

工学硕士学位论文

多媒体家庭网关的系统设计与实现

张世琦

哈尔滨工业大学

2008 年 12 月

国内图书分类号: TN47

国际图书分类号: 621.3.049.77

工学硕士学位论文

多媒体家庭网关的系统设计与实现

硕士研究生: 张世琦

导师: 张岩教授

申请学位: 工学硕士

学科、专业: 微电子学与固体电子学

所在单位: 深圳研究生院

答辩日期: 2008 年 12 月

授予学位单位: 哈尔滨工业大学

Classified Index: TN47

U.D.C: 621.3.049.77

Dissertation for the Master Degree of Engineering

System Design and Implementation of Multimedia Home Gateway

Candidate:	Zhang Shiqi
Supervisor:	Prof. Zhang Yan
Academic Degree Applied for:	Master of Engineering
Specialty:	Microelectronics and Solid-State Electronics
Affiliation:	Shenzhen Graduate School
Date of Defence:	December, 2008
Degree-Conferring-Institution:	Harbin Institute of Technology

摘要

以网络为代表的信息技术不断发展,嵌入式计算机的应用也越来越广,几乎任何人在任何时间和任何地点都可以随时享受到嵌入式计算带来的便捷。相关的很多热点问题吸引了大批的科研人员,其中家庭的数字化和智能化就是其中的热点之一。要想对家庭进行集中地控制,家庭网关是必不可少的。它完成了家庭与外界网络的连接,同时对家庭内部的数字家电进行控制。本文提出的多媒体家庭网关(Multimedia Home Gateway)将在家庭内部负责多媒体数据的传输、存储以及数字家电的控制,同时完成与家庭外部的网络连接,是智能家庭、数字家庭的控制中心。

本文根据目前市场上已有的数字家庭产品和标准以及其不足,首先介绍了多媒体家庭网关的设计目标和设计方案。其中包括系统功能的分析和主要功能模块及其芯片的确定。接着介绍了 MHG 硬件平台的搭建,印刷电路板的制作,以及软件平台的设计与实现。该设计平台契合需求,结构良好,并且具有良好的可扩展性,为后续工作搭建了很好的平台。

此外,针对传统家电无线控制器按键过多,操作复杂的缺点,本文还提出了一种基于惯性器件的新型遥控器。用户通过手持该遥控器,并完成一些很象形的动作,就可以实现传统遥控器的全部功能,无需任何按键的辅助。该控制系统在 J2ME 平台上运行良好,为将来实现无按键的家电控制打下了良好的基础。针对基于加速度计的字符识别,本文还进行了三种特征提取算法的比较。根据大量实验数据,比较得出了在不同条件下,不同方法的优、缺点。

关键词 多媒体家庭网关; 机顶盒; 手势识别; 嵌入式 Linux

Abstract

Information technologies, including internet, have been developing for years, and the applications of embedded computers are also widely used in an unprecedented way. Right now, nearly any one could enjoy the convenience of embedded computing anywhere at any time. Many related hot subjects attracted lots of researcher in this area, and the digitalizing and intelligentizing of modern houses are just one of the hottest subjects. To control a house in a central way, a home gateway is indispensable, because it realize the connection between the inside house and the outside network, and it can also control the digital appliances. MHG (Multimedia Home Gateway) is developed in this thesis, which completes the transferring and storage of multimedia data, the control of digital appliances and the connection with the network outside. So, MHG is the control center of intelligent house and digital house.

In consideration of the existing standards about the products of digital house and their drawbacks now, this design's goals and outline are introduced at first. This section includes the detailed descriptions about system functions, main modules and their chips. After that, the construction of MHG's hardware platform will be presented, which contains how to make the PCB (Printed Circuit Board) and the hardware-related software design and realization. This platform matches the need on market very much, has a good system structure and excellent extensibility at the same time, from this point of view this is a good foundation for the follow-up development.

Moreover, it is noticed that the traditional wireless remote controller has too many keys and is inconvenient to operate, so a novel remote controller based on inertial sensors is introduced in this thesis. Taking this controller in hand, users can give all the well-known instructions through some visual motions, and any key or button is redundant. This control system works stably and shows high performance on J2ME platform, and this is a very good foundation for keyless appliance control in the future. Besides, three feature extraction algorithms are compared on character recognition based on acceleration. According to lots of experiments, the conclusion about the pros and cons for these extraction methods

Error! Use the Home tab to apply 标题,章标题(无序号) to the text that you want to appear here.

is brought out.

Keywords multimedia home gateway (MHG), set top box (STB), gesture recognition, embedded Linux

目录

摘要	I
Abstract	II
第 1 章 绪论	1
1.1 课题背景	1
1.1.1 发展现状	1
1.1.2 多媒体家庭网关课题的提出	2
1.2 相关技术及发展	3
1.2.1 平台标准 MHP	4
1.2.2 嵌入式系统发展现状	7
1.2.3 PowerPC 技术	7
1.3 论文的主要内容	8
第 2 章 MHG 硬件平台及调试	10
2.1 媒体处理芯片 STB04500	10
2.1.1 处理器子系统	10
2.1.2 外设子系统	12
2.1.3 存储器控制子系统	14
2.1.4 多媒体处理子系统	17
2.2 Redwood5 评价板	19
2.3 MHG 硬件平台	20
2.3.1 FLASH 存储模块	21
2.3.2 串口通信接口电路	23
2.3.3 调谐器接口电路	24
2.3.4 以太网接口电路	24
2.4 MHG 硬件调试	24
2.5 本章小结	25
第 3 章 MHG 系统软件平台	26
3.1 OpenBios 介绍	26
3.2 在 OpenBios 中驱动调谐解调器	28
3.3 嵌入式 Linux	30
3.4 应用软件	31

3.5 本章小结	32
第 4 章 基于惯性器件的无按键遥控器	34
4.1 背景介绍	34
4.2 加速度传感器	35
4.3 字符识别算法	36
4.3.1 预处理	37
4.3.2 特征提取	39
4.3.3 离散隐马尔科夫模型	43
4.4 实验结果	45
4.5 本章小结	50
结论	51
参考文献	52
附录 1	56
附录 2	60
攻读学位期间发表的学术论文	64
致谢	66

第1章 绪论

1.1 课题背景

当今社会，网络飞速发展，如何实现家庭的网络化、智能化是一个极其重要的课题。数字家庭的概念在上世纪 90 年代初被提出，当时市场反应冷淡。在经过一段时间的市场培育后，数字家庭在近几年又活跃了起来。数字家庭正成为全球热门话题。

那么，什么是数字家庭呢？广义地讲，它是在家庭内部建立起融合家庭控制网络和多媒体信息网络于一体的家庭信息化平台，是在家庭范围内，实现信息设备、通信设备、娱乐设备、家用电器、自动化设备、照明设备、保安（监控）装置及水电气热表设备、家庭求助报警等设备互连和管理，以及数据和多媒体信息共享的系统。用户的需求可以是有限范围内多个设备之间的信息流通，也可以是有限范围内多个设备与公共网络之间的信息流通，甚至可以是有限范围内的所有设备之间以及这些设备与公共网络之间的信息流通。其实，数字家庭更多的是偏重一种数字化生活的理念。该理念认为，未来的生活应该是一种无所不在（Ubiquitous）的境界，即任何时间，任何地点都能获得你想要的信息。数字家庭是这个理念的缩影。

在可以预见的将来，在线影音娱乐、网络游戏市场等服务内容将不断增加，多个电视台的数字电视节目陆续开通，家用电器控制家居安全监控以及更多 3G 业务应用也变得更加丰富。随着人们对家庭内部共享数字媒体需求的进一步增加以及人们消费方式的逐渐改变，数字家庭的业务与应用前景是相当诱人的。

1.1.1 发展现状

数字家庭的发展主要经历了以下三个阶段：第一阶段主要以信息分享为主；第二阶段关注于信息分享并整合娱乐网络；第三阶段着力于信息分享、整合娱乐网络及控制网络。

数字技术的飞速发展不断催生着众多的数码产品，个人电脑、PDA、手

机、数字电视、DVD、MP3、DV/DC 等数字产品逐步进入普通生活中，与此同时，宽带通信与互联网的普及极大地促进了数字媒体内容的发展。而今，除了家电产品数字化的革新之外，构建一个宽带接入、内部互联、内容共享的家庭网络需求也日趋突显。

从 1994 年数字家庭概念的诞生，到今天数字家庭市场的正式启动；从“维纳斯”计划的推出，到 UPnP 论坛、DLNA 数字家庭组织的成立，历经十余年发展，数字化生活已步入真正的发展期，数字家庭网络离我们又近了一步。

1.1.2 多媒体家庭网关课题的提出

计算机几十年的快速发展的过程中，其体积越来越小、性能越来越强、应用领域越来越广、操作越来越简单，而价格却越来越便宜。最初的计算机体积庞大，管理操作复杂，需要专业的技术人员进行维护，而现在，普通人就可以很快学会基本的计算机操作。IT(Information Technology)技术如此迅猛的发展，使得计算机从财力雄厚的大公司和机构走到个人家庭已成为现实。随着 PC、notebook 以致 Pocket PC、PDA、Smart Phone 已经发展成熟并开始普及，计算机从主机到桌面再到掌上的发展趋势表明计算机的性能价格比迅速提高，计算机能涉及的领域逐渐扩大到了普通家庭，家用电器的一场革命呼之欲出。从老式的唱片机到使用卡式磁带的 Walkman 到家庭录像机 VCR 再到现在的 Digital Camera、Digital Video、Digital TV 以及 VCD、DVD 播放机，多媒体产品从模拟化走向数字化，家庭用户对多媒体处理能力的要求也越来越高。从简单的音频、视频的播放到声、文、图信息综合处理的需求，如何提高家庭内部的多媒体数据处理能力成了一个亟待解决的难题。对于这个问题，计算的嵌入化提供了一个很好的解决方案。所谓计算的嵌入化，就是把计算扩展到原来不具有计算的设备（如电视机、电话、空调机、微波炉等）中，它大大地扩展了“计算机”的含义。随着嵌入式芯片处理能力的提高，便能相应的提高家庭内部多媒体数据处理能力。

另一方面，Internet 的迅速发展将世界互联，无处不在的网，无处不在的计算,成为计算机技术发展的潮流，而家庭在 Internet 中一直扮演着“终

端”的角色。然而随着数字家电的涌现，家庭内部信息交流的需求越来越大，亟待一种家庭网络系统来满足日益增长的信息交流需求。家庭网络系统就是将家庭中各种与信息相关的通信设备、家用电器和保安装置，通过家庭总线技术连接到一个家庭智能化系统上进行集中的或异地的监视、控制和家庭事务性管理，并使家庭里的各种设备之间保持有机联系，保持这些家庭设施与住宅环境的和谐与协调。通过家庭网络系统，家用电器的管理、使用和维护更为集中、便捷和智能化。

结合这两个方面的需求，我们提出了家庭多媒体网关 **MHG** 的概念，即通过嵌入式系统提供的强大的数据处理功能，在家庭内部建立统一的多媒体数据处理中心，并同时通过家庭网络负责对家庭内部多媒体数据的传输、存储进行管理。此外，对于未来可能出现的各种新型的数字家电，也能够在这个平台上通过家庭网络进行集中和异地的监视、控制和事务性管理。**MHG** 起源于 **SOPCA** 平台，然而从机顶盒这一概念开始发展，**MHG** 最终发展目标是要成为真正意义上的家庭服务器（**Home Server**）。多媒体家庭网关的主要应用可以分为以下几类：

1. 数据中心：家庭内部各个设备的数据之间的存贮、传输与共享。
2. 媒体中心：对家庭中多媒体影音视听设备，比如 **TV**、**DVD**、**DC**、**DV**、**mp3** 等设备的播放及控制。
3. 控制中心：通过家庭网络来管理家庭内部各个电器设备，达到远程控制和监控的目的。通过连接 **Internet** 网络和家用网络来控制其它智能家电。
4. 通信中心：主要是家庭内部网络之间的通信，配置和管理家庭内部网络。

