转；占空比低于50%时，电机朝另一个方向转。通过这种方式，可以在程序中实现反向制动，而这对于赛车在直道上提高速度是有帮助的。

4.1.2 核心电路板

核心控制板（如图4-2所示）负责视频采集、获取速度并输出舵机和驱动电机的控制信号。电路板上包括了S12 单片机工作所必需的稳压模块、时钟模块和复位模块，同时还包括了视频信号分离电路、BDM 调试接口、串口以及无线Zigbee接口等。

核心控制板的原理图见附录B，单片机各个端口的分配情况如表4-1 所示：

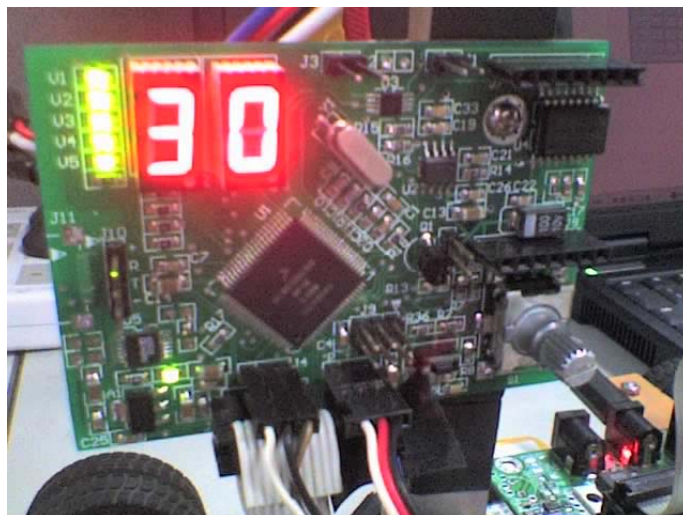


图4-2 控制电路板

表4-1单片机端口分配表

AD 模块	PAD0	摄像头视频信号
	PAD6	电池电压
	PAD7	摄像头电压
外部中断	IRQ	行同步信号
ECT 模块	PT0	速度传感器信号
	PT1	场同步信号
	PT2	奇偶场信号
PWM 模块	PWM0, PWM1	舵机，由 PWM1 输出
	PWM2, PWM3	33886（电机）
	PWM4, PWM5	LED4, LED5
串口通讯	RXD0	串口接收

模块	TXD0	串口发送
无线通讯模块	RX10	无线接收
	TX10	无线发送
IO	PA0—PA7	LED#2
	PB0—PB7	LED#1
	PJ6	33886 状态检测
	PJ7	33886 使能
	PM0	视频选测信号
	PT5-PT7	数码开关信号

电路板上所有的接口功能说明如下：

- 1) 摄像头视频信号接口J1、J2、J3 三个接口，J2、J3用来连接摄像头，另外一个可以外接电视机，以方便信号调试。
- 2) 视频同步分离模块U2 根据视频输入，分离出场同步信号、奇偶场信号等。
- 3) BDM 调试接口J9 按照清华的定义调整了针脚信号定义，以防止误插反时损坏电路。
- 4) MEMS加速度传感器模块 根据赛车加速度的不同，把某一特定方向的加速度转化为信号输入PAD3口。
- 5) 视频信号处理子模块J8 通过外接硬件积分电路，对视频信号进行处理，用以识别路线和控制。
- 6) 舵机控制接口J5 经过测试，在电池电压范围内舵机能可靠工作，所以没有降压而直接用电池电压给舵机供电。
- 7) 串行通讯接口J10 提供了TTL 电平的串口通讯接口，可以配合多种通讯模块与PC 通讯，通过该接口，在调试时可以方便地获取小车的实时运行参数。
- 8) 无线通讯接口J11 使用Zigbee无线传输各种数据，方便地获取小车的实时运行参数。
- 9) 速度传感器接口J4 通过光电编码器产生的脉冲获得赛车的运行速度，该信息可以用于速度闭环控制，保证小车能在不同路段上基本以指定的速度运行，而不受电池电压变化的影响。
- 10) 电池电源接口J12 连接车模供电电池组。
- 11) 运行电机接口J6 将两片33886并联使用驱动运行电机，尽量增加驱动电路的电流驱动能力。
- 12) 数码开关S1 通过左旋、右旋、确认三种方式调节QESA、QESB、QESP，用来

现场调整运行速度及控制策略等。

13) LD0稳压 提供3.3V电压。

14) 复位按钮S2 复位单片机系统。

15) 2位数码管DB1、DB2 通过控制PA0-7，PB0-7用来实时显示系统的工作状态。

16) 状态指示灯 用LED1-LED5显示系统的工作状态。

4.2 电机

4.2.1 33886 使用概述

33886 作为一个单片电路H-桥，是理想的功率分流直流马达和双向推力电磁铁控制器。它的集成电路包含内部逻辑控制,电荷泵,门控驱动,及低读选通(on) 金属-氧化物半导体场效应晶体管输出电路.33886 能够控制连续感应直流负载上升到5.0 安培，输出负载脉宽调制(PWM-ed)的频率可达10 千赫。

一个故障状态输出可以报告欠压,短路,过热的情况。两路独立输入控制两个半桥的推拉输出电路的输出。两个无效输入使H-桥产生三态输出(呈现高阻抗)。

33886制定的参数范围是 $-40^{\circ}\text{C} \leq T_A \leq 125^{\circ}\text{C}$ 、 $5.0\text{V} \leq V_{+} \leq 28\text{V}$ 。集成电路也可以工作在40V通过降低规定的定额值。集成电路能够在表面安装带散热装置电源组件。

33886 Simplified Application Diagram

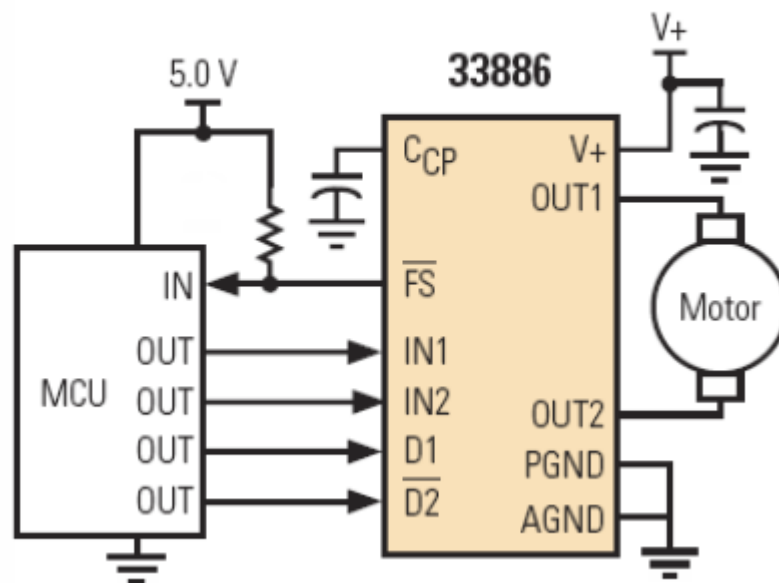


图4-3 33886简化应用图

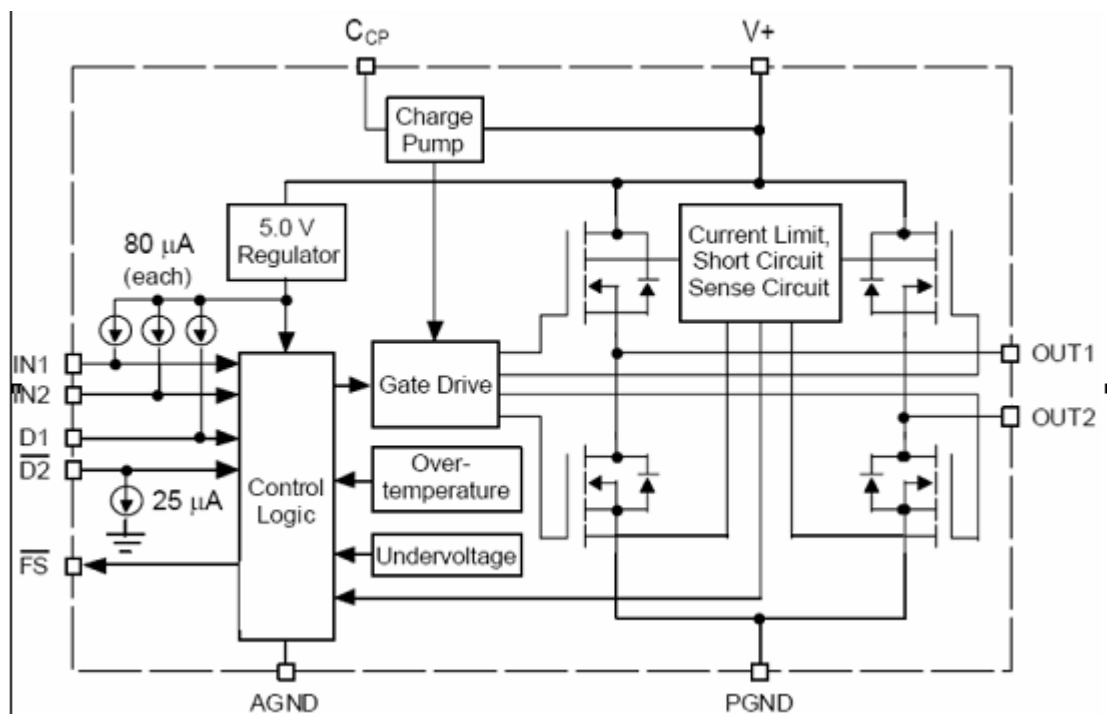


图4-4 33886工作原理图

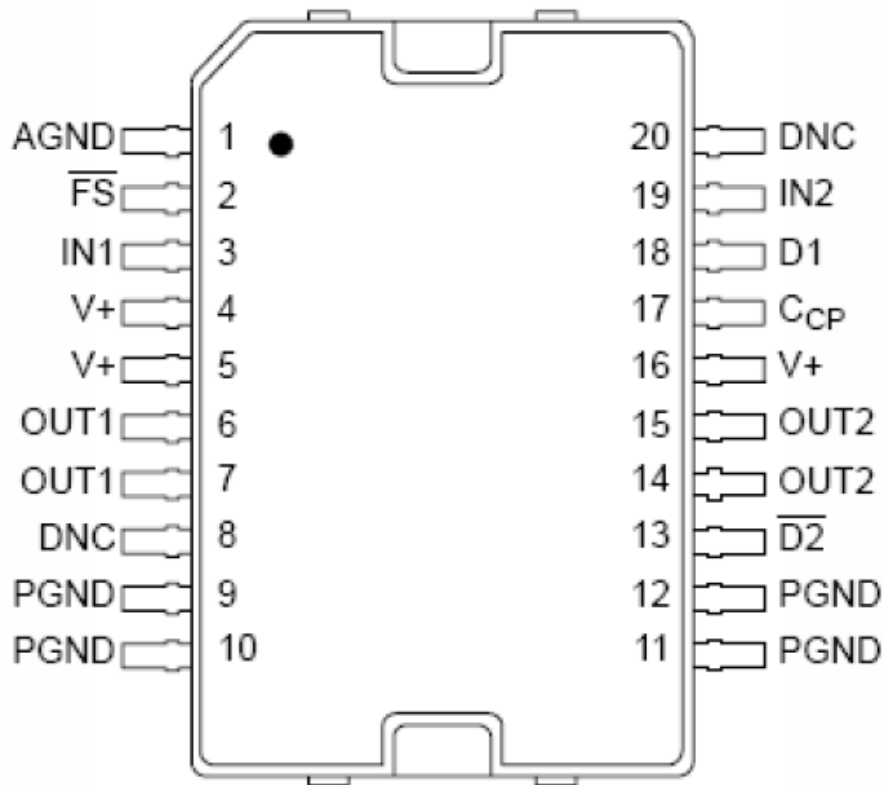


图4-5 33886芯片引脚定义

表4-2 引脚解释

终端功能描述

终端	终端名称	正式名称	定义
1	AGND	模拟接地	低电流模拟信号接地
2	FS	H桥故障状态	故障状态场效应管低电位有效，要求拉电阻上升到5v.
3	IN1	逻辑输入控制1	实际逻辑输入控制的1口
4,5,16	V+	电源供电	正电源连接.
6,7	OUT1	H桥输出1	H桥输出1
8,20	DNC	静止连接	在应用中不连接或接地. 他们仅在制造中用于测试模式终端。
9-12	PGND	电源接地	装置电流高功率接地.
13	D2	无效2	输入低电位有效用于使两个H桥输出同时三态无效. 当D2为逻辑低,输出都是三态。
14,15	OUT2	H桥输出2	H桥输出2
17	CCP	电荷泵电容器	外部充电电容器连接内部电荷泵电容器.
18	D1	无效1	输入低电位有效用于使两个H桥输出同时三态无效. 当D1为逻辑低,输出都是三态。
19	IN2	逻辑输入控制2	实际逻辑输入控制的2口

终端功能描述

PGND 和AGND

电源和模拟接地终端. 电源和模拟接地终端设备应该连接在一起,以非常低的阻抗连接.

V+

V+的终端电源输入装置. 所有V+的终端必须连接在一起, 在印刷电路板上用尽可能短的路径, 终端设备之间的阻抗尽可能低。V+终端有一个低压阈值.如果供电电压降低于V+的低压阈值, 输出功率将会变为三态状态, 同时故障状态标志位被置位, 故障状态终端电压转为逻辑低。当供电电压返回到高于阈值电压, 功率自动恢复正常运作,按照既定条件的输入故障状态标志位自动复位逻辑高。

故障状态(FS)

该终端是故障状态输出装置.该输出一个有效的低开漏极结构,要求拉升电阻上升到5.0V。

IN1, IN2, D1, D2

这些终端用输入控制终端来控制输出.这些终端都是带磁滞的5.0V的CMOS输入. 该IN1和IN2单独控制OUT1和OUT2.D1和D2是相辅相成的输入,用于三态禁用H-桥输出.当D1或D2被置位(D1=逻辑高或D2=逻辑低),在关闭状态, 输出OUT1和OUT2都是三态禁用;然而,其余的电路器件充分运作, 并且供应IQ(待命)目前已减少至几毫安.

OUT1 和OUT2

这些终端设备是带集成输出FET 的体二极管的H-桥的输出. 桥输出用

IN1,IN2,D1,和D2 输入控制.输出的有效电流限制在6.5A。输出也有过关闭(三态锁存),作为磁滞及短路锁存的保护.一个禁用定时器用于在每次输入时侦测电流是否高于有效电流限制, 以方便检测输出硬件短路。

CCP

电荷泵输出端. 一滤波电容器(最多33nF),可接驳CCP 中断和PGND. 该装置能在没有外部电容器时工作, CCP 电容有助于降低噪声,并允许器件工作与最高速率,时间和PWM 频率.

4.2.2 反向制动介绍及其使用

这次使用了两个 33886 并联的方式。工作方式见驱动电路板原理图（附录 A）
在同一个 pwm 输入时，一个是 pwm 波直接输入 33886，另外一个 pwm 波先通过一个三极管再输入 33886。这样可以得到一个取反的效果，即，例如输入 5v 电压将输出一个 0v 电压。这样，通过两片 33886，同一个 pwm 波会输出两个不同电压。通过对 pwm 波的控制可以使电机实现正反转。

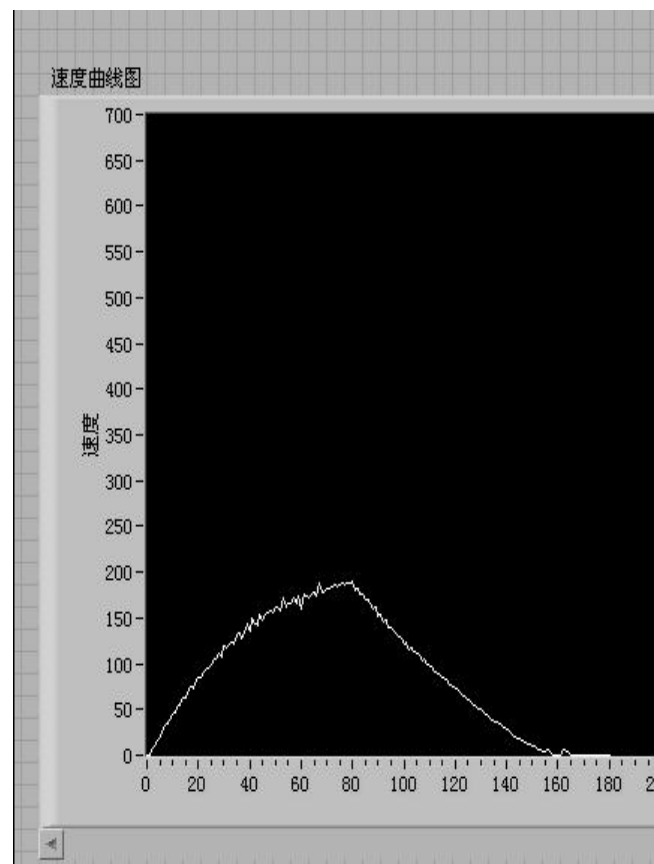


图 4-6 没加反向制动的最佳减速

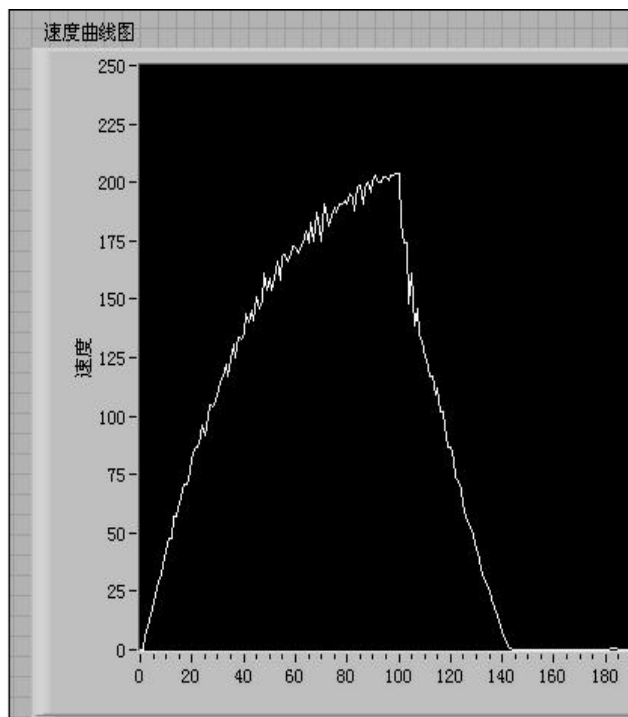


图 4-7 加反向制动的减速

可以看出，反向制动可以比非反向制动更好的减速性能。即加速度更快。

4.3 速度传感器

为了使得赛车能够平稳地沿着赛道运行，需要控制车速，使赛车在急转弯时速度不至过快而冲出赛道。通过控制驱动电机上的平均电压可以控制车速，但是如果开环控制电机转速，会受很多因素影响，例如电池电压、电机传动摩擦力、道路摩擦力和前轮转向角度等。这些因素会造成赛车运行不稳定。通过速度检测，对车模速度进行闭环反馈控制，即可消除上述各种因素的影响，使得车模运行得更稳定。此外，在记忆算法中为了记录道路信息，需要得到赛车运行距离，这也可以通过车速检测来实现。

