

**Design of electromagnetic navigation smart car system and
PIDNN control algorithm applied research**

A Dissertation Submitted to
Nanjing University of Technology
in partial fulfillment of the requirements
for the degree of
Master of Engineering

By
ZHOU Yue

Supervisor: Associate Prof. SHEN Jie

April 2012


Y2400612

摘要

汽车工业已经有 100 多年的发展历史，从 20 世纪 80 年代以来，智能控制理论被越来越多地应用到交通运输领域，各类汽车电子、汽车生产企业以及许多大学研究所都将智能车辆作为重点研究项目。作为当下汽车工业的研究热点以及国家大力发展的智能交通系统中的一部分内容，本课题正是契合此研究方向，以小型实验车模作为初步研究对象，并选用磁导航方式作为车模前进的道路检测手段，通过检测铺设于跑道上的磁信号，由控制算法实现车模自动完成跑道行驶的功能。

本文介绍了智能车自动控制系统的结构以及工作原理，其中包括机械基础和自动控制器两大部分。机械基础方面，简单介绍了实验车模的机械参数及调校方法。自动控制器可分为硬件电路和软件控制两方面，硬件电路方面，以飞思卡尔的 MC9S12XS128 主控芯片为基础，设计了电源管理电路，舵机、电机、速度传感器和 SD 卡的驱动电路，以及从磁场检测原理，磁场检测方法选取，感应线圈放置和检测电路设计四部分详细阐述了如何设计电磁传感器。软件控制方面，主要包括数据存储和控制算法两部分。数据存储部分，介绍了 SD 卡的通信原理、文件系统管理机制；控制算法部分，详细分析了基于 PID 神经元网络（PIDNN）的控制策略。并在这两部分基础上，完成了主控制程序、SD 卡通信程序、PID 神经元网络算法等编写，以及在 Plastid2 软件上进行仿真，比较了 PID 控制与 PID 神经元网络算法在典型跑道上的行驶路径。最后以 MATLAB GUI 作为智能车调试平台，通过读取 SD 卡中存储的实验车模实际行驶数据，绘制出路径偏差曲线，行驶车速曲线，以及舵机转角曲线。调试结果表明该智能车系统，能够正确识别跑道，顺利完成行驶。

关键词：智能车 电磁 SD 卡 文件系统 PID 神经元网络

Abstract

The automotive industry has a history of 100 years of development. Since the 1980s, the theory of intelligent control is increasingly applied to the field of transportation. All kinds of automotive electronics, automobile manufacturing companies, and many university or institute see intelligent vehicle as a key research project. As the research focus of the current auto industry, as well as a part of the Intelligent Transportation System, the subject precisely fit the direction of this study. A small experimental car model is used for a preliminary study, and use magnetic navigation as car models the way forward means of detection. Laying by detecting the magnetic signals on the runway, the control algorithm achieves car model auto-complete runway traveling.

This article describes the structure and working principle of the smart car automatic control system, including mechanical basis and automatic controller. Mechanical basis, briefly introduce mechanical parameters of the experimental car model. And automatic controller can be divided into hardware and software. The hardware circuit, on the base of the Freescale MC9S12XS128 master chip, design drive circuits of the power management circuits, servo, motor, speed sensor drive and the SD card. And from four parts of the magnetic field detection principle, the magnetic field detection method selected, place of the induction coils and detection circuit design, describe in detail how to design a electromagnetic sensor. Software control includes data storage and control algorithm. Data storage part, introduce SD card communication theory and file system management mechanism. Control algorithm part, a detailed analysis of control strategy based on PID neural network (PIDNN) .And on the basis of these two parts, complete the master control program, the SD card communication program, PID neural network algorithms program. In the plastid2 software simulation, compare the travel path of PID control and PID neural network algorithm in a typical runway. MATLAB GUI is used for a smart car debugging platform. Through reading the experiment car real driving data storaging in the SD card, map out the path of deviation curve, driving speed curve, and servo rotation curve. The debug results show that the smart car system can correctly identify the runway, and successfully complete the travel.

KEYWORDS: Smart car; Electromagnetic; SD card; File system; PID neural network(PIDNN)

目录

摘 要.....	I
Abstract.....	II
第一章 绪论.....	1
1.1 引言	1
1.2 智能车辆应用领域	2
1.3 智能车辆国内外研究现状	3
1.3.1 国外智能车辆发展现状	3
1.3.2 国内智能车辆发展现状	4
1.4 论文主要研究内容和章节安排	5
第二章 系统总体结构	7
2.1 智能车系统总体设计	7
2.1.1 硬件电路	7
2.1.2 软件控制	8
2.2 实验车模的主要机械参数	8
2.2.1 车模底盘参数	8
2.2.2 前轮外倾角	9
2.2.3 前轮前束	10
2.2.4 后轮差速	10
2.3 本章小结	11
第三章 系统硬件电路设计	12
3.1 硬件电路总体设计架构	12
3.2 MC9S12XS128 及最小系统	12
3.2.1 MC9S12XS128 芯片	12
3.2.2 最小系统	12
3.3 电源管理模块.....	14
3.3.1 5V 稳压电路	15

3.3.2 12V 稳压电路	15
3.3.3 6V 稳压电路	16
3.4 驱动电路	16
3.4.1 舵机驱动电路	16
3.4.2 电机驱动电路	17
3.4.3 速度传感器驱动电路.....	18
3.4.4 SD 卡驱动电路.....	19
3.5 电磁传感器设计	20
3.5.1 磁场检测的基本原理	20
3.5.2 磁场检测方法	21
3.5.3 双水平线圈的检测方案.....	22
3.5.4 硬件电路设计.....	23
3.6 本章小结	20
第四章 数据存储系统	25
4.1 存储系统结构	26
4.1.1 SD 卡简介.....	26
4.1.2 文件系统.....	27
4.2 SD 卡通信	28
4.2.1 SPI 模式	28
4.2.2 SPI 初始化	29
4.2.3 SD 卡读操作.....	30
4.2.4 SD 卡写操作.....	32
4.3 FAT32 文件系统实现.....	34
4.3.1 FAT32 文件系统结构	34
4.3.2 FAT32 文件系统初始化	37
4.3.3 FAT32 文件系统中文件操作	38
4.4 数据传输测试	39
4.5 本章小结.....	40
第五章 系统控制策略	41

5.1	传统 PID 控制.....	41
5.2	PID 神经元网络(PIDNN).....	42
5.2.1	神经元基本模型结构	43
5.2.2	PID 神经元的计算方法	44
5.2.3	PID 神经元网络结构	45
5.2.4	PIDNN 前向算法	46
5.2.5	PIDNN 反传算法	47
5.2.6	PIDNN 连接权重初值选取	48
5.2.7	PIDNN 收敛性和稳定性	49
5.3	PID 与 PIDNN 阶跃响应测试.....	49
5.4	控制策略设计	52
5.4.1	转向控制	53
5.4.2	速度控制	54
5.4.3	控制策略	57
5.4.4	仿真	58
5.5	本章小结.....	61
第六章	系统调试界面设计	62
6.1	MATLAB GUI 简介	62
6.2	系统调试	63
6.3	本章小结	65
第七章	总结与展望	66
参考文献	67
附录： 设计成果	71
攻读硕士学位期间发表论文与获奖情况	77
致 谢	78

第一章 绪论

1.1 引言

随着社会的快速发展，汽车普及造成了交通供需矛盾的日益严重，道路交通安全形势日趋恶化。近年来，因交通事故造成的伤亡人数直线攀升，但往往专家在分析交通事故的时候，会更加侧重于人与道路的因素，而对车辆性能的提高并不十分关注。如果存在一种高性能的汽车，它可以自动发现前方障碍物，并能自动避让；在陌生地区可以自动导航引路；甚至在特殊情况下能够自动驾驶，那将会使道路交通的安全性大大得到提高。实际上这种车辆，随着科技水平的不断提高，已经不再是梦想，完全可以通过机动车辆的智能化来实现，这种智能化后的车辆通常被称之为智能车辆。智能车辆是一个集环境感知、规划决策、多级辅助驾驶等功能于一体的综合系统，它集中地运用了计算机、现代传感、信息融合、通讯、人工智能及自动控制等技术。智能车辆致力于提高汽车的安全性、舒适性和提供优良的人车交互界面，是目前各国重点发展的智能交通系统中一个重要组成部分，也是世界车辆工程领域研究的热点和汽车工业增长的新动力^[1,2]。

一般情况下，我们对于车辆的驾驶行为可以看作是一个多输入多输出，输入与输出关系复杂多变，并且多干扰源的非线性系统的控制过程。首先，驾驶员需要从视觉上观察到道路，车辆，行人等环境信息，其次，还要把握住驾驶的车辆的车速，转向等车辆信息，之后通过大脑的综合分析，确定出应该对车辆实际的操作。因而，可以看出，智能车辆需要必须借助一些传感器来获取自身的位置信息或外部引导信息，继而通过运动控制系统地分析、决策实现车辆的自我导航行驶。

目前，智能车辆所应用的导航技术主要包括：机器视觉、引导磁钉或引导电缆、惯性导航系统、全球定位系统（GPS）^[3]。磁性导航技术源自 AGVS——Auto Guided Vehicle System 的地下埋线导航方式中的磁性导航技术。它通过在车道上埋设磁性标志来给车辆提供车道的边界信息，磁性材料具有良好的环境适应性，它对雨天、冰雪覆盖、光照不足甚至无光照的情况都可适应，不足之处是需要对现行的道路设施做出较大的改动，成本较高；同时磁性导航技术无法预知车道前方的障碍，因而不可能单独使用；另外，磁性参照物所设定的路线固定，可变性差。

美国和日本在 20 世纪 90 年代中期, 提出在道路中间铺设磁块的方式来进行导航, 并于 1996 及 1997 年在公路上做过实验。美国加州进行的 Path Program 项目即是在高速公路上, 采用安装磁性铁钉来实现道路边界的标识。另外, 美国明尼苏达州的高速公路自动扫雪车也采用磁性参照物的导航方式。目前在工业生产中广泛应用的 AGVS 大都采用这种磁性导航的导航方式。

智能车辆的控制方法有神经网络控制、模糊控制、专家控制、分层递阶智能控制、拟人智能控制、集成智能控制等, 在实际应用中通常采用几种智能控制方法融合在一起, 或是将智能控制和传统控制有机结合而构成的控制方法。在智能控制系统具体设计时的方法则有许多种, 如: 分级递阶智能控制, 实时智能控制, 仿人智能控制, 模糊逻辑控制, 神经网络控制, 自适应模糊控制, 自适应模糊神经网络控制, 遗传优化的模糊网络控制, 基于遗传的自适应模糊逻辑控制, 模糊变结构控制, 神经网络变结构控制, 模糊神经网络变结构控制, 模糊学习控制, 神经网络学习控制, 神经网络遗传学习控制, 模糊专家控制, 神经网络专家控制, 模糊神经网络专家控制等。

本课题是以小型电磁导航式实验车模作为研究对象, 实验车模转向机构采用的是前轮转向后轮驱动的运动方式, 在控制方法方面, 选择传统 PID 控制与神经网络相结合的智能控制方法, 实现分析、决策功能。

1.2 智能车辆应用领域

智能车辆的应用目前主要集中在军事领域^[4]。在 1983 年, 美国国防预先研究计划局就为了发展一种新一代的侦查和作战系统, 提出了自主地面车辆(智能车辆)的研究开发计划, 从而掀起了智能车辆研究的热潮。在此之后的 20 年里, 美国政府有出台了许多相关的后续计划, 包括: 联合机器人计划、Demo II 和 Demo III 计划、战术移动机器人计划、越野机动感知计划、无人地面战斗车辆计划、美国陆军地面机器人技术发展和实验计划等。并在 2003 年 5 月 19 日, 正式启动了世界范围内有史以来最大的一次军事装备计划——未来作战系统(future combat system, FCS)计划, 其投资规模超过 1400 亿美元。随着计划的进行, FCS 最终将发展成由自主平台和无人平台组成的完全无人的作战系统。

1990 年, 智能交通系统(intelligent transport system, ITS)计划被正式提出, 智能车辆技术迅速拓展到民用交通领域。智能车辆致力于提高车辆的安全性、智能化、舒适性以及友好的人车交互界面, 其智能化集中体现在智能驾驶方

面。智能车辆通常可以完成一项或多项驾驶任务，如道路识别和跟踪、障碍物识别和避障、车辆状态估计、车辆检测和跟踪、车辆横向和纵向控制、巡航控制、超车控制等。智能车辆被认为是实现 ITS 人-车-路一体化的关键技术，在改善公路交通安全状况、提高运输效率和环保节能等方面发挥着关键作用。有专家预测，2010 年以后，大多数高级轿车将装配以自动驾驶系统为核心的汽车主动安全系统。智能车辆同时还代表着未来汽车的发展方向，目前，美国、日本、欧洲等竞相开发实用的智能车辆技术，力图抢占这一巨大市场。此外，智能车辆在危险环境（如核、生化等）、采矿、自动化仓库、自动化车间等领域也有广泛应用，其另一个重要的应用是行星探测，如众所周知的“火星探索者”、“机遇号”和“勇气号”等，这一应用具有巨大的战略意义和潜在的经济价值。

1.3 智能车辆国内外研究现状

1.3.1 国外智能车辆发展现状

国外智能车辆的研究开始于 20 世纪 70 年代末，主要集中在美国、日本及欧洲的几个发达国家。世界上第一台智能车辆于 1979 年在日本研制成功，该智能车辆装有两台摄像机及专用的信号处理系统和控制系统，测试时速达 30km/h。20 世纪 80 年代以来，计算机科学、控制科学与工程、光学、仪器科学、力学、通信等技术的发展为智能车辆的研发提供了技术基础，世界发达国家对智能车辆展开了卓有成效的研发工作，特别是进入 20 世纪 90 年代，对智能车辆的研究进入了深入、系统、大规模的阶段，众多的智能车辆被开发出来，其中具有代表性的是美国、德国、意大利和英国等国家的研究成果。

到目前为止，美国是对智能车辆投入最多、研究最为深入的国家。1984 年 9 月，DARPA 与陆军合作，进行了一项投资额为 6 亿美元的智能车辆战略计划，每隔两年双方进行一次研究成果综合演示，其目的就是要研制出一辆能自主导航的车辆，该项任务于 1989 年由 Martin Marietta 公司负责完成，它使用路标识别导航技术，以 10km/h 的车速在较平坦的越野环境中自主行驶了 20km，实现了智能车辆自主导航功能。

DARPA 的介入大大推动了智能车辆中有关视觉和体系结构的研究进展，使一些高校和研究机构在智能车辆研究领域中脱颖而出，如 Martin Marietta 公司、卡内基梅隆大学、FMC 公司、马里兰大学、密歇根州立大学等。20 世纪 90 年代，美国在该领域主要集中在战场 2000 计划中无人地面车辆（unmanned ground

vehicle, UGA) 系统的研究。2002 年 9 月 UGA 在公路上行驶的最高速度达 64km/h, 在无明显植被覆盖的越野环境中最高速度可达到 32km/h, 夜间最高速度可达 16km/h; 同时, UGA 具有战场侦察能力, 在车辆静止时侦察范围为 2km; UGA 具有多平台交互与协同能力, 如 DemoIII 集中了各种传感器信息——CCD、激光、雷达、超声、红外、微波等。

欧洲的陆地移动机器人研究具有鲜明的特点。德国国防军大学研制的高速公路自动驾驶车 VaMoRs 在 1987 年创下 96km/h 的最高速度, 最近达到了 150km/h 以上, 可检测到 700m 远的卡车及 340m 远的小汽车。欧盟制订的发展欧洲信息技术的 ESPRIT 计划也将智能车辆列入重要研究内容。英国、德国、法国和荷兰等国正在合作研究旨在解决三维视觉关键技术的 Vidimus 计划。意大利的 Parma 大学研制的 AGRO 自主车, 只采用摄像机作为传感器; 利用立体视觉来检测道路前方障碍物, 利用单个黑白摄像机来检测道路方向。1998 年 7 月, AGRO 在意大利的高速公路上进行了 2000km 的试验, 其中包括平原、山地和隧道等多种地形, 大约 94% 的路程是自主驾驶的。



图 1-1 卡耐基—梅隆大学的 Navlab 系列



Figure 1-1 Navlab Series of Carnegie - Mellon University

Figure 1-2 the ARGO autonomous vehicle of Parma University of Italy

除欧美外, 日本、加拿大、韩国、泰国及新加波等国也已投入一定的人力和物力开展智能车辆的研究, 并取得了一些研究成果。

1.3.2 国内智能车辆发展现状

我国对智能车辆的研究相对较晚, 主要技术成果是 7B.8 军用智能机器人平台以及清华大学研制的 THMR-III 和 THMR-V 自主车。

我国的第一辆智能车是在“八五”期间由包括北京理工大学在内的国内六所重点大学联合研制的样车 ALVLAB I, 即 7B.8 军用智能机器人平台, 其总体性能

达到了当时国际先进水平。清华大学智能技术与系统国家实验室移动机器人课题组研制了 THMR-III 移动机器人，它的体系结构以垂直式为主，采用多层次感知-动作行为控制和基于模糊控制的局部路径规划及导航控制，在自主道路跟踪时，其速度达到 5-10km/h，避障达 5km/h。在“九五”期间由南京理工大学、国防科技大学、浙江大学、清华大学和北京理工大学继续研究了军用智能车辆 ALVLAB II，全部实验在北京某装甲兵实验厂完成，道路环境下自主行驶最高速度达 15km/h，同时支持临场遥感驾驶及战场侦察等功能。THMR-V 系统是清华大学智能移动机器人课题组于“九五”期间自主研制开发的新一代多功能室外智能移动机器人试验平台，可以应用于高速公路和一般路面。

中国科学院自动化研究所于 2003 年研制出的移动机器人，其基本结构包括传感器、控制器和运动结构，传感器由位于机器人底层的 16 个触觉红外传感器、位于机器人中间两层的 16 个超声传感器和 16 个红外传感器、位于机器人顶部的摄像机等组成。通过多传感器信息融合，能够实时完成运动控制决策、躲避障碍物、寻找最优路径、实现自主移动、定点运动及轨迹跟踪等。另外，浙江大学、中国科学院电子学研究所、中国科学院沈阳自动化研究所、国防科技大学和南京理工大学等各大院校和研究所都对该领域进行了相关的研究，并取得了一定的研究成果。



图 1-3 国防科技大学 CITAVT-IV 自主车

Figure 1-3 CITAVT-IV autonomous vehicle of National Defense University
Figure 1-4 Autonomous driving car of the red flag



图 1-4 红旗自动驾驶轿车

目前，我国正在研究第三代陆地自主车智能车辆 ALVLABIII，除了在“九五”的基础上提高其二维视觉处理功能模块的性能外，又提出了在越野环境下的三维立体视觉障碍物检测的要求。

1.4 论文主要研究内容和章节安排

作为汽车工业的研究热点，智能车辆有着广阔的应用前景，伴随各类传感器的高速发展，车辆自主导航技术已日趋成熟。而本课题选用电磁导航方式控制实验车模完成规定路径的行驶，实验车模利用电磁传感器和速度检测传感器将地面引导线与车模的相对位置信息和车速信息传递给 MC9S12XS128 主控芯片，后者通过 PID 神经元网络算法对舵机及电机输出 PWM 波调节控制，保证车模能够稳定行驶。并在行驶过程中将行驶数据存储至 SD 卡，以便于在调试平台中分析车模行驶状态，对于实际调试工作，具有一定意义。

论文根据智能车辆从环境感知到行为控制过程的特点，并结合实际调试工作，将整个系统的设计与实现分为七章，具体内容组织如下：

第一章——绪论。概述国内外智能车辆的研究背景和发展状况。

第二章——系统总体结构。介绍了实验车模总体系统的组成结构和各模块功能，以及实验车模的主要机械参数。

第三章——系统硬件电路设计。主要包括 MC9S12XS128 的最小系统，电源管理模块，舵机与电机驱动电路以及电磁传感器检测电路等内容。

第四章——数据存储系统。阐述了基于 SD 卡的数据存储系统的组织构成，其主要分为 SD 卡与主控芯片的数据通信以及文件系统的实现。

第五章——系统控制策略。针对实验车模电机调速的 PID 传统控制策略进行优化，结合 PID 控制算法与神经网络学习算法，构成 PID 神经元网络控制策略。

第六章——系统调试界面设计。利用 MATLAB GUI 工具箱设计车模调试界面，通过分析 SD 卡采集的数据，反映出车模行驶过程中存在的问题。

第七章——总结和展望。通过对本文的总结，对下一步工作进行展望。

第二章 系统总体结构

2.1 智能车系统总体设计

作为能够自动识别道路行驶的智能车辆，实验车模与自动控制器可以看成一个自动控制系统。实验车模是这个自动控制系统的机械基础，自动控制器则是核心设计环节。自动控制器的构成分为硬件电路和软件控制两大部分。其中，以主控芯片为核心，配有传感器，执行机构以及驱动电路构成自动控制器的硬件装置；信息处理与控制算法由运行在主控芯片中的控制软件完成，控制车模高速稳定地运行于跑道上。具体功能划分如下：

2.1.1 硬件电路

硬件电路作为自动控制器的基础，在保证车模可靠运行的前提下，电路设计应尽量简洁紧凑，以减轻系统负载，提高车模的灵活性，并以运行稳定性为首要前提^[15,28,30-32]。包括以下四个部分：

(1) 主控芯片 MC9S12XS128 最小系统。可采用已设计好的单片机开发板，也可以自行设计。

(2) 电源管理模块。电源模块为系统其它各硬件电路提供所需电源，整个硬件电路的电源由 7.2V, 2A/h 的可充电镍镉蓄电池提供。由于硬件电路中不同电路模块所需的工作电压和电流容量各不相同，因而电源模块应该包含多个稳压电路，将充电电池电压转换成各硬件电路所需电压。设计时，除了考虑电压范围和电流容量等基本参数，还需在电源转换效率，降低噪声，电路简洁等方面进行优化。

(3) 道路检测电路。检测跑道与车模的相对偏移量，方向，曲率等信息是实现车模自主沿着跑道行驶的信息基础。道路检测方式有很多种，如摄像头检测，红外发射/接受管检测等，本课题选用电磁检测方式，以车体中线为对称轴，布置两个电磁检测单元呈一字型分布于车模前端两侧，通过将车体与引导线的相对位移转化为两个电磁检测单元之间的电压偏差，从而实现道路识别功能。

(4) 驱动电路。驱动电路作为主控芯片与外围设备连接的桥梁，设计时应尽量简洁清晰，驱动电路输出通常包含电源、接地以及信号端口。针对实验车模需要设计了舵机、电机驱动电路，速度检测传感器驱动电路，及 SD 卡接口电路。

2.1.2 软件控制

(1) 信息处理。自动控制器通过检测车模行驶时的各种信息完成自动控制，保证其沿着跑道稳定、快速行驶，这些信息包括车模与跑道之间的位置偏移，车模行驶速度等。对于检测到的信息进行处理，主要包括两方面，一是去除信息中的噪声干扰。检测信息中存在一定的噪声，检测数据中突变的值可以视为噪声，会影响车模行驶的稳定性，通过软件滤波方式，去除这些值可以消除或减小噪声干扰。二是保存有价值的数据信息。通过将数据存储于 SD 卡中，能够方便调试工作。