正是根据以上列出的设计目标，参考了 **IBM Redwood5** 评价板^[5]，实验室自行设计制造了 **MHG** 多媒体家庭网关平台。

1.2 相关技术及发展

随着计算机及通信技术所取得的日新月异的进展，新的技术层出不穷。为了让 **MHG** 平台在适应现有的应用需要的同时并保持一定的技术前瞻性，

我们对与 MHP 密切相关的技术：通信技术、多媒体压缩技术及嵌入式技术进行了调查研究。

1.2.1 平台标准 MHP

MHP 是由 DVB 标准扩展的家庭多媒体平台标准。讨论建立 MHP 标准的工作最早从 1998 年开始，2000 年 1 月公布了 MHP 标准规范的第一个版本 MHP1.0。

DVB 系统为基于 MPEG2 标准的数字视频广播提供了广泛全面的解决方案，同时也制定了一些相关的标准，这些解决方案以及标准使得数字视频广播能够通过包括卫星、有线、微波等等多种媒介传播，同时 DVB 也在对交互式服务应用进行相关的研究。MHP 作为 DVB 项目的扩展，主要研究数字视频广播中的交互式应用，研究范围主要包括应用程序互操作、应用服务下载、系统可扩展性以及系统可升级性等等。

MHP 为数字视频广播终端用户提供技术解决方案，该解决方案使各类服务供应商（包括应用软件开发商、制片商及广播公司等）能共同运转在一个中立的框架下，这个中立的框架还能够兼容以前的运营模式。通过 MHP，多个不同种类的服务供应商所提供的应用服务可以很好的在一个横向的市场中分工合作，应用软件、网络以及 MHP 终端设备等等因素都可以由各个相互独立的供应商提供支持。

MHP 平台的基本结构包括三层：资源层、系统软件层以及应用层。如图 1-1 所示。

资源层：即底层的硬件资源，包括机顶盒的硬件（MPEG 解码、输入/输出设备、CPU、内存、图形显示等）、驱动程序和操作系统等，交互应用通过调用这些资源得以实现，MHP 标准没有规定具体如何实现这些硬件/软件资源。

系统软件层：包括 MHP-API、应用管理器和 Java 虚拟机。该层构成了一个抽象的资源层，并将它提供给应用层。Java 虚拟机为交互应用在不同的硬件和操作系统上的运行提供了标准接口，应用管理器负责管理所有应用程序从启动到结束的生命周期同时负责协调多个应用程序的协同工作，

MHP API 将系统软件层的抽象资源以接口形式提供给上层的应用层。

应用层：应用层包括多个协同工作的应用程序和以及少量应用程序运行时所用到的库，这些应用只能通过 MHP 的 API 访问资源层。

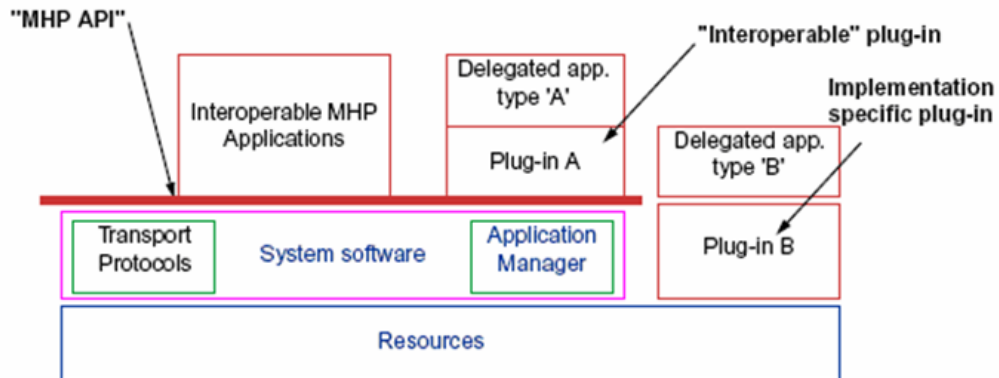


图 1-1 MHP 基本模型结构

Fig. 1-1 General Model Structure of MHP

MHP 标准中定义了 3 个不同应用层次和领域的 profile^[6]，各个 profile 结构以及相互之间的联系如图 1-2 所示。

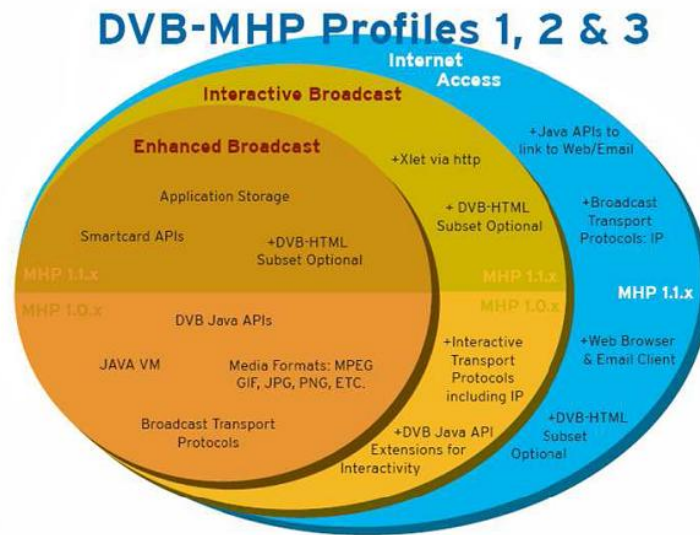


图 1-2 MHP 层次结构

Fig. 1-2 MHP Profiles

其中的虚线框表示了各层次领域间的嵌套关系，各层次领域描述如下。

增强广播：在视音频服务的广播基础之上加入了本地内的交互功能，该层次的应用不需要回传信道，它只需下载应用后，在本地与视音频数据实现交互，这是一种伪交互。

交互广播：是增强广播层的超集，需要额外的回传信道支持，提供独立于视音频广播服务的交互功能。

Internet 接入：在前两者基础之上加入 WEB 浏览、Email 收发、Chating 等 Internet 功能的支持。

通过 MHP 平台，用户可以从多个服务供应商那里获得丰富的应用服务，包括电子节目指南 EPG (Electronic Program Guide)、信息服务 (图文电视、新闻提醒等)、电视内容同步信息 (记分牌、电视互动游戏)、安全可靠的电子商务、交易、家庭银行等等，而软硬件开发商以及服务供应商也可以在 MHP 标准下独立的提供自己的专业产品和服务。

基于 DVB 平台设计的 MHP 获得了很大的成功，于是许多组织希望能够基于自己的广播系统定制和 MHP 兼容的交互式电视标准。于是 GEM (即 Globally Executable MHP) 便应运而生了。

GEM 不是一个能够直接实现的标准，而是一个定义 MHP 兼容标准的框架，它列出了 MHP 中和 DVB 相关的部分，各个组织根据自己的情况对这些部分进行替换，即可定义出和 MHP 兼容的交互式电视标准。基于 GEM 的交互式电视标准包括 OCAP (Open Cable Application Platform)，ACAP (Advanced Common Application Platform) 等，它们和 GEM 的关系如图 1-3 所示。

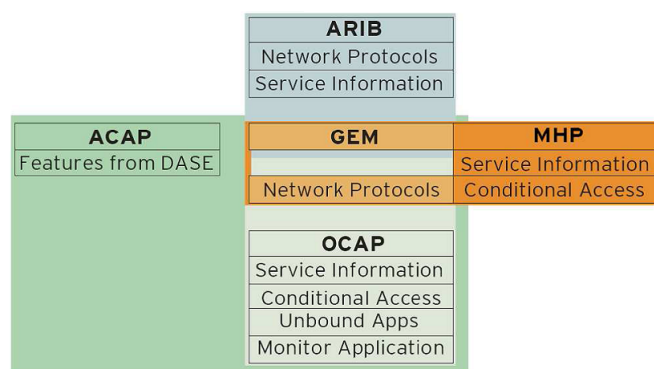


图 1-3 GEM 与交互电视标准的关系

Fig. 1-3 Relationship Between GEM and Interactive TV Standards

1.2.2 嵌入式系统发展现状

嵌入式系统被定义为：以应用为中心、以计算机技术为基础、软硬件可剪裁、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。嵌入式系统的最大特点是其所具有的目的性和针对性，即每一套嵌入式系统的开发设计都有其特殊的应用场合与特定功能，这也是嵌入式系统与普通计算机的最大区别。

随着信息化、智能化、网络化的发展，嵌入式系统技术也将获得广阔的发展空间。美国著名未来学家葛洛庞帝 1999 年 1 月访华时曾预言，4-5 年后嵌入式智能工具将是 PC 和因特网之后最伟大发明。进入 20 世纪 90 年代，嵌入式技术全面展开，目前已成为通信和消费类产品的共同发展方向。在通信领域，数字技术正全面取代模拟技术。在广播电视领域，全球许多国家和地区正在由模拟电视向数字电视转变，欧洲的 DVB 标准已在全球许多国家推广。这些产品都离不开嵌入式系统技术，嵌入式系统带来的工业产值已经超过了一万亿美元。

嵌入式系统技术的迅速发展使之成为当前微电子技术和计算机技术中的一个重要分支，嵌入式系统更是广泛的应用于金融、航天、电信、网络、信息家电、医疗、工业控制、军事等各个领域，甚至有些学者断言，嵌入式技术将成为后 PC 时代的主宰。

1.2.3 PowerPC 技术

嵌入式微处理器 EMPU (Embedded MicroProcessor Unit) 是嵌入式系统的核心部件，目前按指令体系分主要有 ARM、PowerPC、MIPS、X86 系列。其中 PowerPC 是 IBM、Motorola、Apple 三家公司于 1993 年联合开发的新一代微处理器，由于采用了 RISC 技术，其性能价格比从一开始就超越了竞争对手——采用 CISC 技术的 X86 系列处理器。PowerPC 的体系结构的可伸缩性使其既可用于高端服务器，也可用于嵌入式系统。

基于 RISC 技术的 PowerPC 的特性主要有：

- 1、采用固定长度的指令、使机器译码变得简单。不利的影响是完成相同的工作需要更多的指令，但这可以通过快速、价格低廉的存储装置更快的

执行较大的代码段以及流水线技术来弥补。

- 2、采用装载/存储的体系结构，操作数和运算结果不通过存储器直接取回，而通过大量标量/矢量寄存器取回。
- 3、采用硬联控制，用硬件实现指令尽管更加昂贵，但在运行时间上能提供更好的性能平衡，另外还节省了用于存储微代码的空间，并消除了微代码翻译的时间。
- 4、硬联控制还导致了更高的精度，这就促进了溶合指令在 PowerPC 体系结构设计上的实现。将某些经常出现的指令作为溶合指令执行，时钟周期上的节省可以加倍。
- 5、PowerPC 采用流水线执行指令的体系结构，对于流水线执行的空区，PowerPC 采用分支预测的方法，另外除了这个并行机制，每个独立的执行部件还是流水线式工作，具有同时处理多条指令的能力，称为超标量体系结构，在付出了指令执行机制的复杂性的代价后，PowerPC 在指令执行速率上产生很高的效率。

本系统采用的 STB04500 微处理内嵌 PowerPC405D4 内核，集成了 16KB 的指令缓存以及 16KB 数据缓存，通用终端控制器和内存管理单元 MMU。

1.3 论文的主要内容

本文中主要讨论了多媒体家庭网关硬件平台的搭建，相应软件平台的设计与实现，以及一种无按键无线控制器的设计。

首先阐述了 MHG 系统的总体结构设计，详细分析了核心芯片 STB04500 的功能模块，在此基础上定义了 MHG 系统的功能模块，将整个系统划分为 SDRAM 存储模块，FLASH 存储模块，以太网接口模块，EBIU 缓冲接口模块，IDE 接口模块，TUNER 接口模块，音视频输出接口模块，JTAG 接口模块，串口通信模块，红外接收模块等。并据此进行需求分析，确定各个外部模块的重要元件。然后在 EDA 工具 Protel DXP 环境下完成了整个系统硬件电路的原理图设计，并且在此基础上完成了四层印制电路板版图布局布线设计。