车速检测的方式有很多种，例如用测速发电机、转角编码盘、反射式光电检测、透射式光电检测和霍尔传感器检测。经过对去年测速方案和其它学校方案的比较，本次设计中速度传感器采用的是OMRON公司生产的E6A2-CS100型光电编码器。它由5-12V的直流供电，安装方式如图3-12所示。速度传感器用螺钉固定在塑料片上，塑料片固定在后轮支架上，这样固定好之后，就有了较高的稳定性。速度传感器通过后轮轴上的齿轮与电机相连，车轮每转一圈，速度传感器转过

2. 75圈。



图 4-8 速度传感器安装位置

根据不同分辨率，有不同的速度传感器类型

种类				
本体				
输出相	电源电压	输出型号	分辨率(脉冲/旋转)	型号
A相	DC~12V	电压输出	10、20、60、100、200、300、360	E6A2-CS3E
			500	
	DC12~24V	集电极开路输出	10、20、60、100、200、300、360	E6A2-CS3C
			500	
A相、B相	DC5~12V	电压输出	100、200、360	E6A2-CW3E
			500	
	DC12~24V	集电极开路输出	100、200、360	E6A2-CW3C
			500	
A相、B相 Z相	DC5~12V	电压输出	100、200、360	E6A2-CWZ3E
			500	
	DC12~24V	集电极开路输出	100、200、360	E6A2-CWZ3C
			500	
	DC5~12V	电压输出	100、200、360	E6A2-CWZ5E
			500	
	DC12~24V	集电极开路输出	100、200、360	E6A2-CWZ5C
			500	

图 4-9 工作原理图

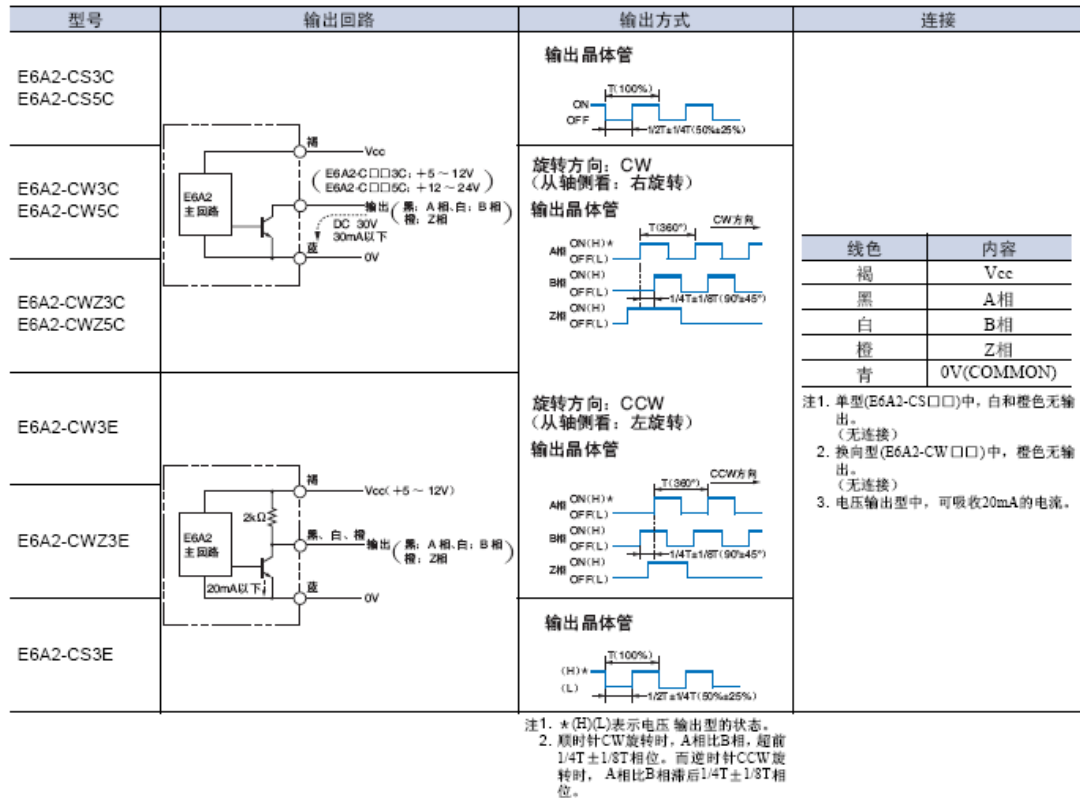


图4-10 速度传感器的型号

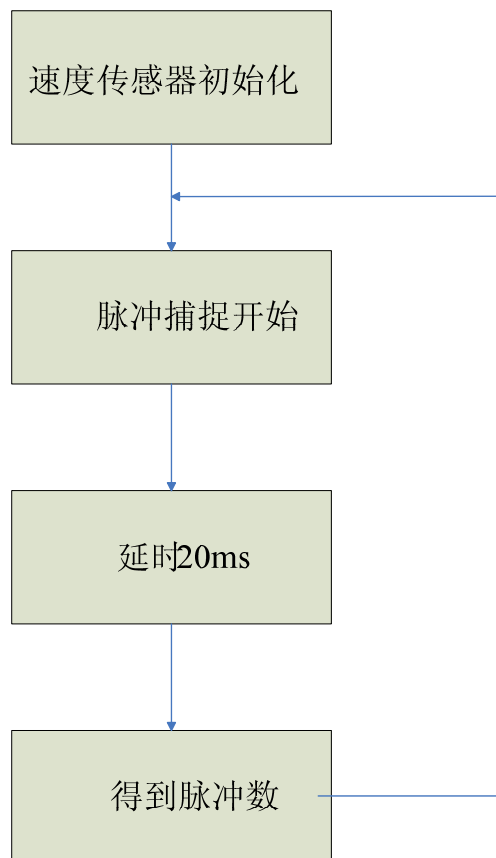


图4-11 速度传感器工作流程图

为了得到实际速度和所得脉冲数的关系。我们做了实验和理论上的分析

(1). 脉冲速度标定

将小车设定为匀速行驶，将速度传感器的数据通过无线每隔40 毫秒发送到电脑上。观察波形，通过设定不同的目标速度，记录每圈的时间，跑道长度约为21.5 米，根据实验所得数据，然后作出下图的曲线图4-12，发现在100个脉冲速度下，速度基本符合线性关系，说明匀速的速度控制效果比较不错。拟合出速度—脉冲曲线，可得周期为40mm 时脉冲为64.7 对应1m/s 的速度。

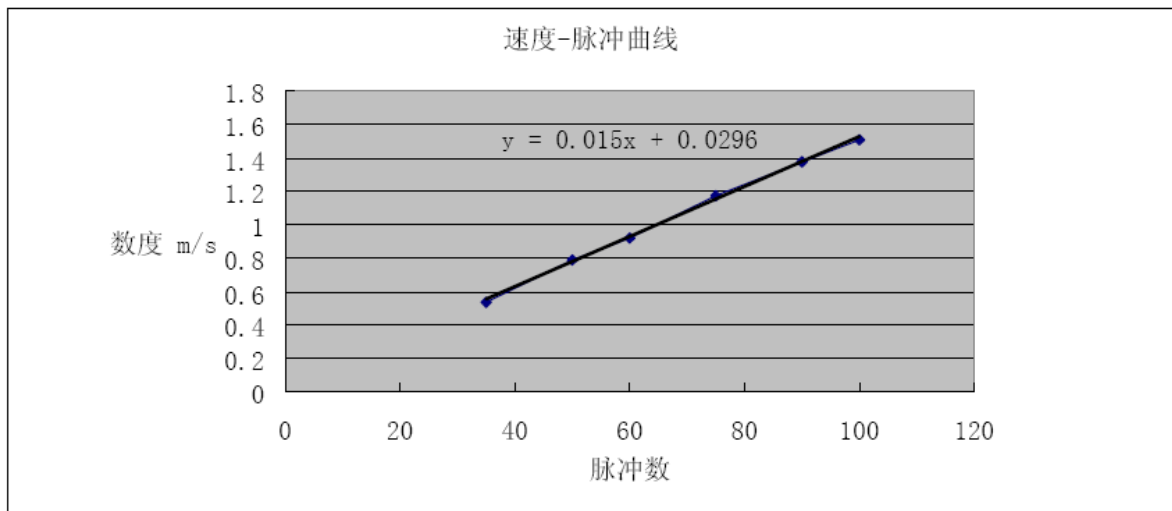


图4-12 速度脉冲对应曲线

通过校验，在小车走线稳定的状态下，每圈里程的记录与跑道的长度的差异均小于0.5米。说明标定的精度已满足目前的使用。

(2). 加速性能标定

反向制动试验中的速度曲线大部分图中的加速部分基本呈线性关系，测量出这些图中加速部分的数据得到下表：

表4-3

增加脉冲数	周期数	加速度脉冲数
130	17	7.647
120	17	7.059
110	15	7.333
130	18	7.222
210	29	7.241
225	31	7.258

200	27	7.407
215	28	7.679

求平均数得每个周期的速度增量为7.356，根据速度标定结果的在全电压下的小车加速度约为 2.84m/s^2 。

(3) 理论分析

首先设置几个变量：

P—速度传感器的精度（单圈内产生脉冲的个数），我们采用速度传感器精度为400， Δt — Δt 时间内，单片机捕获的来自速度传感器的脉冲数为N。

Φ —小车车轮的直径。具体车模而言， $\Phi=0.053\text{m}$ 。

$r=n_1/n_2$ — n_1 为套在车轮上的大齿轮的齿数， n_2 为套在速度传感器上的小齿轮的齿数。

用r记他们之比。对我们使用的三个速度传感器而言， $r=2.75$ 。

Pi—圆周率。一般取3.14。

经推导（过程略）可以得到下面的速度标定公式：

$$v = \frac{N}{P * R * \Delta t} * \pi * \phi$$

当P=100， $\Delta t=0.04\text{s}$ 时，上述公式演化为：

$$v = 0.0153 * N$$

代入N=66, 得到 $v=1.0\text{m/s}$ 。这与之前实验方法得到的结果几乎一致。

4.4 加速度传感器

加速度传感器使用的也是 freescale 公司出品的 MMA2260。

4.4.1 工作方式：

- MMA2260 的端口 4 负责输出会随加速度变化的电压值。
- 端口 8 为自检测（Self-Test）端口。
- 端口 5 为状态位（Status）。

- 端口 4 连接至 S12 的 PAD3，用 A/D 转换端口 3 来转换电压信号。

4.4.2 干扰因素

这块芯片可以测试一个方向上的加速度变化。所以采用了不同的安装方式，其间要考虑到比较多的干扰因素。

- 车体行进过程中的机械振动
- 速度控制不平滑对车体运动状态的影响
- 常规情况下的误差和正常时，加速度传感器产生的输出区别大小

4.4.3 安装方式分析

- 若安装于安装摄像头的立杆上，杆子在小车加减速的时候有比较明显的左右晃动，通过 Labview 观察发现干扰过大。
- 若安装于小车后部，由于电机自身的振动，对 MEMS 产生的影响会较之前更为明显。
- 经测试，如果测量水平 X 方向的加速度，宜将装有 MMA2260 的电路板装载在车前部。
- 若想要测量竖直 Z 方向上的加速度，板子只有采用立式安装的方式。
- X, Z 方向示意图见图 4-14

附图， 加速度传感器引脚定义

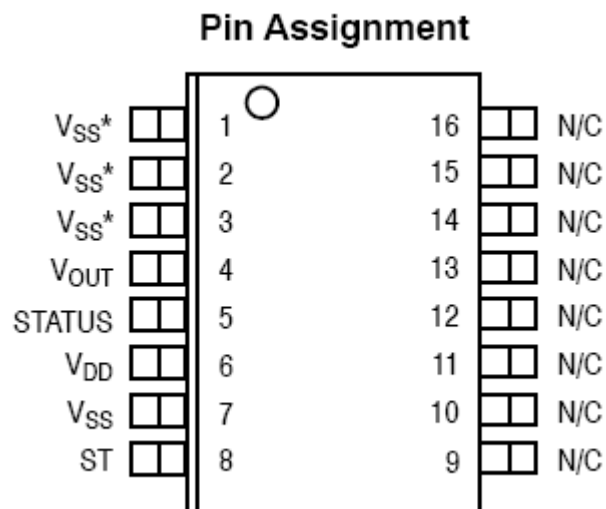


图4-13

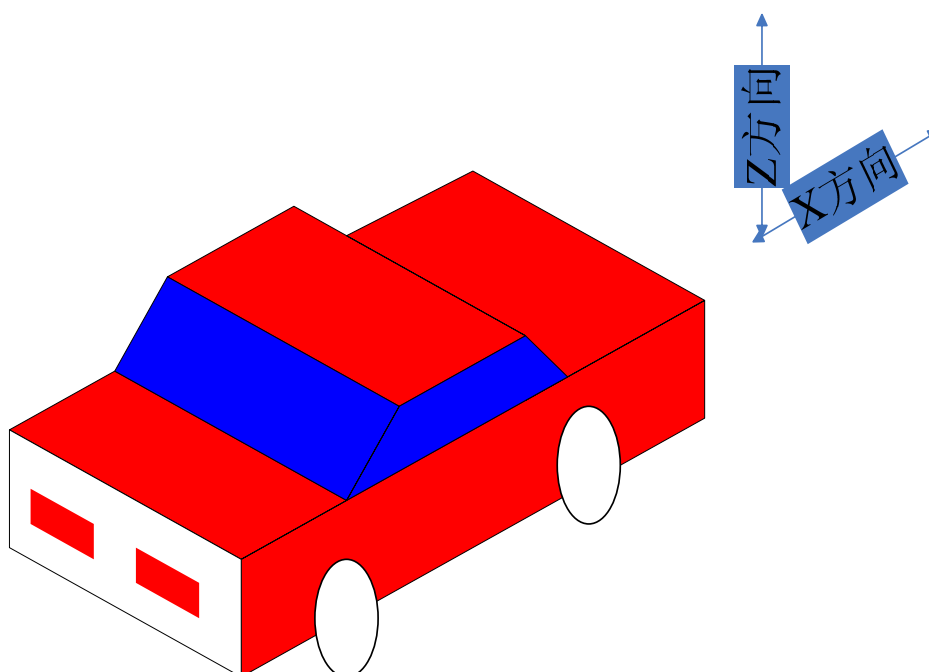


图4-14 X, Z方向示意图

4.4.4 加速度传感器测试

曲线图: (20ms)