(2) 控制算法。自动控制系统可以看作一个车模跟随跑道变化而行驶的随动系统，控制车模行驶的参数包括前轮舵机转角与后轮电机转速。由于建立车模运动学与动力学模型分析过于复杂，传统的控制策略是分别对两者进行 PID 控制调节，本课题对于舵机控制沿用 PID 控制，而电机控制采用 PID 与神经网络相结合的 PID 神经元网络控制算法，利用网络的学习能力，能够更好地调节车速。

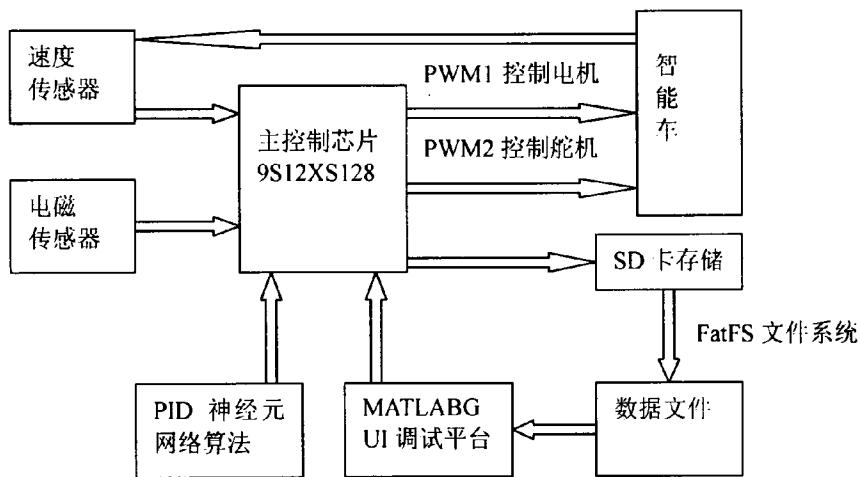


图 2-1 智能车系统模块图
Figure 2-1 Intelligent vehicle system function block diagram

2.2 实验车模的主要机械参数

2.2.1 车模底盘参数

车体：车架长 28.75cm，宽 16.6cm，高 7cm，底盘采用 2.5mm 厚的黑色玻璃纤维板，具有较强的弹性和刚性。前轮调整方式简单，全车滚珠轴承。前后轮轴高度可调(离地间隙 0.75cm/1.65cm)，双滚珠差速。

马达(电机)：DC7.2V RS-380 马达，转速 20000r/min，内装散热风扇。

伺服器(舵机):S-D5 数码伺服器, 工作电压 6V, 带堵转保护电路, 力矩 5.0kg, 回中更准确, 定位更精确, 动作速度 $\leq 0.14 \pm 0.02\text{sec}/60^\circ$ 。

轮胎: 轮胎直径 6.4cm, 前轮宽 2.7cm, 后轮宽 3.7cm, 优质高粘度橡胶发泡材料制作, 坚固耐撞, 摩擦力大, 加强了车子的抓地力。

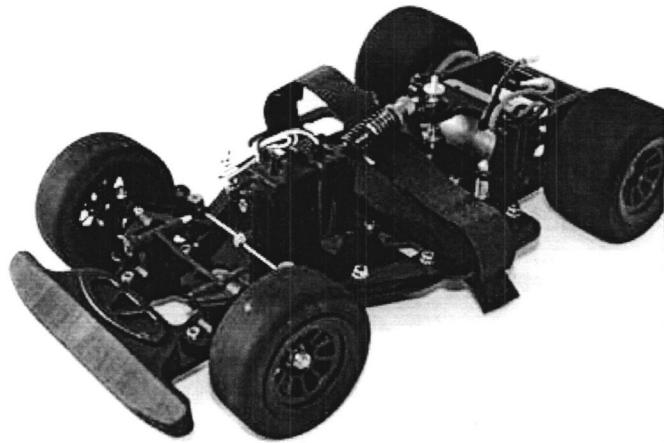


图 2-2 实验车模底盘
Figure 2-2 Experimental Cars Chassis

2.2.2 前轮外倾角

通过车轮中心的汽车横向平面与车轮平面的交线与地面垂线之间的夹角 α , 称为“前轮外倾角”, 如图 2-3 所示。前轮外倾角一方面可以在汽车重载时减小或者消除主销与衬套、轮与轴承等处的装配间隙, 使车轮接近垂直路面滚动而滑动, 同时减小转向阻力, 使汽车转向轻便; 另一方面还可以防止由于路面对车轮垂直反作用力的轴向分力对外端轴承的压迫, 减少轴承及其锁紧螺母的载荷, 从而增加这部分零件的使用寿命, 提高汽车的安全性。一般前轮外倾角为 1 度左右。而现代汽车高速化、急转向等工况要求前轮外倾角减小甚至为负值。模型车可调整前轮外倾角。实验中发现: 由于智能车主要用于竞速, 在设计中必然要求尽可能减轻重量, 所以底盘承重不大, 而且前轮外倾角只有两档可调, 故设定为 0 度即可, 关键是前轮前束要与之匹配。

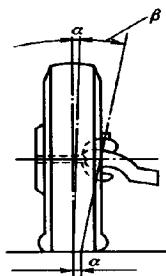


图 2-3 前轮外倾角
Figure 2-3 Front wheel camber angle

2.2.3 前轮前束

当车轮有了外倾角后，滚动就是类似于圆锥滚动，从而导致两侧车轮向外滚开。出于转向横拉杆和车桥的约束使车轮不可能向外滚开，车轮将在地面上出现边滚动边滑移的现象，从而增加轮胎的磨损。在安装车轮时为消除车轮外倾带来的这种不良后果，可以使汽车两前轮的中心面不平行，并使两轮前边缘距离小于后边缘距离，距离之差称为“车轮前束”，如图 2-4 所示。内八字样的前端小后端大的称为“前束”，而外八字样后端小前端大的称为“后束”或者“负前束”。模型车是由舵机带动左右横拉杆实现转向的。主销在垂直方向的位置确定后，改变左右横拉杆的长度可以改变前轮前束的大小。由于前轮前束值须与前轮外倾角匹配，而前面将前轮外倾角设定为 0 度，这里将前束设定为 0 度。

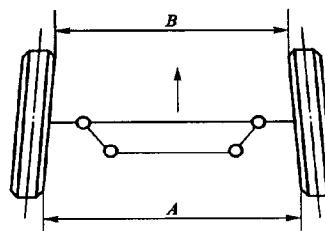


图 2-4 前轮前束
Figure 2-4 Front wheel toe-in

2.2.4 后轮差速

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。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。

2.3 本章小结

本章主要介绍了智能车系统的总体设计组成，展开阐述了硬件电路与软件控制两大控制部分，并简单介绍了本文使用智能车车模的机械参数和调校方法。

第三章 系统硬件电路设计

3.1 硬件电路总体设计架构

主控芯片选用 IC 生产厂商 Freescale 公司 MC9S12 系列 16 位处理芯片 MC9S12XS128，基于此主控芯片，设计实验车模应用硬件电路，硬件电路总体结构如图 3-1 所示，包括六部分：7.2V 电源管理模块，电磁传感器检测电路，速度传感器（编码器）驱动电路，数字舵机驱动电路，直流电机驱动电路以及 SD 卡驱动电路。

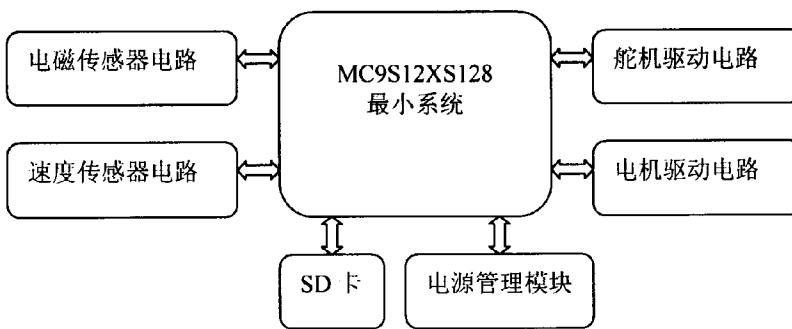


图 3-1 硬件电路总体架构
Figure 3-1 Architecture of hardware circuit

3.2 MC9S12XS128 及其最小系统

3.2.1 MC9S12XS128 芯片

MC9S12XS128 是飞思卡尔公司 MC9S12X 系列 16 位单片机，基于 S12 CPU 内核，总线频率达 40M（超频可到 80M），该芯片有 112 引脚 LQFP 封装，80 引脚 QFP 和 64 引脚 LQFP 封装三种选择，本课题选用 112 引脚 LQFP 封装，其外设集包括 MSCAN，SPI，2 个 SCI 口，1 个 8 通道 24 位周期中断定时器，8 通道 16 位定时器，8 通道 PWM 以及 16 通道 12 位 ATD 转换器。内部包含 8KB 的 RAM 可用于缓存，与多数单片机的仿真调试方式（JTAG）不同，MC9S12X 具有背景调试模块（BDM），BDM 能在单片机运行时对单片机动态调试。

3.2.2 最小系统

硬件电路的搭建采用核心板加扩展板的设计思路，要求核心板尽可能的简洁以及轻巧，所以选用主控芯片的最小应用系统，最小系统包括时钟电路，电源电路，复位电路，串口 RS-232 驱动电路，LED 调试显示以及 BDM 接口电路。各个部

分功能如下：

(1) 时钟电路：时钟电路对主控芯片的运行至关重要，硬件电路虽然简单，但如果出现时钟电路不稳定的情况，时钟电路的高频噪声会引起系统电磁兼容性下降，成为系统隐患。本课题使用的是标准的时钟电路，将 16MHz 的外部石英晶振接在主控芯片的外部晶振输入接口 EXTAL 和 XTAL 上，利用主控芯片内部的压控振荡器和锁相环将此频率提高到 40MHz，作为芯片工作的内部总线时钟。

(2) 电源电路：主控芯片内部使用 3V 电压，I/O 端口和外部供电电压为 5V，进开发板的电源要经过电容滤波去除高频干扰后才能使用。为显示系统已经通电，需要设计指示灯电路。

(3) 复位电路：通过一个复位按键给主控芯片一个复位信号，主控芯片复位是低电平有效，调试过程中非常有用。

(4) 串口 RS-232 驱动电路：RS-232 电平采用负逻辑，即逻辑 1：-3V 至-5V，逻辑 0：+3V 至+15V。RS-232 适用于设备之间的通信距离不大于 15m，传输速率最大为 20KB/s 的数据传输领域。主控芯片通过串口驱动电路中 RS-232 电平转换芯片 MAX232 将 TTL 电平转换成 RS-232 电平，之后通过 9 芯串行电缆和 PC 机进行通信。

(5) LED 调试显示：将调试小灯与 PORTB 口相连，调试按键与 PORTA 口相连。最小系统上 1 位 LED 接到 PK7 口上，可以简单用作信号指示。

(6) BDM 接口电路：单片机片内 Flash 技术在 MCU 上的应用，使得单片机开发脱离了仿真器，在单片机开发与应用上给用户提供了很大的方便。BDM 调试器是用来向目标板下载程序，可将单片机 Flash 中的旧程序擦除，这是 BDM 的编程功能。此外，BDM 接口可以提供很多调试信息，包括 CPU 运行时的动态信息。主控芯片需用一个 6 针插头将信号引出与 BDM 调试器连接，可实现单片机的 BDM 调试。

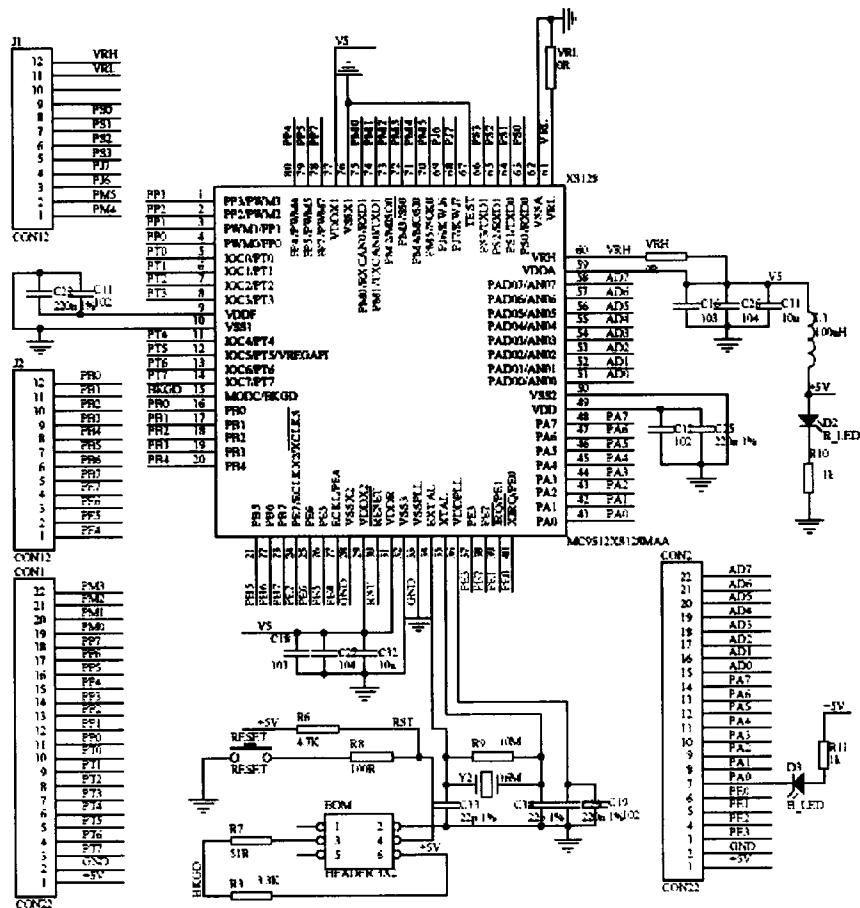


图 3-2 最小系统电路
Figure 3-2 Minimum System circuit

3.3 电源管理模块

实验车模的硬件电路架构是一个较为复杂，需要协调配合的系统，涉及的外围设备较多，所需的供电电源值不一致，因而需要一套较完善的电源管理模块，通过升降压处理，转换成外围设备所需的电压，从而保证整个系统稳定运行。设计电源管理模块时，需要考虑的因素主要包括输入电压，输出电压，电流以及消耗功率。实验车模的电源是 7.2V 镍镉电池，以此为基础，电源管理模块划分如图 3-3 所示。

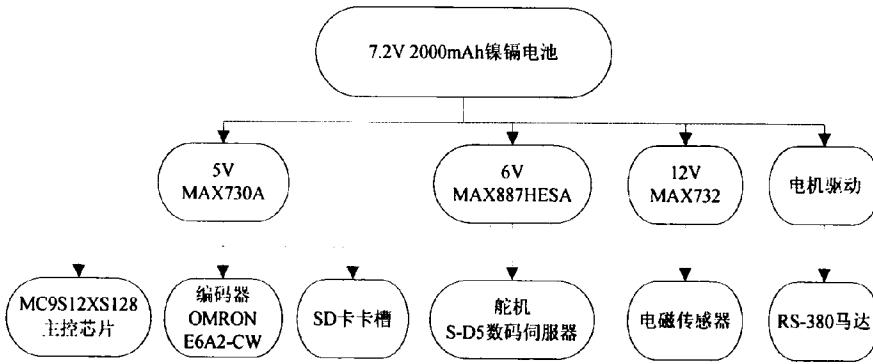


图 3-3 电源模块图
Figure 3-7 Power source block diagram

3.3.1 5V 稳压电路

MAX730A 是美信公司的电流式 5V 稳压芯片，SOP-8 封装，体积较小，输入电压范围 5.2V—11V，输出电压为 $5V \pm 0.25V$ ，优点是压降小，开关频率高，输出续流电感只需小电感即可。通过将 7.2V 电压转换成 5V 电压，以供主控芯片，速度传感器以及 SD 卡卡槽所用。设计电路图如 3-4 所示。

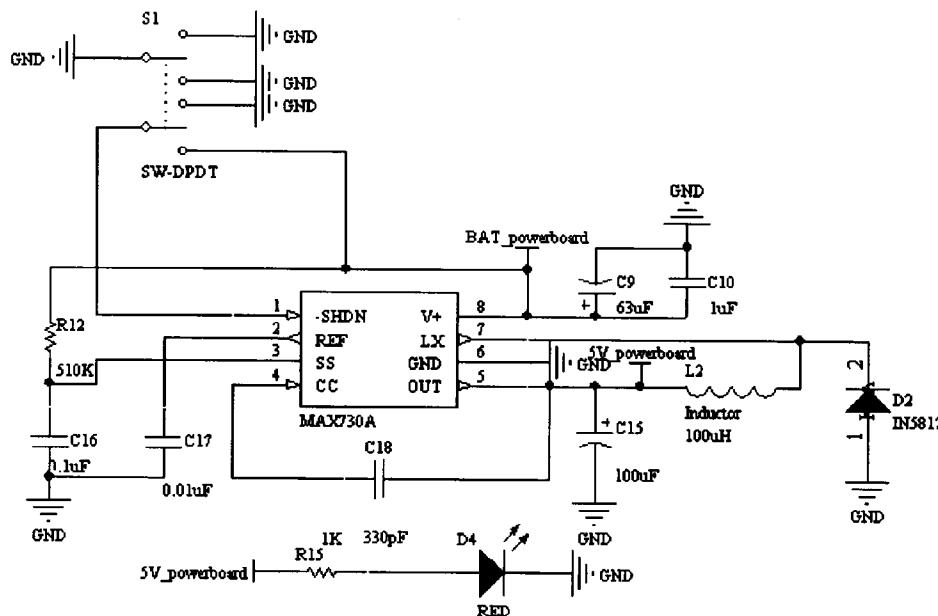


图 3-4 5V 电源电路
Figure 3-4 5V power supply circuit

3.3.2 12V 稳压电路

如图 3-5，12V 稳压电路选用的是 MAX732 升压电路，同样是 SOP-8 封装，输入电压范围 4.0V—9.2V，输出电压 12V，输出范围 11.52V—12.48V。12V 电压用于电磁传感器检测电路，稳定的电压能够保证检测的精确度与灵敏度。

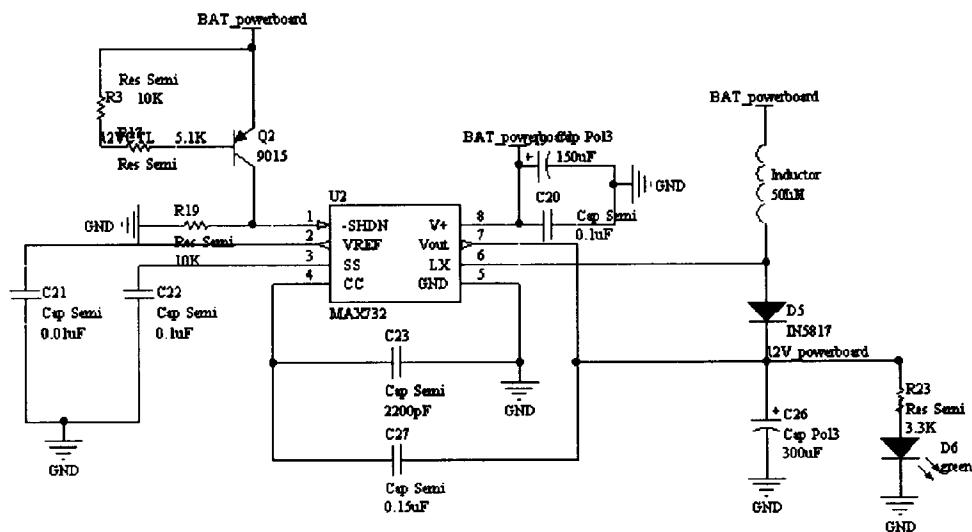


图 3-5 12V 电源电路
Figure 3-5 12V power supply circuit

3.3.3 6V 稳压电路

MAX887HESA 稳压芯片提供 6V 稳定电压，实验车模使用的数字舵机是 S-D5 型，它的额定工作电压是 6V，舵机可以工作于 6V 以上 9V 以下的电压下，虽然较大的电压能提高舵机反应的灵敏度，但同时也会使舵机在工作时过于发热甚至烧毁舵机，因而选用 6V 电压供电。电路搭建见图 3-6。

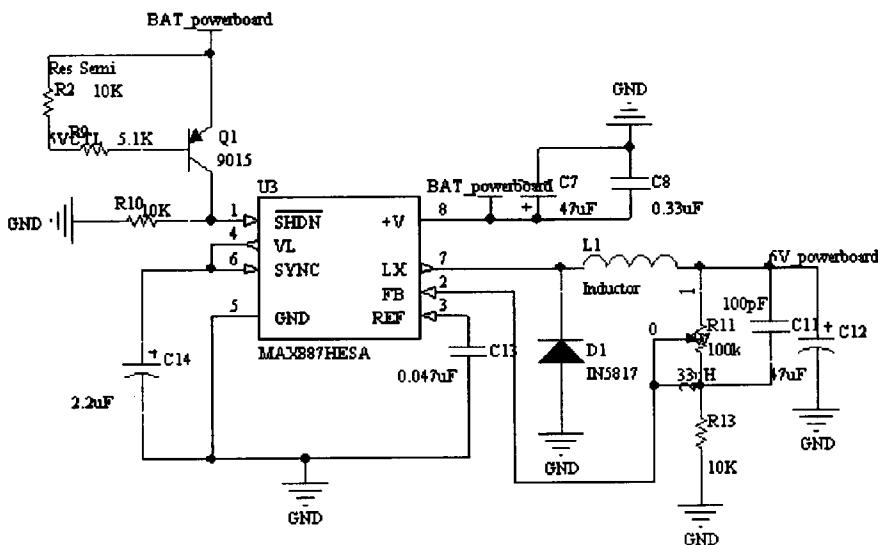


图 3-6 6V 电源电路
Figure 3-6 6V power supply circuit

3.4 驱动电路

3.4.1 舵机驱动电路

舵机最早出现在航模运动中，飞行机的飞行姿态是通过调节发动机和各个控制舵面来实现的，舵机即是控制舵面的伺服电机。标准舵机有3个引脚，分别是电源线，地线，控制线。如图3-7，将控制线连接于PWM口，以实现PWM波对车模转向角度的控制。舵机的规格主要视扭力与速度而定，扭力的单位是kg/cm，指摆臂长度1厘米处所能吊起的物重。速度的单位是sec/60°，指转动60°所需要的时间。

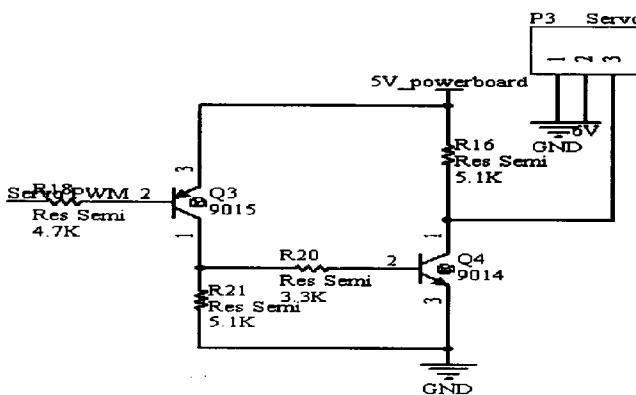


图 3-7 舵机驱动电路
Figure 3-7 Servo drive circuit

3.4.2 电机驱动电路

电机驱动电路选用的驱动芯片是飞思卡尔半导体公司的 H 桥式驱动器 MC33886，MC33886 工作电压为 5-40V，电阻为 120 毫欧姆。MC33886 主要面向小功率直流电机的驱动，最大能够提供连续 5A 的驱动电流，通过 PWM 控制时，频率可以达到 10KHz。相比于自己搭建的 H 桥电路，集成驱动芯片的好处是具有错误状态警告、短路保护、欠压保护、过温保护、欠压关闭等功能。如图 3-8 就是 MC33886 的内部电路模块。