接着介绍了多媒体家庭网关软件平台的设计。基于 IBM Redwood5 评价板及自行设计的硬件平台完成了多媒体家庭网关软件平台的设计。实现了一

个基本的 DVB STB 系统，加入了无线网络支持，为多媒体数据的传输和 3C 产品的整合提供基础。

传统的遥控器按键多、操作不便，对视觉障碍的使用者和老人很不友好。本文设计了基于惯性器件的控制系统，该系统基于三轴微加速度计，具有功耗低、体积小、灵敏度高优点。基于惯性器件的 MHG 控制器没有任何按键，用户只需手持该控制器，完成特定的动作就可以像传统遥控器一样控制 MHG 平台。本文同时对特征提取方法和隐马尔科夫模型结构进行了一系列的比较，基于大量的实验数据，对不同场合的使用提出了建议。

最后一章对作者硕士期间的工作进行了总结。

第2章 MHG 硬件平台及调试

目前 MHG 项目的硬件平台主要是 IBM Redwood5 评价板, 以及实验室自行设计开发的硬件平台, 这两种平台的核心都是基于 PowerPC 的 IBM 多媒体处理器芯片 STB04500。本章首先介绍 IBM Redwood5 评价板, 其中包括对 STB04500 主要功能的说明, 接着介绍了实验室自行开发的 MHG 硬件平台, 最后对硬件调试的基本工具和方法进行简要的说明^[8]。

2.1 媒体处理芯片 STB04500

IBM STB04xxx^[7]系列是专为机顶盒产品设计的集成控制芯片, 是 IBM 公司机顶盒系列第三代产品的一部分。PowerPC 处理器和外部 I/O 结构提供了很强的扩展性, 相比同类产品, 其音频和视频子系统也是很强大的。

STB04500 芯片的各个部件按照功能可以划分为几个模块, 每个模块又对应于芯片的一部分实际部件, 其结构如图 2-1 所示, 其中不同的模块用不同的颜色加以区分。下面将分别介绍 STB04500 芯片中各个功能模块的结构和工作原理。

2.1.1 处理器子系统

处理器子系统是整个硬件系统的核心, 其结构如图 2-2 所示。主要由 PowerPC405D4 中央处理器组成, 另外还包括通用中断控制器 (Universal Interrupt Controller), 以及用于调试的 JTAG 和 TRACE 接口。

中央处理器模块是 MHG 硬件平台的核心控制模块, 它负责控制系统中的所有设备, 监控硬件状态, 协调硬件工作, 以及处理少量的外设数据。PowerPC405D4 有较高的工作性能, 稳定工作频率为 252MHz, 执行指令速率为 350MIPS, 兼容 PowerPC 体系指令集, 并通过流水线等处理手段, 大大提高了指令处理速度。PowerPC405D4 同时具备两路 16Kb 组相连指令缓存, 数据缓存支持回写和直写两种模式。高效高容量的指令缓存和数据缓存大大提高了 CPU 的吞吐能力。同时 PowerPC405D4 还能在低功耗的情况下提供平稳的处理能力, 这种高性能低功耗的特性完全符合 MHG 硬件平台的设计要求。

中央处理器模块主要通过硬件中断监控设置其它硬件设备的工作状态。处理器模块中的通用中断控制器 UIC 负责监视所有中断源的状态，并控制和维护中断源与 PowerPC045D4 核心处理单元之间的痛心。为了减少重要中断的相应时间，UIC 中还提供了一个可选的中断向量，这个向量由两部分组成：位偏移，即由优先级最高并且使重要中断所在的位的位置确定，以及向量基址，通过向量基址寄存器获得。通过这个可选向量，优先级最高的总要中断的中断向量不必再经过 UIC 中复杂的计算，从而缩短了中断响应时间。

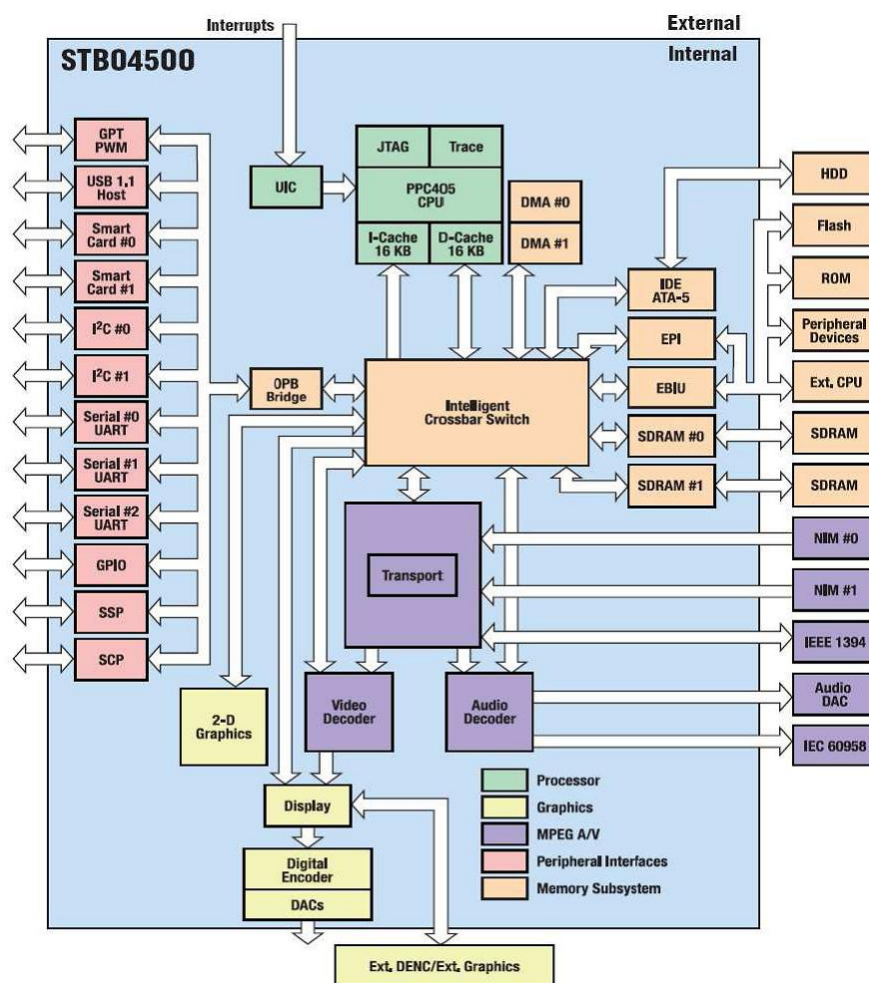


图 2-1 STB04500 芯片结构图

Fig. 2-1 Structure of Processor STB04500

2.1.2 外设子系统

外设子系统主要包含芯片与外部数据传输的接口，其中有 I²C、智能卡、串口、通用 GPIO、红外接收器等，外设子系统总体结构如图 2-3 所示。通过片上外围总线 OPB (On-chip Peripheral Bus)，外设子系统将多个硬件接口连接。下面将选择几个设计调试过程中会涉及的部分进行简要的介绍。

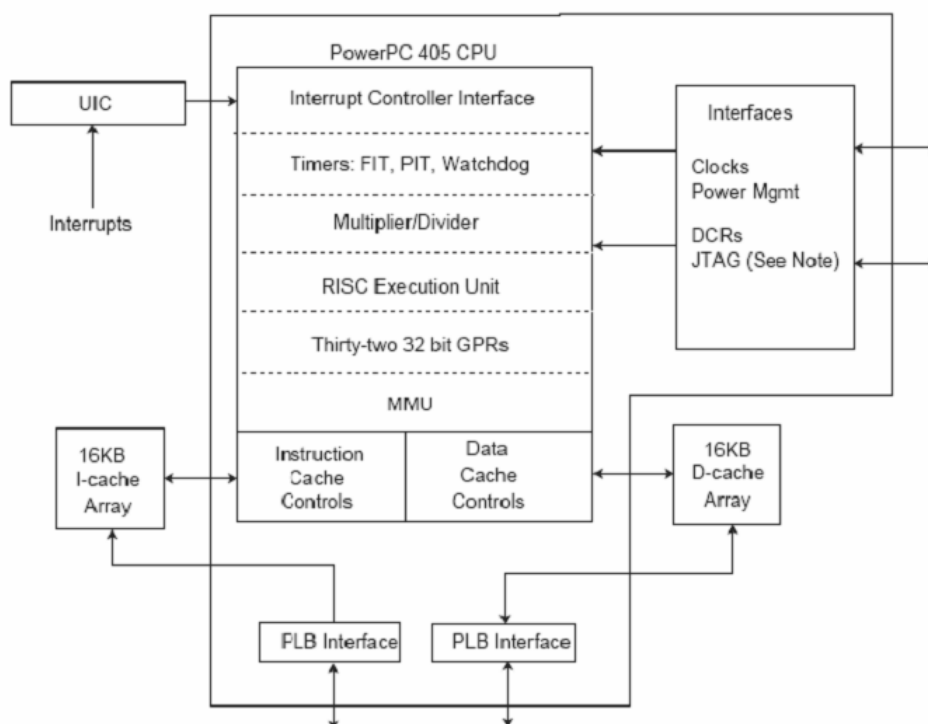


图 2-2 处理器子系统结构图

Fig. 2-2 Structure of Processor Subsystem

1、通用定时器 GPT (General Purpose Timer)

通用定时器是片上外设总线 OPB 的一个功能单元，它提供了一个独立的时钟计数器和独立于 PowerPC405 处理器的系统计时器。对该定时器进行配置和编程，用户可以调整其同步控制边沿检测和输出电平。GPT 内部包含三个计时器，它们接收其他单元的计数信号输入。该单元包含两个独立的复位输入，一个用来复位整个 GPT 单元，另一个用来做时间同步。

2、I²C (Inter-Integrated Circuit) 单元

I²C 总线为简单的双向 2 线串行总线，由数据线 SDA 和时钟 SCL 构成。I²C 总线在 CPU 与被控 IC 之间、IC 与 IC 之间进行双向数据传送。各种被控制器件均挂载到这条总线上，每个芯片都有唯一的地址。CPU 发出的控制信号分为地址码和控制量两部分，其中地址码用来选中要控制的电路；空置量决定要执行的操作量。通过这些简单的标准，就实现了多器件在同一总线上的互相控制。

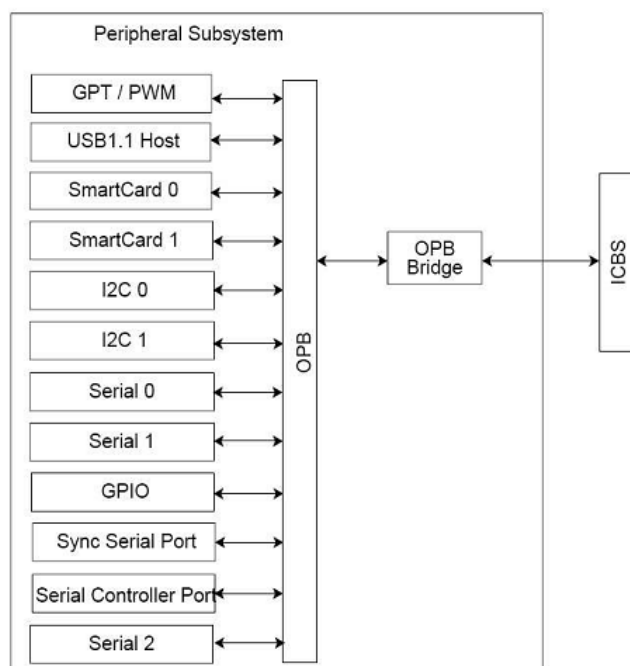


图 2-3 外设接口子系统结构图

Fig. 2-3 Structure of Peripheral Devices Subsystem

两个 I²C 单元提供了两个独立的 I²C 接口，并且在 OPB 和符合工业标准的 I²C 总线之间建立了高度可编程接口。他们完成了对所有 I²C 总线协议的管理，与飞利浦半导体制定的 I²C 标准完全兼容。挂载在 I²C 总线上的设备可以做主设备、从设备。I²C 单元提供了 100kHz 和 400kHz 的工作频率，支持 8bit 数据传输、7bit 和 10bit 的地址生成和解析。