当装载于静止小车上时

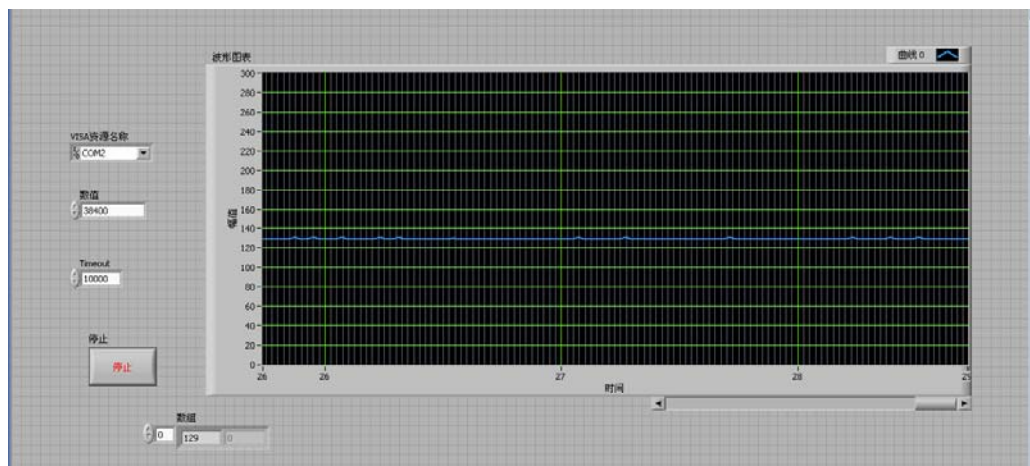


图 4-15

当装载于运动状态不变小车上时

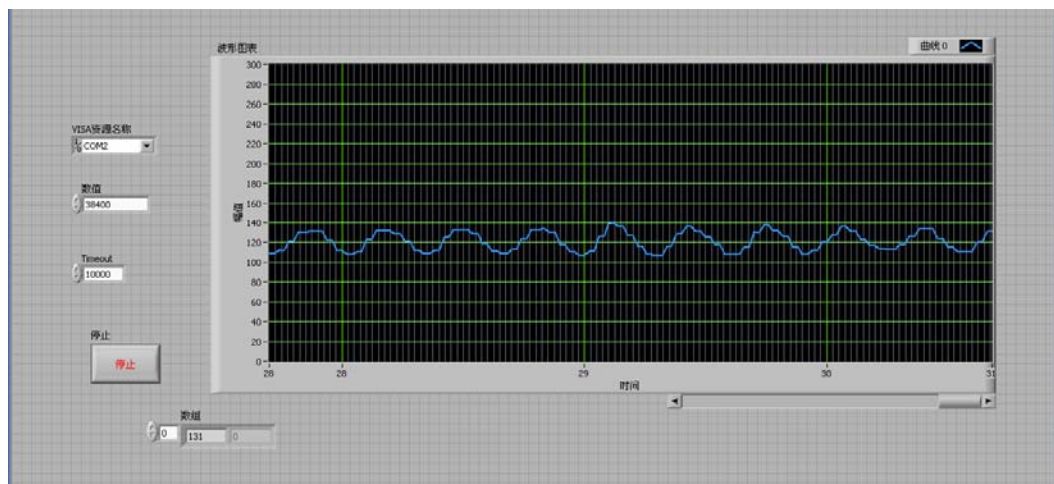


图 4-16

小车角度变化时

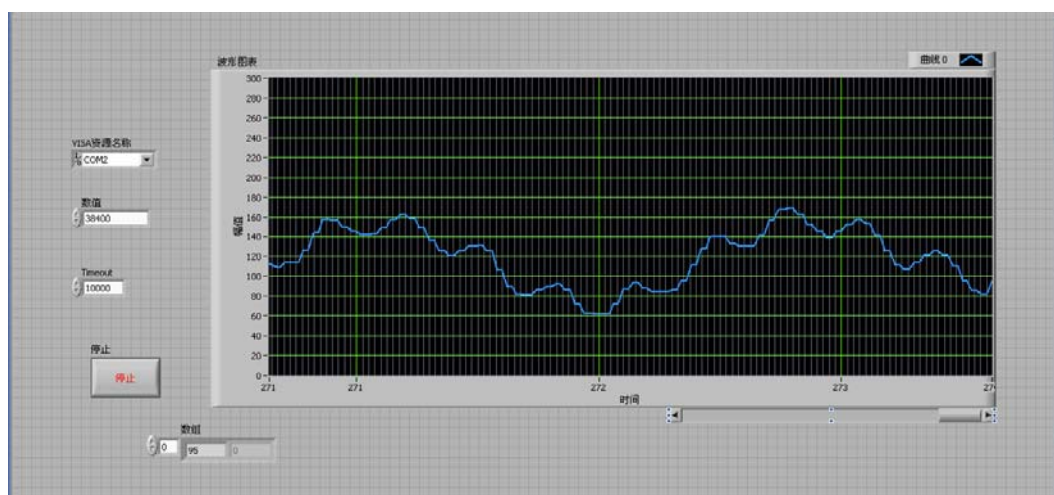


图 4-17

角度 0 到-15 变化

加速度+a 到-a 变化(100ms)

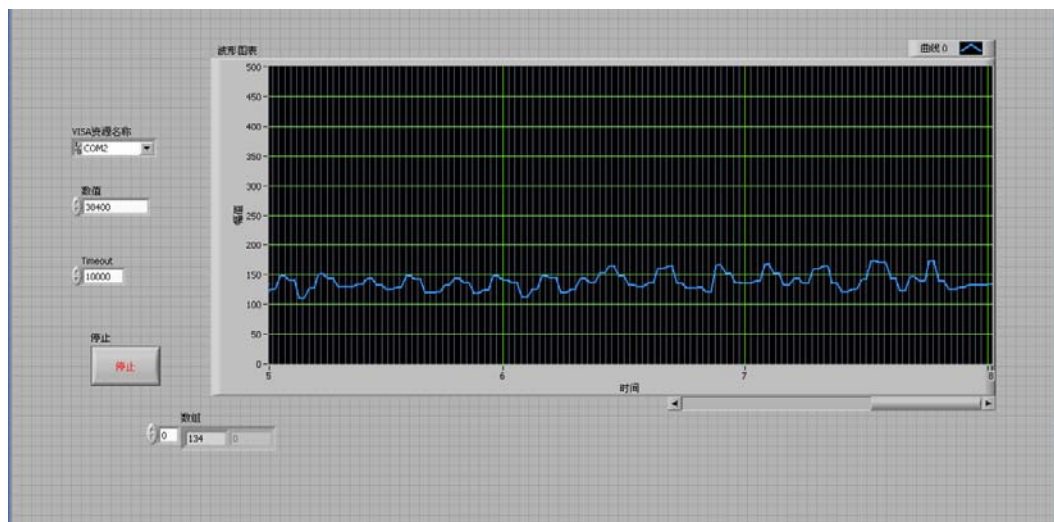


图 4-18

通过测试可以看出。对于车体角度的变化，加速度传感器的值都有一定的变化。不过同时也看到车子的振动对加速度传感器也会产生一些干扰。不过两者还是可以比较好的区分的（比较图 4-16，图 4-17）。

4.5 计时器

计时器芯片是 DP8573ARTC 芯片。引脚定义如图

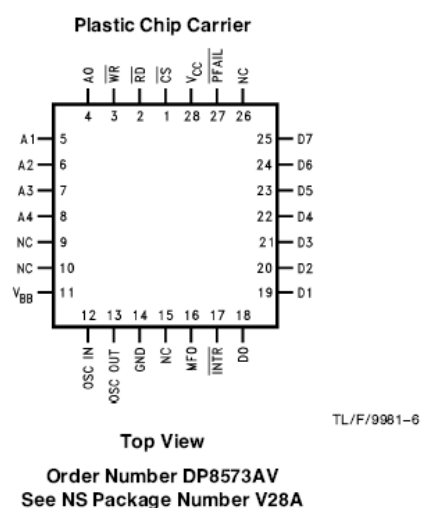
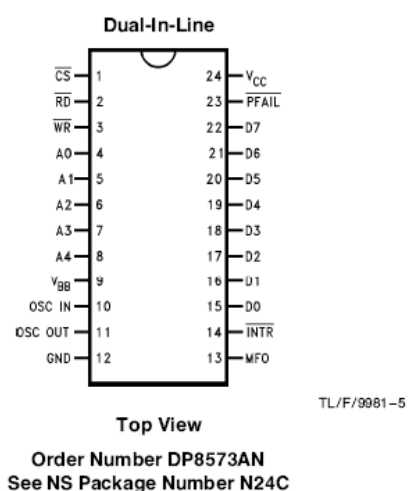


图4-19

完成计时器程序大致分为两个部分，一个是计时功能实现，另一个就是计时器初始化（端口初始化和显示），最后将光电触发功能加入计时器。在按键扫描中增加了对 P1.2 口的扫描，为了防止小车通过时多次触发，加入了延时程序（在

计时开始后程序主体经过 20 次循环即 0.53s 后才再次响应其他触发), 最终的程序比较稳定。

计时器操作方法:

1. 接线: 发射管接 VIN01 和 VIN02 不分极性
接受管: 灰—GND 蓝—GND 棕—VPP 黑—SIN 白—GND
显示屏将白色接口插上即可
2. 操作: 接上电源, 屏幕显示 smartcar 即正常工作, 注意对射装置要对准。若不正常工作, 断开电源在插上。若要重启可按下右边第一个按键 reset。计时器也可手动操作, 右边第二个按键是开始, 第三个是停止。
3. 注意事项: 最好不要长时间通电, 防止板子过热。

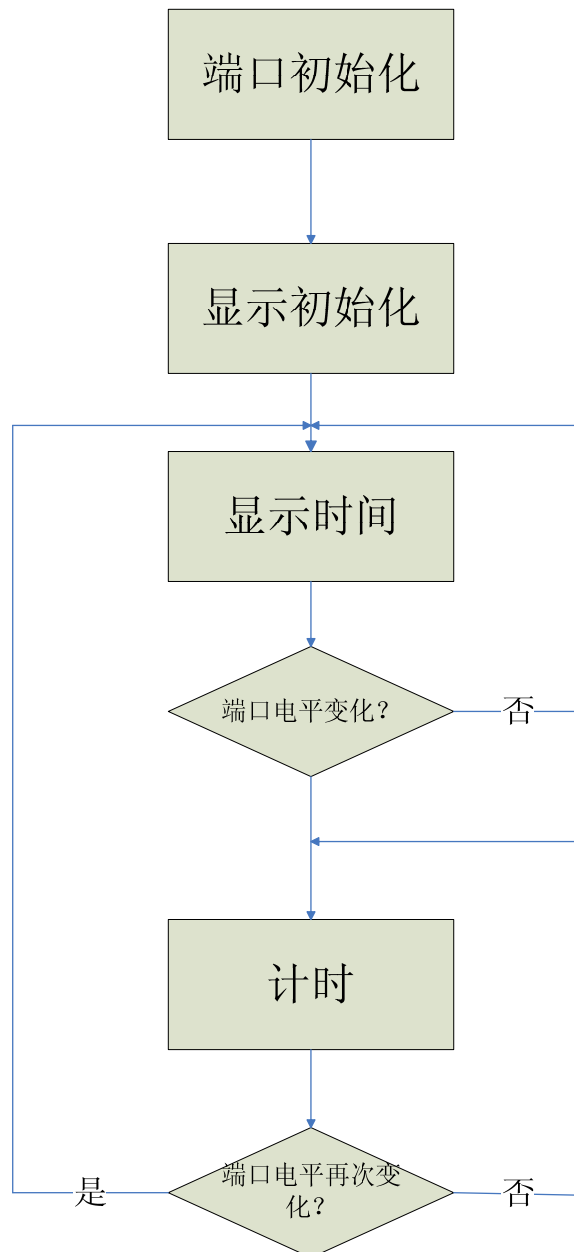


图 4-20 计时器程序流程图



图 4-21 计时器实物图

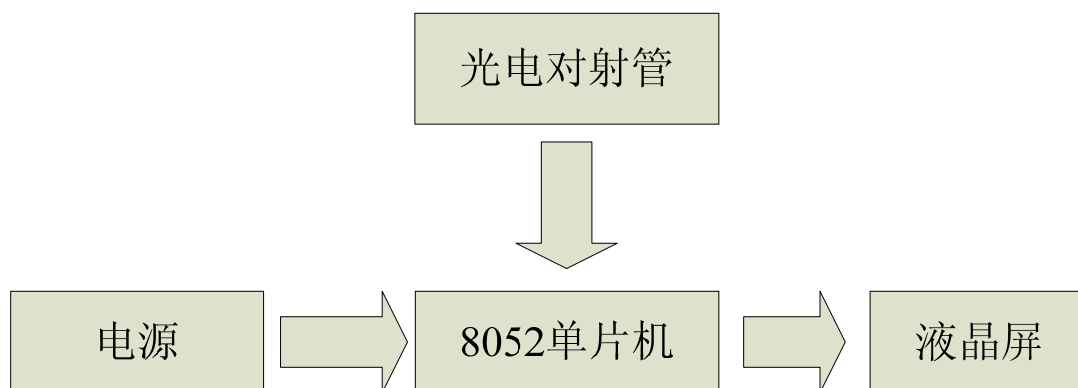


图 4-22 计时器系统框图

第五章 赛车系统软件设计

5.1 s12 系统片内资源

MC9S12DG128 微控制单元作为MC9S12 系列的16 位单片机，由标准片上外围设备组成，包括16 位中央处理器、128KB 的Flash 存储器、8KB 的RAM、2KB 的EEPROM、两个异步串行通信接口、两个串行外围接口、一组8 通道的输入捕捉或输出捕捉的增强型捕捉定时器、两组8 通道10 路模数转换器、一组8 通道脉宽调制模块、一个字节数据链路控制器、29 路独立的数字I/O 接口、20 路带中断和唤醒功能的数字I/O 接口、5 个增强型CAN 总线接口，同时，单片机内的锁相环电路可使能耗和性能适应具体操作的需要。S12 的片内资源如图5-1所示。

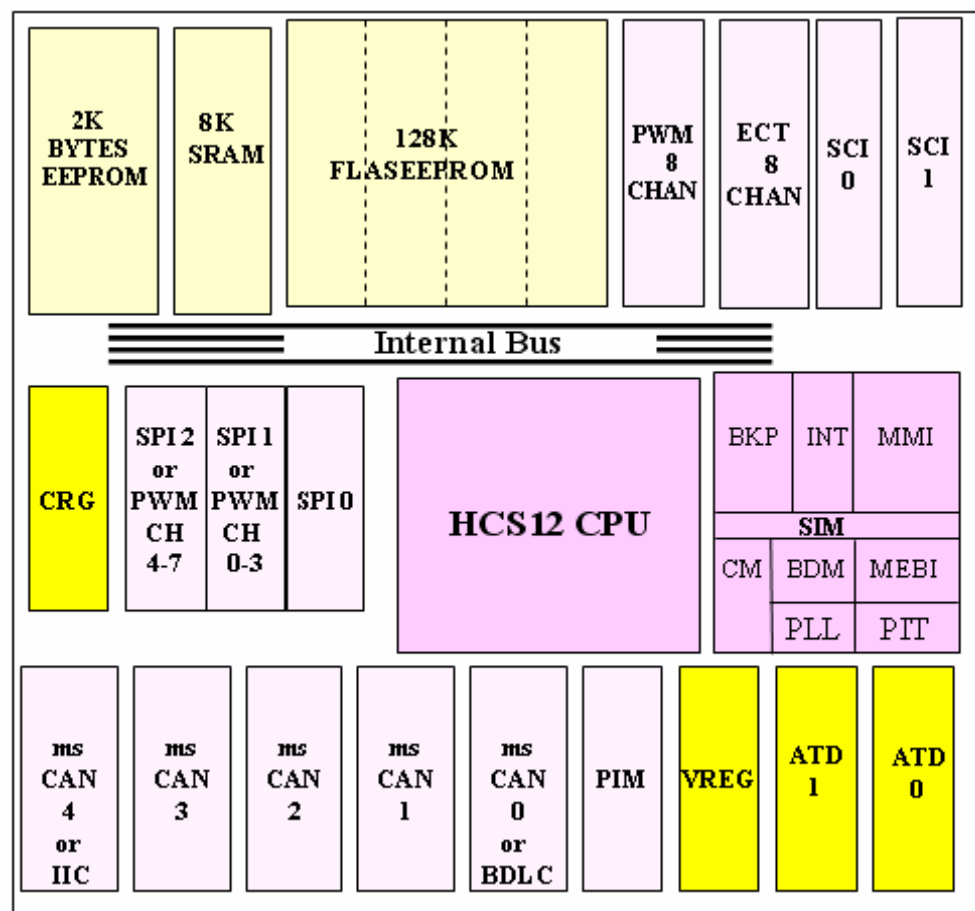


图5-1

具备了以上这些卓越的性能，MC9S12DG128 完全能够满足我们设计的需要，并使得电路更加紧凑。以下将首先介绍系统设计中用到的各个软件功能模块的设

置，然后再讨论黑线提取及车体控制算法。

5.2 系统初始化

5.2.1 摄像头初始化

由于摄像头行同步信号接到了S12 单片机的外部中断IRQ 引脚，所以当每行视频信号到来时，会产生一个中断。此时如果需要采集该行，就开始进行AD 转换。摄像头的初始化包

括初始化中断和AD 模块，具体过程为：

```
DDRS_DDRS2=0; //奇偶场信号输入
INTCR_IRQE=1; //外部IRQ使能
INTCR_IRQEN=1; //IRQ选择为边沿触发
ATD0CTL2=0xc0; //AD上电，快速清零，无等待，关闭外部触发
ATD0CTL3=0x08; //每通道转换一次，无 FIFO
ATD0CTL4=0x81; //8位精度，2个时钟周期，ATD时钟=8MHz
ATD0CTL5=0xa0; //右对齐，无符号，单通道，通道0
ATD0DIEN=0x00; //关闭数字输入
```

5.2.2 PWM 初始化

脉宽调制（PWM）模块有8 路独立的可设置周期和占空比的8 位PWM 通道，每个通道配有专门的计数器。该模块有4 个时钟源，能分别控制8 路信号。通过配置寄存器可设置PWM的使能与否、每个通道的工作脉冲极性、每个通道输出的对齐方式、时钟源以及使用方式（单独输出还是合并输出）。