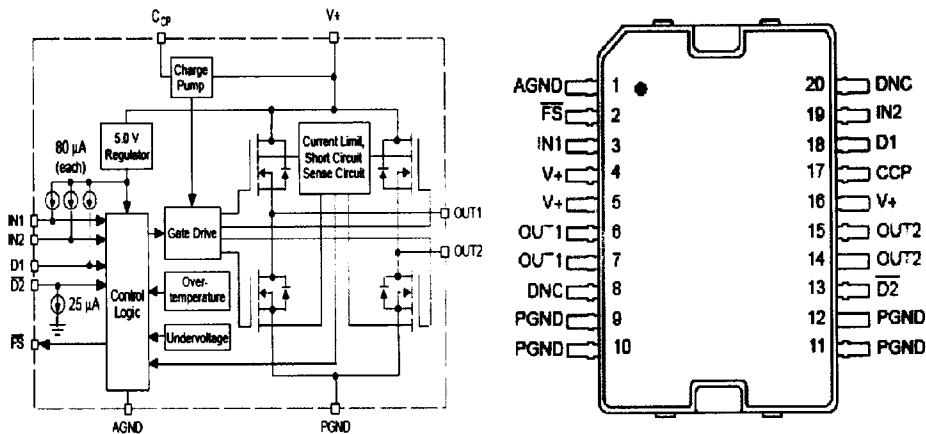
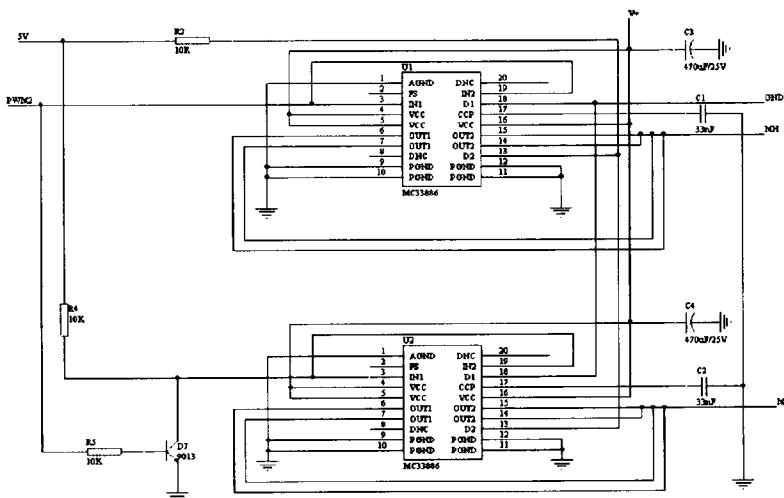


图 3-8 MC33886 内部电路模块

Figure 3-8 MC33886 circuit modules

考虑到车模的紧急制动问题，所以必要的时候需要对电机进行反向制动，本驱动电路使用两片 MC33886 芯片并联的方式。工作原理是同一 PWM 信号输入时，一路 PWM 信号直接传输到一片 MC33886，另外一路 PWM 信号先通过三极管反向再传输到另一片 MC33886。设计目的是使两片驱动片平均分担，并且两个芯片都是同一方向驱动，避免电机换向驱动形成过大的电流，造成芯片发热，具体的连接图如图 3-9 所示。

图 3-9 MC33886 电机驱动电路
Figure 3-9 MC33886 motor driver circuit

3.4.3 速度传感器驱动电路

为保证实验车模能够平稳地沿着跑道行驶，除了控制前轮舵机转向之外，还需要调控车速，使车模在过大弯道时不至于速度过快而冲出跑道，通过调节后轮直流电机的平均电压可以限制车速。开环控制电机的转速会受很多因素影响，如电池电压，电机传动摩擦力，道路摩擦力以及前轮转向角度等，这些因素都会造

成车模行驶不够稳定。因此，通过速度检测装置，对车速进行闭环负反馈控制，能够消除上述因素的影响，使得车模行驶更加稳定。

在车轮没有打滑的情况下，车速正比于电机的转速，通过检测电机转速来实现车速的检测。车速检测方法较多，如测速发电机、转角编码盘、反射式光电检测、透射式光电检测以及霍尔传感器检测等。本课题选用OMRON公司生产的E6A2-CW型光电编码器，与后轮轴齿轮相连，输出的脉冲信号通过I/O口输入到主控芯片的计数器模块中进行测量。

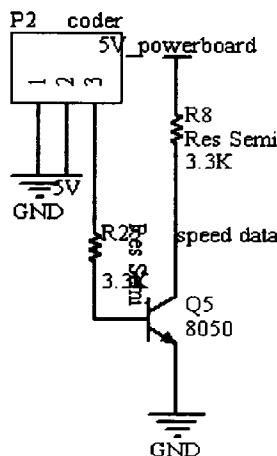


图 3-10 光电编码器驱动电路
Figure 3-10 Encoder circuit

3.4.4 SD 卡驱动电路

根据 SD 卡数据手册可知，SD 卡使用 3.3V 稳压电源供电，故 SD 卡数据端口电平应该是 3.3V 的 CMOS 电平，但是由于本课题中选用的主控芯片 MC9S12XS128 普遍采用 5V 稳压电源供电，故 I/O 口输出也为 5V CMOS 电平，根据 CMOS 电平定义，3.3V CMOS 电平输入到 5V CMOS 器件中识别时没有问题，而 5V CMOS 高电平输入到 3.3V CMOS 的器件中将会导致元器件灌电流过大而烧毁，一般电平转换可以考虑使用专门的电平转换芯片，也可以考虑使用三极管电路进行电平转换，但是由于本课题设计中需要的 SPI 速率较高，而使用一般三极管通带频率要求显然无法达到，故放弃三极管电平方案。并且选用电平转换芯片不仅增大模块体积，而且元器件成本很高。由 MC9S12XS128 技术资料知，该型号单片机拉电流不大，故只要在 SD 卡的输入端接一个上拉电阻即实现适当分流，保证 SD 卡输入端的灌电流不大，一般上拉电阻选择 4.7K 到 10K，本设计中选用 10K 的上拉电阻。此设计方案不仅降低了成本，同时还能够保证 SPI

接口在较高传输速率下数据波形不变形从而保证了数据传输时的正确率，同时根据 SD 卡协议，在 SD 卡使用 SPI 模式进行通信时，引脚需使用 10K-100K 电阻上拉，正好符合上述分流的目的。

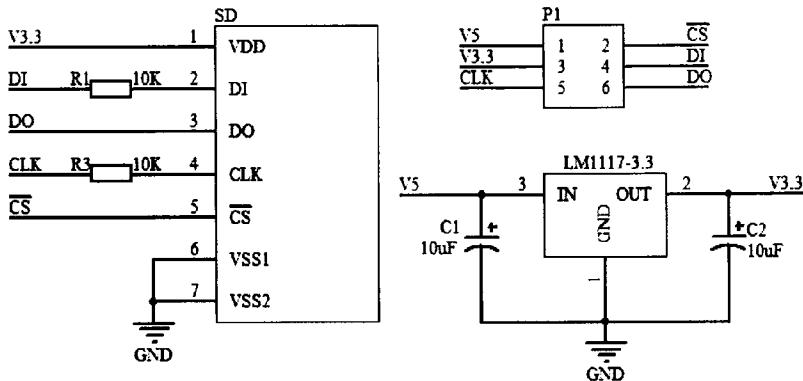


图 3-12 SD 卡驱动电路
Figure 3-12 SD card driver circuit

3.5 电磁传感器设计

3.5.1 磁场检测的基本原理

实验车模跑道是一条通以 100mA，频率为 20kHz 交变电流的闭合导线，根据麦克斯韦电磁场理论，如图 3-13，交变电流会在周围产生交变的电磁场，产生的电磁波属于甚低频，频率介于工频和低频电磁波之间（3kHz–30kHz）。通过检测相应的电磁场强度和方向，能够反向推导出检测点与导线中心线的空间距离，正是利用此原理，通过检测导线磁场强度确定车模与导线的相对位置，从而进行电磁导航^[32,36]。

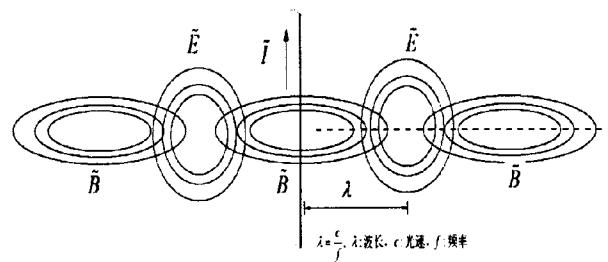


图 3-13 电流周围磁场

Figure 3-13 The magnetic field around the current

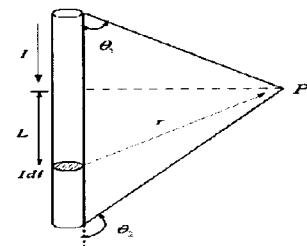


图 3-14 直线稳恒电流

Figure 3-14 Linear constant current

磁场的测量包括直流稳恒磁场和交变磁场两类。由于道路导航电线和实验车模的尺寸都远远小于电磁波波长 λ ($\lambda = \frac{c}{f}$, c : 光速, f : 频率)，电磁场辐射

能量很小，所以将导线周围变化的磁场近似看成缓慢变化的磁场，用检测静态磁场的方法获取导线周围的磁场分布，从而确定实验车模的行驶位置。

如图 3-14, 由毕奥-萨法尔定律知：通有稳恒电流 I 长度为 L 的直导线周围会产生磁场，感应磁场的分布是以导线为轴的一系列同心圆，同一个圆上的磁场强度大小相同，并随距离导线的半径 r 增加成反比下降。距离导线 r 处的 P 点磁感应强度为：

$$B = \int_0^{\theta_2} \frac{\mu_0 I}{4\pi r} \sin \theta d\theta \quad (\mu_0 = 4\pi \times 10^{-7} TmA^{-1}) \quad (3-1)$$

$$\text{求得: } B = \frac{\mu_0 I}{4\pi r} (\cos \theta_1 - \cos \theta_2) \quad (3-2)$$

3.5.2 磁场检测方法

磁场传感器是利用物质与磁场之间的各种物理效应来检测磁场强度和方向，很多磁场测量方法在二十世纪初期才得到突飞猛进的进展，如磁电效应（电磁效应、霍尔效应等）、磁机械效应、磁光效应、核磁共振等。而本课题利用电磁效应设计了电磁检测传感器，它具有原理简单、体积小、频率响应快、电路实现简单的特点。

通电导线周围产生的磁场是一个矢量场，如果在通电直导线周围放置两个轴线相互垂直并位于导线垂直面内的线圈，则可以感应到磁场向量的两个垂直分量，进而可以求得磁场的强度与方向。导线中的电流按一定规律变化时，导线周围的磁场也将发生变化，线圈将感应出一定的电动势。感应电动势的方向可以通过楞次定律来确定，而感应电动势的大小根据法拉第定律，线圈磁场传感器的内部感应电压 E 与磁场 $B(t)$ 、线圈匝数 N 、横截面积 A 的关系有：

$$E = NA \cdot \mu_0 \mu_r \cdot \frac{dB(t)}{dt} = -\frac{d\Phi(t)}{dt} \quad (3-3)$$

由于本设计中导线通过的电流频率较低且线圈较小，令线圈中心到导线的距离为 r ，可以认为小范围内的磁场分布是均匀的。导线周围磁场分布规律，则线圈中感应电动势可近似认为：

$$E = -\frac{d\Phi(t)}{dt} = \frac{k}{r} \cdot \frac{dI}{dt} = \frac{K}{r} \quad (3-4)$$

线圈中感应电动势的大小正比于电流的变化率，反比于线圈中心到导线的距离。其中常量 K 为与线圈摆放方法、线圈面积和一些物理常量有关的一个量，具

体的感应电动势常量须实际测定来确定。

3.5.3 双水平线圈的检测方案

不同的线圈轴线摆放方向，可以感应不同的磁场分量。本课题使用一种较简单的线圈分布方案，双水平线圈检测。在实验车模前方水平方向固定两个相距 L 的线圈，两个线圈的轴线水平，高度为 h ，如下图。

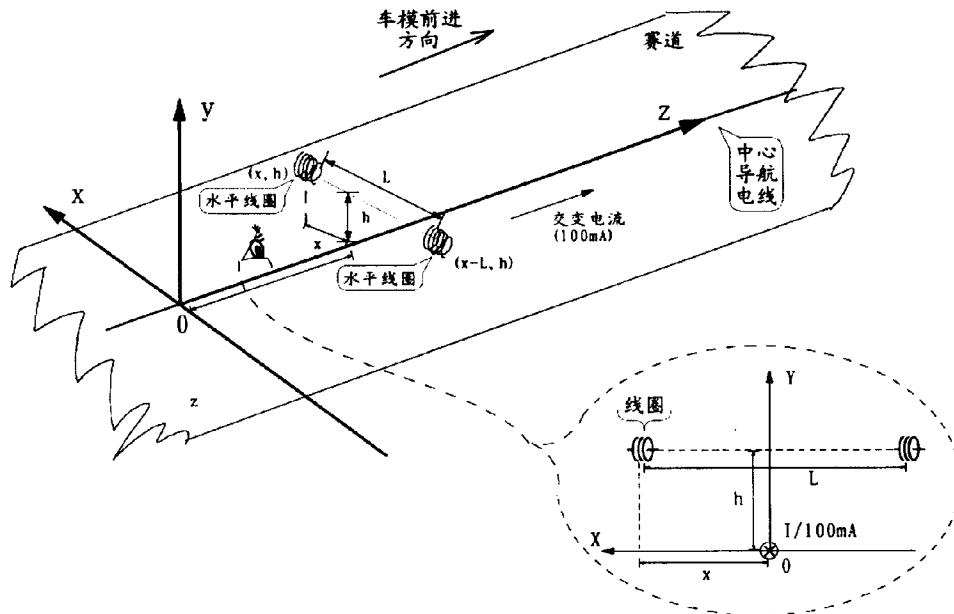
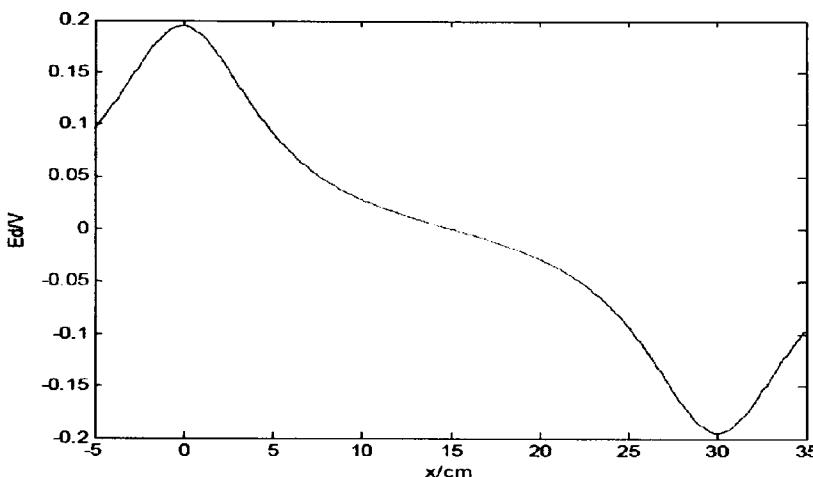


图 3-15 感应线圈分布方案
Figure 3-15 Induction coil distribution program

在跑道上建立如下的坐标系，假设沿着跑道前进方向为 z 轴，垂直跑道往上为 y 轴，跑道平面内垂直于跑道中心线为 x 轴。xyz 轴满足右手方向。左边的线圈坐标为 (x, h, z) ，右边的线圈坐标为 $(x-L, h, z)$ 。由于磁场分布是以 z 轴为中心的同心圆，在计算磁场强度的时候我们仅仅考虑坐标 (x, y) 。由于线圈轴线是水平的，所以感应电动势反映了磁场的水平分量。根据公式 3-4 可以知道感应电动势大小与 $\frac{h}{h^2 + x^2}$ 成正比。

$$E_d = E_1 - E_2 = \frac{h}{h^2 + x^2} - \frac{h}{h^2 + (x-l)^2} \quad (3-5)$$

图 3-16 感应电动势差值 E_d 与距离 x 的函数关系Figure 3-16 Function of induced electromotive force difference E_d and distance x

从图3-16可以看出，当左边线圈的位置x在15cm处，此时两线圈的中心恰好处于跑道中央，感应电动势差值 E_d 为0。当线圈往左偏移， $x \in (15, 30)$ ，感应电动势差值小于零；反之，当线圈往右偏移， $x \in (0, 15)$ ，感应电动势大于零。因此在位移0~30cm之间，电动势差值 E_d 与位移x是一个单调函数。可以使用这个量对于小车转向进行负反馈控制，从而保证两个线圈的中心位置跟踪赛道的中心线。通过改变线圈高度h，线圈之间距离L可以调整位置检测范围以及感应电动势的大小。

3.5.4 硬件电路设计

从上面检测原理可以知道，测量磁场的核心就是检测线圈的感应电动势E的幅值。下面将从感应线圈、信号选频放大、整流与检测等几个方面讨论电路设计的问题，最后给出电路设计系统框图和实际电路。

(1) 感应磁场线圈

检测线圈可以自行绕制，也可以使用市场上能够比较方便购买的工字型100mH的电感。这类电感体积小，Q值高，具有开放的磁芯，可以感应周围交变的磁场。

(2) 信号选频放大

使用电感线圈可以对其周围的交变磁场感应出响应感应电动势。这个感应电动势信号具有以下特点：

信号弱：感应电压只有几十个毫伏。在检测幅值之前必须进行有效的放大，放大倍数一般要大于100倍（40db）。

噪声多：一般环境下，周围存在着不同来源、不同变化频率的磁场。如下表所示：

表3-1 典型环境下的磁场强度范围

Table 3-1 The magnetic field intensity range of typical environment

磁场环境	磁场性质	磁场强度（高斯）
家用电器周围一米范围	50Hz	$10^{-3} - 10^{-2}$
地表面地球磁场	恒定	0.2-0.5
工业电机和电缆周围十米范围	50Hz	1-100
长波通讯	>30kHz	$10^{-6} - 10^{-3}$
跑道中心导线周围0.5米范围	20kHz	$10^{-4} - 10^{-2}$

跑道选择 20kHz 的交变磁场作为路径导航信号，在频谱上可以有效地避开周围其它磁场的干扰，因此信号放大需要进行选频放大，使得20kHz 的信号能够有效的放大，并且去除其它干扰信号的影响。可以使用 LC 串并联电路来实现选频电路，如下图所示：上述电路中，E是感应线圈中的感应电动势，L是感应线圈的电感量， R_0 是电感的内阻，C是并联谐振电容。上述电路谐振频率为：已知感应电动势的频率 $f_0 = 20\text{kHz}$ ，感应线圈电感为 $L = 100\text{mH}$ ，可以计算出谐振电容的容量为：

$$C = \frac{1}{(2\pi f_0)^2 L} = \frac{1}{(2\pi \times 20 \times 10^3)^2 \times 100 \times 10^{-3}} = 6.33 \times 10^{-10} (\text{F}) \quad (3-6)$$

选用最为接近的 680pF 的电容作为谐振电容。

电路示意图

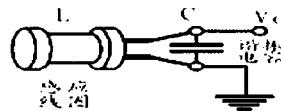


图 3-17 LC 并联谐振电路
Figure 3-17 LC parallel resonant circuit

(3) 整流与检测

采用简单的三极管共射极放大电路，如图3-18所示，为了增加电压输出范围、提高放大倍数，整个传感器采用12V供电，电路分为三级：电压放大级、电流放大级以及后级检波输出级。信号采集部分的电感并联了一个680pF的瓷片电容，目的是让LC电路的谐振频率落在导线中电流频率的附近，增加信噪比。第一极是

基本的共射极反相放大器，发射极电阻R4用来调节放大倍数，以抵消因电感的差异导致的左右电压值的差异。第二极是由Q1和R5组成的射极跟随器，目的是减小输出阻抗，增大驱动力，以便向第三级提供足够的电流。第三级检波电路是将交流信号的振幅大小转换成直流信号的电压高低，以便让微控制器进行AD转换，得到相应的电压。

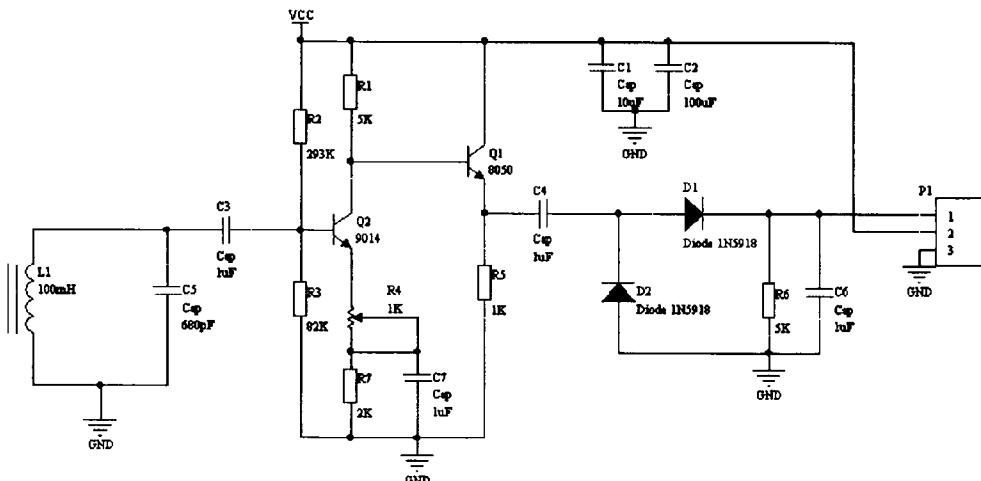


图 3-18 电磁传感器检测电路

3.6 本章小结

本章主要介绍了 MC9S12XS128 芯片及其最小系统，并设计了电源管理模块，将 7.2V 直流电源通过升降压处理转换成 5V, 12V, 6V 电压。并设计了舵机、电机、速度传感器以及 SD 卡驱动电路。最后详细说明了电磁传感器的检测原理，包括磁场检测的基本原理，磁场检测方法的选择，磁场线圈的分布方案，以及硬件电路设计。