1、UART570 串口控制器

Universal Asynchronous Receiver and Transmitter，通用异步收发器。STBx25xx 向外提供 3 个 UART570 串口通信接口。接口通过 FIFO 接收和发

送数据，每个接口控制都可以通过编程实现，同时具有完备的 modem 控制功能和状态反馈功能。STB04500 芯片内部 UART570 串口通信控制器同时还支持 DMA 传输模式。在 DMA 模式下，每个 UART 可以使用两条 DMA 通道进行数据传输，在 DMA 通道的后端是一个 16 字节的 FIFO 作为数据缓冲，DMA 模式可以应付大量数据的传输工作。在 UART570 串口通信控制器内部有一个自闭的回路，可以用来对自身的工作状态进行诊断。控制器内还包含一个可交互的超时检测装置，该装置通过 GPT 来为控制器提供时钟信号的支持。

2、USB 接口控制器

Universal Serial Bus version 1.1 host Controller，控制器通过一个集成的根集线器和 USB 端口给外部 USB 设备提供连接。控制器在软件层由 Host Controller Driver(HCD)管理，通过 HCD 获得要执行的指令，并向 HCD 反馈执行结果和当前的状态，接口同时支持全速率和低速率传输的设备（12Mbps 和 1.5Mbps）。通过 USB 接口我们又可以扩展出丰富的外部设备，尤其是用于无线网络通讯的设备，比如蓝牙卡，通过这些无线网络通讯设备，就能比较方便的搭建家庭内部的控制网络。

2.1.3 存储器控制子系统

STB04500 芯片提供多种存储设备接口，支持 SRAM、ROM、FLASH、SDRAM、IDE 等多种存储设备。如图 2-4 所示，该子系统一方面为数据存储提供了空间，如设备启动代码、操作系统、文件系统、应用软件、媒体文件等；另一方面为系统程序和应用程序的运行提供物理存储空间，STB04500 最大可支持 256MB SDRAM，尤为重要它是充当了系统各个模块间的相互通信提供与数据传输的媒介。尤其是存储模块中的处理器本地总线 PLB(Processor Local Bus)，它直接负责连接 PowerPC 405 D4 CPU 与其他的几个主要模块，例如存储模块中的存储设备，视音频数据处理模块中的视音频解码器等等。所有这些模块的控制信息都通过 PLB 由 CPU 发出，接收到控制信息的设备再经由 PLB 向 CPU 发送自身的状态信息或者请求中断等等。在 STB04500 芯片中的 PLB 总线构架中，包含两条 PLB 总线以及一个总线切换开关(Cross Bar Switch)，两条 PLB 总线分别简称为 PB0 和 PB1，给芯片内部的数据通信与数据传输提供了很高的带宽，能够满足多个

设备并发地进行通信或者数据传输的带宽需求。

在芯片内部各个功能模块与外部存储设备接口之间的数据通信和数据传输也通过 PLB 来完成，这些外部存储设备包括 SRAM、ROM、Flash、HDD、SDRAM 等。其中 SRAM、ROM、Flash 和 HDD 的 IDE 接口都通过一个外部总线接口单元 EBIU(External Bus Interface Unit)连接到 PLB 上。EBIU 是一个 PLB 到外部存储设备的接口扩展，它一端通过多种接口与各种外部存储设备连接，另一端通过一个接口与 PLB 连接，因此从 CPU 和存储模块以外的各个功能模块看来，EBIU 是将多种多样的外部存储设备接口抽象为单一的接口，再通过 PLB 提供给 CPU 和各功能模块访问。

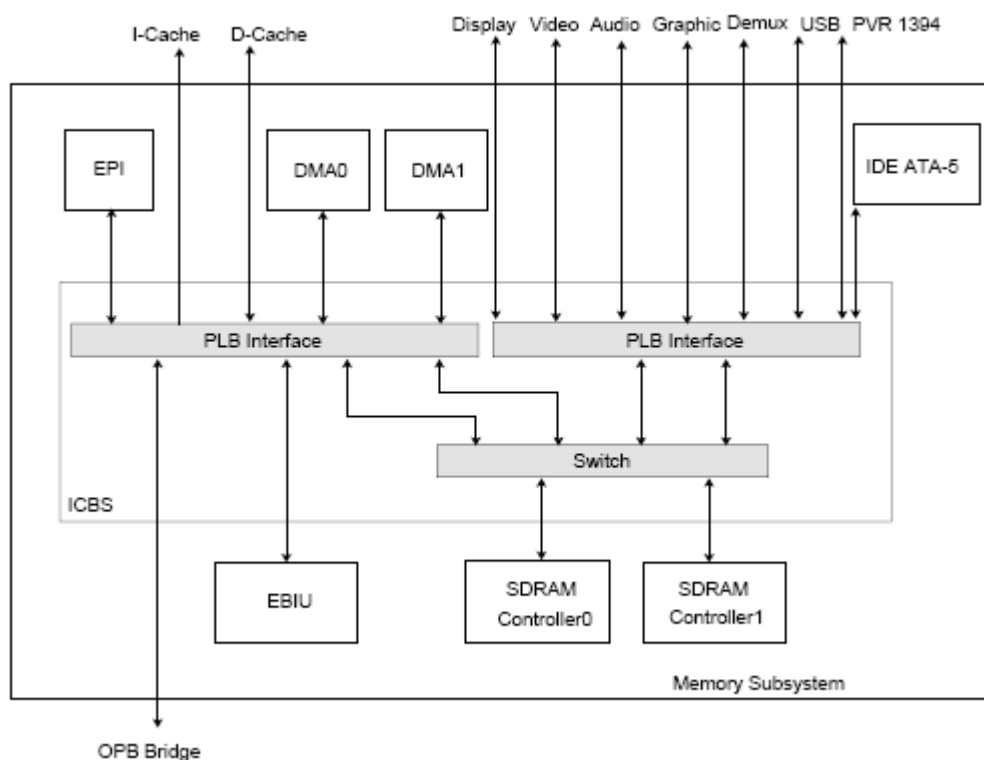


图 2-4 存储器控制子系统

Fig. 2-4 Structure of Memory Control Subsystem

存储器控制子系统芯片外的两片 SDRAM 通过两个 SDRAM Controller 连接到 PLB 的总线切换开关上，SDRAM Controller 可以支持不同大小的 SDRAM 芯片，它除了为 PLB 提供与 SDRAM 的接口之外，还为 SDRAM

的访问提供其他一些功能支持，这些功能支持包括：16 位窄接口的高速率访问、可编程的地址选择等。页面的交叉存取也是通过 SDRAM Controller 完成。页面的交叉存取(Interleaving)是指存储体的奇数地址和偶数地址部分分开，这样当前字节被刷新时，可以不影响下一个字节的访问。

总线切换开关连接两个 SDRAM Controller 与 PLB，它分配与 PLB 总线通信的 SDRAM Controller，同时与 SDRAM Controller 一起，完成对 SDRAM 的交叉访问，提高 SDRAM 的访问效率。

在存储模块中还有另外一个很重要的接口设备：DMA 控制器(Direct Memory Access Controller)。DMA 控制器控制外设与内存之间通过 PLB 进行数据传输，在实际的数据传输过程中不需要 CPU 或者其他程序的管理控制，数据成块传输，从而得到很高的数据传输速率。由于在 DMA 模式下传输数据时不需要 CPU 的介入，因此在传输数据的同时，CPU 可以同时执行其他指令而不受到数据传输的干扰，大大提高了 CPU 的工作效率。在 STBx25xx 中的 DMA 控制器支持三种传输路径，分别是内存-内存、内存-外设、外设-内存，三种传输路径都有各自不同的应用场景。

STB04500 芯片内 DMA 控制器控制 4 条 DMA 通道，每条通道都有自己独立的一套寄存器组，在寄存器组中存储的数据包括控制命令、数据源地址、数据目的地址、传输数据量以及通道状态。在给定了这些寄存器的值以后，就能通过 DMA 通道开始独立的数据传输。

所有的 4 条 DMA 通道都支持链式 DMA 传输(Chained DMA)。链式 DMA 传输的关键在于传输的块数据当中包含一段要写入 DMA 通道寄存器组的数据。这段数据被称作传输控制块 TCB(Transmission Control Block)，传输的过程中能在没有 CPU 的干涉下自动写入到 DMA 通道寄存器组中。在链式 DMA 传输模式下，DMA 通道寄存器组中的传输数据量寄存器会被作为链计数寄存器使用，同时数据源地址寄存器中会被为链地址寄存器使用，写入链(chain)的开始地址。TCB 中包含的数据包括控制命令、数据目的地址、通道状态以及下一个 TCB 数据块的地址。在上一个数据块传输完毕以后，下一个 TCB 数据块会被自动写入通道寄存器组中，从而开始下一个块的数据传输。如此 DMA 传输的块数据通过各自的 TCB 连成了一条数据块链，一旦数据块链的起始地址写入到链地址寄存器中，整个数据块链就能在没有 CPU 干涉的情况下独立的通过一个 DMA 通道进行传输。这无疑大大提高了传输效率，减轻了 CPU 的工作负载。在大量多媒体数据的处理中，DMA 模式的数据传输对于系统的工作效率非常重要。

2.1.4 多媒体处理子系统

MHG 系统的基本显示设备是普通的 TV，无论是播放数字电视节目，或者浏览网络等，都需要将信号处理并输出到 TV 上，因此多媒体处理功能对于 MHG 系统来说是必不可少的一个重要环节，有不少的媒体处理系统采用 DSP 芯片加外部可编程逻辑芯片（FPGA or CLPD）或者专用芯片（ASIC）的结构实现这一功能。STB04500 芯片内部集成了多媒体处理模块，负责视音频数据的处理，包括传输流解复用、MPEG2 视音频解码，多种制式编码。多媒体处理模块工作时对 CPU 的要求很低，原本工作量最大的 MPEG2 解码工作基本由该模块独立完成，很大程度的释放了 CPU，为 MPG 系统的其它功能提供了更多的 CPU 负荷。

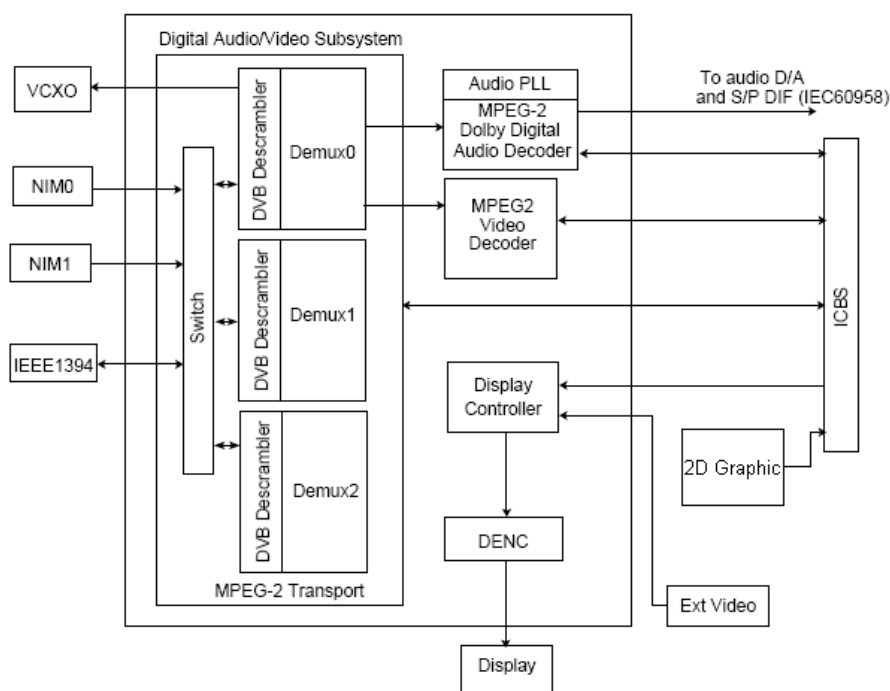


图 2-5 多媒体处理子系统结构

Fig. 2-5 Structure of Multimedia Processing Subsystem

从图 2-5 可以看出，视音频处理模块包括几个主要组成部分：三路解复用器（Transport DEMUX）、视音频解码单元、制式编码单元。

解复用器负责将复用的传输流中的各种数据流检索出来，分别输出到各个响应的解码处理单元中。检索出来的数据流称为 PID 流，PID 流既可以是视频流，也可以是音频流，还可以是时钟信息或者其他用户关心的数据流。在解复用器前端还有 NDS 的 ICAM 模块支持以及 DES 解密模块支持，可以用于条件接收的功能的实现。STB04500 提供了三路相对独立的解复用通道，大大方便了个人视频录像 PVR (Personal Video Recording) 功能的实现。