为了提高控制精度，将PWM0、PWM1 两路8 位通道合并为一个16 位通道来控制舵机，这样可使舵机的控制精度从1/255 提高到1/65536。PWM 模块的初始化设置过程为：

```
DDRS_DDRS7=1; //33886使能信号输出
PTS_PTS7=0; //33886使能
PWME=0x00; //关闭pwm
```

```
PWMCTL_CON01=1; // PWM01合并为16位输出
PWMPRCLK=0x34; //时钟A=2M, 时钟B=4M
PWMSCLA=100; //时钟SA=时钟A/2/100=10k
PWMSCLB=1; //时钟SB=时钟B/2/1=2000k
PWMCLK= 0b00011100; //pwm0,1-A; pwm2,3-SB; pwm4-SA
PWMPOL=0xff; //极性位=1, duty=高电平时间
PWMCAE =0x00; //左对齐
PWMPER01=40000; //舵机pwm周期=时钟A/40000=50Hz
PWMDTY01=3000; //舵机中心位置
PWME_PWME1=1; //舵机使能
PWMDTY2=100;
PWMPER2=200; //电机频率=时钟SB/200=10K
PWME_PWME2=1; //电机使能
```

5.2.3 速度传感器初始化

通过对每个控制周期内速度传感器产生的脉冲数进行累加，可以测出这一周期内的平均速度。由于目前速度传感器的精度较高，所以用了16 位的脉冲累加器PB 进行脉冲计数。

其初始化过程为：

```
TIOS=0x00; //所有通道设置为输入捕捉
TSCR1_TEN=1; //主计时器使能
TCTL4=0x03;
PBCTL_PBEN=1; //16 位脉冲累加器B 使能
```

5.2.4 串口初始化

S12 有两个串行通信接口SCI0 和SCI1，均为TTL 电平输出。使用时，可以对波特率、数据格式（8 位或9 位）、发送输出极性、接收唤醒方式等进行选择。另外，发送和接收可分开使能，模块中还提供了多种避免传输错误的选项。在系统调试时，利用其中一个串口SCI0 通过Zigbee 和上位PC 通

信，其初始化过程如下：

```
SCI0BD=52; //波特率=总线时钟/(16*BD)=32MHz/(16*52)=38400Hz
```

```
SCI0CR1=0x00; //正常工作，8 个数据位，1 个停止位
```

```
SCI0CR2=0x0c; //允许收发
```

5.2.5 超频设置

S12 单片机中有四个不同的时钟，即外部晶振时钟、锁相环时钟、总线时钟和内核时钟。目前电路采用的是16MHz 的外部晶振，默认设置下，锁相环时钟为32MHz，总线时钟为8MHz，内核时钟为16MHz。通过将PLLSEL 置位，可选择总线时钟从锁相环时钟获得，而锁相环时钟与外部晶振时钟的关系由SYNR、REFDV 两寄存器决定。通过寄存器设置，将总线时钟超频到了32MHz。由于总线时钟用作片上外围设备的同步，而内核时钟则用作CPU 的同步，它决定了指令执行的速度，所以超频后AD 的采样性能大大提高。具体设置过程为：

```
REFDV=3;
```

```
SYNR=7; //总线时钟=16MHz*(SYNR+1)/(REFDV+1)=32MHz
```

```
while(0==CRGFLG_LOCK); //等待VCO 运行稳定
```

```
CLKSEL=0x80;
```

5.2.6 参数设置模块初始化

系统参数设置模块由1 个数码开关、2 个数码管和5 个LED 指示灯组成。通过数码开关选择要设置的参数类型和该参数的具体值，同时利用LED 灯和数码管进行显示，可以方便地对各种参数进行设置。这部分的初始化过程为：

```
DDRA=0xff; //A 口输出
```

```
DDRB=0xff; //B 口输出
```

```
PORTA=0x00;
```

```
PORTB=0x00;
```

```
TIOS=0x00; //设置为输入捕捉
```

```
TSCR1_TEN=1; //计时器使能
```

TCTL3=0b01001000; //捕捉PT7 的上升沿和PT5 的下降沿

DDRE=0xff;

DDRP=0xff;

5.3 视频采集和黑线提取

5.3.1 视频信号时序

由于摄像头产品说明上通常只会给出有效像素和分辨率（分辨率即为每场信号中真正为视频信号的行数），但不会具体介绍视频信号行的持续时间、它们在场信号中的位置、行消隐脉冲的持续时间等参数，而这些参数又关系到图像采样的有效实现，因此利用S12 的AD 和定时器对这些参数进行了实际测量。具体结果见表5-1。

表5-1视频信号时序参数

信号属性	行序数	行持续时间	行同步脉冲持续时间	消隐脉冲持续时间
场消隐区	1-4	23us		3.5us
	5	27.3us		8us
	6	37.3us		3.5us
	7-10	29.8us		3.5us
	11-22	64us		4.7us
视频信号区	23-310	64us	4.7us	
场消隐区 (同步脉冲)	311-314	64us		4.7us
	315	64us		3.5us
	316-319	29.8us		3.5us
	320	53.4us		28us

从上表可以看出，该摄像头扫描的每场中有320 行信号，其中第23 行到310 行是视频信号，第311 行到下一场的第22 行是场消隐信号。在视频信号区，每行信号持续的时间相同，约为62us；每行的行同步脉冲持续时间也相同，约为4.7us。而在场消隐区，每行持续的时间会有所变化，每行对应的消隐脉冲持续时间，尽管大多数为3.5us，但也有变化。场消隐区中第320 行的消隐脉冲持续时间远长于其他消隐脉冲，此脉冲即为场同步脉冲。

5.3.2 视频信号采集

在利用LM1881 分离出了行同步信号、场同步信号和奇偶场信号后，可以采用查询方式或者中断方式对视频信号进行采样。结合前面的电路原理图，主程序采用I/O 查询方式判别奇偶场信号跳变，并用中断方式处理行同步引发的外部中断。当奇偶场信号发生跳变（即新的一场到来）时，对行同步信号计数器清零。在行中断服务函数中，每来一个行同步则行计数器加1，当行计数等于所需采样行时，开始对此行信号进行AD 转换，直到下一个行同步信号到来。行中断服务函数的流程为：

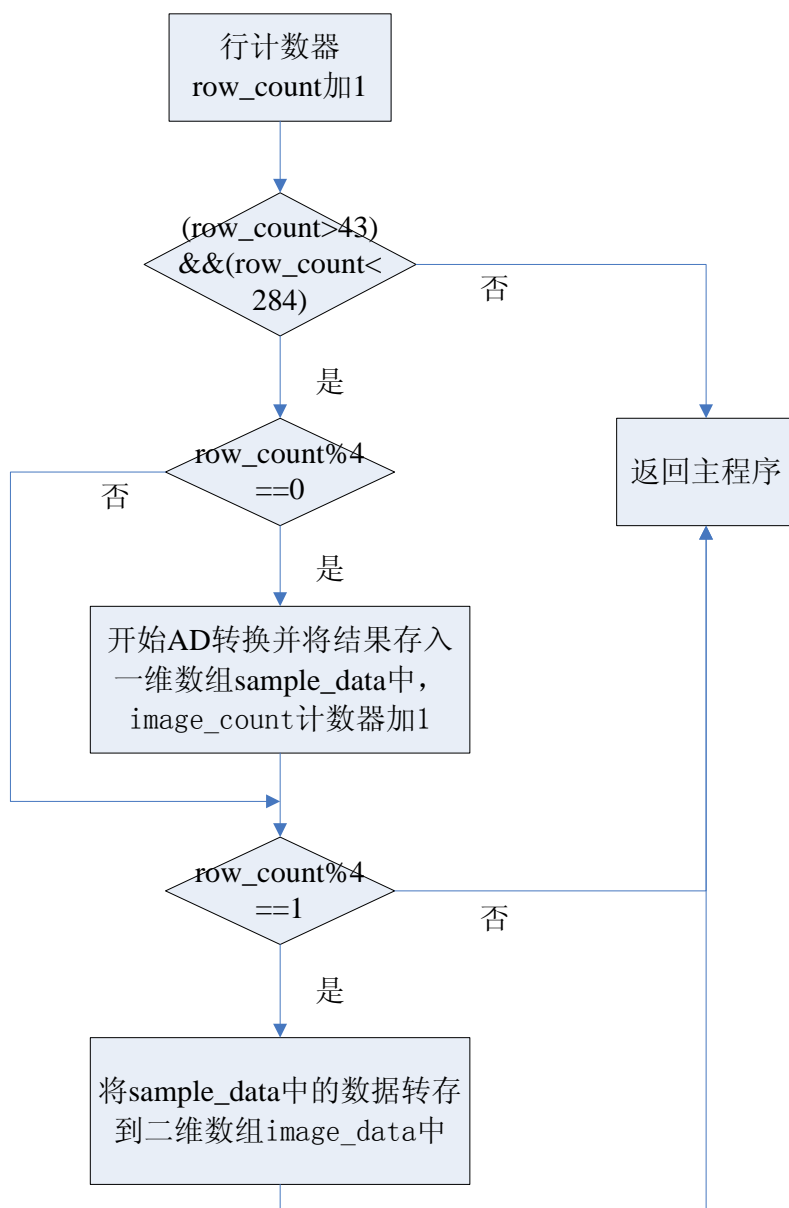


图 5-2 行中断服务函数流程

由于实际中没有必要对280 行视频信号全部进行采集，所以选择第43 行到284 行间等间隔的60 行视频信号进行采集。为了提高采样精度，每次AD 转换时先将视频信号存入一个一维数组sample_data 中，然后在紧接着的下一行时间里将该行信号转存到二维数组image_data 中。目前，单行视频采集的点数为80（其中有效点数为65），这一采样精度使得系统可以针对车体偏差，对舵机位置进行精确调整，以提高系统稳定性。将采集到的图像通过串口发送至PC 机，然后用Labview 程序进行显示，可以清楚地看到视频采样的效果：

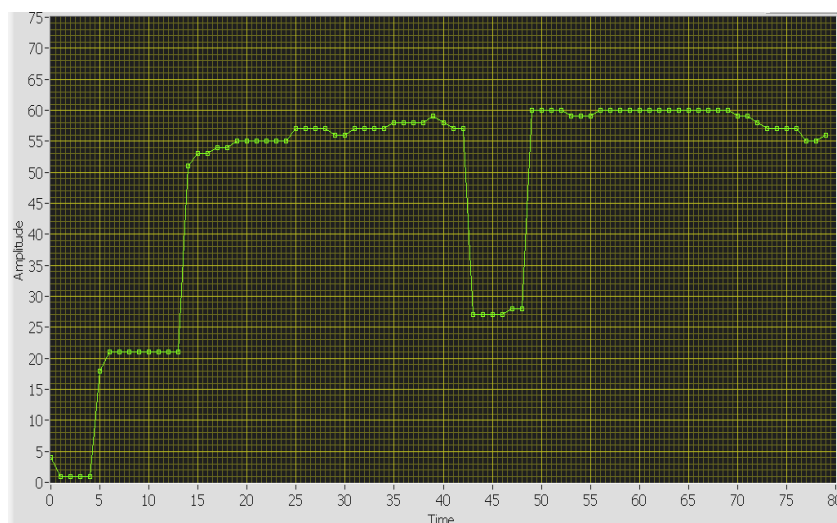


图 5-3 单行 80 点：前 0—13 点为行消影、凹槽为黑线所处位置

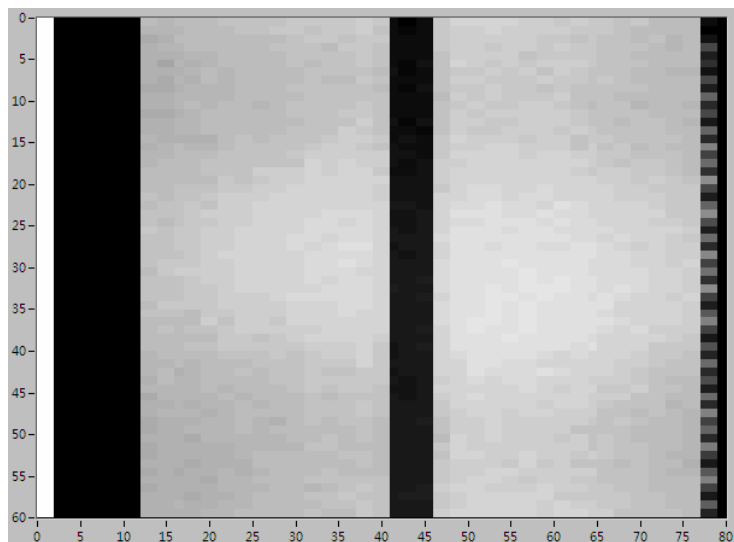


图 5-4 60 行视频采集效果

5.3.3 黑线位置提取

得到一场视频信号后，可以从中提取出黑线位置、是否是起始线交叉线

等各种有效信息。这些信息可以用于对舵机、速度进行控制，也可用于赛道记忆。下面先对黑线位置提取算法进行介绍。目标导引线（即赛道上的黑线）的宽度相对整个赛道较窄，因此只要提取目标导引线的某些特征点，就能反映出导引线的形状。如图5-5，可以取每行的中间点或边缘点作为该行的特征点。

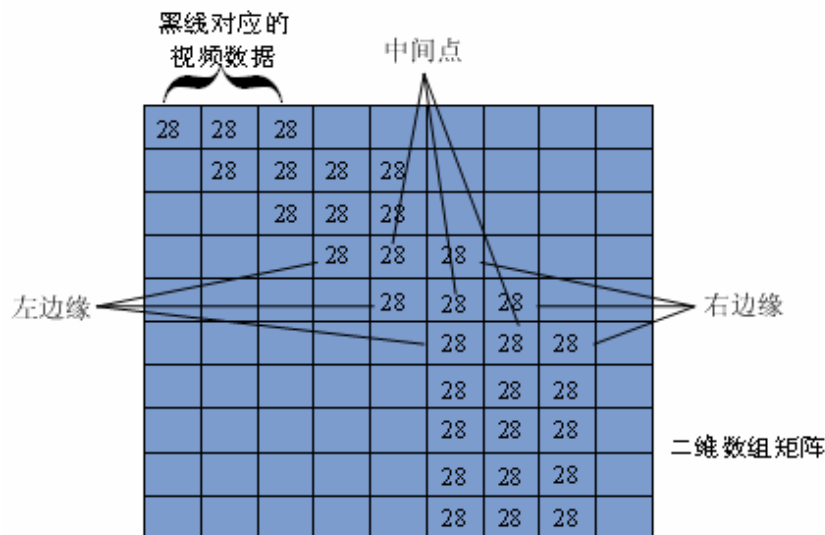


图 5-5 黑线特征点示意图

5.4 速度闭环控制

5.4.1 速度控制介绍

通过速度传感器获取的当前速度值调整电机的PWM 占空比，可以实现对于速度的闭环控制。这样做改变了通过直接设置PWM 占空比调整电机转速的开环控制方法，通过对速度的闭环控制，去掉电源电压和车身重量对车速的影响，采取了最可靠的方法，保证赛车各段速度较为稳定。

速度闭环控制有很多种方法，比如PID 控制、bang-bang 控制。经过实际测试，由于车模电机属于低负载简易永磁直流电机，其加速减速特性较差，因此如采用PID 控制，P 参数只能控制在很小的范围，否则很容易发生超调。P 参数的减小，导致电机速度响应变慢，影响实际赛车控制效果。而选用 bang-bang控制方法，让电机动态稳定在一定转速下，实际在赛道上表现很

稳定，而且速度响应很快。

5.4.2 实验依据

pwm 波与实际速度的关系

首先，测定开环时不同PWM 占空比对应的稳定速度值。由于电机启动时，初始角速度为0，此时转动惯量很小，因此角加速度很大。电机在直流电压的驱动下不断加速，同时转动惯量也逐渐增大，使得角加速度减小。最后角加速度为0，电机速度稳定在某一水平上。这一速度值反映的是该PWM 占空比下，电机所能达到的最高速度。实验测得的结果如图5-6所示。