第四章 数据存储系统

4.1 存储系统结构

现今对于数据采集系统要求日趋增高，大量的采集数据使得很多采集系统都需要借助外部存储介质进行数据存储，因此对于外部介质存储容量以及传输速度的要求，尤为突显。SD卡作为一种基于半导体闪存工艺的存储卡，1999年由日本松下、东芝及美国SanDisk公司共同研制完成。其体积小、容量大、数据传输快等优点，是众多电子设备理想的外部存储介质。在众多厂商推动下，SD卡已成为目前消费数码设备中应用最广泛的一种存储卡。利用SD卡采集智能车行进过程中跟踪偏差、运动速度、控制参数等信息。MC9S12XS128主控芯片自带SPI接口，SD卡与SPI口相连，可以较方便地进行数据通信，并且SD卡储存空间大，能够保证存储足够多的信息。为使SD卡中采集数据可直接在PC上读取、处理，需建立文件系统的基本文件操作函数，使文件系统组织数据格式，文件系统即操作系统用于明确磁盘或分区上文件的方法和数据结构，是对文件的管理规则，也是操作系统的重要组成部分。而操作系统对处理器要求较高，且占用一定的存储空间，对于许多无法使用嵌入操作系统的处理器，需要自己来实现文件系统——在处理器中内嵌文件系统。

主控芯片通过SPI口与SD卡相接，驱动程序实现两者之间数据通信，采集数据在Fat32文件系统管理下，以文件形式存储于SD卡，如.txt、.dat等。SD卡存储系统结构如图4-1所示：

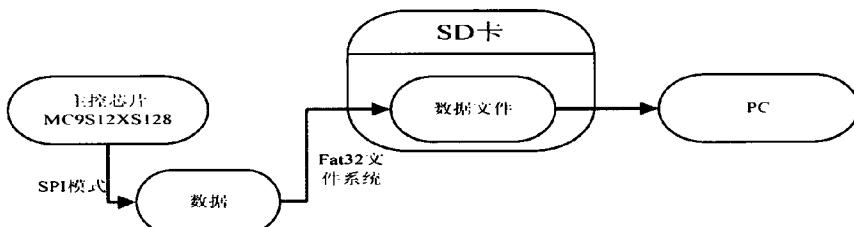


图4-1 存储系统结构
Figure 4-1 storage system architecture

4.1.1 SD 卡简介

SD 卡的技术是基于 MultiMedia 卡 (MMC) 格式上发展而来，大小和 MMC 卡差不多，尺寸为 32mm × 24mm × 2.1mm。SD 卡在 24mm×32mm×2.1mm 的体积内结合了 SanDisk 快闪记忆卡控制与 MLC (Multilevel Cell) 技术和 Toshiba

(东芝) 0.16u 及 0.13u 的 NAND 技术, 通过 9 针的接口界面与专门的驱动器相连接, 不需要额外的电源来保持其上记忆的信息。而且它是一体化固体介质, 没有任何移动部分, 所以不用担心机械运动的损坏。

SD 卡有两种总线模式, 即 SD 总线模式和 SPI 总线模式^[5,16,22,61,65]。SD 总线模式采用 4 条数据线实现并行传输数据, 数据传输速率较高, 但是其传输协议相当复杂, 并且大部分单片机不提供 SD 总线模式接口。若使用软件方法模拟 SD 总线模式又会比较繁琐, 反而容易降低 SD 卡的数据传输速率。SPI 总线模式只需要一条数据线传输数据, 数据传输速率相对较低, 但大部分单片机均提供 SPI 总线模式接口, 故本课题 SD 卡与单片机通讯时选择 SPI 总线模式。SD 卡有 9 只引脚, 工作电压为 3.3V, 其引脚定义如下:

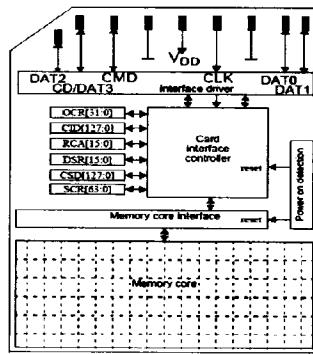


图 4-2 SD 卡引脚图

Figure 4-2 SD card pin

表 4-1 引脚定义
Table 4-1 Pin Definitions

引脚	名称	描述
1	CS	片选 (低通)
2	DataIn	数据输入
3	VSS1	接地
4	VDD	电源供应
5	CLK	时钟
6	VSS2	接地
7	DataOut	数据输出
8	RSV	保留
9	RSV	保留

4.1.2 文件系统

选择使用 SD 卡存储数据，除了速度以及接口上的优势，主要原因还在于 SD 卡作为通用存储设备其数据很容易在 PC 端读取出来，但并不是所有写入 SD 卡的数据都能够顺利通过 PC 恢复，PC 端要能够识别 SD 卡上的文件，首先 SD 卡需要有 PC 端支持的文件系统格式，SD 卡上存储的文件也必须根据文件系统规定的文件格式进行存储，否则写入的数据无法利用 PC 读取。目前常用的文件系统有两类，一类是基于微软 windows 操作系统的 FAT16、FAT32、NTFS、exFAT 文件系统，另一类是 Linux 系统下的 EXT3、EXT4 等。Windows 操作系统使用广泛，且 SD 卡容量较大，因此选用 Fat32 文件。

4.2 SD 卡通信

4.2.1 SPI 模式

SD 卡 SPI 模式下 Command 从 CMD0 到 CMD63，SPI 命令分为 11 个组，各组是多个命令的集合，每组中的命令均有相似的功能。CMD0 命令即 0，CMD1 命令即 1，依次类推^[5]。所有的命令格式均是 6 字节长度，首先发送 MSB。第一个字节为命令，第二到第五个字节为命令参数(有的命令不带命令参数，有的命令带命令参数，但格式均是一样的)，第六个字节为 CRC 校验码。事实上 SD 卡在进入 SPI 模式后，SD 卡不再通过 CRC 码来确认指令的传输正确与否，而 CRC 校检，只在 SD 模式下起作用。因此仅 SD 卡上电后的第 1 条切换 SPI 模式指令 CMD0 需要校检码，而此校检码是固定的 0x95，其他指令的 CRC 均置 1 即可。

SPI 命令分为 11 组，但由于未使用特殊用途，本课题设计中只是频繁使用到了一些基本指令、数据单块读取指令、数据单块写入指令、数据多块写入指令，而其他指令均作为特殊用途使用，此处并未涉及，故仅介绍本设计中频繁使用到的上述四类指令。

表 4-3 中涉及到了命令的应答方式，SPI 总线模式下共有四种命令应答方式，分别为 R1,R1b，R2，R3，由于 R2、R3 应答方式本设计中未使用，故不作具体介绍^[1]。R1b 格式与 R1 相同，只是附加一个可选忙信号，这个忙信号可以是任意数量的字节，一个 ‘0’ 表示 SD 卡处于忙状态，而非 ‘0’ 则表示 SD 卡已经准备好接收下一命令。R1 应答由一个字节组成，其字节内各比特位的意义如图 4-3 所示。

表 4-2 SD 卡 SPI 命令格式
Table 4-2 SD card SPI command format

Byte 1								Bytes 2-5					Byte 6		
7	6	5	0	31	0	7	6	0				
0	1	Command			Command Argument					CRC			1		

表4-3 SD卡SPI总线模式下命令集描述
Table 4-3 SD card SPI bus mode command set description

命令索引	参数	应答方式	简写	描述
CMD0	无	R1	GO_IDLE_STATE	复位 SD 卡
CMD1	无	R1	SEND_OP_CONO	激活 SD 卡的初始化处理
CMD12	无	R1b	STOP_TRANSMISSION	多数数据块操作时强制中止
CMD16	块长度	R1	SET_BLOCK	设定读写的块长度
CMD17	数据地址	R1	READ_SINGLE_BLOCK	从指定地址读取一个数据块
CMD18	数据地址	R1	READ_MULTIPLE_BLOCK	读取多个数据块
CMD24	数据地址	R1	WRITE_SINGLE_BLOCK	向指定地址写入一个数据块
CMD25	数据地址	R1	WRITE_MULTIPLE_BLOCK	写入多个数据块

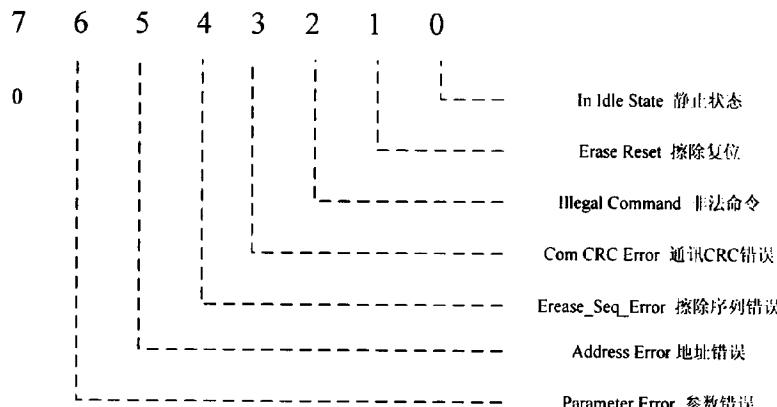


图 4-3 R1 应答格式结构
Figure 4-3 R1 response format structure

4.2.2 SPI 初始化

SPI初始化的主要目的是设置SD卡的数据传输模式以及读写数据块大小(SD卡以块为存储单位，数据块大小为512Byte的整数倍，具体倍数视具体的SD卡而定)。查阅SD卡工作手册，SD卡上电过程需有大于74个时钟周期的等待时间，待

电压平稳后，才可向SD卡发送启动命令。由于SD卡在上电后默认是工作在SD模式，要使用SPI模式进行数据传输则必须先将SD卡设置成为SPI模式，为保证稳定操作，SD卡协议规定对SD卡进行SPI初始化操作时SPI总线频率必须低于400KHz，当然，总线速率越低越能够保证稳定初始化。

待发送完74个时钟周期高电平之后，SD卡默认工作在SD模式状态，然后片选SD卡有效再发送16个时钟周期。SD卡要初始化为SPI模式则是在CS片选为低时发送CMD0命令，收到CMD0命令后若SD卡回复R1指令为0x01（即回复SD卡处于激活状态且发送的命令无错误），此时SD卡已经工作在SPI模式。SD卡初始化时序图见图4-4。当收到CMD0命令的有效应答后，只有再发送CMD1命令激活初始化进程才能够真正完成SD卡初始化进入能够读写的状态。SD卡初始化流程图见图4-5。

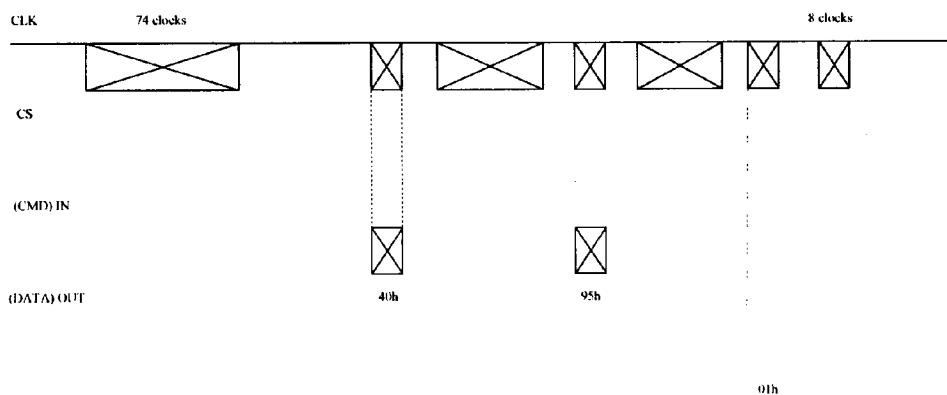


图4-4 SD卡初始化时序图

Figure 4-4 Timing diagram of SD card initialization

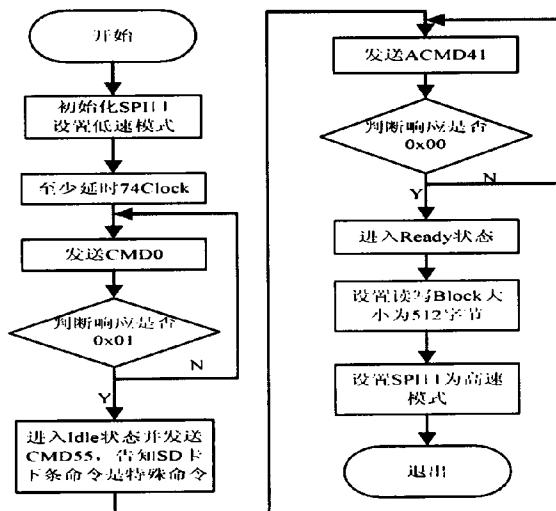


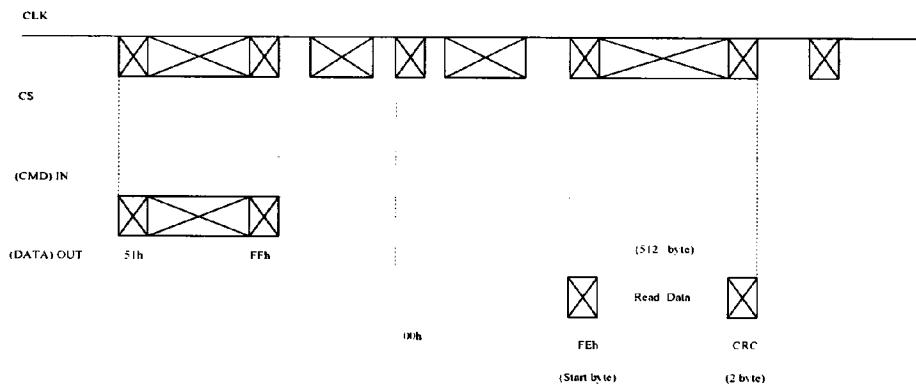
图4-5 SD卡初始化流程图

Figure 4-5 The flow chart of SD card initialization

4.2.3 SD 卡读操作

读/写操作以扇区为单位，扇区容量为512个字节。SD卡的存储单位是块，是扇区的整数倍。初始化完成后，使用CMD16命令设置SD卡读/写块长度(512B)，发送CMD17或CMD18进行SD卡的单块或多块读操作，发送CMD24或CMD25进行SD卡的单块或多块写操作^[11]。

SPI模式可支持单块和多块的读操作(SD卡的CMD17或CMD18)。在接收到一个合法的读命令后，SD卡将用一个应答标记来应答，读操作的开始地址可以是SD卡合法地址范围内的任何字节地址，在数据接收错误时，SD卡将不传送任何数据，而是将一个报告错误数据的标志位传送给主控制器。需要注意，当每次单块或多块读操作结束且片选SD卡无效后，都要再补送8个时钟周期信号，便于SD卡稳定工作。SD卡单块读操作时序图见图4-6，SD卡单块读模块框图如图4-7所示。



4-

图4-6 SD卡单块读时序图

Figure 4-6 SD card single block read timing diagram

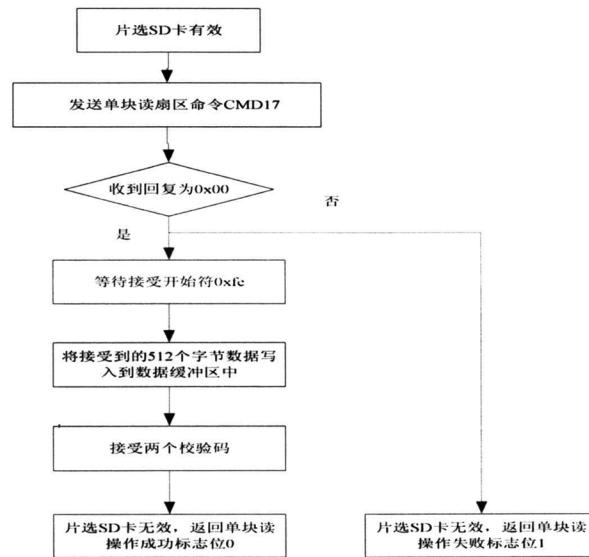


图4-7 SD卡单块读模块

Figure 4-7 SD card single block read module

SD卡多块读操作与SD卡单块读操作时序图相似，多块读命令的响应标志位也是0x00，但传送完毕后需要发送停止命令CMD12，以结束数据传输。在多块数据读取情况下，传输的数据块都包含1个16位的CRC校验码后缀，在SPI模式下CRC虽不被用来校验数据，但每次都要接受两位校验。SD卡多块读操作流程如图4-8所示^[13]。

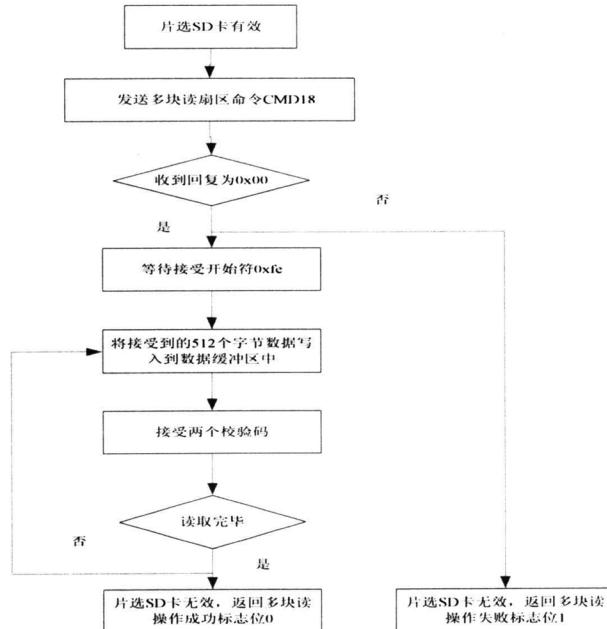


图4-8 SD卡多块读模块

Figure 4-8 SD card multi-block read module

4.2.4 SD 卡写操作

SPI模式下，SD卡支持单块和多块写操作(CMD24, CMD25)。在接受到一个合法的写命令时，SD卡将响应一个应答标记，随后等待主控制器发送数据块^[1]。每个数据块均由一个字节的起始令牌、512个字节数据、两位CRC校验码组成。CRC后缀和起始地址的限制与读操作一致，且块长度必须是512字节，如果设置了小于512字节的块长度，会导致下一条写操作命令发生错误。当主控制器发送完数据块后，SD卡将回应一个应答标记(该字节低五位为00101)，然后SD卡处于烧录编程状态(即始终保持DATAOUT信号线为低电平)，一旦烧录编程完毕，DATAOUT将变成高电平，之后SD卡才能执行其他命令^[14]。SD卡单块写时序图见图4-9。

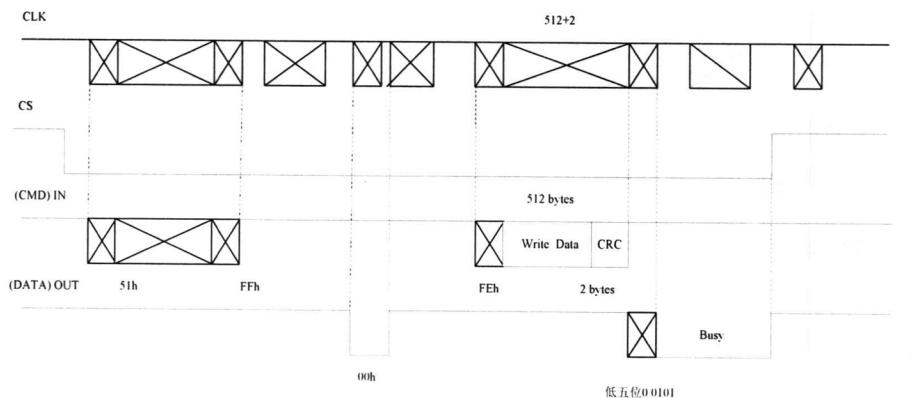


图4-9 SD卡单块写时序图

Figure 4-9 SD card single block write timing diagram

写单块数据需要向SD卡发送写命令CMD24，在收到有效应答信号0x00后，开始发送数据块。该数据块前的起始令牌为0xfe，要与多块写数据时的起始令牌区分开来。SD卡单块写操作流程如图4-10所示。

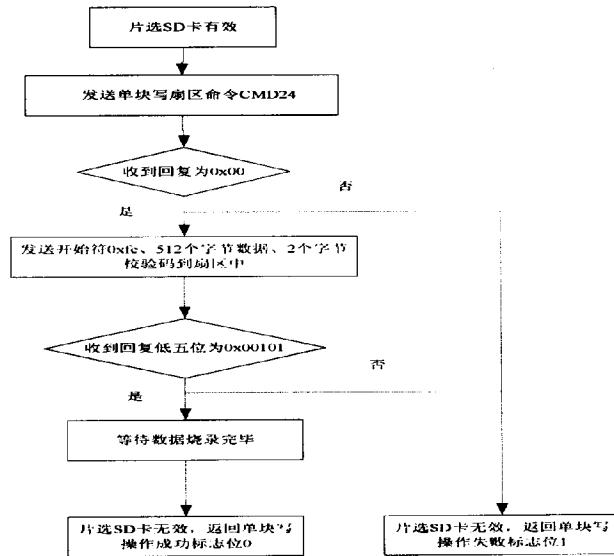


图 4-10 SD 卡单块写操作流程图

Figure 4-10 SD card single block write flow chart

多块写操作是向SD卡发送CMD25, 收到有效应答信号0x00后, 开始发送数据块。写多块数据时, 数据块起始令牌是0xfc, 这与单块写操作有区别。另外多块数据写成功后, 要发送停止符0xfd。发送完停止符0xfd后, SD卡将会响应一个应答标志位0xff。SD卡多块写操作流程如图4-11所示。

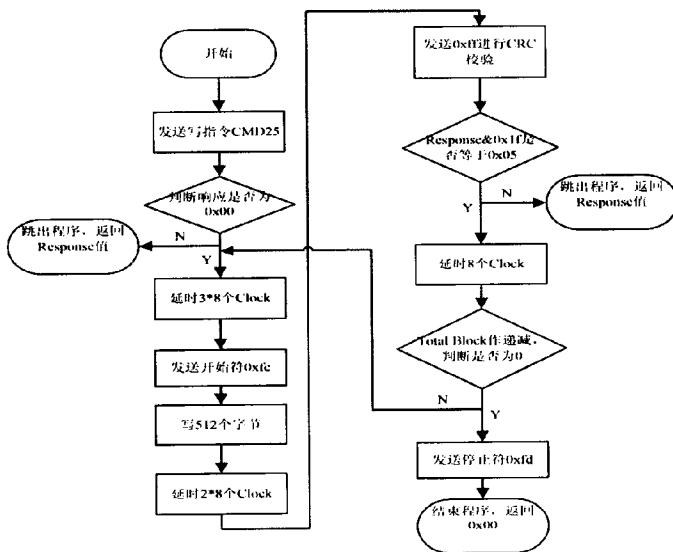


图 4-11 SD 卡多块写模块
Figure 4-11 SD card multi-block write module

4.3 FAT32 文件系统实现

4.3.1 FAT32 文件系统结构

Fat32 文件系统以簇为单位, 簇的大小视具体 SD 卡而定。其管理容量 2^{32} Bit, 相当于 32GB。要在 SD 卡上实现对 FAT32 文件系统的移植必须深入了解 FAT32 文件系统的结构及格式。FAT32 文件系统的结构包含主引导记录区、系统引导记录区、文件分配记录表以及数据区 4 个部分。FAT32 文件系统的存储结构如表 4-4 所示。

表 4-4 Fat32 组织结构
Table 4-4 the Fat32 organizational structure

MBR (主引导记录)
包含 DPT (磁盘分区表)
DBR (系统引导记录区)
包含 BPB (参数块)
保留区
FAT 表 1 (文件分配表)
FAT 表 2
数据区
包含 FDT (文件目录表)