视音频解码单元紧接着解复用器之后，处理解复用器送来的视音频流。STB04500 芯片中集成有 MPEG2 硬件解码器，其中，MPEG2 视频解码器支持 ISO/IEC 13818-2 MP@ML，并且兼容 MPEG1 标准，提供 MPEG2 标准视频流的解压缩，解码以及同步回放，支持的输入 MPEG2 压缩数据流平均达到 15Mbps，最终输出交织视频信号。对于所有 16 像素的整数倍分辨率的视频，只要在 $720 \times 480 \times 30$ 或者 $720 \times 576 \times 25$ 以内，STB04500 的硬件 MPEG2 视频解码器能够提供能实时解码。MPEG2 视频解码器支持欧洲的数字广播电视 DVB 标准，可以接收 MPEG2 PES 流和 ES 流，在 PES 流中，可以通过视频解码获得包时间戳，为视频的同步回放等功能提供参照。MPEG2 视频解码器的输入有两个来源，一是直接从解复用器送入的视频流，另一个是直接从系统内存输入的用户数据。输出也有两种，一是直接输出单一的视频数据，另一种可以输出嵌入 OSD 数据的视频。

OSD (On Screen Display) 模块和 2-D Graphic 模块一起生成 MHG 系统中面向用户的交互界面，如开机画面，节目菜单等等，这些图像数据与解码模块输出的视频流数据混合在一起，经由 DENC 模块编码转换成为模拟的 PAL 或 NTSC 制式的电视信号。

音频解码单元接收并解码 PES 或者 ES 音频数据流。音频处理单元主要由一个通用的数字信号处理器组成，能对 MPEG 压缩音频，16、18 或 20 比特无格式化的 PCM 或线性 PCM 音频数据进行解码。支持符合 ISO/IEC13818-2 的层 1 和层 2 标准的声道 MPEG1 和 MPEG2 音频。

2.2 Redwood5 评价板

该评价板主要是为方便实现机顶盒产品（Set Top Box products）而设计，Redwood5 板上支持 IBM 多媒体处理芯片（STB 系列）的大部分功能，而这些功能并不是每一个机顶盒产品都要保留的。针对不同的设计目标，我们可以量身定做不同的 STB 产品，完整的评价板接口如图 2-6 所示。其中的主要部件列表如表 2.1 所示。

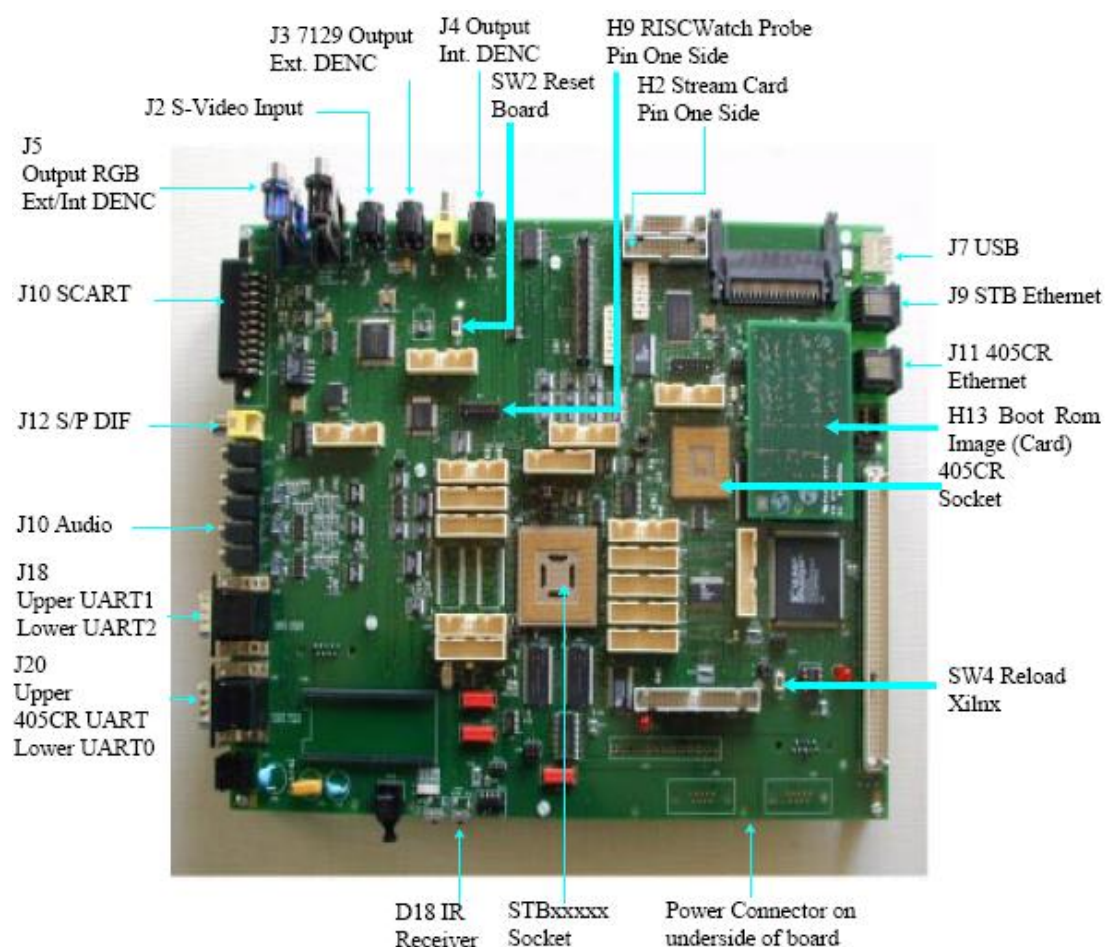


图 2-6 Redwood5 评价板接口

Fig. 2-6 Junctions of Redwood5 Evaluation Board

2.3 MHG 硬件平台

以 Redwood5 平台为参考，去掉了一些不常用的功能，我们从硬件上裁剪出一个更小的系统，该系统较评价板体积小、成本低，适合在市场上广泛推广。为了实现 MHG 平台的设计目标，具体来说，MHG 平台应该支持表 2.2 所示应用。

表 2.1: Redwood5 目标板主要组件

组件	型号	数量	描述
机顶盒集成控制器	STB04500	1	该组件为评价板的核心，完成了多种控制功能，提供了包括音、视频传输流，串口和 SDRAM 等接口
视频输入处理器	SAA7111	1	接收 CVBS 或 S-Video 模拟信号，同时产生 8-bit 的 YUV 格式的数字信号
音频数模转换器	CS4329	3	每个 CS4329 可以将一路 IIS 音频信号转化为两路独立的模拟音频信号
SDRAM	48LC16M16A2	2	SDRAM0 和 SARAM1，各 64MB，位宽为 16bits，3.3V 工作电压
以太网控制器	LAN91C111	2	提供了 10M/100M 的网络连接
可编程逻辑器件	XC9536	1	控制以太网接口
27MHz 压控振荡器		1	3.3V 工作电压，用来同步块解码器
红外接收器	TDFS650E	1	为评价板提供了红外遥控器接口

为了实现对表 2.2 上述应用的支持，相应的我们硬件上应该满足表 2.3 所示的要求。通过前面功能需求分析步骤的工作，我们明确了 MHG 系统的功能，接下来就是根据整体功能的需求绘制电路原理图。在绘制电路原理图前，我先对整个系统的绘图区域进行合理的规划。MHG 系统电路逻辑框架

如图 2-7 所示。

2.3.1 FLASH 存储模块

Flash 算是内存的一种，我们知道内存可以简单的分为两类：**RAM** 和 **ROM**。随机访问存储器 **RAM**（**Ram Access Memory**）的最大特点是可随机读写，而且读写速度很快，但存储的数据在掉电后丢失；只读访问存储器 **ROM**（**Read Only Memory**）的特点是只能读取数据，不能写入数据，但掉电后数据不会丢失，而且成本相对较低，用于保存代码和永久性数据。**Flash** 可以说是介于两者之间，同时拥有两者的优点。

表 2.2: MHG 平台功能

数据中心	媒体库管理 数字版权管理（ DRM ） 系统验证及授权 程序管理及软件更新
媒体中心	DVB-S/DVB-C 接收 视频点播 视频会议 视频电话 个人录像（ PVR ） 媒体间转码
控制中心	家电控制 家庭外部的授权访问 家居监视
通信中心	Web 浏览器 Email 客户端 IM 客户端 无线路由 无线媒体数据分发

Flash 技术根据不同的应用场合及发展方向，分为擅长储存代码的 NOR Flash 和擅长存储数据的 NAND Flash。MHG 系统中 Flash 的主要作用是存放硬件初始化代码和内核镜像，另外 STB04500 芯片初始化时从底地址获取第一条指令，因此我们选用了 AMD 的 Bottom Boot NOR Flash 芯片 AM29LV320D^[9]。

表 2-3 MHG 硬件功能

Table 2-3 Functions of MHG hardware

多媒体处理	Video Decoder (MPEG 1,2 and 4,H.264)
	Video Encoder (MPEGpeg1,2 and 4,H.264)
	Video Transcoder (MPEG1,2 to MPEG4, MPEG1,2,4 to H.264)
网络通信	Ethernet (IEEE 802.3)
	Bluetooth 2.0
	IEEE 802.11 b/g
输入输出及外设	DVB Tuner
	IDE
	IEEE 1394
	USB Host 1.1/2.0
	Memory Card Slot
	IR Receiver and Remoter

STB04500 理论上可以寻址的空间是 4GB，其中 3GB 的空间留给了处理器内部的寄存器和其他设备了，留给外部可寻址的空间是 1GB，也就是 0x60000000~0x7FFFFFFF 和 0xE0000000~0xFFFFFFFF，这时地址线的 Bit 1 和 Bit 2 都置 1。STB04500 将这 1GB 的空间等分为 8 份，每一份空间 128MB，这每一份空间就称为一个 Bank。为了方便操作，STB04500 芯片独立的给每个 Bank 提供了一个片选信号（BI_CS0~BI_CS7），这 8 个片选信

号可以看作是 STB04500 内部高位地址线地址译码的结果。

Flash 占用了其中一个 Bank，即片选信号 BI_CS0 有效时选中 Flash 芯片。AMD 的 AM29LV320D 的数据线宽度 16Bit 或者 8Bit 可选，容量为 4MB，是一个 Bottom Boot 的 NOR Flash 芯片。其读写时序和 STB04500 EBIU 的读写时序基本吻合，具体在绘制原理图时比较简单，只要注意数据线和地址线的最高位即可。

2.3.2 串口通信接口电路

串口简单的说就是串行收发数据的接口。串口通信是一种异步通信方式（也有同步通信的标准，但是实际应用很少），其中负责异步串行收发数据的模块叫 UART（Universal Asynchronous Receiver/Transmitter）。由于异步通信的数据发送方和数据接受方的系统时钟不是同步的，数据通信过程中也不携带任何时钟信息，因此它的速率受到了很大限制，目前较常用的串口速率最快的也只能达到 115200bit/S。