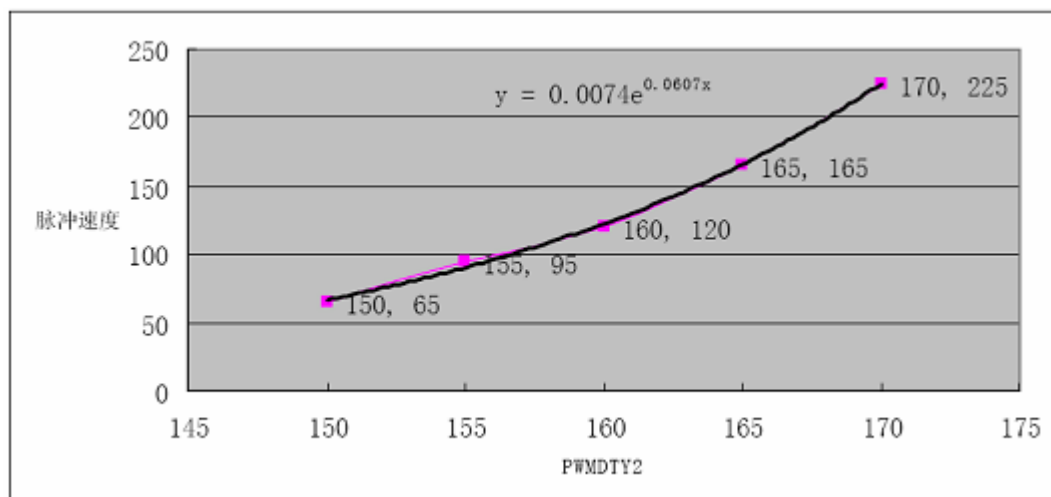


图5-6 pwm占空比与稳态速度的对应关系

实验例图

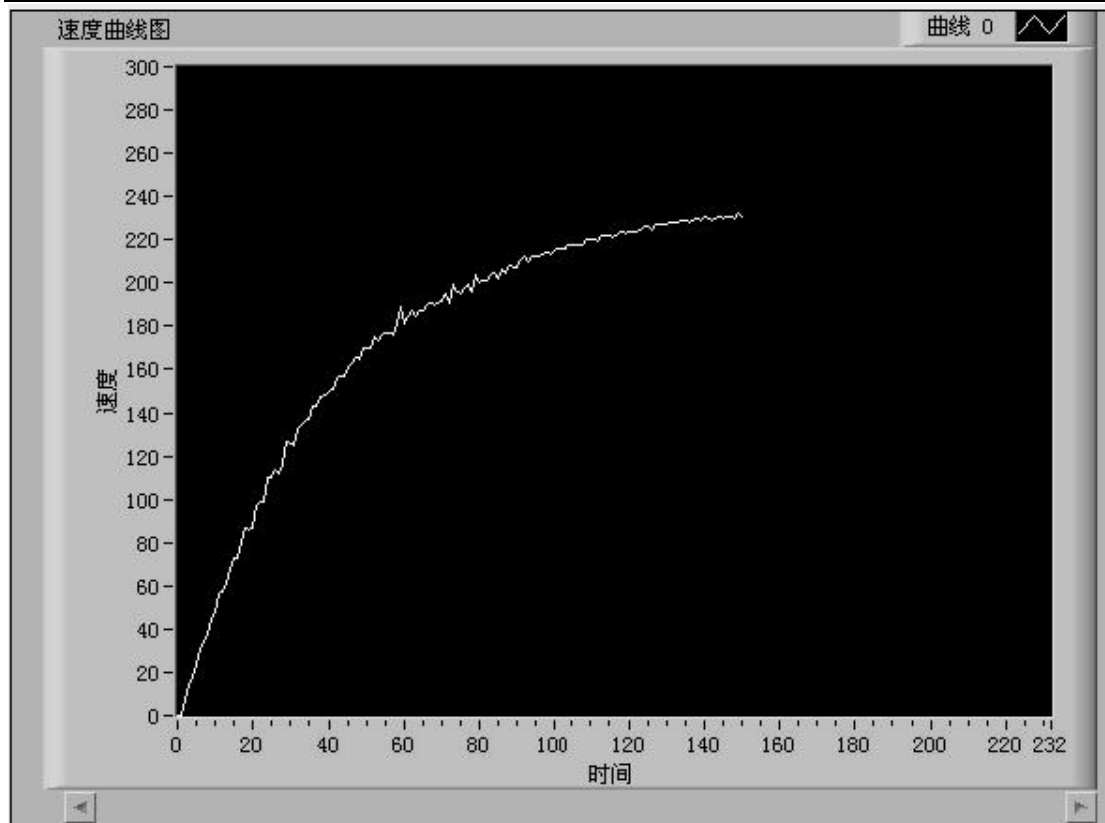


图5-7 PWMDTY=170

5.5 EEPROM

由于单片机芯片的Ram容量有限，而记忆算法需要大量的存储空间记录赛道的信息，第一圈记录的数据越详细，第二圈可用于控制的信息就越丰富，因此如何记录和存储足的数据也是运用记忆算法的基础。

算法研究者可以对算法进行优化处理，改变保存的格式以节省内存空间。例如采用一定的压缩算法，把直道，弯道分成若干级别，分别记录他们的距离和顺序。优化压缩虽然可以节省存储空间，但同时对于赛道信息多少都有丢失，不利于记忆算法的准确使用，因此如果可以获得更大的存储空间，就可以完整记录赛道的特征信息。S12提供了2K EEPROM，EEPROM就是electrically erasable programmable，是一种电可擦除可编程只读存储器，并且其内容在掉电的时候也不会丢失。在平常情况下，EEPROM与EPROM一样是只读的，需要写入时，在指定的引脚加上一个高电压即可写入或擦除，而且其擦除的速度极快！通常EEPROM芯片又分为串行EEPROM和并行EEPROM两种，串行EEPROM在读写时数据的输入/输出是通过2线、3线、4线或SPI总线等接口方式进行的，而并行EEPROM的数据输入

/ 输出则是通过并行总线进行的。

小车的平均速度一般都在2 米左右，对于一个50M 长的赛道，25 秒内就可以跑完，由于摄像头50Hz 的限制，小车的控制周期一般都是20ms, 由此可以得出跑完一圈的才运行了1000 多个周期，就算每个周期都记录一定的信息，EEPROM 也足够满足需求。在对EEPROM 进行编程和擦除时，要严格地遵守下列步骤：

首先，访问ECLKDIV 寄存器，进行时钟分频，EEPROM 需要编程时钟在150-200KHz 之间。如果没有设定时钟分频，对EEPROM 的一切操作无效。检查CBEIF 标志位，确定地址、数据和命令缓冲区是否为空。然后，进行如下的三步操作（注意，操作顺序不能改变，中间也不可插入其他操作）：向需要编程的地址写入一个字（2Byte）。如果是全擦除操作，可向EEPROM 中任意一个地址写入数据；如果是向量擦除，则二进制地址的最后两位将被忽略。向ECMD 寄存器写入命令字。向CBEIF 位写入1 开始命令。CCIF 位置1 表示命令被执行完毕。用户可以通过中断机制确定命令的完成。如果出现无效的命令字或是企图修改保护区内容的操作，ACCERR 和PVIOL 位会置位。

测试表明，对EEPROM 每次进行写操作所需的时间大约为0.9ms，而每次读的时间仅需2us。写操作的时间较长，这应引起足够的重视，因为对于摄像头控制方案，保证控制周期在20ms 以内是非常必要的，否则赛车在高速行驶时将来不及反应。除此以外，对于记忆信息的处理时间也应尽量压缩。由于第一圈控制算法通常较为简单，这给信息处理提供了一定的时间裕量，所以只要算法得当，赛道记忆算法的条件能得到满足。

EEPROM程序代码见附录C

第六章 系统开发与调试

6.1 软件调试平台

Codewarrior 是由Metrowerks 公司提供的专门面向Freescale 所有MCU 与 DSP 嵌入式应用开发的软件工具。其中包括集成开发环境IDE、处理器专家、全芯片仿真、可视化参数显示工具、项目工程管理、C 交叉编译器、汇编器、链接器以及调试器。其中在本设计方案中最为重要的部分就是集成开发环境IDE 以及调试器，所以接下来将主要介绍该两部分。

6.1.1 Codewarrior IDE 功能介绍

CodeWarriorIDE 能够自动地检查代码中的明显错误，它通过一个集成的调试器和编辑器来扫描你的代码，以找到并减少明显的错误，然后编译并链接程序以便计算机能够理解并执行你的程序。每个应用程序都经过了使用像CodeWarrior 这样的开发工具进行编码、编译、编辑、链接和调试的过程。具体到比赛所应用MCS12DG128B 的一个显著特点就是片上外围设备众多，每个外围设备对应的寄存器也较多，有的甚至达到三十多个。MetrowerksCodewarrior IDE 中的mc9s12dg128.h 文件对所有寄存器对应的存储映射地址都进行了宏定义，开发者在软件开发时直接调用这些宏就可以了（如图6-1）。而且，这些宏的名称都与说明文档上相应寄存器的名称相同或类似，这样，很便于对MC9S12DG128 的开发。

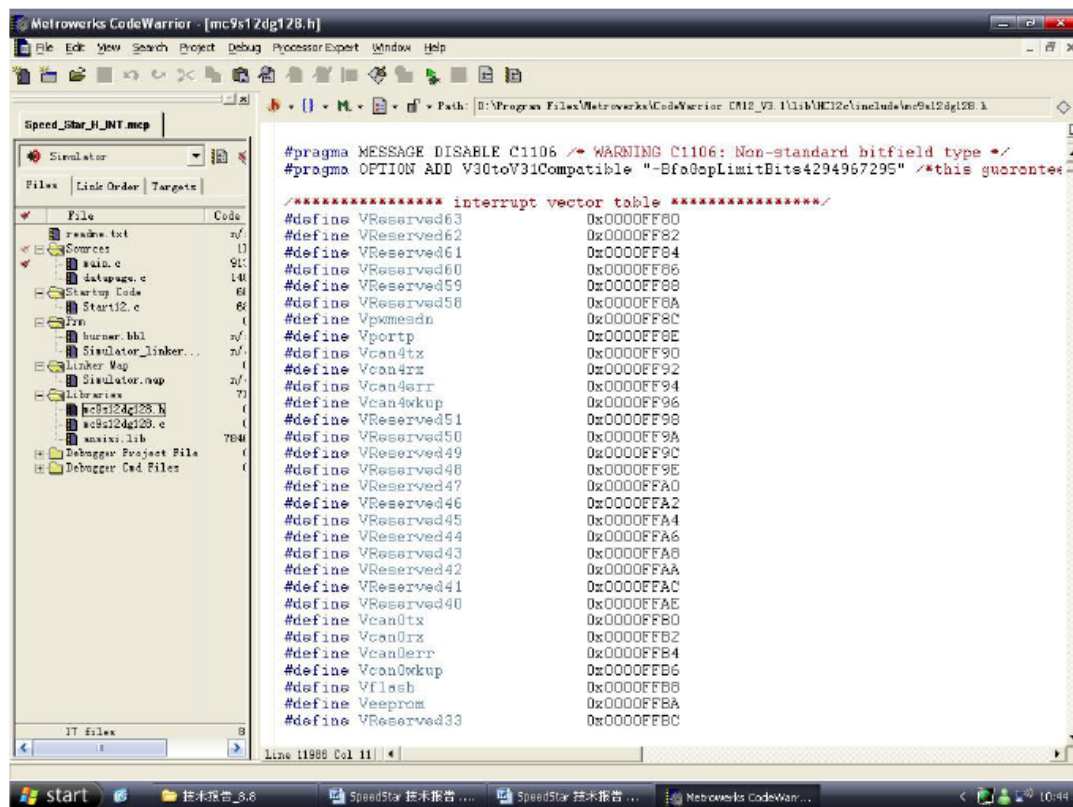


图6-1

6.1.2 Codewarrior IDE 基本使用方法

A 运行“开始菜单—>所有程序—>Metrowerks CodeWarrior—>CW12V3.1—>CodeWarrior IDE”，选择“File—>New”，出现如下的对话框

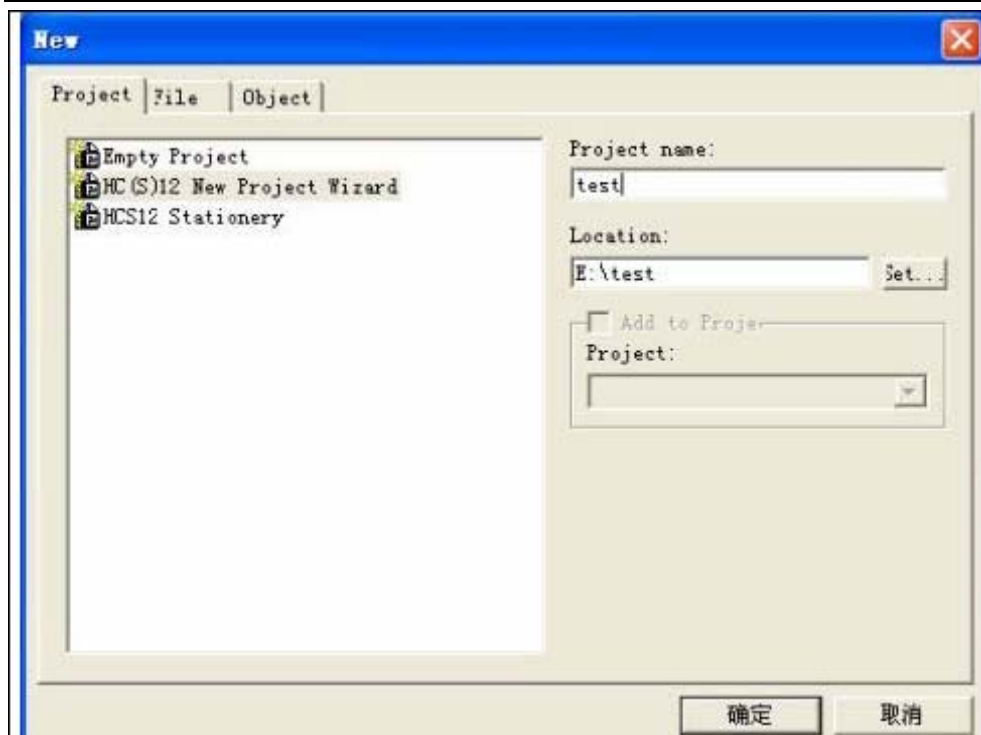


图6-2

此后选择MC9S12DG128B，按照提示选取期望的选项。直至建立工程文件。如下：

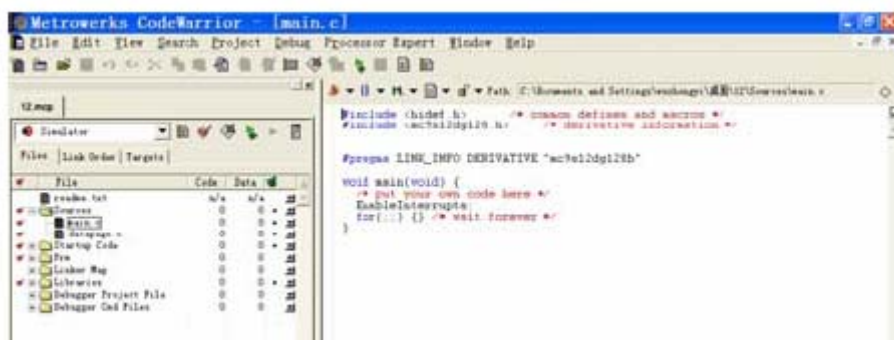


图6-3

打开Start12.c 文件，找到代码

```

#ifdef _HCS12_SERIALMON
#define __INITRM (*(volatile unsigned char *) 0x0010)
#define __INITRG (*(volatile unsigned char *) 0x0011)
#define __INITEE (*(volatile unsigned char *) 0x0012)
#endif
  
```

修改成：

```
//#ifndef _HCS12_SERIALMON  
  
#define __INITRM (*(volatile unsigned char *) 0x0010)  
  
#define __INITRG (*(volatile unsigned char *) 0x0011)  
  
#define __INITEE (*(volatile unsigned char *) 0x0012)  
  
//#endif
```

之所以要进行这个步骤是因为，初始化代码是认为程序中有监控程序的，但是因为我们使用的是bdm 调试工具，所以就不需要监控程序了。如果不注释掉这两条语句，一旦将程序烧写到片内的flash，会发现程序跑不起来，因为初始化过程在等待监控程序的响应，如果没有监控程序，那么程序就会进入死循环的状态。

C 添加代码：将自己的代码加入到工程文件中。

D 如果文件编译通过，通过BDM 下载。启动CodeWarrio 4.1 自带的Hiwave.exe 程序，该程序一般位于CodeWarrio 4.1 的安装目录“\Metrowerks\CWfor HCS12X V4.1\prog”中，是用来往单片机下载程序和调试单片机的程序。启动后程序界面如图6-4 所示。

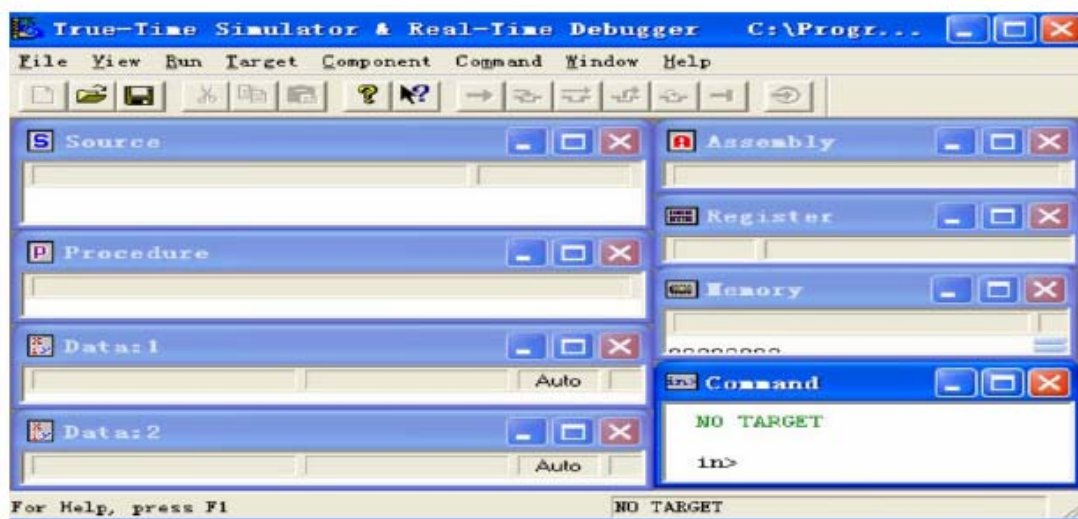


图6-4 HIWAVE程序烧写环境

装好相关驱动后，界面如图6-5 所示：

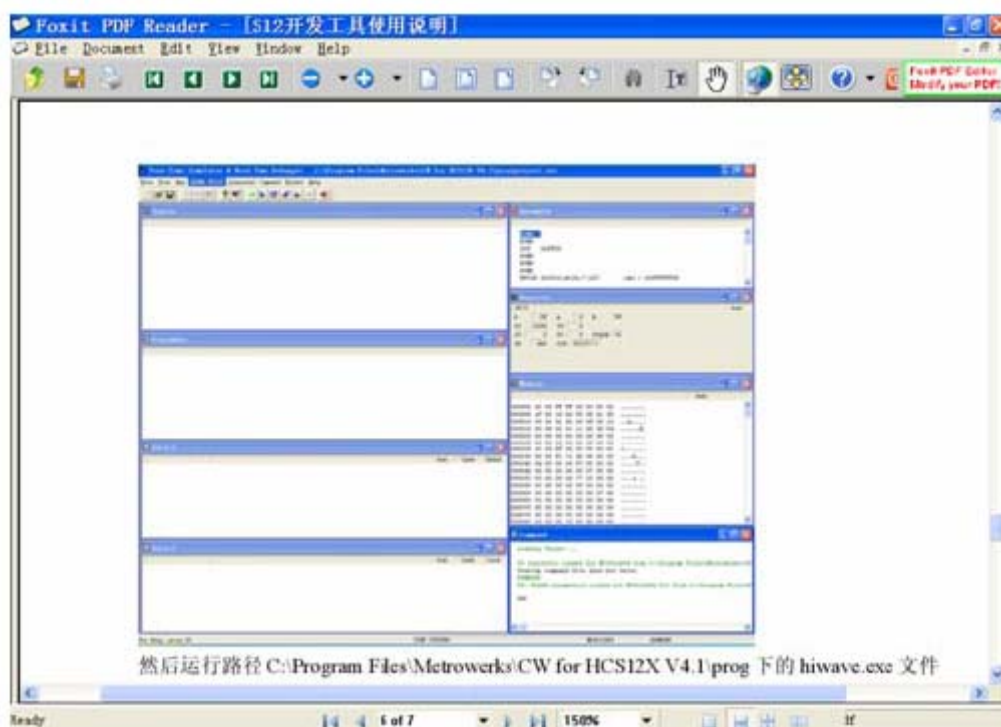


图6-5 烧写环境建立界面

之后运行路径C:\Program Files\Metrowerks\CW for HCS12X V4.1\prog 下的 hiwave.exe 文件选择TBDML HCS12，在下拉菜单中选择“set speed”，输入晶振频率16.00，然后这个下拉菜单中会出现“Flash”选项，点击这个选项，出现如下的对话框：