MBR, 位于 SD 卡物理结构的第 0 扇区(0 柱面 0 磁头 1 扇区), 该区域分为两大部分, 第一部分为预启动区, 占用 446 字节; 第二部分为 DPT, 占用 66 个字节(64 字节的硬盘分区表和 2 字节的分区结束标志)。MB MBR 分区最终以 0X55AA 结束, 故读取 MBR 分区时以是否以 0x55 与 0xAA 作为结束标志来判断是否读取成功的标志。

DBR, 一般占用 512 字节空间, 这 512 字节是由跳转指令(3 个字节)、操作系统名称和版本号(8 个字节), BPB 区(71 个字节), 扩展 BPB 区, OS 引导程序区, 结束标志(0x55, 0xAA)几部分组成。BPB 记录本分区的每簇扇区数、每扇区字节数、FAT 表数、FAT 表扇区数、根目录簇号等重要参数。

FAT 表 1, 是 Fat32 文件系统中用来实现数据文件索引和定位的链式结构表。Windows 操作系统中, 数据文件不是连续地存放在硬盘的某一区域, 往往是分成若干簇, 如同链子一般存放。FAT 表就是记录数据文件簇与簇之间的连接关系, FAT 表中的每个簇项对应于数据存储区中相应的簇区。

FAT 表 2,FAT 表 1 的备份, 防止 FAT 表 1 损坏。

数据区,记录着每个数据文件的目录项信息(文件大小、文件起始簇号、文件创建时间)以及文件的内容。数据区从 2 号簇开始,而 2 号簇一般用来存储文件目录项信息。FAT32 目录项 32 个字节定义格式见下表 4-5。

表4-5 FAT32目录项32个字节定义格式
Table 4-5 of the FAT32 directory 32 bytes defined format

字节偏移 (16进制)	字节数	定义	
0x0~0x7	8	文件名	
0x8~0xA	3	扩展名	
0xB	1	属性字节	00000000 (读写)
			00000001 (只读)
			00000010 (隐藏)
			00000100 (系统)
			00001000 (卷标)
			00010000 (子目录)
			00100000 (归档)
0xC	1	系统保留	
0xD	1	创建时间的10毫秒位	
0xE~0xF	2	文件创建时间	
0x10~0x11	2	文件创建日期	
0x12~0x13	2	文件最后访问日期	
0x14~0x15	2	文件起始簇号的高16位	
0x16~0x17	2	文件的最近修改时间	
0x18~0x19	2	文件的最近修改日期	
0x1A~0x1B	2	文件起始簇号的低16位	
0x1C~0x1F	4	文件的大小	

4.3.2 FAT32 文件系统初始化

向 SD 卡写文件的前提是通过文件目录表 FDT 寻找文件存储位置的第一个簇地址,然后通过 FAT 表 1 按簇查找该文件所有存储地址从而找到文件所有簇地址。而 SD 卡并不以簇为单位存储数据,因此必须将簇地址转换为扇区地址。

在FAT表中查找到的簇地址时无法利用单片机进行操作的,因为SD卡本身并不使用簇为单位进行数据存储,故查找到的簇地址需要转换为SD卡进行物理操作的物理扇区地址才行,而簇地址转换为扇区地址只能通过参数记录块BPB查

找到该FAT分区中每个簇占用的扇区数以及保留扇区数才能够算出，即FAT卷中扇区号计算应为：当前簇所在扇区号=(当前簇-根目录簇号)*每个簇所占扇区数+根目录所在扇区号。而根目录簇号、每个簇所占扇区数、根目录所在扇区号这些信息均是通过FAT32文件系统初始化模块获得的。

综上所述，正确获取SD卡上FAT分区数据的方法是首先读取MBR，接着根据MBR中隐藏扇区个数计算出BPB所在物理扇区位置，然后读取并保存BPB，然后根据BPB中得到的信息获取FAT表大小及位置，以及FDT目录大小及位置。至此，便可以很方便的对文件进行添加，查找以及操作。

FAT32文件系统初始化程序见下所述。

```
void mfFAT_Init(void){
    struct mfFAT_BPB *bpb;
    //将数据缓冲区指针转为struct znFAT_BPB 型指针
    bpdb=(struct mfFAT_BPB *)(mfFAT_Buffer);
    //znFAT_FindBPB()可以返回BPB所在的扇区号
    pArg->BPB_Sector_No=mfFAT_Find_DBR();
    SD_Read_Sector(pArg->BPB_Sector_No,mfFAT_Buffer);
    //装入FAT表占用的扇区数到FATsectors中
    pArg->FATsectors=LE2BE((bpdb->BPB_FATSz32),4);
    //装入根目录簇号到FirstDirClust中
    pArg->FirstDirClust=LE2BE((bpdb->BPB_RootClus),4);
    //装入每扇区字节数到BytesPerSector中
    pArg->BytesPerSector=LE2BE((bpdb->BPB_BytesPerSec),2);
    //装入每簇扇区数到SectorsPerClust中
    pArg->SectorsPerClust =LE2BE((bpdb->BPB_SecPerClus),1);
    //装入第一个FAT表扇区号到FirstFATSector中
    pArg->FirstFATSector=LE2BE((bpdb->BPB_RsvdSecCnt),2)+pArg->BPB_Sector_No;
    //装入第一个目录扇区到FirstDirSector中
    pArg->FirstDirSector=(pArg->FirstFATSector)+(bpdb->BPB_NumFATs[0])*(pArg->FATsectors);
    //磁盘的总容量，单位是字节
}
```

```

pArg->Total_Size=LE2BE((bpb->BPB_TotSec32),4)*pArg->BytesPerSector;
}

```

FAT32文件系统初始化过程如图4-12所示。

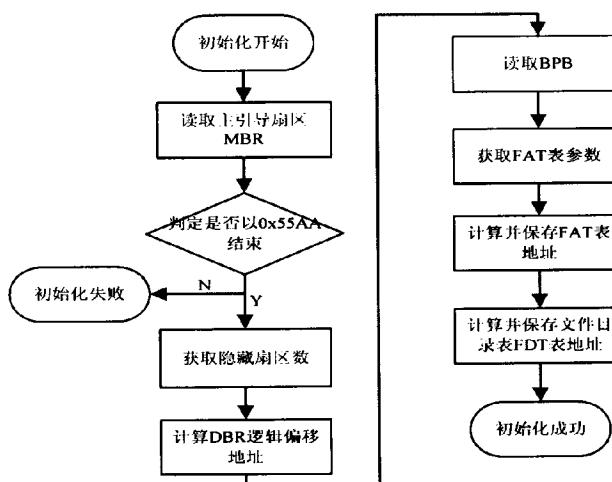


图4-12 FAT32文件系统初始化流程图

Figure 4-12 the FAT32 file system initialization flow chart

4.3.3 FAT32文件系统中文件操作

正确初始化完文件系统之后就可以对文件进行操作了，本课题需要实现文件的创建和写操作。

(1) FAT32中文件的创建操作

在FAT32文件系统中创建文件(或目录)的过程就是在文件目录表FDT中申请登记项的过程^[19]。登记项中包括文件名、文件长度和起始簇号等内容。文件名一般占用8字节，文件名的首字节又表明该文件的状态，00H表示该目录项未使用，E5H表示该文件(或目录)已被删除。创建文件时，属性值设置为20H(表示归档)，文件长度为0。FDT中每个文件占用32个字节，分别存储文件名扩展名等属性，每个登记项结构如前面所述的表4-5。

文件创建的具体方法如下所述。先将文件的信息填充到一个数组中，然后再将这个数组的数据写入到存储设备相应扇区的相应位置上。由于对SD卡写操作时，只能写入512个字节整数倍的数据量，故不能只能对要写操作的扇区某处进行写操作。鉴于此，先借助SD卡单块读把整个扇区内数据读取上来存放到512个字节的缓冲区里，再在该缓冲区内修改数据，从而实现间接申请登录项，最后把缓冲区内的数据再写入到原先的扇区中。需要注意，在FAT32文件系统中，文件

创建之初文件系统并不为文件分配起始簇，且文件的长度默认设置为0。

(2) FAT32中文件的写操作

FAT32文件系统上文件都是以簇为单位存取的。当读取FAT32上的文件时，首先要根据文件名查找到该文件的目录登记项。根据目录登记项中的起始簇号既可找到文件在数据区DATA中第1簇的内容，又可在FAT 表中找到第2个簇号。根据第2个簇号又能找到第2簇的内容和FAT表中的第3个簇号。这样，就可以根据FAT 表中的簇号读取到全部文件数据。故我们每写完一个簇就得及时更新FAT表，并能为文件的继续写操作提供空闲簇。写文件时要保证FAT1和FAT2中内容的一致性，即对两块都要进行同样的写操作。写操作结束后，还得对文件目录项信息中的文件长度信息及时更新。

FAT表相当与数据区的映射，每个FAT表项存放的是该FAT项所指向的簇号的下一个簇号。在FAT文件系统中，文件的存储依照FAT表制定的簇链式数据结构来进行。同时，FAT文件系统将组织数据时使用的目录也抽象为文件，以简化对数据的管理。

此外，为了提高文件写操作的速度，写数据时尽量使用SD多块写操作。正常情况下，多块写操作的数据长度不会超过SD卡一个簇的大小。

4.4 数据传输测试

本文采用直接方法实现 Fat32 文件系统，以 SD 卡 SPI 模式存储数据，将此系统应用于电磁式导航智能车数据采集系统中，每隔 20ms 记录电磁传感器采集值、测速编码器采集值和 PID 值，实践证明，此系统存储数据稳定、可靠。并且利用 MC9S12XS128 内部定时器 PIT 来估算写入速度，通过设置 PIT 参数，将 1 次中断定时设定为 1ms,而变量 a 与 b 分别对应写操作前与写操作后的时间，其差值乘以 1ms 即写入数据耗时，数组 filebuff[]大小为写入数据大小。从图 4-13、图 4-14 看出，写入速度随数据量增加呈现递增趋势，约 200KB/s 的写入速度能够满足传输速度的要求。

图4-13 写数据测试结果

Figure 4-13 Writed data test results

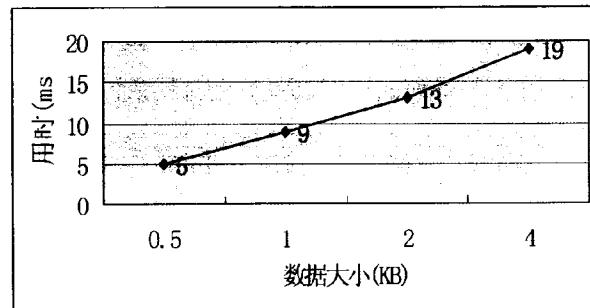


图4-14 测试结果折线图

Figure 4-14 Test results line chart

4.5 本章小结

本章首先介绍了 SD 卡与主控芯片之间的通信，包括 SD 卡的初始化和读写操作，然后阐述了 FAT32 文件系统的格式要求，并实现了 FAT32 文件系统。此系统移植性很强，适用于其他使用 SD 卡采集信息的环境。但对于要求高速传输的数据采集环境，可进一步研究 SD 卡 SD 模式，以提高传输速率，并且注意文件系统对传输速度的影响。

第五章 系统控制策略

智能车辆是一个复杂的被控对象，主要体现在：模型的不确定性，系统的非线性，以及控制量之间的协调关系等。智能车控制系统的两个控制量是舵机的转角和电机的转速，两者相互配合才能够保证智能车稳定行驶。传统的控制策略分别对这两个控制量进行 PID 控制，这就带来 PID 参数整定的问题，通过不断的实际调试固然能够整定出较优的 PID 值，但十分不便。因此设计出参数自整定的控制器显得十分重要，本课题采用 PID 与神经网络结合的控制算法，将 PID 融入神经网络中，构成比例，积分，微分神经元，并通过误差反向传播调整网络的权值，实现参数自整定的过程。整个控制器，前向算法体现出 PID 控制，反传算法体现神经网络的学习能力，两者结合构成动态调整的控制网络。

5.1 传统 PID 控制

由于建立智能车运动学或动力学模型进行分析再设计控制策略过于复杂，通常对于这样的黑箱模型，使用较多的控制方法是 PID 控制和对 PID 控制进行改进的控制算法，以及专家控制、模糊控制、神经网络控制等智能算法。而 PID 控制作为最早发展起来的控制策略之一，其算法简单、鲁棒性好和可靠性高，不应完全摒弃 PID 控制方法，将 PID 控制方法与智能算法作结合，可以达到优化 PID 控制的目的。

(1) PID 控制

PID 控制器由比例、积分和微分单元组成，它以结构简单、稳定性好、工作可靠、调整方便的优点成为实际工程中应用最广泛的一种闭环控制方法。在系统调试前期，简单而实用的 PID 方法自然而然成为首选，被应用于智能车系统的控制。PID 控制器的数学表达式如下：

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (6-1)$$

比例调节(K_p)按比例反映系统的偏差情况，系统一旦出现一定的偏差，比例单元立即发挥调节作用，以减少偏差值。比例系数较大可以加快调节速度，减少误差值，但是过大的比例系数，反而会对系统造成负面的影响，使系统的稳定性下降，甚至造成系统运行不稳定的现象。

积分调节(K_i)消除系统的稳态误差，提高无差度。只要系统存在误差，积分单元就进行调节，调节直至消除误差方才停止。积分作用的强弱由积分时间常数 T_i 来决定。 T_i 值越小，积分作用则越强；反之 T_i 值越大，则积分作用越弱。加入积分环节后，会使系统运行时的稳定性下降，以及动态响应速度变慢。

微分调节(K_d)反映系统偏差情况的变化率，能够预见偏差变化的趋势，因而在系统偏差还没有形成之前，能够产生一个超前的控制量，从而消除偏差的影响。因此微分单元能够改善系统的动态性能。

由于电子芯片的广泛应用，PID 控制方法在离散系统中被越来越广泛的使用，离散化处理过的 PID 控制方法基本形式可分为两种：一种是位置式 PID 控制方法，另一种是增量式 PID 控制方法。两者本质相同，但增量式 PID 控制方法更易于编程实现，在工程中被普遍使用。首先推导出位置式 PID 控制算法，设定系统的采样时间为 T ，得到位置式 PID 控制在任意 k 时刻的表达式：

$$u(k) = k_p e(k) + k_i \sum_{i=1}^k e(i) + k_d [e(k) - e(k-1)] \quad (6-2)$$

其中， $e(k) = r(k) - y(k)$

其中 k_p 、 k_i 、 k_d 分别为比例，积分，微分系数。若令：

$$\left\{ \begin{array}{l} \Delta u(k) = u(k) - u(k-1) \\ \Delta e(k) = e(k) - e(k-1) \\ \Delta_2 e(k) = \Delta e(k) - \Delta e(k-1) = e(k) - 2e(k-1) + e(k-2) \end{array} \right. \quad (6-3)$$

则增量式PID控制在任意k时刻的表达式：

$$\Delta u(k) = k_p \Delta e(k) + k_i e(k) + k_d \Delta_2 e(k) \quad (6-4)$$

(2) 分段 PID 控制

针对不同的弯道情况，设置 2-3 个与之对应的 PID 参数值，在行驶时切换 PID 参数值，在车模由直道入弯道以及弯道入直道时，能够在一定程度上改善车模的控制性能，提高行驶时的稳定性。

(3) 结合 Bang-Bang 的 PID 控制

在智能车行驶控制过程中，车速的动态响应要快，在直道高速转入弯道以及弯道进入直道时，这点更加突出。速度闭环控制有很多种方法，其中响应最快的是 Bang-Bang 控制，但是 Bang-Bang 控制的稳态性能比较差，容易造成系统的抖动。在速度控制时采用 Bang-Bang 与 PID 相结合的方式改变电板电压，实现调速

跟随，达到了闭环控制的效果。

5.2 PID 神经元网络（PIDNN）

近年来，随着神经网络理论的发展，将应用最广泛的 PID 控制器与具有自学习功能的神经网络相结合，已成为智能控制研究的一个新方向。并且，在这个方向上已取得了一些研究成果。其主要结合方式分为三类：

(1) 单神经元网络控制，神经元输入权值一一对应 PID 参数，但它不承担比例，积分，微分处理工作，它的输入信号为经过比例、积分、微分处理的偏差值，这种单层网络（只有输入与输出层， $3*1$ 结构）只具备线性分类能力，无任意函数逼近能力，在复杂系统控制中难以达到良好的性能。

(2) 在常规 PID 控制器基础上附加一个神经网络模块（如 BP 神经网络，RBF 神经网络），神经网络根据系统的输入与输出信息，通过反复的学习调整，在线不断整定 PID 参数值。其缺点是网络结构比传统 PID 控制器复杂很多，较难应用于实际工程。

(3) 将 PID 控制规律融入神经网络中，实现神经网络和 PID 控制规律的结合，构造出 PID 神经元网络（PIDNN）。PIDNN 属于多层前向网络，它与一般的多层前向网络又不完全相同，一般的多层前向网络中全部神经元的输入输出特性都是相同的，而 PIDNN 隐含层是由比例、积分、微分 3 个单元组成。

本课题选用第三种控制方法（PIDNN）作为智能车的控制策略。神经网络和传统 PID 控制器均有其优点和薄弱之处，将二者结合起来，可以取长补短。先将 PID 功能引入神经元网络的基本神经元中，分别构成 PID 神经元，再按照 PID 控制规律的基本模式，用这些基本神经元构成新的神经元网络。网络的反传学习算法借鉴 BP 神经网络的反传思想，使用最快下降法进行权值调整，以达到学习的目的。

5.2.1 神经元基本模型结构

人工神经元网络的结构实体包括两个重要组成部分：一个是神经元网络的互联结构模式；另一个是神经元网络的最基本处理单元，即神经元。尽管神经网络有不同类型的神经元，但它们具有一些共同的特征和共同的性质，图 5-1 即为神经元的基本模型结构形式。图 5-2 为采用控制领域通用的符号和描述方法，得到的模型框图形式^[62]。

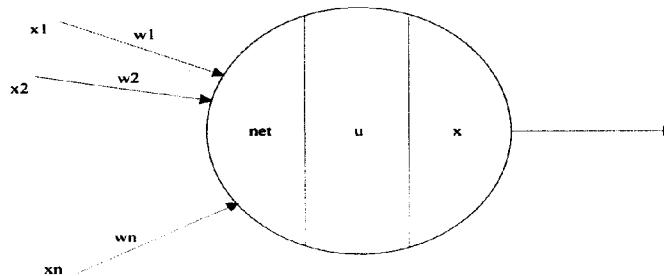


图 5-1 神经元模型结构形式

Figure 5-1 Neuron model structure

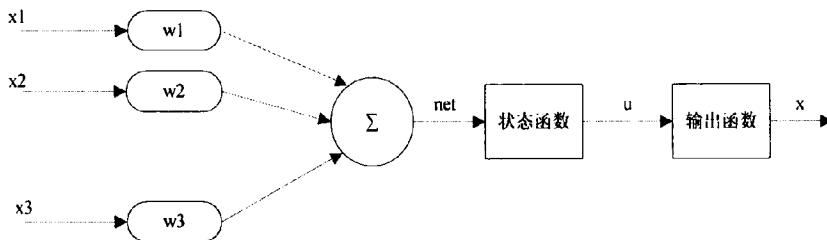


图 5-2 神经元模型结构框图

Figure 5-2 neuron model block diagram

(1) 神经元的输入

神经元的输入体现了生物神经元的空间总和作用。在任意时刻 k , 对于神经元网络的第 j 个神经元, 其总输入 net_j 等于与其相连的各支路输出量 x_1, x_2, \dots, x_n 分别乘上权重值 $w_{1j}, w_{2j}, \dots, w_{nj}$ 后的总和, 即

$$net_j(k) = \sum_{i=1}^n w_{ij} x_i(k), i \neq j \quad (6-5)$$

(2) 神经元的状态

神经元 j 的状态 u_j 由其状态函数 $g(\cdot)$ 决定。用神经元当前输入 net_j 和当前状态为自变量, 按其状态函数就可产生神经元的下一个状态, 即

$$u_j(k+1) = g(net_j(k), u_j(k)) \quad (6-6)$$

(3) 神经元的输出

神经元 j 的状态 x_j 由此神经元的输出函数 $f(\cdot)$ 决定。用神经元状态 u_j 为自变量, 按其输出函数就可产生神经元的输出值, 即

$$x_j(k) = f(u_j(k)) \quad (6-7)$$

神经元的输出函数有很多形式, 包括恒等函数、阈值函数、统计函数、S 形函数、拟线性函数、随机函数等。

5.2.2 PID 神经元的计算方法

通过以上对神经元模型的分析可知，神经元的输入是对所有输入信息的总和，而神经元状态函数和输出函数则有不同的形式，从而决定了不同类型的神经元模型。分别选择状态函数为比例、积分、微分函数，从而建立比例元、积分元、微分元。

(1) 比例元的状态函数 $g_p(\cdot)$ 为比例函数，其比例系数等于 1。在 k 时刻的状态式为

$$u_j(k) = g_p[\text{net}_j(k)] = \text{net}_j(k) \quad (6-8)$$

(2) 积分元的状态函数 $g_i(\cdot)$ 为积分函数，其积分系数恒为 1。在 k 时刻的状态式为

$$u_j(k) = g_i[\text{net}_j(k), u_j(k-1)] = u_j(k-1) + \text{net}_j(k) \quad (6-9)$$

(3) 微分元的状态函数 $g_d(\cdot)$ 为微分函数，其微分系数恒为 1。在 k 时刻的状态式为

$$u_j(k) = g_d[\text{net}_j(k), \text{net}_j(k-1)] = \text{net}_j(k) - \text{net}_j(k-1) \quad (6-10)$$

5.2.3 PID 神经元网络结构

PIDNN 的基本形式是 2*3*1 的三层网络结构，包括输入层的 2 个神经元，隐含层的 3 个神经元，输出层的 1 个神经元。网络输入与输出层的神经元为一般神经元，而隐含层的 3 个神经元为比例，积分，微分神经元，神经元的状态函数不同。PIDNN 网络结构简单，易于工程实现。在软件编程时，PID 神经元的计算工作量均不大。PIDNN 变量控制系统结构如图 5-3。

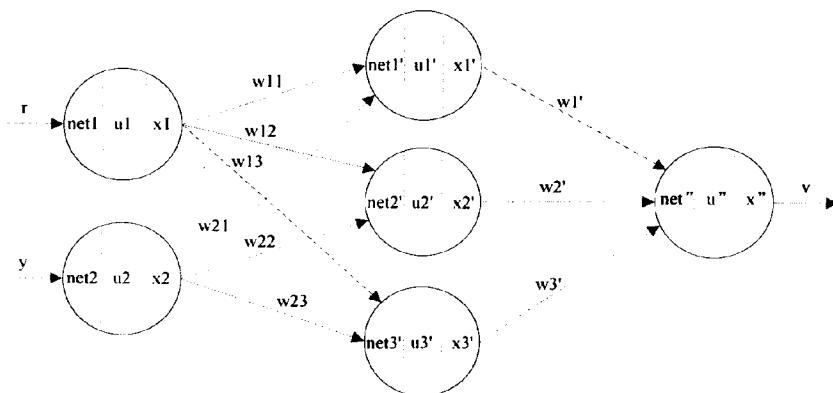


图 5-3 PIDNN 结构形式

Figure 5-3 PIDNN structure

5.2.4 PIDNN 前向算法

(1) 输入层

PIDNN 的输入层有 2 个神经元，在构成控制系统可分别输入被调量的给定值 $r(k)$ 和实际值 $y(k)$ 。在任意采样时刻 k , 其输入

$$net_1(k) = r(k) \quad (6-11)$$

$$net_2(k) = y(k) \quad (6-12)$$

输入层神经元的状态函数：

$$u_i(k) = net_i(k) \quad (6-13)$$

输入层神经元的输出函数：

$$x_i(k) = \begin{cases} 1, & u_i(k) > 1 \\ u_i(k), & -1 \leq u_i(k) \leq 1 \\ -1, & u_i(k) < -1 \end{cases} \quad (6-14)$$