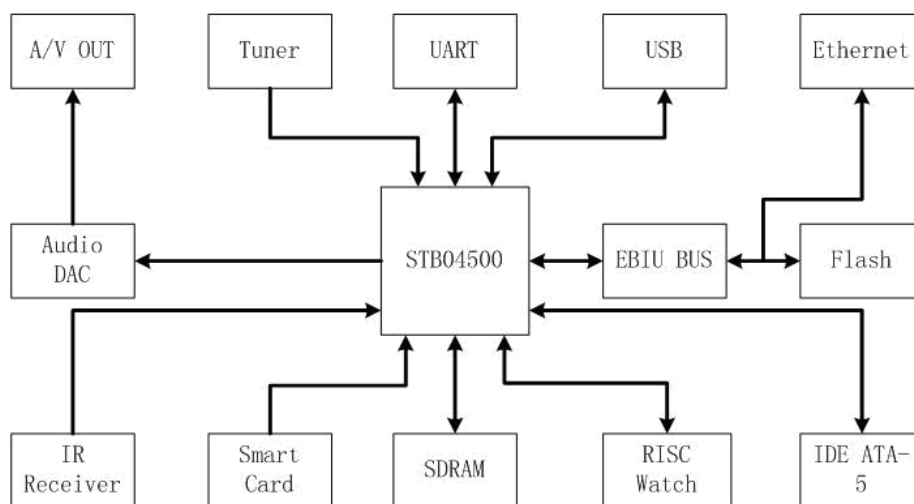


图 2-7 MHG 目标板电路逻辑框架图

Fig. 2-7 Skeleton of MHG board's circuits

串口设备的实际使用过程中，人们为了让它的信号可以传输得更远，并

不会直接将逻辑电平放到传输线上，而是先作一个电平转换。这个电平标准就是 RS232，它规定逻辑 1 的电平范围是-5~-15V；逻辑 0 的电平范围是 5~15V。在这个电平标准下，如果有足够好的屏蔽电缆，甚至可以将 19200bit/S 的数据传输到 10000 英尺以上。

串口电路的设计比较简单，STB04500 内置串口收发模块，但是串口模块都不会直接输出 RS232 的串口电平，所以需要有一个串口电平转换电路。

2.3.3 调谐器接口电路

调谐器（TUNER）是用来接收数字电视信号的前端设备。Redwood5 开发板不具备 TUNER 接口，仅支持现成的传输流输入，而 MHG 系统的一个基本功能是接收播放数字广播电视，因此 TUNER 接口是必不可少的。我们在这里选用了 SHARP 的 BS2F7VZ0624 立式 TUNER^[10]，支持一路 RF 输入/输出，8 位传输流接口，具有 I²C 总线接口，外部控制芯片通过读写 I²C 总线对 TUNER 的行为进行控制。我们将 I²C 总线的数据线以及时钟信号线连接到 STB04500 的 I²C 总线上，然后将传输流接口接到 STB04500 上对应的传输流接口上，另外遵照 0624 的规格书接上合适的滤波电容。

2.3.4 以太网接口电路

MHG 系统需要实现浏览网页，收发 Email 的基本功能，因此网络接口是必不可少的，STB04500 没有内嵌以太网控制器，因此需要外部电路搭建一个以太网接口电路，市场上常见的以太网控制芯片非常多，如 RTL8019、RTL8029、RTL8039、CS8900、DM9008。

本课题选用了 LAN91C111 以太网控制芯片，包含了 MAC 控制器和物理层接口，该芯片支持全/半双工模式 10/100Mbps 双速率自适应切换，支持 8、16、32bit 三种数据宽度，并且支持多种体系结构的 CPU，其中包括 PowerPC 体系结构。

2.4 MHG 硬件调试

对 MHG 进行硬件调试的过程中，除了用到一些基本工具之外，主要使用了 IBM 专为 PowerPC 系列处理器开发的 RISCWatch 调试器^[11]。

RISCWatch 可以实现实时追踪、以太网硬件接口监视，C++支持，扩展命令文件支持和片上在线调试。

RISCWatch 的连接方式如图 2-8 所示，其中包含一台 Linux 主机、一台 Windows 主机、以及一个用来连接这些操作系统的网络交换机。RISCWatch 通过 JTAG 接口与 Target (MHG 平台) 连接，实时监控平台的运行状态，也随时可以暂停或者停止系统的运行。给调试带来极大方便的是，暂停之后通过 RISCWatch 可以随意浏览和修改 PowerPC 处理器中寄存器的值，这就给开发人员在硬件调试方面很大帮助。

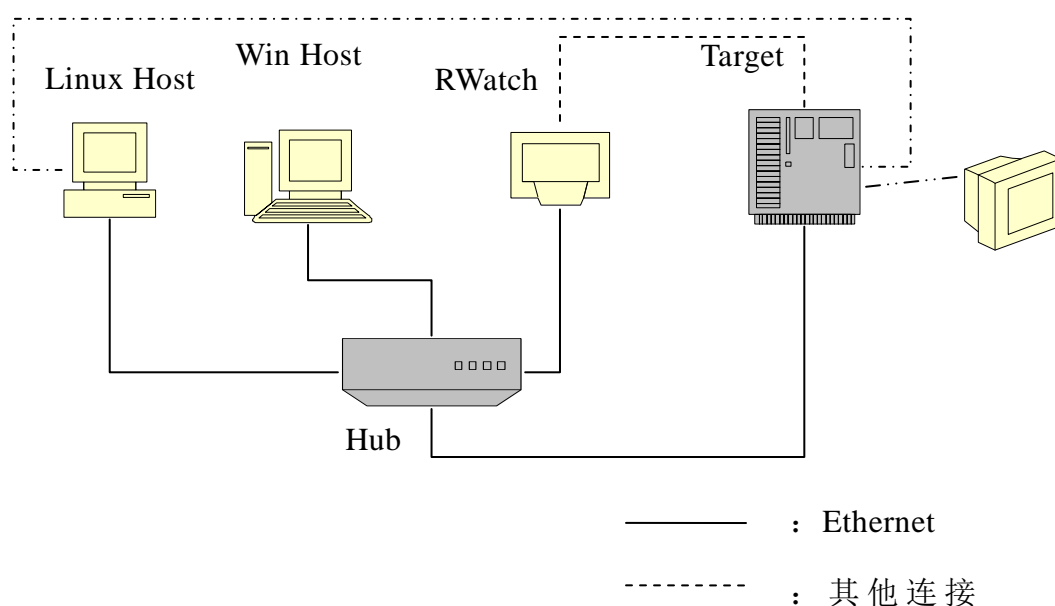


图 2-8 RISCWatch 的接入方式
Fig. 2-8 Connection manner of RISCWatch

2.5 本章小结

本章首先介绍了 Redwood5 和 MHG 的核心组件——STB04500 芯片，对其中各个模块的功能进行了简单的阐述，接着介绍了 MHG 平台的硬件参考平台 Redwood5 评价板。对 Redwood5 评价板性能和我们对于 MHG 的功能定位，我们设计出我们的 MHG 平台。在介绍完 MHG 平台的硬件之后，最后说明了主要调试工具 RISCWatch 的主要功能。

第3章 MHG 系统软件平台

在 MHG 硬件平台已经搭建完成之后，我们还需要搭建相应的软件平台，向下通过驱动程序调用硬件系统资源，向上提供丰富的接口给应用软件提供支持，整个软件系统从下到上依次是 OpenBios，操作系统，中间件，应用软件，如图 3-1 所示。

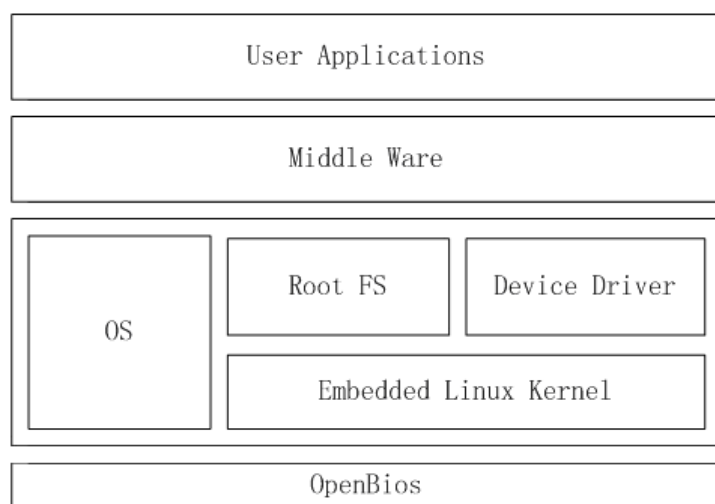


图 3-1 MHG 软件体系结构图

Fig. 3-1 Software skeleton of MHG

整个开发过程采用主机——目标板模式进行，主机上配置交叉编译的开发环境，主机和目标板通过 JTAG、串口以及以太网进行通信（JTAG 用于调试以及最初的裸板代码下载，串口负责通信以及监控，以太网则用来下载文件），在主机上进行程序开发，交叉编译，然后下载到目标板上测试、运行。

3.1 OpenBios 介绍

OpenBios 在 MHG 系统中充当了操作系统向导的角色。在 PC 系统中，系统上电后后，最先执行的程序是主板上的 BIOS。BIOS 程序负责对底层

硬件进行初始化，并给后续程序提供访问硬件的接口函数。在硬件初始化过程中，某些 PCI 卡上自带的 BIOS 程序也会执行，完成对自己的初始化（如显卡）。当 BIOS 程序完成后，会将硬盘 MBR 中的启动代码拷贝到内存中并执行，从这段代码到操作系统启动之间的程序一般被人称为 BootLoader。到这里 BIOS 程序功成身退，将系统的控制权交给了 BootLoader，Linux 操作系统下的 Lilo 和 Grub 都是 BootLoader 的一种。BootLoader 一般会传递一些系统参数给操作系统内核，让操作系统可一在这个特定的系统环境中顺利地启动。例如，BootLoader 会告诉操作系统硬盘的哪一个分区才是主分区，要从哪个分区上启动。

对于 MHG 这样一个嵌入式系统来说，通常不存在 BIOS 那样的固件（FirmWare）程序，整个系统得加载启动任务就完全由 BootLoader 来完成，如图 3-2 所示。而且各种嵌入式系统得硬件千差万别，而这些硬件某些初始化工作也必须在操作系统启动之前完成，这也一定程度上增加了 BootLoader 程序的复杂度。在实际的使用当中，为了系统调试的方便，BootLoader 更是增加了如下的内容。

1. 读写内存。
2. 通过以太网或者串口电缆上载二进制文件到内存。
3. 将内存中的程序或是数据烧写到 Flash 中。

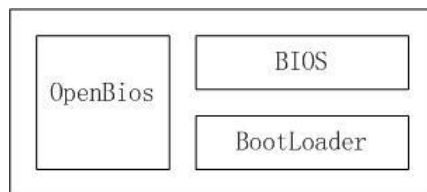


图 3-2 OpenBios 示意图

Fig. 3-2 Snapshot of OpenBios in system

这里的 OpenBios 就担当了 BIOS 和 BootLoader 的角色，完成系统硬件的初始化工作并且引导操作系统的启动。将交叉编译后的 OpenBios 映像文件下载到目标板的 Flash 上，当 STB04500 上电以后，PowerPC405 会从地

址 0xFFFFF0FC 读取第一条指令，此时下载到 Flash 底地址处的 OpenBios 镜像中初始化代码便开始顺序执行，首先初始化 MUXIO 和 GPIO 的控制寄存器，然后顺序初始化 EBIU、SDRAM 等其他功能寄存器，在完成基本的硬件初始化后就进入菜单等待用户选择操作，或者直接根据配置启动内核。

3.2 在 OpenBios 中驱动调谐解调器

MHG 中的 OpenBios 大部分继承于 Redwood5 评价板，并相应做了裁剪。但由于 Redwood5 并不支持数字卫星电视接收，而 MHG 的主要功能之一就是接收数字电视，于是在 Redwood5 附带的代码的基础上增加了对数字卫星电视调谐器（Tuner）的初始化代码。该代码的编写主要根据相应的 Tuner 的数据手册，也就是 Sharp BS2F7VZ0624。为了说清楚针对 Tuner 的代码的主要功能，这里首先介绍一下 Tuner 的模块组成。该调谐器主要由两个大模块组成：前端的调谐器和后端负责解调的 LINK IC。