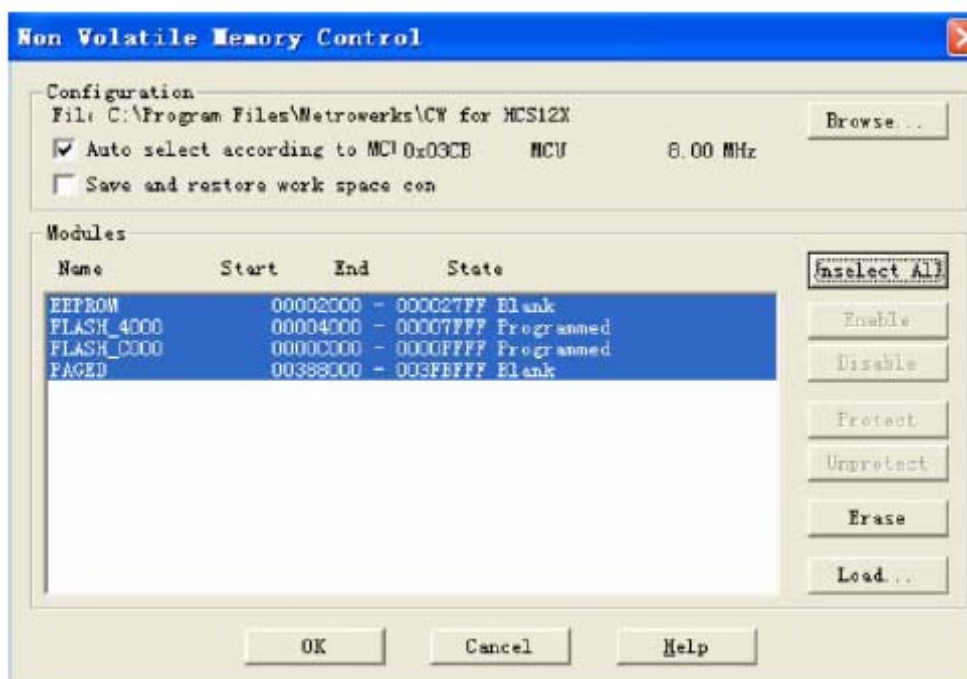


图6-6 程序烧写擦除界面

在下载自己的程序前应该将板子中已经存在的程序擦除，选择有程序的部分，点击Erase，擦除完成后，点击Load，选择自己所建立工程的文件夹中bin文件夹下后缀为“.abs”的文件，点击打开，下载完成。至此已经基本完成了从程序代码编写和设置过程，到下载烧制进S12 芯片的主要过程；其中也包括程序开发和调试的基本步骤。

6.2 辅助调试工具

6.2.1 无线通讯模块

在智能车的制作和调试过程中，需要将赛车检测到的路面信息以及速度等参数实时地发送给PC，以便对算法进行有针对性的分析。由于小车在行驶时不能通过有线的方式获得其运行参数，就需使用无线方式。

这里选择的是Zigbee 无线通信模块，它可以设置传输速率，以数据包的形式对数据进行收发。下图为Zigbee 模块的示意图。



图6-7 无线传输示意图

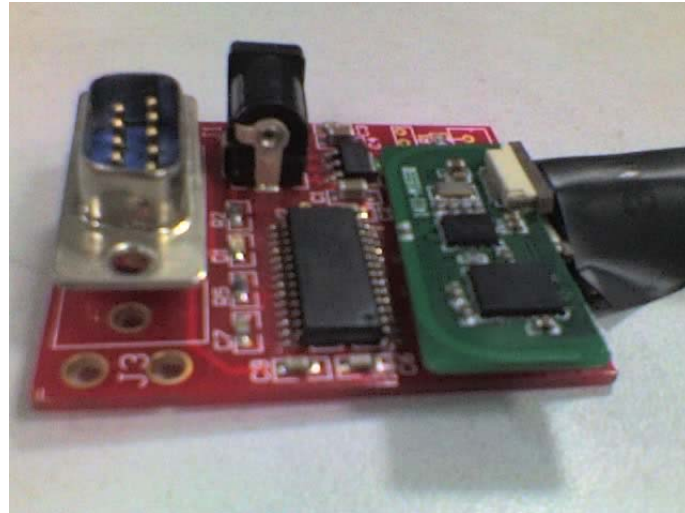


图 6-8 Zigbee 无线通信模块



图 6-9 ZigBee 模块安装图示

1、IP-Link1270 型模块支持信息

产品名称	备注	配件
IP-Link1270 模块	空模块，不带任何固件	14pin 软排线（送） USB 连接板（售）
IP-Link1270 模块	带有 Freescale 提供的 S-MAC 或 Full-MAC(须 Freescale 授权)	14pin 软排线（送） USB 连接板（售）

2、模块简介

IP-Link1270是完全符合IEEE802.15.4标准与ZigBee规范的2.4GHz无线

收发模块。射频部分使用Freescale的MC13191/ 13192 / 13193芯片，MCU使用的是Freescale 的MC9S08GT60 芯片。



图6-10 IP-Link1270 模块的照片

3、IP-Link1270 模块简化功能框图

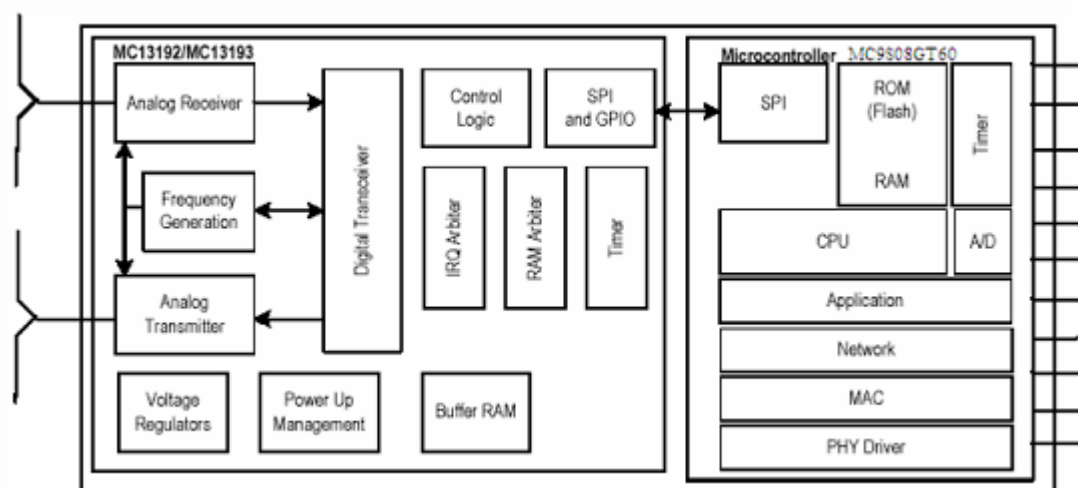


图 6-11 IP-Link1270 模块简化功能框图

4、接口定义

表 6-1

Pin No.	Name	Type	Function Description
1	GND	Power	Supply Ground
2	/RESET	RESET	
3	GPO/ BDM	Digital, O	General Purpose O(PTG0) ; download program for chip
4	RXD	Digital, I	UART1 Input RXD1(PTE1)
5	TXD	Digital, O	UART1 Output TXD1(PTE0)
6	GPIO1	Digital, I/O	General Purpose I/O(PTA7) ; interrupt for external
7	GPIO2	Digital, I/O	General Purpose I/O(PTA6) ; interrupt for external
8	GPIO3	Digital, I/O	General Purpose I/O(PTA5) ; interrupt for external
9	GPIO4	Digital, I/O	General Purpose I/O(PTA4) ; interrupt for external
10	GPIO5	Digital, I/O	General Purpose I/O(PTD3) ; Timer/PWM
11	GPIO6	Digital, I/O	General Purpose I/O(PTD0) ; Timer/PWM
12	GPIO7	Digital, I/O	General Purpose I/O(PTB0) ;ADC
13	GPIO8	Digital, I/O	General Purpose I/O(PTB1) ; ADC
14	VCC	Power	supply (2.5VDC ~ 3.4VDC)

其中，我们只用到了 1.4.5.14 号接口，1 号口是接地口，4 号是接受端口，5 号是发送端口，14 号口接 VCC。

5、软件支持

AT 命令概述

模块初次使用时，应采用 AT 命令通过 UART0 对模块进行配置，配置结果保存在模块内部非易失性存储器中（模块断电后配置结果仍然保留）。正确配置后，模块应进入数据状态。

AT 命令的格式是：ATxxx.....\r

其中“AT”为 AT 命令的前导字符，不区分大小写，用 ASCII 码表示；

“xxx.....”是由字母和数字组成的功能命令，字母和数字都是由 ASCII

表示的，其中的字母不区分大小写，数字反映十进制数值。

\r表示在AT命令的末尾要加上回车符（ASCII码为0DH，这里H表示16进制数）

IP-Link1270模块在收到AT命令后，将在10ms内由UART0向外设返回信息。若返回的信息是“Em\r”或在规定的时间内没有出现反馈信息，表明模块没有正确接收该命令或没有完成该命令所要求的功能；若返回的信息是“O\r”或其它相关内容，表示该AT命令对模块已经生效。

AT命令中的返回信息也是由ASCII码描述的，“\r”表示回车符。
“Em\r”中的“m”表示返回信息错误类型代码（错误类型代码参见附录1：AT命令中返回值错误类型代码），返回信息中的字母和数字都是由ASCII表示的，其中的字母不区分大小写，数字反映十进制数值。

例如，外设要通过模块UART0查询模块的节点号，可通过UART0向模块输入以下数据

（首字节）

（尾字节）

41H 54H 2FH 40H 0DH

对应于 A T / @ \r

查询模块的节点号， 模块的返回信息可能是

（首字节）

（尾字节）

4EH 4FH 44H 45H 3DH 35H 36H 0DH

对应于 N O D E = 5 6 \r

表示模块的节点号是56(十进制)；

模块的返回信息也可能是

（首字节）

（尾字节）

45H 31H 30H 30H 0DH

对应于 E 1 0 0 \r

表示模块没有完成此项功能，出现第100类型的错误；

模块也可能在规定时间内不会返回信息，表示模块没有接收此项AT命令。

6. 数据帧结构

采用AT命令通过UART0配置IP-Link1270模块，再用ATW\r命令使模块进入数据状态。在数据状态下，利用IP-Link1270模块组成的网络可以进行数据交换。数据交换必须按照IP-Link1270数据帧结构的格式来发送、接收、解析网络中的数据。

IP-Link1270模块的数据帧结构由数据模式、目标地址、数据长度、数据信息与校验和五部分构成，格式如下（数据帧结构中的数据都是16进制度数）：

（首字节）	（尾字节）
数据模式	目标地址
数据长度	数据1……数据n
数据信息，共n个字节	
校验和	

“数据模式” 占用一个字节，可以是43H（‘C’）、44H（‘D’）、52H（‘R’）和57H（‘W’）四种数值之一，分别表示数据帧结构是正确返回型、数据型、错误返回型和命令型。

“目标地址” 表示数据帧结构要发送的目标位置（网络中的节点号），“目标地址” 占用一个字节，取值范围是 0——255。当“目标地址”与自身模块的节点号一致时，表示数据帧结构发送到自身的模块内；当“目标地址”与自身模块的节点号不同时，表示数据帧结构要无线发送到目标地址所指定的网络节点中。

“数据长度” 表示数据数据帧结构中从“数据1”到“数据n”所占据的字节数。数据长度占据一个字节，取值范围是0——100。

“数据信息”表示用户要通过UART0传送的命令或有效数据，占据的字节数由“数据长度”规定，范围是0——100字节。

“校验和”是对帧结构中的全部数据（校验和字节除外）进行的校验，采用字节逐位异或的方式实现。校验和占据一个字节。

应当说明，命令型和数据型两种类型的数据帧结构都是由用户通过UART0送入IP-Link1270模块的，但这些数据帧结构的作用以及最终到达的目标是不同的。命令型数据帧结构的作用是对IP-Link1270模块进行设置或获取模块的有关信息（目前只是设置模块进入两种省电模式），最终到达的目标是IP-Link1270模块本身的内部电路，是直接与用户相连的IP-Link1270模块。

采用命令型数据帧结构的通讯方式实质上是外部设备与IP-Link1270模块之间的通讯，具体工作过程是：发送端把数据帧结构通过UART0送到直接连接的IP-Link1270模块上，该模块接收这些信息自行处理，并把处理结果发送回外部设备。

6.2.2 数码开关

一. 原理简介

新的 S12 电路板上将拨码开关换成了数码开关，数码开关可以左右旋转并可以按下，如 S12P80V1.mdi 图所示，开关有三个输出端口 QESA, QESB, QESP，它们分别与单片机的 PT5-PT7 相连。

如下图所示，当数码开关顺时针旋时，A，B 口的电平从右向左变化，依次是 10，11，01，00，而当数码开关逆时针旋时，A，B 口的电平从右向左变化，依次是 00，01，11，10。

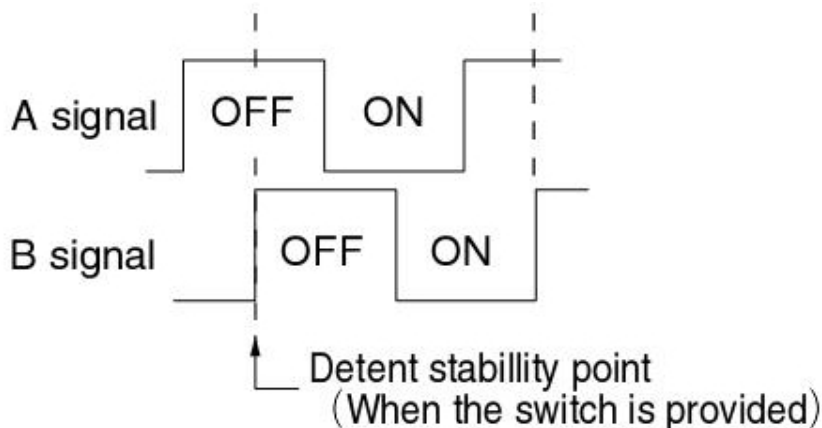


图 6-12

因此可通过不断读 PT5 和 PT6 口判断开关是否在旋转并确定方向，通过软件或硬件计脉冲数确定旋转量（由于一圈有 32 个脉冲，所以我将脉冲数除以 5，操作效果较好）。而 QESP 在位按下时为高电平，按下后即变为低电平，所以可用排脉冲捕捉，通过读标志位判断是否按下。

板子上有两个八段数码管和 5 个并行的 LED 灯可用于数码开关的操作和程序运行时的显示。

二. 程序使用

程序写成了函数，可移植到其它程序中直接使用，只要执行 void operate(void)函数即可。

程序开头对一些端口做的宏定义，还定义了一些全局变量，其中 LEDvalue[5]存储操作后所定的参数值（未操作前初值为零，若要改变初值，可在 void operate(void)函数的开头部分对 LEDvalue[]数组赋值），中除第一个元素外，其余四个均可赋值给控制程序中的任何参数。

若要单独使用数码管显示可直接调用 void dis_digital(char number)函数。

程序体见附录 D。

三. 操作

数码开关顺时针旋转为增加，逆时针为减少，按下为确定。

进入操作程序，最上方一个 LED 灯亮，数码管显示为“0”，此时若按下数码开关，即不改变任何参数，跳出操作界面。若旋转数码开关可选择第几个 LED 亮（即选择要调整的参数），移动到所需要的灯后，按下即可调整该

灯所对应的参数，调整完，按下确定后修改即生效。然后可继续调整其它参数。若要退出，选择第一个 LED 按下确定键即可。

四. 问题

在修改参数时，可能会发生跳变，这可能和数码开关的内部结构有关。

6.3 仿真平台 LABVIEW

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。接下来将主要介绍如何将虚拟仪器作为辅助的调试开发工具。