式中 $i=1, 2$ 。

(2) 隐含层

隐含层神经元的输入函数：

$$net_j(k) = \sum_{i=1}^2 w_{ij} x_i(k) \quad (6-15)$$

隐含层神经元的状态函数：比例元状态函数

$$u_j(k) = net_j(k) \quad (6-16)$$

积分元状态函数：

$$u_j(k) = u_j(k-1) + net_j(k) \quad (6-17)$$

微分元状态函数：

$$u_j(k) = net_j(k) - net_j(k-1) \quad (6-18)$$

隐含层神经元的输出函数：

$$x_j(k) = \begin{cases} 1, & u_j(k) > 1 \\ u_j(k), & -1 \leq u_j(k) \leq 1 \\ -1, & u_j(k) < -1 \end{cases} \quad (6-19)$$

式中 $j=1, 2, 3$ 。

(3) 输出层

输出层神经元的输入函数：

$$\text{net}_j(k) = \sum_{j=1}^3 w_j x_j(k) \quad (6-20)$$

输出层神经元的状态函数:

$$u''(k) = \text{net}_j(k) \quad (6-21)$$

输出层神经元的输出函数:

$$x''(k) = \begin{cases} 1, & u''(k) > 1 \\ u''(k), & -1 \leq u''(k) \leq 1 \\ -1, & u''(k) < -1 \end{cases} \quad (6-22)$$

5.2.5 PIDNN 反传算法

这里的反向传播算法即误差反向传播算法, 借鉴了 BP 神经网络算法的精华, 它使得网络权重值可以修改, 完成学习和记忆的功能。PIDNN 训练和学习的目的是使网络实际输出和理想输出的偏差平方平均值为最小。

$$J = E = \frac{1}{2} [r(k) - y(k)]^2 = \frac{1}{2} e^2(k) \quad (6-23)$$

按照梯度下降法来调节 PIDNN 权重值, 经过 k 步训练和学习后, PIDNN 各层权重值的迭代方程式为

$$W(k+1) = W(k) - \eta \frac{\partial E}{\partial W} \quad (6-24)$$

隐含层至输出层的权重值 W_j 和输入层至隐含层的权重值 W_{ij} 的具体计算方法分别如下:

(1) 隐含层至输出层的权重值迭代公式为

$$w_j(k+1) = w_j(k) - \eta_j \frac{\partial E}{\partial w_j} \quad (6-25)$$

$$\text{式中 } \frac{\partial E}{\partial w_j} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial v} \cdot \frac{\partial v}{\partial x''} \cdot \frac{\partial x''}{\partial u''} \cdot \frac{\partial u''}{\partial \text{net}''} \cdot \frac{\partial \text{net}''}{\partial w_j}$$

$$\frac{\partial E}{\partial y} = \frac{\partial}{\partial y} \left\{ \frac{1}{2} [r(k) - y(k)]^2 \right\} = -[r(k) - y(k)]; \quad \frac{\partial y}{\partial v} \approx \frac{y(k) - y(k-1)}{v(k) - v(k-1)}$$

$$\frac{\partial v}{\partial x''} = 1; \quad \frac{\partial x''}{\partial u''} = 1; \quad \frac{\partial u''}{\partial \text{net}''} = 1; \quad \frac{\partial \text{net}''}{\partial w_j} = \frac{\partial}{\partial w_j} [\sum_{j=1}^3 w_j x_j(k)] = x_j(k)$$

带回原式

$$\frac{\partial E}{\partial w_{ij}} = -x_j(k) \cdot \frac{y(k) - y(k-1)}{v(k) - v(k-1)} \cdot [r(k) - y(k)] \quad (6-26)$$

(2) 输入层至隐含层的权重值迭代公式为

$$w_{ij}(k+1) = w_{ij}(k) - \eta_{ij} \frac{\partial E}{\partial w_{ij}} \quad (6-27)$$

式中

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial v} \cdot \frac{\partial v}{\partial x} \cdot \frac{\partial x}{\partial u} \cdot \frac{\partial u}{\partial net} \cdot \frac{\partial net}{\partial x_j} \cdot \frac{\partial x_j}{\partial u_j} \cdot \frac{\partial u_j}{\partial net_j} \cdot \frac{\partial net_j}{\partial w_{ij}} \\ \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial v} \cdot \frac{\partial v}{\partial x} \cdot \frac{\partial x}{\partial u} \cdot \frac{\partial u}{\partial net} &= -\frac{y(k) - y(k-1)}{v(k) - v(k-1)} \cdot [r(k) - y(k)] \\ \frac{\partial net}{\partial x} &= \frac{\partial}{\partial x} \sum_{j=1}^3 w_{ij} x_j = w_{ij}; \quad \frac{\partial x_j}{\partial u_j} = 1 \\ \frac{\partial u_j}{\partial net_j} &\approx \frac{\Delta u_j}{\Delta net_j} = \frac{u_j(k) - u_j(k-1)}{net_j(k) - net_j(k-1)}; \quad \frac{\partial net_j}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} [\sum_{i=1}^2 w_{ij} x_i(k)] = x_i(k) \end{aligned}$$

带回可得

$$\frac{\partial E}{\partial w_{ij}} = -w_{ij} \cdot x_i(k) \cdot \frac{y(k) - y(k-1)}{v(k) - v(k-1)} \cdot \frac{u_j(k) - u_j(k-1)}{net_j(k) - net_j(k-1)} \cdot [r(k) - y(k)] \quad (6-28)$$

5.2.6 PIDNN 连接权重初值选取

(1) 输入层至隐含层的连接权重初值选取

根据 PID 控制算法的特点，使其能够完成由 $(r, y) \rightarrow e$ 的映射功能，选取 $w_{1j} = 1, w_{2j} = -1$ 。

(2) 隐含层至输出层的连接权重初值选取

为使 PIDNN 连接权重取初值时的输出等价于 PID 控制器输出，选取 $w_1 = K_p, w_2 = K_I, w_3 = K_D$ 。由于 PID 控制器在被大量的使用，人们已经掌握了大量的参数整定规律和经验，利用这些经验确定 K_p, K_I, K_D 值，使 PIDNN 初始等价于 PIDNN。在这基础上，再通过在线训练和学习，调整网络连接权值，PIDNN 就可以具有较优异的控制效果。

5.2.7 PIDNN 收敛性和稳定性

稳定性是控制系统最重要的性能指标之一，PID 神经元网络也必须满足稳定性的要求，否则将失去使用价值。PID 神经元网络的结构和连接权重初值决定系

统是否初始稳定，神经元网络控制系统的学习步长则决定了收敛性。利用李雅普诺夫（Lyapunov）稳定性原理确定 PIDNN 控制收敛时的学习步长范围。

李雅普诺夫稳定性原理：如果一个 PIDNN 控制系统的输入序列 $r(k) \in R^n$ 在 R^n 中是长期激励的，且学习步长 η 满足 $0 < \eta < \frac{1}{\varepsilon^2}$ ，式中 $\varepsilon = \frac{1}{2\sqrt{E}} \cdot \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial v} \cdot \frac{\partial v}{\partial W}$ ，则 PIDNN 学习 k 步后的权值调整算法： $W(k+1) = W(k) - \eta \frac{\partial E}{\partial W}$ 。保证 PIDNN 控制系统在学习过程中收敛。式中 W 表示 PIDNN 的连接权值 w_{ij} 和 w'_{ij} 。

如果一个 PIDNN 控制系统在 PIDNN 连接权值取初值时是稳定的，而且 PIDNN 的学习步长满足李雅普诺夫稳定性原理的条件，则 PIDNN 控制系统整个过程都是稳定的。

5.3 PID 与 PIDNN 阶跃响应测试

选取直流电机为被控对象，分别对PID控制和PIDNN控制进行阶跃响应测试，通过比较两者的仿真结果，分析两者的调节性能，直流电机模型的传递函数

$$G(s) = \frac{1}{Js^2 + Bs}, \quad J = 0.0067, B = 0.10.$$

测试方法采用Simulink模块与M函数相结合的方式，分别对两个算法进行测试。给定信号为阶跃信号，初始值为0，阶跃时间为1s，阶跃值为0.5。采样时钟 Clock=10ms，仿真时间5s，仿真结果通过Scope1显示出来。其中PID-Simulink的 MATLAB Function模块链接一个M文件，M文件的程序如下，实现PID功能，其中PID系数选取，Kp=10，Ki=0.01，Kd=0.5。

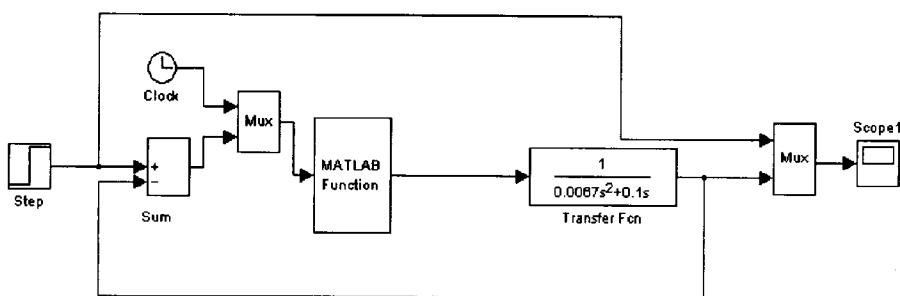


图5-4 PID-Simulink结构图

Figure 5-4 PID-Simulink block diagram

```
function [u]=pidsimf(u1,u2)
```

```

persistent errori error_1
if u1==0
    errori=0
    error_1=0
end
kp=10;
ki=0.01;
kd=0.5;
error=u2;
errord=error-error_1;
errori=errori+error;
u=kp*error+kd*errord+ki*errori;
error_1=error;

```

可以看出PID-Simulink结构图与PIDNN-Simulink结构图在MATLAB Function模块上输入端口个数不一样，这是由两者算法的差异所造成的。PIDNN-Simulink中的MATLAB Function模块同样也链接一个M文件，实现PIDNN控制算法，学习步长steplength1,steplength2定为0.01，学习精度j=0.01,权值初值选取，w11=w12=w13=1, w21=w22=w23=-1, w_1=10, w_2=0.01,w_3=0.5。使得其初始状态等同于PID控制。程序篇幅过长，不罗列在此，可从附录中查看完整程序。

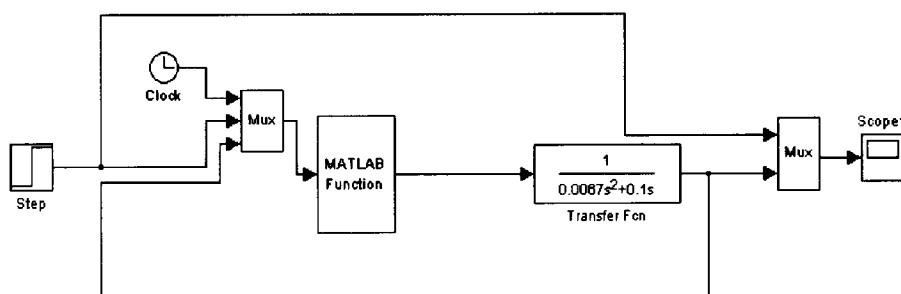


图5-5 PIDNN-Simulink结构图

Figure 5-5 PIDNN-Simulink block diagram

PID与PIDNN阶跃信号的仿真输出如图5-6,5-7，比较两者的仿真结果，PIDNN控制下，系统的上升时间（tr）和峰值时间（tp）增长，但超调量明显变小，并且稳定时间（ts）也变短。从总体性能上比较，PIDNN控制要优于PID控制。

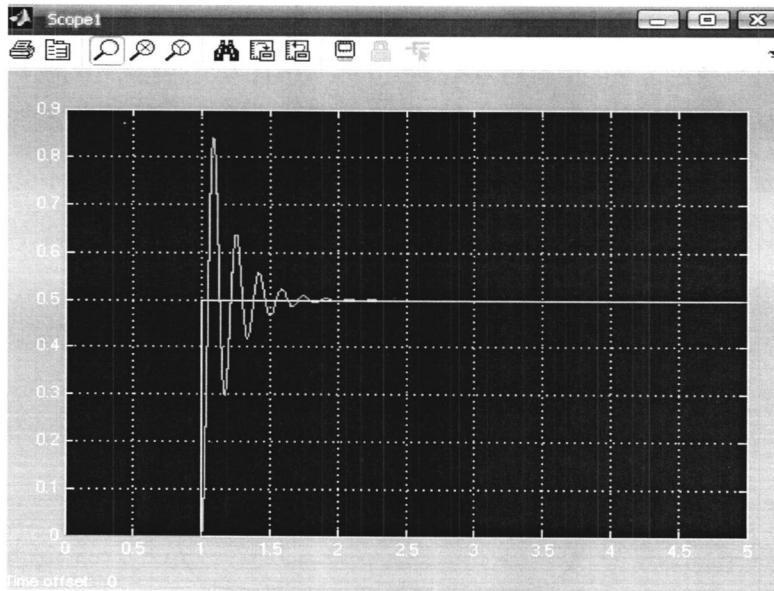


图5-6 PID控制仿真结果

Figure 5-6 PID control simulation results

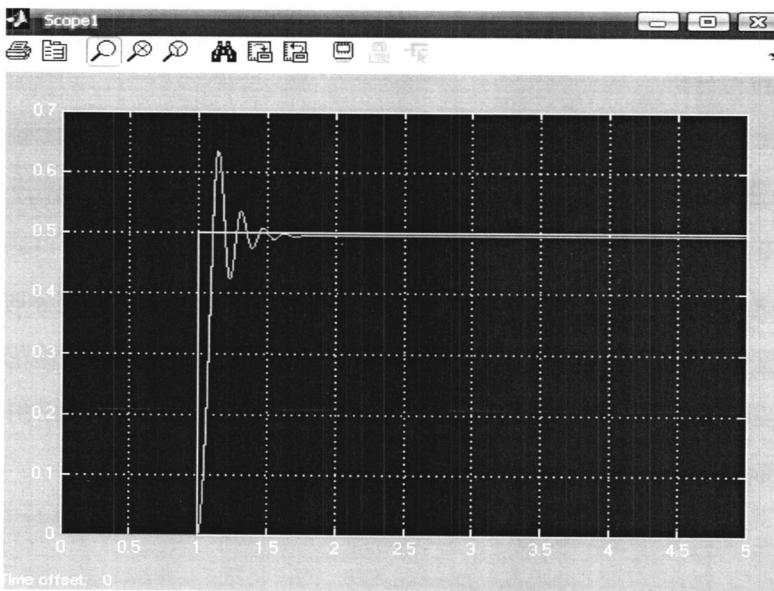


图 5-7 PIDNN 控制仿真结果

Figure 5-7 PIDNN control simulation results

5.4 控制策略设计

实验车模是一个两输入，两输出的被控对象。对于被控对象，控制量为输入量，其被控量为输出量。则控制系统输入量为作用于调节舵机和电机的 PWM 波值，控制系统输出量为电磁传感器检测车模与跑道的相对位置信息，以及光电

编码器检测到的电机转速。多输入—多输出系统不一定是多变量系统。如果被控对象的输入量与一个以上的输出量之间存在关联耦合关系，则该系统属于多变量系统。而如果每个输入变量分别唯一地和一个输出变量有关系，则该系统应属于多个单变量系统的组合体，而不是多变量系统。实验车模的舵机改变不会影响电机转速，电机改变也不会影响车模的转角，显然实验车模控制系统是两个单变量系统的组合体。

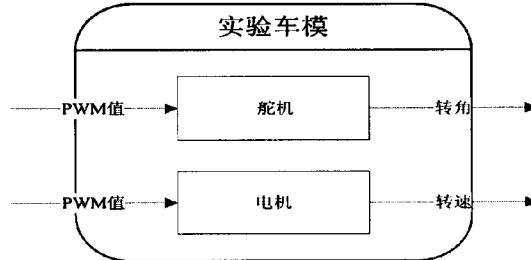


图 5-8 单变量系统组合体

Figure 5-8 Single variable combination

5.3.1 转向控制

(1) 转向分析

跑道主要由直道、大弯道、小弯道以及 S 型弯道组成，为了保证实验车模在不触碰跑道两边边界的情况下，沿着跑道方向前进，需要能够检测到跑道中心线相对于车模中心的偏移量，此外还包括中心线的方向、弯曲程度、车模速度、加速度和电池电压等辅助参数。当车模与跑道中心引导线出现偏离距离 d 时，如图所示，车模需要对其前轮行驶方向进行调整。车模与跑道的偏移会导致左右电磁传感器检测值的变化，通过将两传感器检测差值反馈给控制器，计算出相应的舵机输出控制角 A ，进而改变前轮转向。

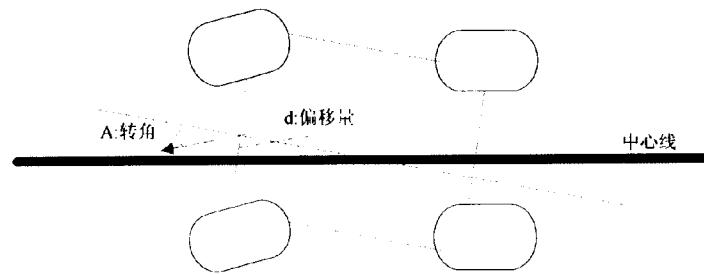


图 5-9 转角与偏移量

Figure 5-9 Corner and offset

从上面简化的车模运动模型图 5-9 可以得到，实验车模与跑道中心线的相对位置的改变量 d 近似等于舵机输出转角 A 随着距离的积分，如式所示。

$$d(l) \approx kc \int_0^l A(\tau) d\tau \quad (6-29)$$

其中 $d(t)$ ——车模与跑道中心线的偏移量

$A(t)$ ——舵机输出转角

k ——比例因子, 与车模尺寸、舵机至前轮转角的传动比有关

c ——舵机转向传动比

(2) 舵机分析

实验车模转向执行机构包括舵机和传动机构, 舵机由舵盘、减速齿轮组、位置反馈电位计、直流电动机和控制电路组成, 输出轴带动线性位置电位计作位置检测, 电位计使舵盘输出转角正比于给定的控制信号, 并反馈给控制电路, 控制电路将其与输入的脉冲信号进行比较, 驱动直流电机正反转, 使齿轮组的输出位置与期望值相符。由于舵机内部带反馈环节, 因此它的精确模型很难求得。

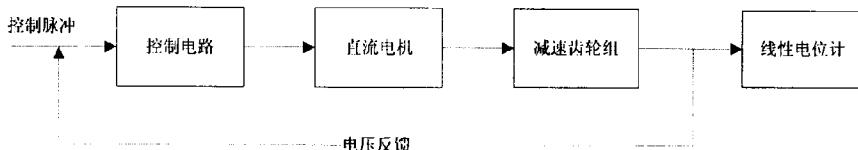


图 5-10 舵机工作原理
Figure 5-10 Servo Working principle

舵机的内部结构决定了其输出转角与外部控制电压成正比关系, 但舵机本身是一个滞后环节, 其滞后时间正比于输出角的大小, 因此舵机模型可以近似认为:

$$\theta(u) = k_u (u - u_0) e^{(t-\tau)} \quad (6-30)$$

k_u ——转角增量和控制电压增量的比例系数

u_0 ——当转角为 0° 时的控制电压值

e ——为单位阶跃信号

τ ——舵机输出的滞后时间, 它与舵机输出转角大小成正比关系

实验采用的舵机型号为S-D5, 塑胶齿, 双含油轴承, 尺寸49.0mm, 重量44g, 额定电压为6V, 额定电压下速度0.14sec/60°, 扭力5kg/cm。正常工作电流200mA, 频率300Hz, 堵转电流800mA。舵机的控制电压采用PWM波进行控制, 当输出转角为 0° 时, PWMPER23=6666; PWMDTY23=2900; 控制电压为2.61V。表为车模与跑道中心线偏离距离实测极值以及与之对应的电磁传感器AD采样值。

表 5-1 偏差距离与电磁传感器采样值

Table 5-1 Deviation distance and electromagnetic sensors sampled values

车体与跑道偏离距离	左电磁传感器 AD 采样值	右电磁传感器 AD 采样值
-12cm (左偏)	64	255
0	148	147
+12cm (右偏)	255	65

表 5-2 PWM 值与转角
Table 5-2 PWM value and corner

实验车模	PWM 波值	电压	转角
左转极值	3700	3.33V	-35°
中值	2900	2.61V	0°
右转极值	2100	1.89V	+35°

5.3.2 速度控制

(1) 速度分析

在实际调试过程中，速度的大小在很大程度上影响车模的转弯效果，因此为了使车模在转弯时不出现冲出道路的情况，需要对车模的行驶速度进行调整。

首先分析一下车模的运动模型，显然在行驶转角调整之后，且在下一次行驶转角调整之前，车头和车尾地行驶可以看成是一个匀速圆周运动，为了简便起见，将车模两前轮连线的中间点的运动轨迹视为车模的运行轨迹，其速度方向与车模本身方向的夹角为 A ，两后轮连线的中间点的运动轨迹视为车模尾部的运行轨迹，其速度方向与车模自身方向相同，如图 6.8 所示，由于车模长度不变，因此头部和尾部运行轨迹应该是共圆心的，且速度方向与它们到圆心的方向是垂直的，这样便可以通过转角的大小来寻找圆心。

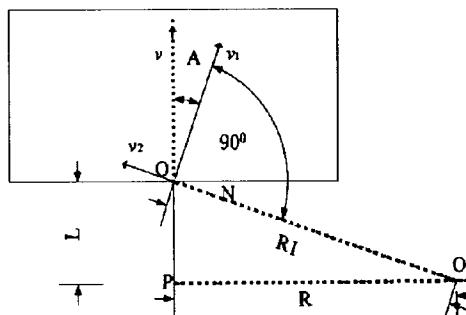


图 5-11 转向速度分析
Figure 5-11 steering velocity analysis

在图中， O 点为车模头部， P 点为车尾，后轮的速度为 v ，前轮转弯方向的速度为 v_1 ，因此有

$$v_1 = v \times \cos A \quad (6-31)$$

由图可以得出车模头部运行圆弧的半径为 R_1 ，并且有

$$R_i = \frac{L}{\sin A} \quad (6-32)$$

车模尾部运行圆弧的半径为 R ，并且有

$$R = \frac{L}{\tan A} \quad (6-33)$$

为了方便计算，可认为车模的重心落在尾部，设道路的摩擦系数为 μ ，为使不出现侧滑情况，因此转弯速度必须满足

$$v^2/R \leq g\mu \quad (6-34)$$

设车轮半径为 r ，根据式 (6-17) 可知转弯时电机速度必须满足

$$n < \frac{1}{2\pi \cdot r} \sqrt{\frac{Lg\mu}{\tan A}} \quad (6-35)$$

(2) 电机分析

实际上对车速的控制就是对直流电机的转速进行控制，通过对直流电动机进行建模，分析电机转速与控制电压之间的关系。额定他励直流电机的等效模型如图 6.1。

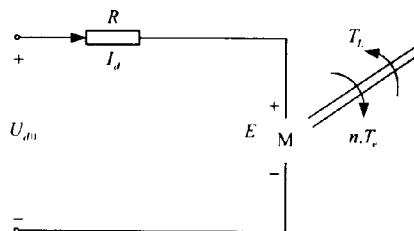


图 5-12 他励直流电动机在额定励磁下的等效电路
Figure 5-12 Excited DC motor equivalent circuit under rated excitation