调谐器主要由前置放大器、变频器、锁相环（PLL）、带通滤波器、中频放大器、AGC 电路和移相器组成。卫星信号首先经过前置放大器进行信号放大，同时要保证低噪声。调谐器中的本振频率通常有 CPU 通过 I²C 控制的 PLL 电路产生。调谐时，CPU 读出预置在存储器中的频道参数，通过 I²C 总线通知 PLL 电路的分频比，从而将输入的射频数字信号转换成中频信号。如图 3-3 所示，信号经过自动增益控制（AGC）来保持信号强度的稳定。调谐器的输出端有一个 90 度移相器，产生的相移信号与原始信号同时输出，即 I、Q 信号，被送到 QPSK 解调器进行解调。

QPSK（四相相移键控）解调器主要包括时钟产生电路、QPSK 解调器、前向纠错（FEC）单元和输出控制电路部分。简单来说该部分主要完成了两件事，如图 3-4，首先将地面卫星信号发射塔向卫星发射的信号还原出来，其次，为了增强信号的稳定和效率，上传之前增加了交织和冗余码，调谐器后端需要去掉这些，从而得到我们阵中需要的 MPEG2 数据流。

参考文献[10]中说明了通过 I²C 初始化调谐解调器的过程。由于 MHG 中使用了 PowerPC 处理器，在驱动程序中只能使用 PowerPC 汇编语言编写，给开发过程带来了很大不便。汇编语言可读性差，与硬件关联性极强，但效率高，编写过程可以让我们对处理器的工作过程有深入的了解。

调谐解调器的初始化是通过 RISCWatch 的命令行开始的，首先会调用一段 PowerPC 可执行的汇编代码。首先卸载掉处理器中寄存器的所有设

置，重新启动系统，接着通过 RISCWatch 的“halt off”命令使处理器暂停工作。这时需要根据系统的硬件配置，对处理器的所有模块进行寄存器的初始化，包括多路视频寄存器、外部总线接口（EBIU）、通用输入输出（GPIO）、SDRAM 和一些其它的处理器内部寄存器。处理器初始化之后需要对串口进行初始化，并检测串口是否能正常执行命令。在运行这段命令的同时，可以通过连接在 MHG 板子串口上的 PC 的 Minicom 程序显示出一系列的命令行。最后，将 OpenBios 程序加载到板载内存中，该程序包含了调谐解调器的驱动程序。RISCWatch 命令参考附录 1，调谐解调器的驱动程序参考附录 2。

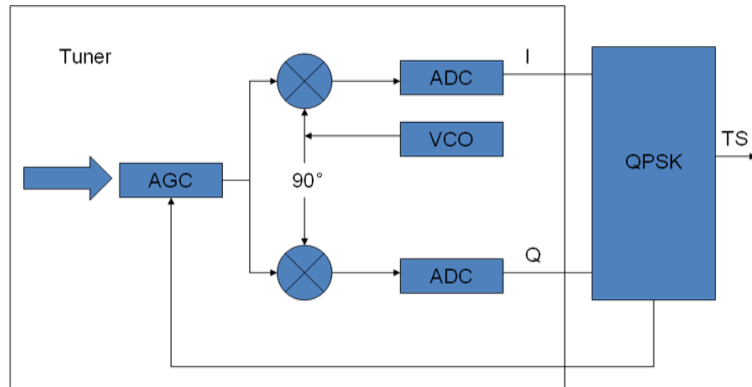


图 3-3 一体化调谐解调器的前端

Fig. 3-3 Frontend of Tuner

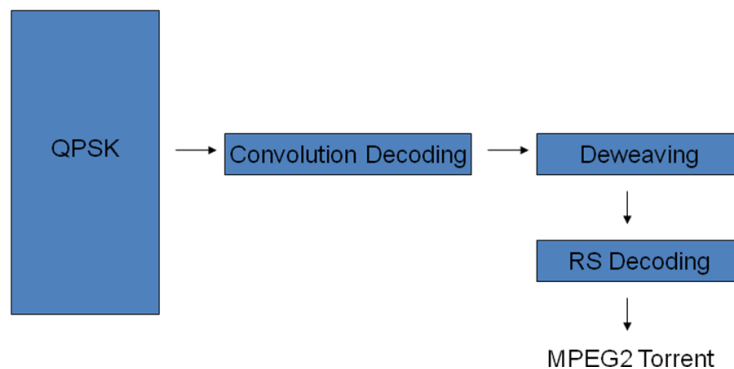


图 3-4 简化的后端解调器

Fig. 3-4 Simplified back end decoder

3.3 嵌入式 Linux

许多从事过 8/16 位单片机开发的技术人员，可能对于嵌入式操作系统会有些陌生，很多人开发了多年单片机，却也从来没有在上面运行过操作系统。很多人心里有这样的疑问，操作系统有什么用处？什么时候必须要用操作系统？确实，操作系统不是必须的，即使对于一个复杂的 32 位处理器的复杂嵌入式系统也是如此。有些系统仅仅要求的是高性能的 32 位处理，以满足系统对于实现复杂算法时所需要的高速处理能力，但是本身的任务并不复杂，系统也不需要扩展能力，对于这样的系统就没有必要安装操作系统，安装操作系统反而会带来新的系统开销，降低系统的性能（运行操作系统本身就需要大量的硬件资源；启动操作系统需要时间）。

当系统需运行较多任务，任务调度、内存分配复杂，或需要大量协议（如网络协议、文件系统）支持，或需要很好的图形用户接口 GUI，或需要系统有很好的可扩展性或可移植性时，操作系统就是个不错的选择。

总的来说，引入操作系统的目的有以下几个：

1. 有效性

有效的管理和分配硬件、软件资源，合理的组织计算机的工作流程；

2. 方便性

给上层应用提供良好的、一致的用户接口，弥补硬件系统的类型和数量差别；

3. 可扩充性

硬件的类型和规模、操作系统本身的功能管理策略、多个系统之间的资源共享和互操作；强调系统的开放程度。

MHG 系统运行的任务种类较多，播放数字电视节目，浏览 WEB，充当家庭监控的中转站等。MHG 系统必须支持相当种类的协议，如网络协议是实现浏览 WEB、发送 Email 的基础，文件系统又是实现 PVR 功能所必不可少的。MHG 系统的基本定位是个多媒体中心，其用户对象是家庭，鲜明易操作的图像界面也就必不可少。可见，一个好的操作系统是 MHG 系统所需要的。

嵌入式操作系统种类繁多，WinCE，VXWorks 等由企业主持开发的嵌入式操作系统，在稳定性，硬件支持，软件支持等方面都非常成熟，对于加快开发进程，缩短投入市场的周期非常有利，但是由于是商用操作系统，获取操作系统的使用权以及技术支持需要支付高昂的费用，同时由于其内部细节并不公开，不利于研究性的开发工作。

标准 Linux 操作系统经过多年的发展，以其开源性、稳定性和免费的特点在服务器和桌面系统获得了成功，并且在嵌入式系统中也获得了越来越广泛的应用。由于 Linux 在嵌入式应用中的活跃，其重要性已经无法取代，就连大名鼎鼎的 WindRiver 公司（VXWorks 的拥有者）也宣布开始支持 Linux 操作系统，提供对 Linux 开发的技术支持。

Linux 操作系统主要有用户程序、操作系统服务、内核和硬件控制器四个子系统构成。

1. 用户程序

即上层应用服务，如网络浏览器、Telnet 服务器等

2. 操作系统服务

Linux 操作系统提供的服务，如图形界面、控制台等

3. Linux 内核

这是 Linux 操作系统的核心，包含了给上层的虚拟硬件接口，以便于上层程序可以访问硬件资源

4. 硬件控制器

各类硬件驱动程序

3.4 应用软件

现有的机顶盒设备有很多不足，如多媒体数据处理能力不够、对智能家电的支持不足以及缺少安防等功能。为了解决这些问题，MHG 平台应该具有更强大的多媒体数据处理能力，可以和数字电器进行交互，而且具有视频监控等功能。根据以上对 STB04500 的介绍，我们知道该芯片支持这些功能，我们需要做的只是在已有的平台上设计上层应用软件。总体来说，MHG 的功能需求主要包括五个部分：媒体应用，网络应用，智能家庭，内容管理和系统管理。各个应用的具体内容如表 3-1 所示。

表 3-1 MHG 应用软件
Table 3-1 Software applications of MHG

分类	功能	阶段		
		1	2	3
媒体应用	卫星数字电视接收	√		
	有线数字电视接收	√		
	视频点播			√
	视频电话			√
	转码			√
	视频会议			√
网络应用	网页浏览	√		
	电子邮件收发			√
	即时通信			√
	无线媒体数据分发		√	
智能家庭	家电控制		√	
	家庭监控		√	
	房屋准入控制		√	
内容管理	硬盘录像机	√		
	条件接收	√		
	数字版权管理			√
系统管理	软件升级		√	
	用户权限管理		√	

3.5 本章小结

本章对 MHG 软件平台进行了简要的介绍。首先简述了 OpenBios 在系统中的位置和功能，接着从嵌入式 MHG 的 OpenBios 说起，详细说明了 OpenBios 对 MHG 硬件的初始化过程。OpenBios 是系统软件和硬件接触的

第一层，在 MHG 中完全用 PowerPC 汇编语言编写。接着介绍了 MHG 平台中运行的操作系统——嵌入式 Linux，对 Linux 系统中的层次做了简要的说明。最后说明了 MHG 平台已实现的应用软件，并对将来的开发做了展望。

第4章 基于惯性器件的无按键遥控器

传统家电的遥控器有一些缺点，比如按键数量多、操作复杂、体积大等。本章中将提出一种基于惯性器件的新型控制器，这种遥控器通过内部的惯性器件可以感知自身的加速度状态，从而识别出用户完成了什么动作。在识别方向键、确定键和取消键之后，继续讨论了识别 10 个数字的方法，从而让其识别范围几乎覆盖了传统遥控器的所有按键。

4.1 背景介绍

在上世纪，惯性器件（包括加速度计和陀螺仪等）主要用在武器装备和高档轿车之类的一些奢侈品，用来识别其运动方向和感知猛烈的撞击。近年来由于传感器性能的提高和价格的大幅下降，越来越多的民用产品中也加入了惯性传感器。比如 adidas 的“miCOACH”^[2]和 Nike 的“nike+”^[3]系列产品，这些产品的面世使得“计算”不再局限在办公桌上，人们可以随时随地享受到生活的便捷。

根据输入的不同，字符识别可以分为 online 和 offline 两种^[1,4,19,35]。Online 的字符识别要求被识别的轨迹曲线是时间的函数，而 offline 的情况只需要有轨迹在空间域的图像即可。如图 4-1 所示，同一组数据分别以 online 和 offline 的形式被表示。

Online 类型的字符识别包含了大量的时间信息，在图中是通过相邻点之间的距离表示的，当然还有很多其它的表示方法，甚至可以直接给出轨迹与时间的函数。基于惯性器件的字符识别属于 online 的范畴，因为实时记录的加速度信息已经包含了大量的时间信息。

很多算法已经成功应用在字符和动作识别领域，比如隐马尔科夫 HMM（Hidden Markov Model）^[16,20,21,36]，DWT（Dynamic Timing Warping）算法^[33,36,39]和贝叶斯网络（Bayesian Networks）^[33,34]。针对最常用的 HMM 模型，其又被分为离散隐马尔科夫模型（DHMM）和连续隐马尔科夫模型（CHMM）。综合考虑嵌入式系统的处理能力、计算复杂度和准确性，本系统最终选择 DHMM 作为字符识别的算法。

4.2 加速度传感器

系统选用了飞思卡尔半导体公司生产的三轴加速度传感器 MMA7260T，该传感器的有四个可以选择的量程 1.5g/2g/4g/6g，分别对应着不同的精度，1.5g 时的精度最高为 800mV/g。同时该传感器采用 QFP 的封装，尺寸为 6mm×1.45mm。