6.3.1 虚拟仪器 Labview 和串口通讯

摄像头方案相比于光电管方案，它的底层信息更为复杂，所以只有在充分了解和掌握视频信号特点的情况下才有可能将其应用至赛车的实际控制。如果能够将 S12 采集到的视频信号以及中间算法处理得到的数据通过串口发送至 PC，不仅能够直观地看出视频信号的特点，还能够检验算法的准确性和可靠性。

虚拟仪器中对串口的配置如同超级终端一样的方便快捷，只需要选择串口号、波特率、数据位、校验位、停止位和控制位就可以了。主要的子 VI 就是 VISA 中的 SERIAL。

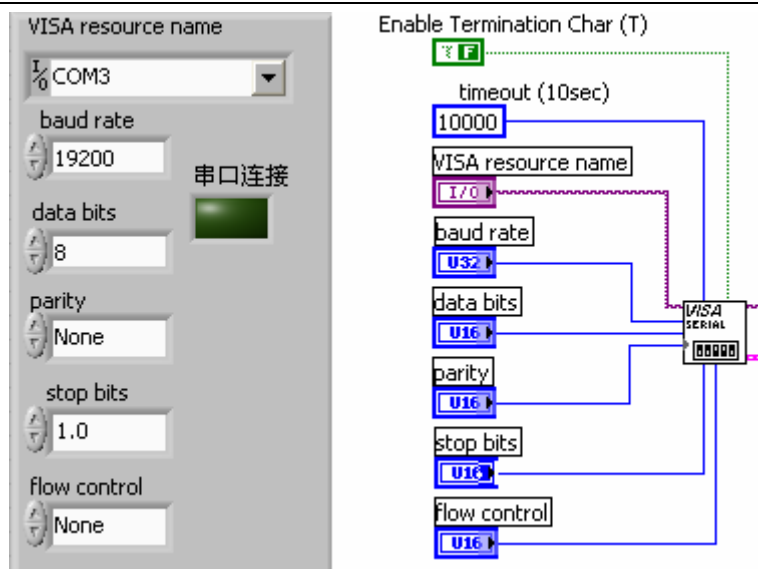


图 6-13 Labview 串口配置程序

配合 Labview 中的串口配置程序，S12 的程序也需要进行相应的串口配置，其初始化设置在前面已经介绍过，这里不再赘述。

6.3.2 Labview 测试程序

(1) 虚拟仪器 Labview 单行视频显示

设置好串口通讯协议之后，就可以将 S12 采集到的任何数据通过串口发送到 PC 上进行观察和测试。首先，需要测试的是摄像头信号单行的数据特点。虽然通过查阅说明书可以了解一部分信号特征，比如行同步、行消隐、场同步和场消隐，但是具体到 AD 采样以后的数据信号特点还是需要通过直接观察才能有一个比较明确的认识和了解。所以，首先得到了单行的采样数据：

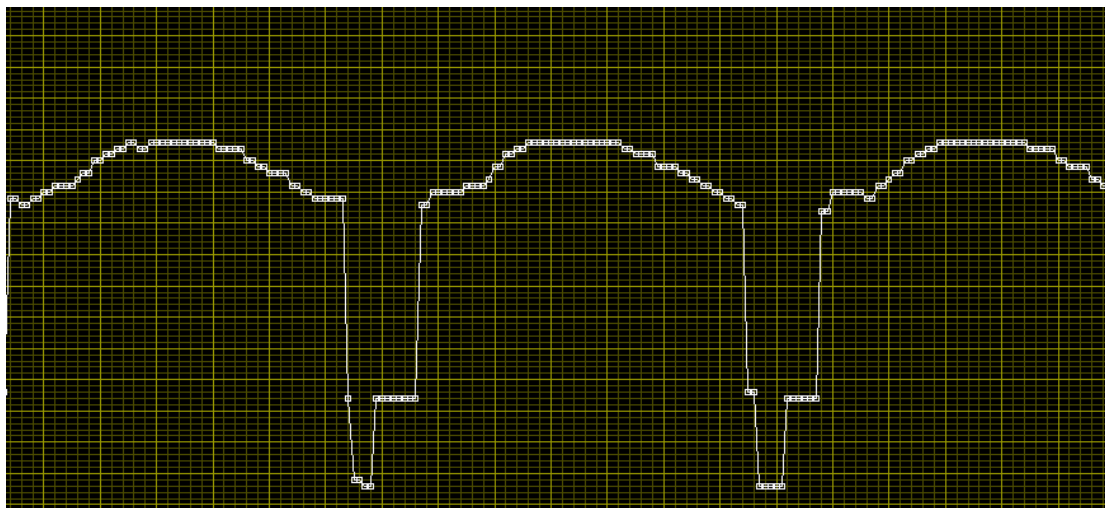


图 6-14 单行视频信号

通过观察图 6-14 中的数据曲线，可以很明确地看到行同步信号和行消隐所需要维持的时间。相比与示波器而言，这样不仅能看到波形，而且可以具体得到哪些点是实际上的有效点，有效数据点到底有几个等等。

(2) 虚拟仪器 Labview 40×54 灰度图显示

在得到了单行的数据信号之后，就可以将多行数据通过串口发送到 PC 上，然后在 Labview 中用灰度图来显示，这样就得到了整幅图像的信号特点：

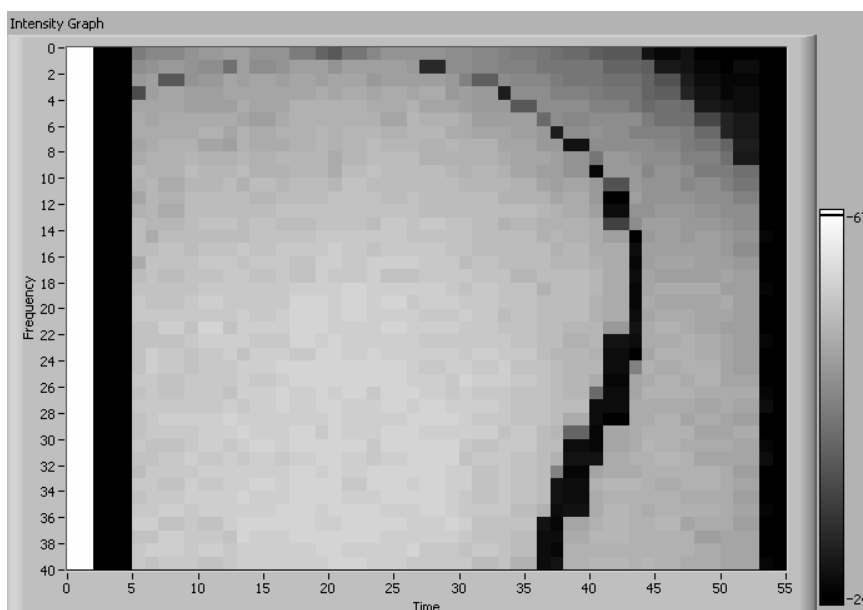


图 6-15 虚拟仪器用灰度图显示 40×54 视频信号

虽然最终算法和控制不需要分辨率这么高的图像,但是通过对图像的观察可以进一步加深对图像采集和数据转换的认识,以便在后面的算法控制和策略研究中应用地更加灵活自如。

(3) 虚拟仪器 Labview 舵机测试、速度测试等

虚拟仪器除了简单的对 S12 发送上来的数据进行处理之外,也可以对赛车进行实时控制,这对于进行类似舵机测试和速度测试来说是非常重要的。但是由于 S12 控制舵机的 PWMDTY01 是 16 位的,所以需要将其转化为两个 8 位的整型数据。具体的程序可以参见下图:

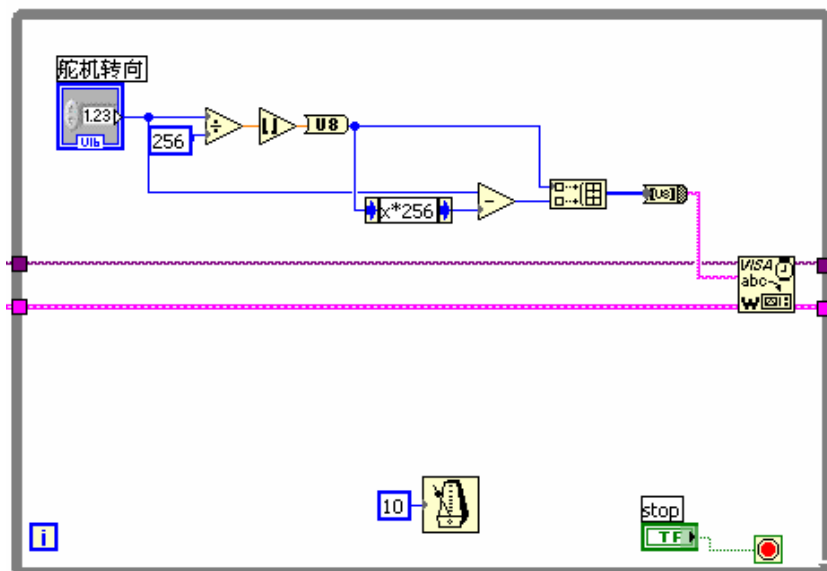


图 6-16 虚拟仪器 Labview 舵机测试

6.3.3 智能车仿真平台

由于最终比赛赛道是不提前公布的,这就给系统可靠性测试提出了更高的要求。一个可靠的系统应该能在各种赛道上稳定运行。但是实际中受到场地时间等的限制,不可能铺设太多赛道进行测试,因此参赛队员们开发了一套智能车仿真平台,以提高测试效率。

这套仿真系统包括离线仿真和在线仿真两大部分。其中离线仿真可以通过 MapCreator.exe 对所需测试的地图进行生成,如图 6-17 所示。生成地图后,打开

图 6-18 所示的离线仿真平台再将地图导入，然后就可以选择相应的控制算法对系统进行仿真了。该平台同时支持光电管方案和摄像头方案的仿真，并且可以实时地显示导引线相对于光电管的位置、舵机方向角命令、当前速度以及行驶时间等主要参数值。值得一提的是，针对某一方案编写控制算法时只需采用 C 语言进行编程即可。这和实际环境中的软件编写是十分接近的，因此代码具有较高的移植性。为了使仿真效果更贴近实际，系统对赛车进行了高阶建模，并加入了侧滑的模拟。在准备比赛的最初阶段，通过离线仿真系统测试了许多不同赛道上赛车行驶的情况。

在线仿真平台在离线仿真的基础上，将数据采集和算法执行直接交给了实际赛车，但控制算法仍只需在 PC 机上用 C 进行编写。其界面如图 6-19 所示。S12 单片机通过 Zigbee 无线通讯模块和 PC 进行通信，将图像和速度信息发送上来，然后 PC 机的控制程序对这些信息进行处理，返回相应的舵机转角和速度控制命令。由于赛车在实际赛道上运行，通过这样的仿真过程能更加逼真地对算法进行测试。