假定电路电流连续，则动态电压方程为

$$U_{d0} = E + I_d(R_a + R) \quad (6-36)$$

式中 R_a 为电枢回路电阻，包括电枢回路串联各绕组电阻与电刷接触电阻之和； R 为接在电枢回路中的调节电阻。

$$R_a = \frac{1}{2} \cdot \frac{U_N \cdot I_N - P_N}{I_{N}^2} \quad (6-37)$$

$$n = \frac{U_{d0} - I_d(R_a + R)}{C_e \cdot \Phi} \quad (6-38)$$

额定励磁下的感应电动势和电磁转矩分别为

$$E = C_e \Phi n \quad (6-39)$$

$$T_m = C_m \Phi I_d \quad (6-40)$$

式中 T_m —— 直流电机的电磁转矩 ($N \cdot m$)；

C_m —— 额定励磁下电动机的转矩系数 ($N \cdot m/A$)；

C_e —— 额定励磁下电动机的机电常数 ($N \cdot m/A$)， $C_m = \frac{30}{\pi} C_e$ ；

实验车模选用 RS380 直流电机作为驱动，RS380 电机的具体参数如表 6-1 所示。

表 5-3 电机参数
Tab.5-3 Parameter of motor

电压 (V)		空载		最高效率 (65%)				堵转	
工作 范围	正常	速度 (r/min)	电流 (A)	速度 (r/min)	电流 (A)	扭矩 (g.cm)	输出 (W)	扭矩 (g.cm)	电流 (A)
3-9	7.2	16200	0.5	14060	3.29	111	16	839	21.6

5.3.3 控制策略

(1) 数据归一化处理

由于 PIDNN 的神经元输出函数采用如下函数形式：

$$f(x) = \begin{cases} 1, & x > 1 \\ x, & -1 \leq x \leq 1 \\ -1, & x < -1 \end{cases} \quad (6-41)$$

输出函数使得输出控制在 -1 至 1 之间，如果对输入样本数据不预先进行归一化处理，很可能出现样本输出超出神经网络输出范围的情况。常用的归一化方法有三种：线性转换函数、对数函数转换、反余切转换函数。

线性转换函数：

$$f(x) = (x - Minvalue) / (Maxvalue - Minvalue) \quad (6-42)$$

对数转换函数：

$$f(x) = \log_{10}(x) \quad (6-43)$$

反余切转换函数：

$$f(x) = \text{arc cot}(x) * 2 / \pi \quad (6-44)$$

本文选用第三种归一化方法，通过反余切转换函数，将样本数据控制在 -1 至 1 之间，对输出结果作反归一化也较为方便。

(2) PIDNN 控制实现

如何在保证实验车模在稳定运行的情况下,又能以较快的速度行驶?一方面是协调舵机转向与电机转速的控制值,使两者能够较好的配合,保证行驶的稳定性。另一方面提高车模在通过弯道时所允许的最大速度。

提高车模的稳定性,相对于不同曲率的跑道,需要在车模调试过程中不断的测试较优的转角以及车速,并以此作为调整的基准。

提高车模的行驶速度,但车模过弯时速度是受到限制的,通过提高道路检测前瞻距离,可以提高车模通过弯道的最大允许速度。提高舵机响应速度以及舵机的转向传动比都可以提高车模通过弯道的允许速度。

常用的控制策略对于横向控制(舵机)采用离散增量式PD算法,对于纵向控制(电机)采用进行PID或PD控制。为了提高电机的调节速度,有时会忽略积分环节(I)。通过现场调试发现,实验车模在过弯道时发生滑出跑道的情况,主要原因并不是转角的问题,而是车模行驶速度太快,调节不够快捷,导致车体冲出赛道。过弯道时需要控制适当的转角以及相应的车速,才能保证车模平稳地通过弯道。

因而本文对转向控制保留PD控制,而对速度控制进行算法改进,利用PID神经元网络算法代替PID控制,通过PID神经元网络算法的自适应调整功能,实现更及时,有力地车速调节。图中, r_1 为车模与跑道中心线相对位置的目标值,给定为 0, 就是要求车模居中行驶; y_1 为两个电磁传感器 AD 检测差值; v_1 为舵机的控制量, PWM 波值。 r_2 为车模速度的目标值,根据不同的跑道曲率,给定不同的速度值; y_2 为车速检测值,通过速度传感器检测得到; v_2 为电机的控制量,也为 PWM 波值。

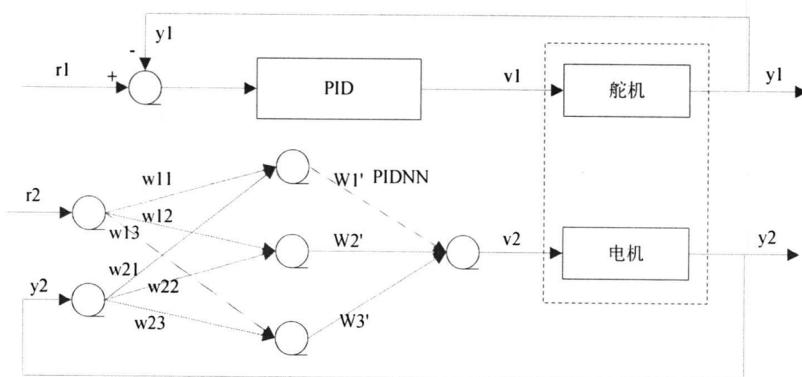


图 5-13 控制系统结构

Figure 5-13 Control System Architecture

5.3.4 仿真

在 Plastid2 软件上对改进后的控制策略进行仿真，并对行驶情况以及性能进行对比分析。Plastid2 软件是由清华大学于 2007 年开发的智能车仿真平台，它包括四项基本功能：赛道设计、赛车设计、仿真模拟、结果回放。

赛道设计，通过直线与弧的设计，将终点坐标与起点坐标首尾相连，构成闭合赛道。并可以调整赛道宽度，中心线宽度，以及赛道摩擦系数。如图。

赛车设计，包括车体的基本设置，车长，车宽，轮距，轴距，前轴距中心线距离，轮径，轮宽，以及整车质量和传动比；电机和舵机的参数设置。如图。

仿真模拟，通过加载赛道信息，赛车信息，以及控制策略 3 个文件，模拟车模行驶，记录行驶速度，偏差距离以及时间，并能够保存数据。

结果回放，回放仿真情况。

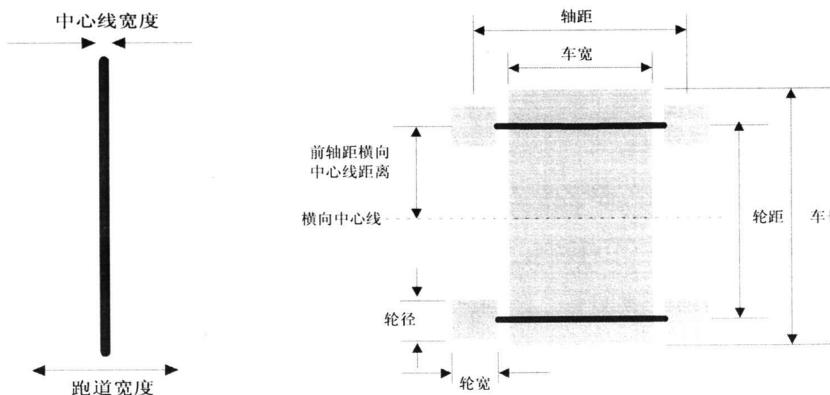


图 5-14 赛道参数

Figure 5-14 track parameters

图 5-15 赛车参数

Figure 5-15 racing parameters

在控制界面中设计跑道，生成.ppf 文件；设计赛车，生成.pcf 文件；并在 VC 中编写控制策略的 C 语言程序，生成.dll 文件用于决策程序加载。表为设定的跑道和赛车参数，并在下面给出了 PIDNN 控制部分的 C 语言程序。赛车在高速行驶时，为了提高电机的速度调节能力，因此控制策略设计时忽略积分神经元，这样也简化了网络结构，便于编程实现，同样在实际应用中，也将减少主控芯片的处理量。

表 5-4 参数设置
Table 5-4 Initial parameter settings

赛道宽度	中心线宽度	摩擦系数	车长	车宽	轮距	轴距	轮径
400mm	30mm	0.02	400mm	200mm	280mm	280mm	50mm
轮宽	前轴距横向中心线	整车质量	传动比	电机额定电压	电机额定电流	电机额定转速	电机额定转矩
50mm	130mm	2000g	0.5	7.2V	0.2A	3000rpm	10N·m
电枢回路电阻	电枢回路电感	舵机最大转角	转角变化速率				
0.2Ω	15mH	±35°	35°/s				

```
void PIDNNforward () //前向算法
```

```
{
```

```
//输入层
```

```
x_a1=u_a1=net_a1=r;
```

```
x_a2=u_a2=net_a2=y;
```

```
//隐含层
```

```
net_b1=x_a1*w11+x_a2*w21;
```

```
net_b3=x_a1*w13+x_a2*w23;
```

```
u_b1=net_b1;
```

```
u_b3=net_b3-net_b3;
```

```
x_b1=u_b1;
```

```
x_b3=u_b3;
```

```
//输出层
```

```
u_c=net_c=x_b1*w_1+x_b3*w_3;
```

```
x_c=u_c;
```

```
*MotorPWM=unsigned short(x_c);
```

```
}
```

```
void PIDNNback () //反传算法
```

```
{
```

```
y_=y-ynext;
```

```
if y_>0 y_=1;
```

```
if y_<0 y_=-1;
```

```
x_=x_c-x_cnext;
```

```
if x_>0 x_=1;
```

```
if x_<0 x_=-1;
```

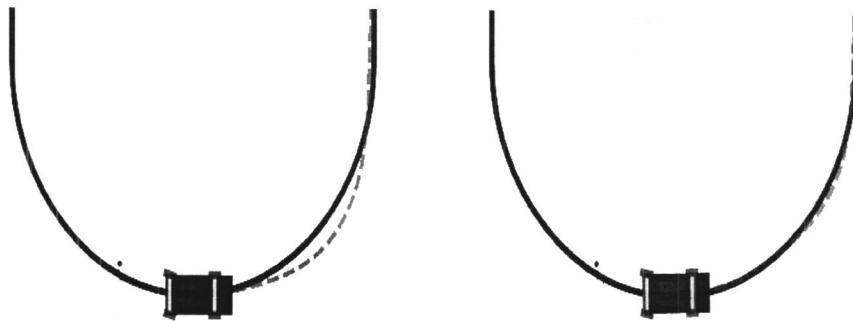
```

u_1=u_b1-u_b1next;
if u_1>0 u_1=1;
if u_1<0 u_1=-1;
u_3=u_b3-u_b3next;
if u_3>0 u_3=1;
if u_3<0 u_3=-1;
net_1=net_b1-net_b1next;
if net_1>0 net_1=1;
if net_1<0 net_1=-1;
net_3=net_b3-net_b3next;
if net_3>0 net_3=1;
if net_3<0 net_3=-1;
dw_1=-(r-y)*(y_*x_)*x_b1;
dw_3=-(r-y)*(y_*x_)*x_b3;
w_1=w_1-dw_1*steplength1;
w_3=w_3-dw_3*steplength1;
dw11=-(r-y)*(y_*x_)*w_1*(u_1*net_1)*x_a1;
dw13=-(r-y)*(y_*x_)*w_1*(u_3*net_3)*x_a1;
dw21=-(r-y)*(y_*x_)*w_3*(u_1*net_1)*x_a2;
dw23=-(r-y)*(y_*x_)*w_3*(u_3*net_3)*x_a2;
w11=w11-dw11*steplength2;
w13=w13-dw13*steplength2;
w21=w21-dw21*steplength2;
w23=w23-dw23*steplength2;
ynext=y;
x_cnext=x_c;
u_b1next=u_b1;
u_b3next=u_b3;
net_b1next=net_b1;
net_b3next=net_b3;
}

```

通过选取典型的大弯，小弯进行赛车行驶仿真，对比不同控制策略下赛车的行驶状况，仿真结果下图所示，对于比例系数(P)设置 0.11, 微分系数(D)设置 0.009。在经过大弯和小弯时，PID 控制下的赛车出现滑出赛道的情况，而 PIDNN

控制算法下赛车能很好地贴合跑道中心黑线。

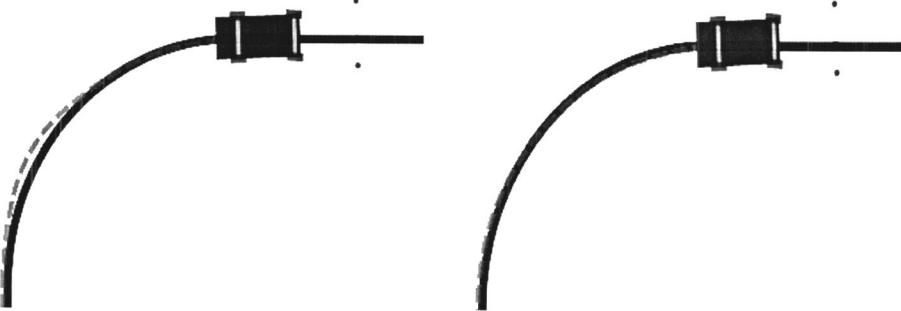


(1) PID 控制

(2) PIDNN 控制

图 5-16 仿真中大弯过弯路径

Figure 5-16 Big curve passing-path on simulation



(1) PID 控制

(2) PIDNN 控制

图 5-17 仿真中小弯过弯路径

Figure 5-17 Small curve passing-path on simulation

5.5 本章小结

本章由传统 PID 控制的介绍过渡到 PID 与神经网络结合的控制算法，并简单介绍了 PID 与神经网络相结合的三类应用，选取了其中易于工程实现，且具有学习能力的 PID 神经元网络作为控制算法，之后详细描述了 PIDNN 的构成，前向和反传算法，以及收敛性和稳定性。最后，将此算法用于 Plastid2 智能车仿真软件，通过与 PID 控制对比，证明此控制策略确实有效。

第六章 系统调试界面设计

6.1 MATLAB GUI 简介

通过图形实现用户与系统之间的交互界面称为图形用户界面(GUI)，图形用户界面系统是在 1983 年由苹果公司设计的 Lisa 电脑开创的，该电脑首次将图形用户界面 (GUI) 和鼠标结合起来，例如拖放鼠标技术、文件系统中的文件用图标来代替等，其操作理念被应用至今。MATLAB(Matrix Laboratory)计算机软件是由美国 MathWorks 公司于 1984 年推出的，通过自身良好的开放性和运行可靠性，如今已是国际公认的最优秀的数学应用与科学计算软件之一，经过多年的发展与完善，近几年 MATLAB 更成为在国内外相当流行的可视化科学计算软件。

MATLAB 句柄图形控制系统是 MATLAB 数据可视化的核心部分。使用 GUIDE 工具可以方便地使用句柄图形创建自己的图形用户界面 (GUI)，GUIDE 是一个专用于 GUI 程序设计的快速开发环境，具体实现一个 GUI 包括两个基本任务：一是 GUI 的组件布局，另一个是 GUI 组件编程，另外，用户还必须能够保存并发布自己的 GUI，使得用户开发的图形界面能够真正得以应用，所有这些功能都能够通过图形用户界面开发环境 GUIDE 来完成。GUIDE 首先是一个组件布局工具集，能够生成用户所需的组件资源并保存在一个 FIG 文件中；其次，GUIDE 还将生成一个包含 GUI 初始化和发布控制代码的 M 文件，该文件为回调函数(用户在图形界面中激活某一控件时要执行的函数)提供了一个框架。事实上，用户也可以通过编写调用组件函数的 M 文件来实现 GUI 中所有组件的布局，但是使用 GUIDE 交互式的组件布局功能将会大大减小工作量。GUIDE 可以为 GUI 同时生成 FIG 和 M 这两个文件^[60]。

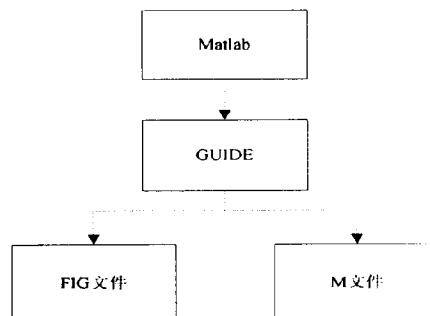


图 6-1 GUI 构成
Figure 6-1 the GUI Constitutes

FIG 文件包括 GUI 图形窗口和所有子对象（包括用户控件和坐标轴）的完全描述以及所有对象属性值。FIG 文件是一个二进制文件，调用 hgsave 命令或界面设计编辑器的 File 菜单 Save 选项保存图形窗口时将产生该文件。FIG 文件包含序列化的图形窗口对象，在用户打开 GUI 时，MATLAB 能够通过读取 FIG 文件重新构造图形窗口及其后裔。所有对象的属性都被设置为图形窗口创建时保存的属性：缺省情况下，即使用户使用 hgsave 和 hgload 命令保存系统缺省的图形工具条和菜单，FIG 文件也不保存这些缺省信息。FIG 文件最有用的地方之一就是对象句柄的保存和引用。

M 文件包括用户用来发布和控制界面和回调函数（这里作为子函数）的各种函数，主要用于控制 GUI 展开时的各种特征，该文件不包含任何组件的布置信息。M 文件可分为 GUI 初始化和回调函数两个部分，用户控件的回调函数根据用户与 GUI 的具体交互行为分别调用。GUIDE 可以根据用户 GUI 的版面设计过程直接自动生成 M 文件框架，这样就简化了 GUI 应用程序的创建工作，用户可以直接使用这个框架来编写自己的函数代码。这样的编写方法具有以下几个优点：应用程序 M 文件已经包含实现一些有用的函数编写代码，无需用户自行编写；可以使用该 M 文件生成的有效方法来管理图形对象句柄并执行回调函数子程序；提供管理全局数据的途径；文件支持自动插入回调函数原型，确保当前 GUI 与未来发布版本的兼容性。

6.2 系统调试

在实验车模调试过程中，为把行驶性能调到较好的状态，需要对车模的自适应性调速，跟踪精度，以及动态响应控制算法的稳定性等方面进行观测，采集数据以相应格式显示在图形用户界面中，通过对数据分析处理，调试界面能够以函数曲线等形式直观显示车模的各参数性能，发现车模行驶过程中可以改进的地方。整个调试过程，从数据采集至 MATLAB GUI 读入数据以及以函数曲线形式显示出来，构成智能车调试平台。调试平台的工作界面如下：

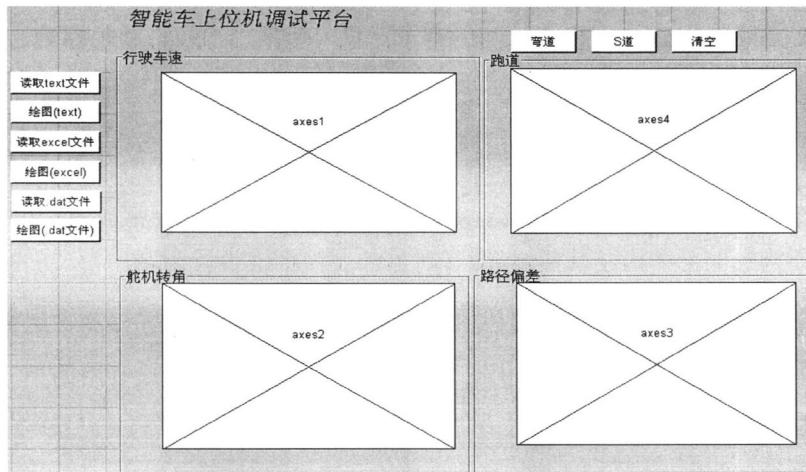


图 6-2 智能车系统调试界面
Figure 6-2 Smart car system debugging interface

在对SD卡进行写入时，需要考虑SD卡的写入形式，SD卡是以扇区为单位进行操作，即每次传送至少一个扇区的内容，而本设计中一个扇区定义的是512字节，故SD卡的操作单位是512个字节。MC9S12XS128有8KB的RAM空间，可将其中一部分作为数据缓冲区，当达到一定量的大小(512字节整数倍)后，再传输给SD卡。数据存储过程中一定要注意数据的类型，以保证在上位机恢复的时候不会出错，由于智能车控制中对数据精度要求不是很高，故大部分数据均以整型格式传输。

SD卡数据保存文件格式，一般有二进制文件(.DAT)和文本文件 (.TXT) 两类形式，文本文件需要注意，保存时需将保存的数据格式转换成字符型，直接保存数据会出现文本不识别的状况。本课题选用文本文件 (.TXT) 保存数据。

数据采集周期，通过A/D转换器的频率设置，可得车模的实际数据采集周期。而数据存储时间的设定，通过内部定时器PIT可以设置数据传递给数据缓冲区(数组)的时钟间隔，从而确定数据存储的时间周期。本文设置5ms时间为数据存储周期。数据组织格式如表6-1：

表6-1 数据组织格式
Table 6-1 Data organization format

数据大小： 10B	左电磁传 感器检测 值 (1B)	右电磁传 感器检测 值 (1B)	车模与跑 道位置偏 差值 (2B)	速度传感 器检测值 (2B)	舵机PWM 波控制值 (2B)	电机PWM 波控制值 (2B)
--------------	------------------------	------------------------	-------------------------	----------------------	-----------------------	-----------------------

当TXT数据文件成功写入SD卡之后，便需要考虑文件在上位机端的恢复问题，基于SD卡智能车调试平台使用Matlab作为上位机开发平台较为方便。上位机的功能是对下位机MC9S12XS128写入SD卡的TXT文件进行提取，分析，处理和显示。将

数据提取并显示出来，同时要能够绘制出行驶车速曲线，舵机转角控制曲线以及路径偏差曲线。

选取实验车模在 5.1S-6S 时间内通过弯道的行驶情况作分析，通过行驶路径图 6-6 可以看出，选取的是一段典型的由弯道入直道的行驶路径，在这段弯道上，舵机转角由-15° 逐渐增大到 0°，而行驶车速保持在 1.4m/s，车模与跑道中心线的偏差控制在 15mm 内，实验车模能够顺利完成跑道寻迹行驶。

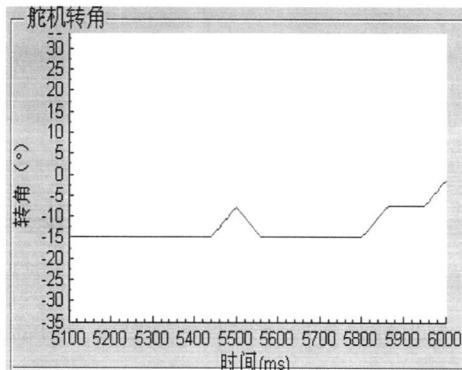


图 6-3 舵机转角
Figure 6-3 Servo corner

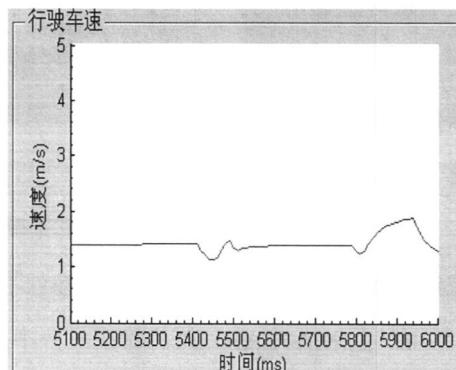


图 6-4 电机转速
Figure 6-4 motor speed

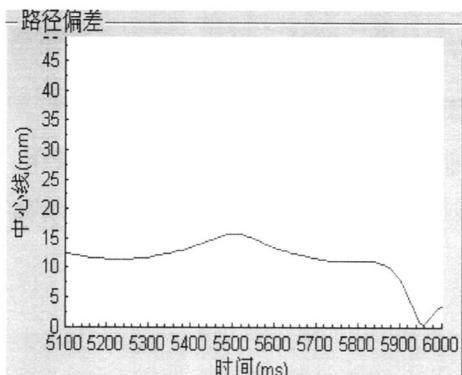


图 6-5 偏差变化
Figure 6-5 deviation change

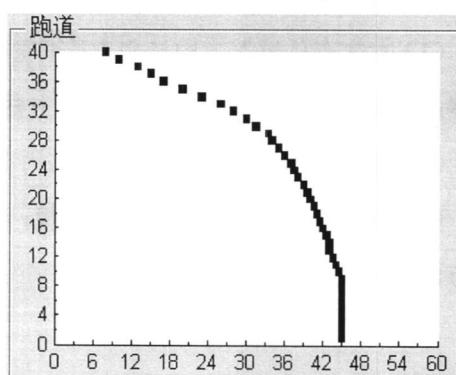


图 6-6 行驶路径
Figure 6-6 driving path

6.3 本章小结

本章简要地介绍了 Matlab 及其 GUI 的构成，并在此调试平台上，利用 SD 卡中储存的实验车模行驶数据信息，绘制出曲线图，以便直观地分析。

第七章 总结与展望

本文从智能车辆的应用与国内外的研究状况开始着手介绍，之后联系到实际课题，以实验车模作为研究对象，设计了整个运动控制系统。所做的工作如下：

- (1) 设计了硬件电路，其中包括电源管理模块，舵机、电机等驱动电路，以及电磁传感器。
- (2) 开发了基于 SD 卡的数据存储系统，可以将实验车模行驶过程中的数据保存下来，以便上位机调试时使用。分析了 SD 卡与主控芯片之间的数据通信，以及 FAT32 文件系统管理机制。
- (3) 设计了基于 PIDNN 的控制策略设计，重点讨论了 PIDNN 控制算法，并在仿真环境中，与 PID 控制方法进行了对比。
- (4) 在 MATLAB GUI 调试平台中，对行驶的数据进行了曲线描述，以便发现行驶中出现的问题。

由于能力有限，本文要存在许多不足之处，需要在以后进一步的提高：

- (1) 本文在传感器方面选用电磁传感器作为单一的检测手段，而导航技术的发展趋势是多传感器的交叉检测，这方面有许多工作值得研究。
- (2) SD 卡在应用时，仍有部分实用功能可以完善，这样更方便于实际使用。
- (3) 研究的实验车模只是小型车模，希望有机会在以后工作中能够做载人实车方面的研究。

参考文献

- [1] 徐友春,王荣本,李兵,李斌.世界智能车辆近况综述[J].汽车工程,2001,23(5):289-295.
- [2] 胡海峰,史忠科,徐德文.智能汽车发展研究[J].计算机应用研究,2004,6:78-80.
- [3] 李旭,张为公.智能车辆导航技术的研究进展[J].机器人技术与应用, 2007,4:25-27.
- [4] 付梦印,邓志红,刘彤.智能车辆导航技术[M].北京: 科学出版社,2009 年.
- [5] SanDisk Secure Digital Card – Product Manual Version 2.2[S]. Document No.80-13-00169 September,2004.
- [6] 张明亮,张宗杰.浅析FAT32文件系统[J].计算机与数字工程,2005,33(1):56-59.
- [7] 范寒柏,王少仙,彭安.大容量 SD 卡在海洋数据存储中的应用[J].单片机与嵌入式系统应用,2010,4:30-32.
- [8] 张华,吴欣,王大星.基于ARM7的SD卡读写控制在数据采集系统中的应用[J].电子技术应用,2009,7:38-41.
- [9] 陈卫明,廖成旺,邓涛,丁炜,王浩.基于ARM和SD卡的微动数据采集系统数据存储技术研究[J].大地测量与地球动力学,2009,29:114-119.
- [10] 韩辉,丛培田,郭颖.基于SD卡的多通道数据采集系统[J].工业仪表与自动化装置, 2010,1: 65-67.
- [11] 杨明极,陈方县,吴学君.嵌入式系统中SD卡的FAT32文件系统的设计[J].器件与电路,2010, 34(4):36-39.
- [12] 王建,张晓炜,杨锦,曾鑫,刘小勇.基于SD卡的智能车调试系统设计[J].工业控制计算机, 2010,23(8):48-49.
- [13] 楚彦君,巨林仓,刘军辉. PID神经网络控制器的设计及仿真研究[J].工业仪表与自动化装置,2001,2:3-6.
- [14] 吴全玉,张晓东,晁晓琪,徐宇宝.基于经典PID算法的智能车系统研究[J].自动化与仪器仪表,2011,1:25-26.
- [15] 王凯,谈英姿,周怡君.基于S12XS单片机的智能车研究——智能车调试平台的设计与实现[D].东南大学本科毕业论文,2010年.
- [16] 罗小刚,张炎,彭承琳.基于MSP430和SD卡的FAT16文件系统的设计[J].电子技术应用, 2008,11:65-68.
- [17] 何义,姚锡凡.基于BP网络自整定的PID控制在加工过程中的应用[J].控制与检测,2009, 10:48-51.
- [18] 何军旗,万少松,齐峰.基于BP遗传算法的PID控制器参数设计[J].弹箭与制导学报,2007, 27(5):53-58.
- [19] 张世峰,李鹏.基于神经网络的自整定PID控制器设计[J].自动化仪表, 2009,30(7):64-66.

- [20] 汪蓓蕾.基于鱼群算法和Hopfield网络的PID参数寻优[J].南京信息工程大学学报:自然科学版, 2009, 1(2) : 179-182.
- [21] 杜建凤,崔勇,张寒松,严忠慧.神经网络PID控制[J].北京科技大学学报,1998,20(6):571-575.
- [22] 高永吉,周洪利.基于SD卡的嵌入式文件系统的设计与实现[D].北京交通大学硕士学位论文,2010年.
- [23] 白金,韩俊伟.基于MATLAB/Simulink环境下的PID参数整定[J].哈尔滨商业大学学报(自然科学版),2007,23(6):673-676.
- [24] 何东健,刘忠超,范灵燕.基于MATLAB的PID控制器参数整定及仿真[J].西安科技大学学报,2006,26(4):511-514.
- [25] 高林中.基于MATLAB的PID控制器设计[J].山西煤炭管理干部学院学报,2007,4:150-151.
- [26] 白继平,许德辉.基于MATLAB下的PID控制仿真[J].中国航海,2004,4:77-80.
- [27] 王元哲,谈英姿.磁导航智能车路径信息采集系统的设计与实现[J].微型机与应用,2011,30(1):19-22.
- [28] 陈懂,刘璟.智能小车运动控制系统的研究与实现[D].东南大学硕士学位论文,2005.
- [29] 徐华水,解永生.基于FM357-2及路径记忆技术的探测车驱动系统.微计算机信息,2003,19(12):50-51.
- [30] 段颖康.基于MC9S12XS128单片机的智能循迹小车的硬件设计[J].新特器件应用,2010,12(1):33-35.
- [31] 刘伟.基于MC9S12XS128微控制器的智能车硬件设计[J].电子设计工程,2010,18 (1):102-105.
- [32] 张吴飚,马旭,卓晴.基于电磁场检测的寻线智能车设计[J].电子产品世界,2009,11:48-50.
- [33] 苏敏,梁银丽,汪道辉.基于路径记忆算法的智能小车控制系统的设计[J].哈尔滨理工大学学报, 2011,16(1):64-68.
- [34] 袁三男,汪剑光,张云峰,吕尧.基于智能车非匀速行驶记忆算法的研究和实现[J].电子产品世界,2008,7:152-153.
- [35] 杨秀春,游林儒.具有路径记忆功能的智能车控制系统设计[J].计算机工程, 2011,37(4):187-189.
- [36] 尹继武.智能车磁导航中的信号调理电路设计[J].电子设计工程,2010,18(7):122-124.
- [37] 艾宁,谭启寅,马文军,李宗洋.智能车运动状态实时监测系统的设计及实现[J].电子产品世界,2007,8:156-157.
- [38] 卜建国,徐友春,戴书泉,韩骏.智能车自主导航的神经网络转向控制算法[J].装甲兵工程学院学报,2007,21(4):66-71.
- [39] 史久根,王跃,韩江洪.exFAT文件系统的优化模型[J].计算机工程, 2009,35(16):54-58.
- [40] 李良,王绍国.FAT存储结构理论研究[J].电脑知识与技术,2009,5(21):6048-6049.

- [41] 张国军,侯玉峰,章振海,李书斌.基于ARM9和FAT的高速海量数据采集系统[J].计算机测量与控制,2011,19 (2):462-464.
- [42] 雷振伍,居鹤华,崔平远.BH-1型月球车的运动学和动力学研究[J].北京工业大学学报,2011,37(1):
- [43] Zhiyun Lin, Mireille Broucke, and Bruce Francis. Local Control Strategies for Groups of Mobile Autonomous Agents[J]. IEEE TRANSACTIONS ON AUTOMATIC CONTROL, 2004, 49(4):622-629.
- [44] Xiaohai Li, Jizhong Xiao, Jindong Tan. Modeling and Controller Design for Multiple Mobile Robots Formation Control[J]. Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics, 2004,8:838-843.
- [45] 巩书财,石磊,赵福全,李慧军,刘胜利,袁爽.柴油机曲轴系运动学和动力学仿真分析[J].拖拉机与农用运输车,2011,38(1):82-84.
- [46] 李伟.基于GPS导航的智能车转向的控制策略[J].现代计算机,2011,1:140-141.
- [47] 赵多兴.基于单片机的智能车底层控制系统设计[J].农机使用与维修,2011,1:28-30.
- [48] 李仕伯,马旭,卓晴.基于磁场检测的寻线小车传感器布局研究[J].电子产品世界,2009,12: 41-44.
- [49] 林焕新,胡跃明,陈安.基于自适应模糊控制的智能车控制系统研究[J].计算机测量与控制, 2011,19(1).
- [50] 孙绍杰,齐晓慧.一种基于模糊控制的智能车转向控制算法研究[J].微计算机应用. 2011, 32(1):26-30.
- [51] 甄红涛,齐晓慧.智能车参数自校正方向模糊控制器的设计[J].自动化仪表, 2010,31(9): 57-59.
- [52] 孙辉,楼佩煌,唐敦兵,陈旭玲.自适应摩擦驱动系统弯道运动特性分析[J].中国制造业信息化,2011,40(5):35-39.
- [53] 于娟,王子斌,范方灵.提高智能车速度的研究[D].电子科技大学工程硕士学位论.2010 年.
- [54] 刘啸森,陈昌.基于 MC9S12DG128 的智能模型车系统设计与仿真研究[D].大连理工大学硕士学位论文,2010 年.
- [55] 朱政合,潘学军.飞思卡尔智能赛车及调试平台开发研究[D].大连理工大学硕士学位论文,2010 年.
- [56] 王鹏宇,冯林.基于摄像头的智能小车决策系统研究[D].大连理工大学硕士学位论文,2009 年.
- [57] 刘祥庆,孙汉旭.水下球形机器人嵌入式运动控制系统研究[D].北京邮电大学硕士研究生学位论文,2011 年.
- [58] 徐建功,赵捷.Cortex 系统中FAT32文件读写的硬件及软件设计[D].山东师范大学硕士学位论文,2011 年.

- [59] 卓晴,黄开胜,邵贝贝.学做智能车——挑战“飞思卡尔”杯[M].北京:北京航空航天大学出版社,2007年.
- [60] 苏金明,阮沈勇.MATLAB实用教程[M].北京:电子工业出版社,2007年.
- [61] 马培娇,张宇光,姚永兴等.高速大容量存储系统的关键技术实现[J].化工自动化及仪表,2011,38(7):869-872.
- [62] 舒怀林.PID神经元网络及其控制系统[M].北京:国防工业出版社,2006年.
- [63] 孙同景,陈桂友.Freescale 9S12十六位单片机原理及嵌入式开发技术[M].北京:机械工业出版社,2008年.
- [64] 孙振平,贺汉根.自动驾驶汽车智能控制系统[D].国防科学技术大学博士学位论文,2004.
- [65] 李飞,张井岗.基于LabVIEW环境下的可移植数据采集系统[J].化工自动化及仪表,2011年,38(9):994-997.

附录

```
% PIDNN 阶跃响应测试 M 文件程序
function [u]=pidnnnsimf(u1,u2,u3)
persistent errori error_1 w11 w12 w13 w21 w22 w23 w_1 w_2 w_3 u_b2
ynext x_cnext u_b1next u_b2next u_b3next net_b1next net_b2next
net_b3next steplength1 steplength2
if u1==0
w11=1;
w12=1;
w13=1;
w21=-1;
w22=-1;
w23=-1;
w_1=10;
w_2=0.01;
w_3=0.5;
u_b2=0;
net_b3next=0;
steplength1=0.001;
steplength2=0.001;
ynext=0;
x_cnext=0;
u_b1next=0;
u_b2next=0;
u_b3next=0;
net_b1next=0;
net_b2next=0;
net_b3next=0;
end
r=u2;
y=u3;

%前向算法
net_a1=r;
u_a1=net_a1;
x_a1=u_a1;
net_a2=y;
u_a2=net_a2;
x_a2=u_a2;
net_b1=x_a1*w11+x_a2*w21;
net_b2=x_a1*w12+x_a2*w22;
```

```

net_b3=x_a1*w13+x_a2*w23;
u_b1=net_b1;
u_b2=u_b2+net_b2;
u_b3=net_b3-net_b3next;
x_b1=u_b1;
if x_b1>1
x_b1=1;
end
if x_b1<-1
x_b1=-1;
end
x_b2=u_b2;
if x_b2>1
x_b2=1;
end
if x_b2<-1
x_b2=-1;
end
x_b3=u_b3;
if x_b3>1
x_b3=1;
end
if x_b3<-1
x_b3=-1;
end
net_c=x_b1*w_1+x_b2*w_2+x_b3*w_3;
u_c=net_c;
x_c=u_c;
if x_c>1
x_c=1;
end
if x_c<-1
x_c=-1;
end
u=x_c;

```

%反传算法

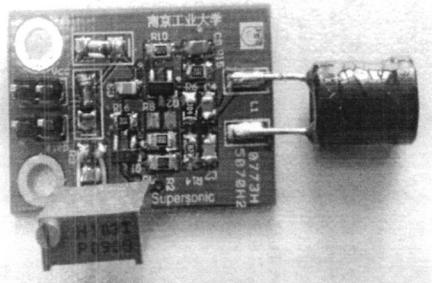
```

y_=y-ynext;
j=(r-y)*(r-y)/2;
if y_>0
y_=1;
end
if y_<0
y_=-1;

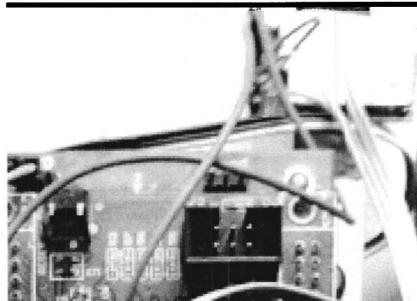
```

```
end
x_=x_c-x_cnext;
if x_>0
x_=1;
end
if x_<0
x_=-1;
end
u_1=u_b1-u_b1next;
if u_1>0
u_1=1;
end
if u_1<0
u_1=-1;
end
u_2=u_b2-u_b2next;
if u_2>0
u_2=1;
end
if u_2<0
u_2=-1;
end
u_3=u_b3-u_b3next;
if u_3>0
u_3=1;
end
if u_3<0
u_3=-1;
end
net_1=net_b1-net_b1next;
if net_1>0
net_1=1;
end
if net_1<0
net_1=-1;
end
net_2=net_b2-net_b2next;
if net_2>0
net_2=1;
end
if net_2<0
net_2=-1;
end
net_3=net_b3-net_b3next;
```

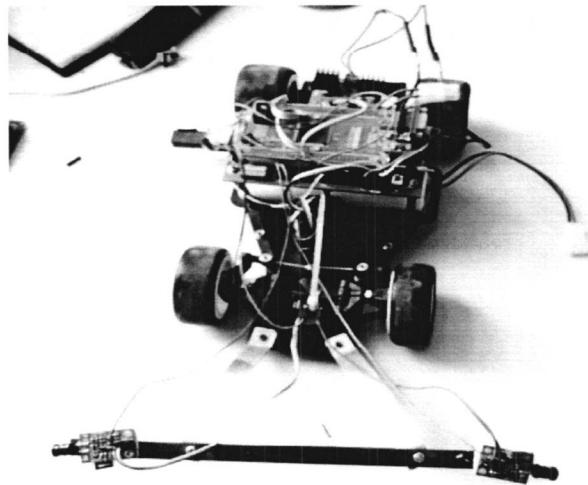
```
if net_3>0
net_3=1;
end
if net_3<0
net_3=-1;
end
if j>0.01
dw_1=-(r-y)*(y_*x_)*x_b1;
dw_2=-(r-y)*(y_*x_)*x_b2;
dw_3=-(r-y)*(y_*x_)*x_b3;
w_1=w_1-dw_1*steplength1;
w_2=w_2-dw_2*steplength1;
w_3=w_3-dw_3*steplength1;
dw11=-(r-y)*(y_*x_)*w_1*(u_1*net_1)*x_a1;
dw12=-(r-y)*(y_*x_)*w_2*(u_2*net_2)*x_a1;
dw13=-(r-y)*(y_*x_)*w_3*(u_3*net_3)*x_a1;
dw21=-(r-y)*(y_*x_)*w_1*(u_1*net_1)*x_a2;
dw22=-(r-y)*(y_*x_)*w_2*(u_2*net_3)*x_a2;
dw23=-(r-y)*(y_*x_)*w_3*(u_3*net_3)*x_a2;
w11=w11-dw11*steplength2;
w12=w12-dw12*steplength2;
w13=w13-dw13*steplength2;
w21=w21-dw21*steplength2;
w22=w22-dw22*steplength2;
w23=w23-dw23*steplength2;
ynext=y;
x_cnext=x_c;
u_b1next=u_b1;
u_b2next=u_b2;
u_b3next=u_b3;
net_b1next=net_b1;
net_b2next=net_b2;
net_b3next=net_b3;
end
```



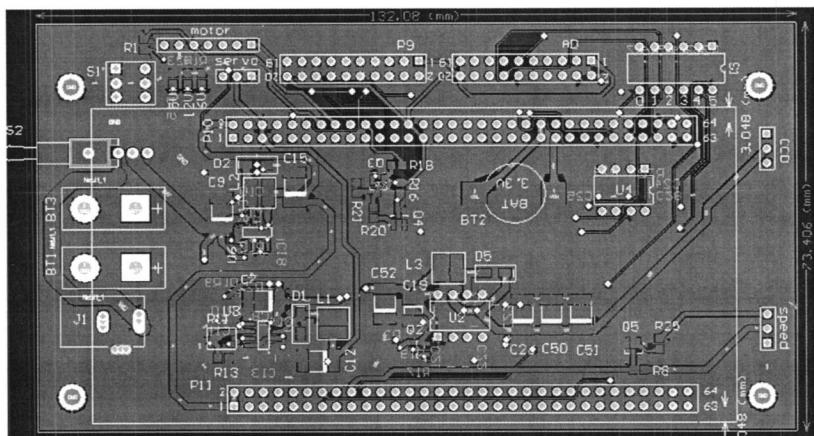
(1) 电磁传感器



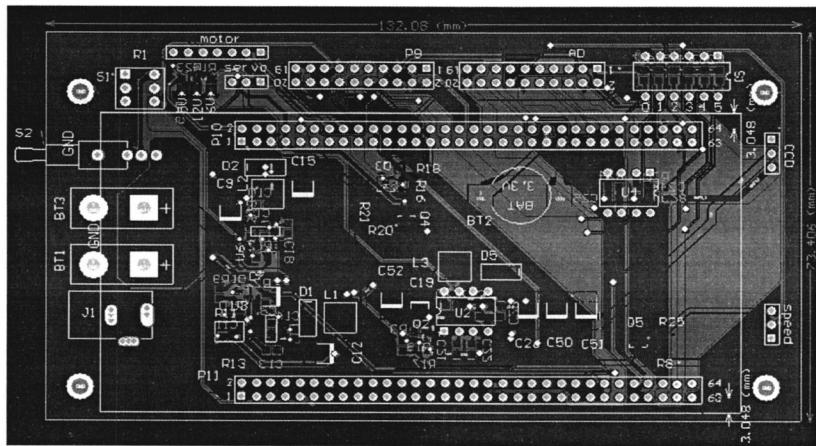
(2) SD 卡卡槽



(3) 实验车模



(4) 主控芯片扩展板 PCB 图 (上层)



(5) 主控芯片扩展板 PCB 图（下层）

发表论文

[1] 周跃,沈捷,花魁.基于 SD 卡的数据存储系统设计[J].化工自动化及仪表,2012,39(1):95-98.

致 谢

研究生三年生活一晃即过，已经到了毕业的时刻，回想这三年，自己在学习和生活上有许多地方做的不够好。导师和宿舍同学对我帮助甚大，他们时不时地督促我，使我能够顺利完成课题研究和论文撰写。

首先要感谢导师沈捷对我的悉心指导和帮助，沈老师扎实的知识，严谨的治学态度和认真的工作精神给我留下了深刻的印象。我也从中学习到，应该抱着怎样的态度来面对知识。

其次要感谢宿舍同学，王守振，黄锋，庞龙，谢谢你们陪我度过艰难的一段日子。感谢峰哥，阿贵，石头，董科，韩丰涛，焦敏，张林宝，小贝，孙振兴等等，谢谢你们对我学业和生活上的帮助，跟你们在一起很愉快。

感谢实验室的师弟们，花魁，周柱，孙文林，以及在南理工教书的师兄李盛辉，谢谢你们对我课题上的帮忙。

感谢父母，谢谢你们一直默默的支持我，也真心地感谢刘林在研一，研二阶段对我的帮助。

最后感谢自动化与电气工程学院的老师们！