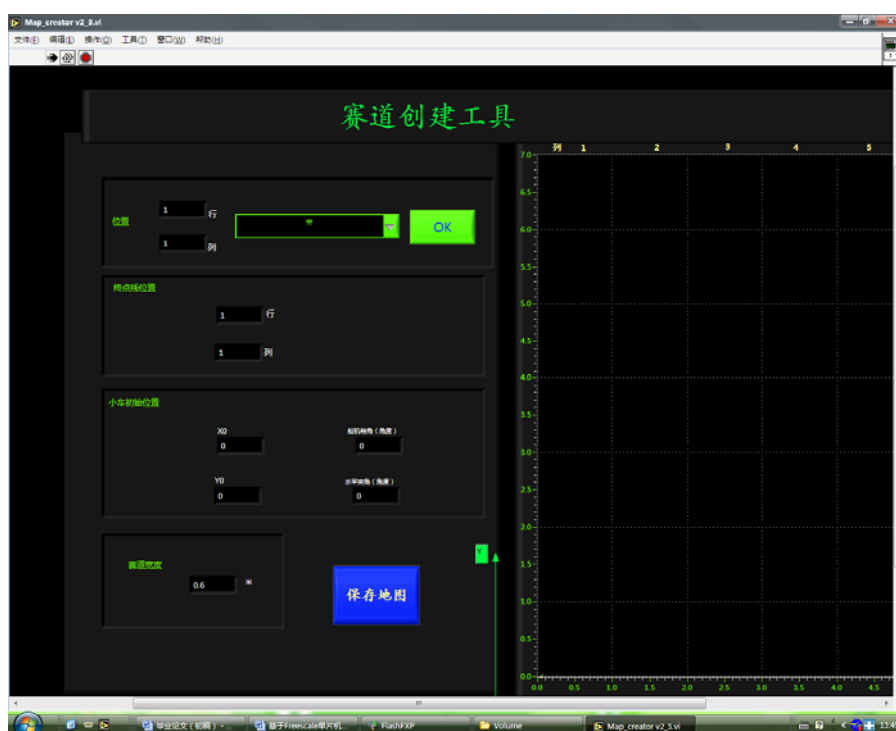


图 6-17 赛道创建工具界面

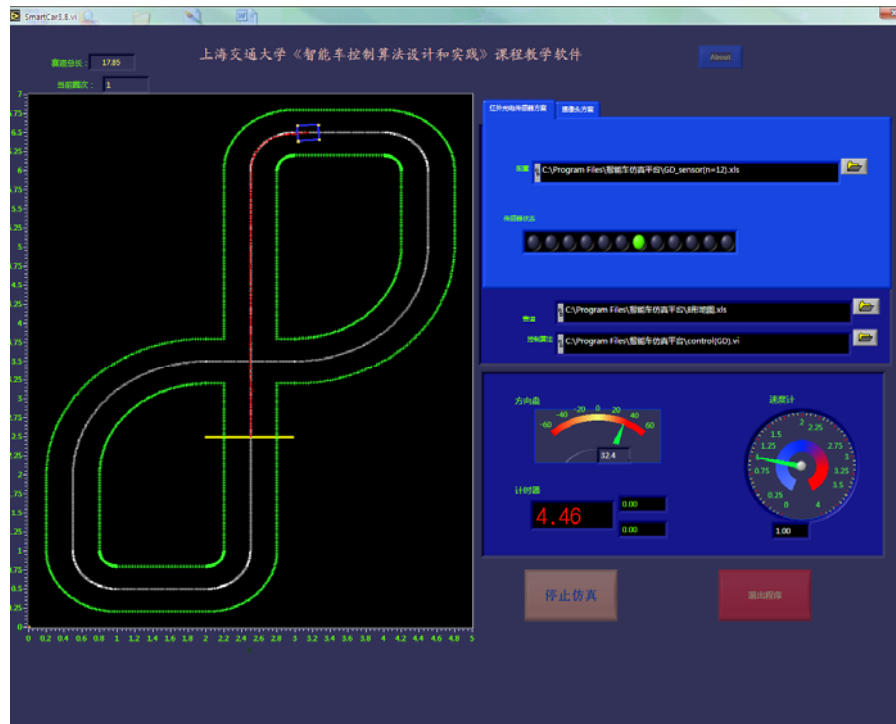


图 6-18 离线仿真平台界面

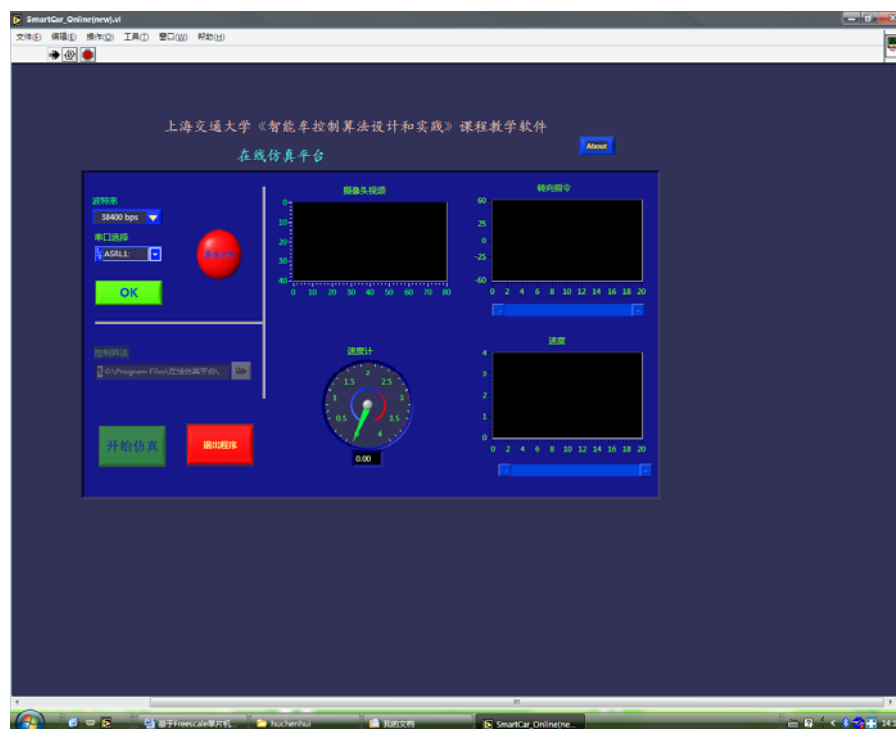


图 6-19 在线仿真平台界面

图 6-20 为在线仿真平台的程序流程图。

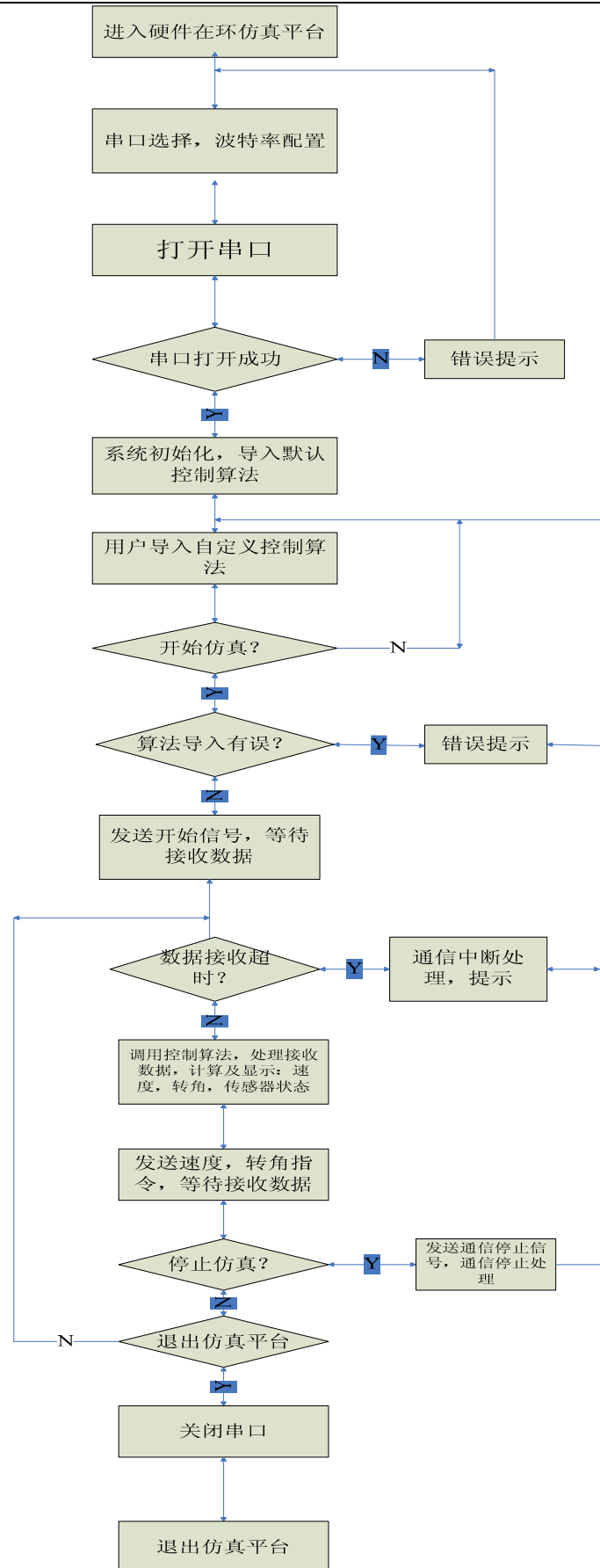


图6-20 在线仿真平台LABVIEW实现程序框图

第七章 结论

本文主要介绍了基于 Freescale 单片机的智能车设计流程、摄像头寻线方案各个主要模块的工作原理和设计思路。回顾整个设计过程，先通过光电管和摄像头方案的对比，选择了摄像头为赛道检测的传感器，然后基于这一方案从总体上确立了硬件和软件的基本框架。接下来分别对系统的机械结构调整、硬件设计和软件设计进行了详细的介绍。

虽然在智能车快速寻线方面可借鉴的经验不多，但是有了去年的一些基础和这段时间的进一步探索，目前系统性能已有较大提高。为了使成绩有所突破，今年尝试了许多新想法，主要体现在以下几个方面：

（1）舵机安装方式

考虑到弯道速度是制约赛车平均速度的主要因素，因此系统采用了和去年不同的舵机安装方式，将舵机安在了更靠后的位置上。这样做的好处是：一、可以使转向更加灵敏，提高过弯速度；二、可以使左右转向的极限相等，简化控制程序。

（2）反向制动

理论上，如果赛车的制动性能越好，直道速度就可以设置得越高，这样能大大提高平均速度。为此设计时，采用两片 33886 使电机实现了正反转，这样高速入弯时适当采用反向制动即可平稳过弯。

（3）速度控制

速度控制是智能车比赛的关键所在，如何使赛车速度响应迅速、平稳是设计的焦点。本次比赛采用了精度更高的速度编码器，同时结合 Bang-Bang 控制和 PID 控制各自的优点做了大量测试工作，最终使速度控制达到了满意的效果。

在比赛准备过程中，小组成员们对赛车各个模块进行了测试、改进。但由于受知识水平和时间上的限制，本方案还有一些不足之处，例如：需要改进控制策略使赛车在高速行驶时舵机仍能够转向平滑；速度控制目前只是简单分级，如能结合记忆算法将速度级别进一步细分，效果应该会更好；除此以外，机械振动的消除、系统稳定性和适应性的提高也需要下一步做更多的工作。

谢辞

今年能将顺利参加第二届“飞思卡尔”全国大学生智能车邀请赛，首先要感谢电信学院杨明老师和机械与动力工程学院王冰老师的悉心指导。两位老师多次询问研究进程，并为我们指点迷津，帮助我们开拓研究思路，精心点拨、热忱鼓励。其严谨的治学态度、渊博的学术知识、诲人不倦的敬业精神以及宽容的待人风范使我们获益颇多。

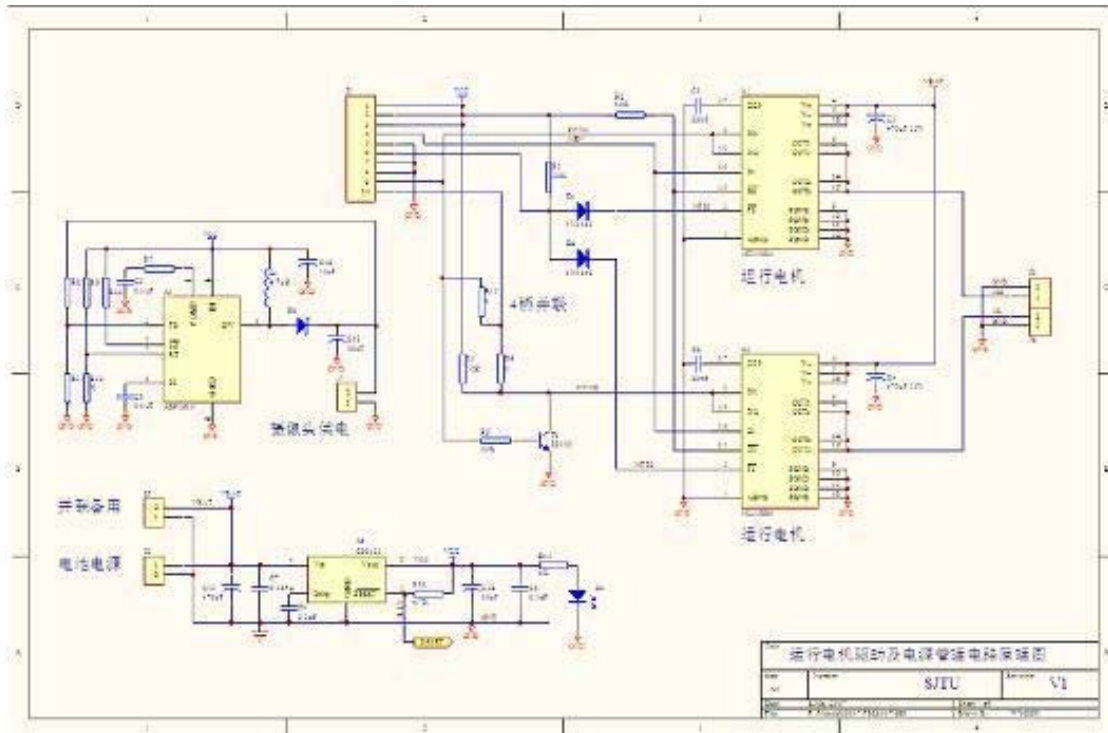
在比赛准备期间，实验室的贾秀江、关军学长尽管已经毕业，但仍然为本次比赛积极提供帮助和建议，这里要特别感谢他们。感谢一起准备比赛的队员方兴、郭思明、陆佳南、马明煌、黄铎佳、贺嘉良、陈立刚，没有他们的配合和帮助，不可能完成工作。除此以外，也要感谢实验室的其他同学和前辈，他们为我们提供了许多便利。

参考文献

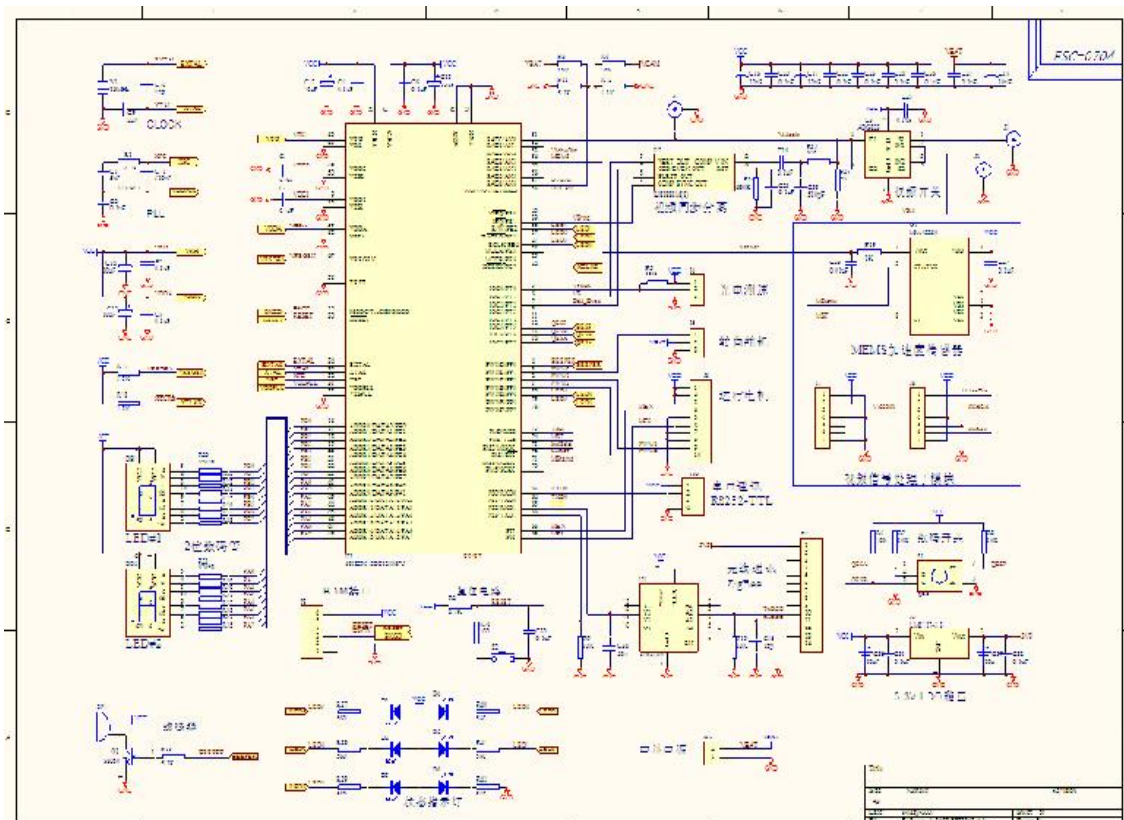
- [1] <http://www.zgsz.org.cn/view.asp?num=zgsz-20056211551>.
- [2] 鲍晓东, 张仙妮. 智能交通系统的现状及发展. 交通论坛, 2006 年 8 月: 15-18.
- [3] <http://www.smartcar.org.cn/web/rule.asp?Area=0>.
- [4] 黄开胜, 金华民, 蒋狄南. 韩国智能模型车技术方案分析. 电子产品世界, 2006(5): 150-152.
- [5] 许欢, 唐竞新. 调制光寻线系统及其在自动行走机器人中的应用. 清华大学学报 (自然科学版), 2002: 第 42 卷, 第 1 期.
- [6] 万永伦, 丁杰雄. 一种机器人寻线控制系统. 电子科技大学学报, 2003: 第 32 卷, 第 1 期.
- [7] 张晓飞, 袁祥辉. 基于 DSP 成像系统的视频图像采集部分的实现. 压电与声光, 2002(6): 第 24 卷, 第 3 期.
- [8] 卓晴, 王璿, 王磊. 基于面阵 CCD 的赛道参数检测方法. 电子产品世界, 2006(4): 141-143.
- [9] 关军, 吴中仪, 胡晨晖. 上海交通大学 CyberSmart 队技术报告.
- [10] 贾秀江, 李颢, 戴丰, 田兴华. 上海交通大学 SpeedStar 队技术报告.
- [11] 邵贝贝. 单片机嵌入式应用的在线开发方法. 清华大学出版社, 2007: 158-159.

附录

附录A speedstar电路板原理图



附录B 核心电路板原理图



附录C EEPROM程序代码

```
#ifndef _ee_prom_h
#define _ee_prom_h

#define EE_BASE ((volatile unsigned int*)(0x0800))

unsigned char ee_write(int relative_addr, int data)
{
    volatile unsigned int* ee_addr = EE_BASE;
    ee_addr += relative_addr;

    ECLKDIV = 0x4b;
    while(!(ECLKDIV&0x80));
    while(!(ESTAT&0x80));
    while(!(EPROT&0x80));
    *ee_addr = data;
    ECMD = 0x20;
    ESTAT |= 0x80;
    while(!(ESTAT&0x40));
    return 0;
}

int ee_read(int relative_addr)
{
    volatile unsigned int* ee_addr = EE_BASE;
    return(*(ee_addr+relative_addr));
}

#endif
```

附录D 数码开关程序代码

```
#define QESB      PTT_PTT6
#define FLAG_A    TFLG1_C7F
#define FLAG_P    TFLG1_C5F

#define LED1      PTP_PTP5
#define LED2      PTP_PTP4
#define LED3      PORTE_BIT4
#define LED4      PORTE_BIT3
#define LED5      PORTE_BIT2
```

```
char temp=0;           //the number display on the board
char state=0;          //cancle the modify mode
char quit=0;
char LED=1;
char LEDvalue[5];      //存储设定数据的数组

void ini_port() {
    /* set PORT A and B */
    DDRA=0xff;          // define port A as output
    DDRB=0xff;          // define port B as output
    PORTA=0;
    PORTB=0;

    /* set PORT T */
    DDRT=0x00;          // set PORT T as input
    TIOS=0x00;          //the corresponding channel acts as input capture
    TSCR1_TEN=1;        //Timer Enable
    TCTL3=0b01001000;   //capture rising edge on PT7 and falling edge on PT5

    /* set PORT E */
    DDRE=0xff;

    /* set PORT P */
    DDRP=0xff;
}

void turnoffLED(void){
    LED1=1;
    LED2=1;
    LED3=1;
    LED4=1;
    LED5=1;
}

char convert_digital(char n){           //convert digital for dis_digital
    char result;
    switch(n)
    {
        case 0: result=0xc0;
            break;
        case 1: result=0xF9;
            break;
        case 2: result=0xA4;
            break;
```

```
case 3: result=0xB0;
        break;
case 4: result=0x99;
        break;
case 5: result=0x92;
        break;
case 6: result=0x82;
        break;
case 7: result=0xF8;
        break;
case 8: result=0x80;
        break;
case 9: result=0x90;
        break;
default:result=0x00;
        break;
}
return result;
}

void dis_digital(char number){
char H,L;
L=number%10;
H=(number-L)/10;
PORTA=convert_digital(H);
PORTB=convert_digital(L);
}

/*****

*****/

void dis_LED(char n){
switch(n)
{
case 1: turnoffLED();
        LED1=0;
        break;
case 2: turnoffLED();
        LED2=0;
        break;
case 3: turnoffLED();
        LED3=0;
        break;
case 4: turnoffLED();
        LED4=0;
```

```
        break;
    case 5: turnoffLED();
        LED5=0;
        break;
    default:
        break;
}
}
```

```
void modifynumber(unsigned char a) {
    int i=a*5;
    dis_digital(temp);
    while(state==0){
        while((FLAG_A!=1)&(FLAG_P!=1)){ };
        if(FLAG_A==1){
            if(QESB==0) {
                i++;
                FLAG_A=1;
            }
            else if(i>0){
                i--;
                FLAG_A=1;
            }
            if((i/5)!=temp) {
                temp=i/5;
                dis_digital(temp);
            }
        }
        if(FLAG_P==1){
            FLAG_P=1;
            state=1;
        }
    }
}
```

```
void modifyLED(void) {
    int i=0;
    dis_LED(LED);
    dis_digital(LEDvalue[LED-1]);
    while(state==0){
        while((FLAG_A!=1)&(FLAG_P!=1)){ };
        if(FLAG_A==1){
            if(QESB==0&i<27) {
                i++;
            }
        }
    }
}
```

```
        FLAG_A=1;
    }
    else if(i>0){
        i--;
        FLAG_A=1;
    }
    if((i/5)!=LED) {
        LED=i/5;
        dis_LED(LED);
        dis_digital(LEDvalue[LED-1]);
    }
}
if(FLAG_P==1){
    FLAG_P=1;
    state=1;
}
}
}

void operate(void){
    char i;
    for(i=0;i<5;i++) {
        LEDvalue[i]=0;
    }
    ini_port();
    while(quit==0){
        modifyLED();
        state=0;
        if(LED>1){
            temp=LEDvalue[LED-1];
            modifynumber(LEDvalue[LED-1]);
            if(state==1){
                LEDvalue[LED-1]=temp;
            }
            state=0;
        }
        if(LED==1) {
            quit=1;
        }
    }
    dis_digital(99);
}
```