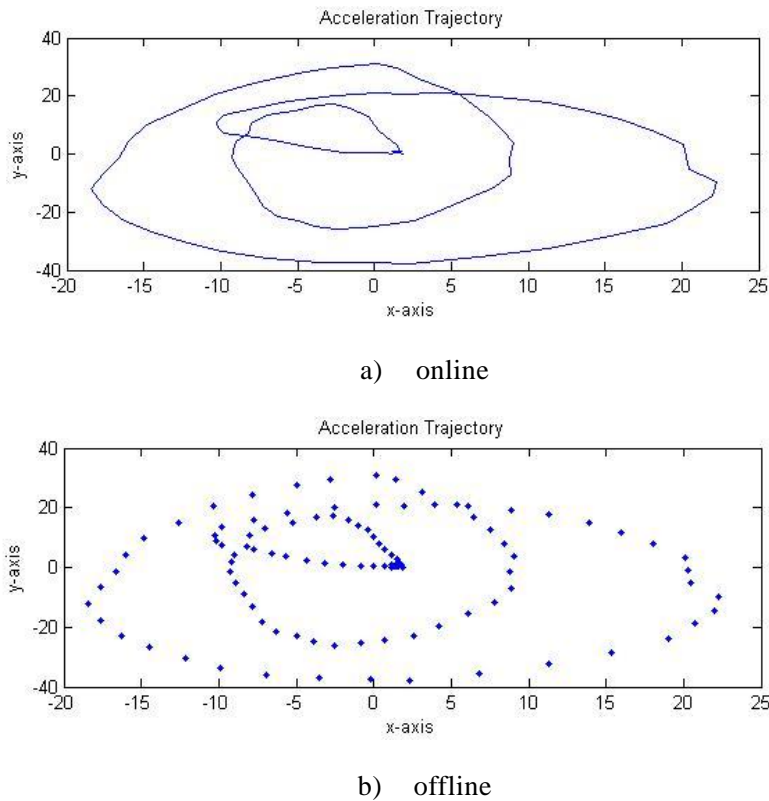


图 4-1 数据的存储形式

Fig. 4-1 Storage manner of data

该传感器有两个量程输入端口 g-select，分别输入高低电平时可以产生四种组合，分别对应四种不同的量程。当睡眠模式（Sleep Mode）被选中时，传感器将处于低功耗的保护状态，相对于传统的梳齿式加速度计，这种睡眠模式能大幅度降低耗电量。结构框图如图 4-2 所示。

三轴加速度计被固定在加速度计模块上，然后再将这个加速度计模块整

体插在 CT298 评价板上，结构如图 4-3 所示。

尽管这个迷你评价板的尺寸为 $47\text{mm} \times 72\text{mm}$ ，对于嵌入式设备来说显得有些大。但实际上用户并不需要担心这些，因为该评价板被做成产品时会删除像 JTAG、串口和所有的按键这样的多冗余的功能模块，从而大大降低尺寸。

数据将通过串口传输至 PC 端，这里我们使用运行在 Windows 平台的 Matlab 完成字符识别系统的分析，如图 4-4 所示。

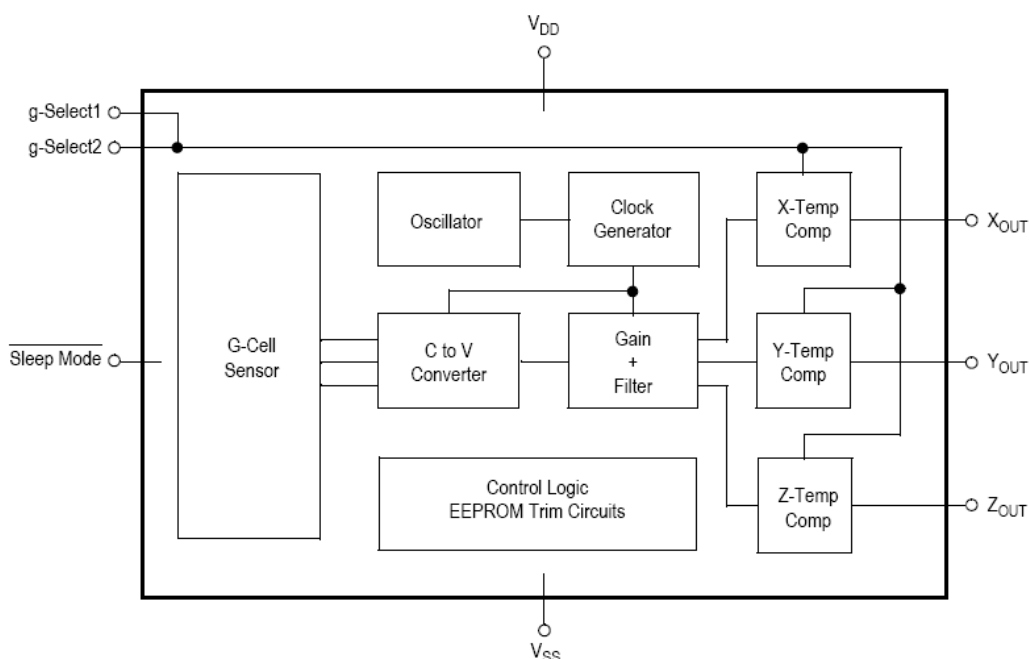


图 4-2 MMA7260T 内部结构

Fig. 4-2 Inner structure of MMA7260T

4.3 字符识别算法

这一部分中我们将介绍基于加速度计的字符识别的过程，包括有效区段提取、零偏补偿、滤波、特征提取和分类器。算法流程如图 4-5 所示。

前三部分属于预处理，因为它们的目的是为了后两步准备数据。考虑到不管用户在什么方向“书写”字符，这个字符都一定落在一个二维平面上，我们忽略掉 z 轴方向的加速度，这样可以减小嵌入式系统的运算量。但这样做的同时用户必须按照一定的规则完成字符的书写^[13~17]。

1. 保证传感器的 x 轴和 y 轴在运动过程中是保持不变的

2. 传感器的运动轨迹就在上述 x 轴和 y 轴所在的平面

3. 传感器的重心只能在一个平面内运动，也就是 x 、 y 轴所在的平面

在保证以上前提的条件下，所有的操作将从三维简化到二维空间，否则将导致系统数据的混乱。

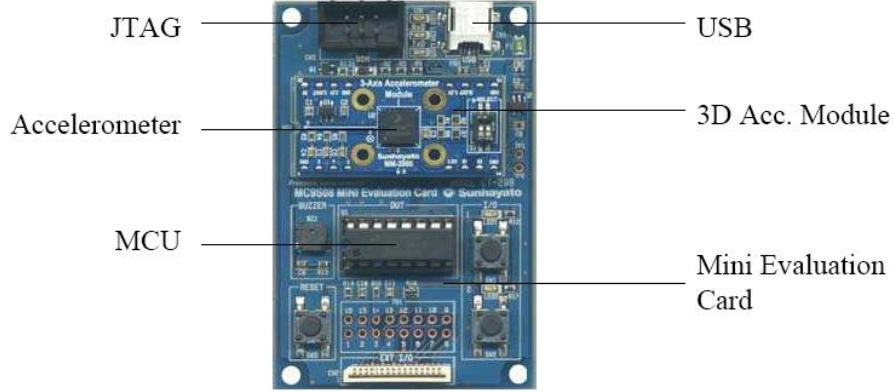


图 4-3 装有加速度计的迷你评价板

Fig. 4-3 Mini Evaluation Board With Accelerometer Implemented

4.3.1 预处理

加速度信号以信号流的形式通过串口从迷你评价板传输到 PC 端，所以在后续处理之前，我们必须把数据流分割成很多段，同时保证每一段包含且仅包含一个字符的信息。提取有效区段，唯一的任务就是找到一个字符的开始和结束位置。

当加速度计停止运动的时候，加速度计在 x 方向和 y 方向的输出都将是不变的，但不一定保证为零。为了检测到字符的起始位置，我们设定一个经验阈值。无论 x 方向还是 y 方向的加速度超过这个阈值，都将被视为一个字符的开始。同样的原理，当 x 方向和 y 方向的加速度值都保持不变一段时间之后，我们就定义一个字符的结束。

$$\begin{cases} |a_x(T+1) - a_x(T)| < \theta \quad \text{and} \quad |a_y(T+1) - a_y(T)| < \theta & T < T_0 \\ |a_x(T+1) - a_x(T)| > \theta \quad \text{or} \quad |a_y(T+1) - a_y(T)| > \theta & T = T_0 \end{cases} \quad (4.1)$$

对于， $T = T_{end}, T_{end} + 1, \dots, T_{end} + N$ ，我们要求：

$$\begin{cases} |a_x(T_{end}+1)-a_x(T_{end})|>\phi \text{ or } |a_y(T_{end}+1)-a_y(T_{end})|>\phi \\ |a_x(T+1)-a_x(T)|<\phi \text{ and } |a_y(T+1)-a_y(T)|<\phi \end{cases} \quad (4.2)$$

式中 T —— 加速度向量的时间序号；
 T_0 —— 字符的起始时间序号；
 T_{end} —— 结束位置的时间序号；
 π —— 起始位置的加速度阈值；
 θ —— 结束位置的加速度阈值；
 N —— 结束时的保持时间长度

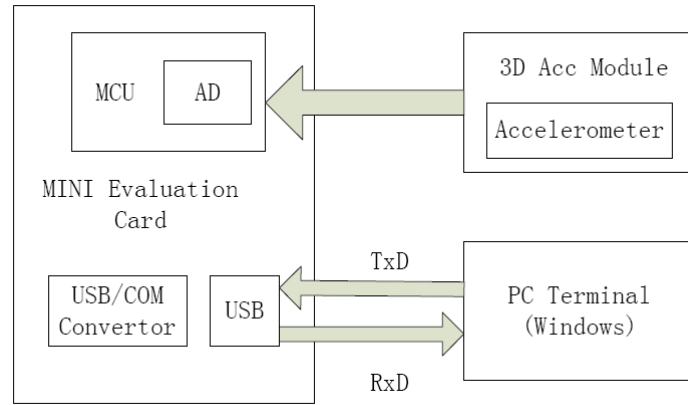


图 4-4 系统数据流向
 Fig. 4-4 Data Flow in System

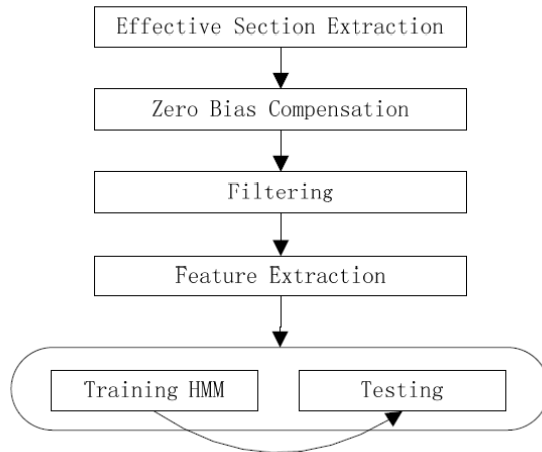


图 4-5 字符识别流程
 Fig. 4-5 Flow Chart of Character Recognition

微电子机械系统 MEMS (Micro-electromechanical System) 的制造工艺在近年来已经有了很大的提高, 但仍然有很多无法解决的问题, 比如随机漂移和噪声。加速度计在任何方向都是静止的状态下的输出被称为零偏。这个常量是重力加速度与一个漂移的叠加。然而这个漂移却受到很多不可估计的环境因素的影响, 比如温度和电磁场。该漂移可以被理解为随机的, 但在短时间内其值却变化甚微, 于是输入数据首先被零偏补偿模块处理。

$$a_0 = \frac{1}{T_{end} - T_0} \sum_{i=T_0}^{T_{end}-1} a_r(i) \quad (4.3)$$

$$a(i) = a_r(i) - a_0 \quad i = T_0, \dots, T_{end} \quad (4.4)$$

式中 a_0 —— 有效区段内的平均加速度;

$a_r(i)$ —— 原始的加速度输入;

$a(i)$ —— 和处理后的加速度

这时的加速度计输出为一系列离散的加速度值, 由于加速度计结构的内在缺陷和一些随机噪声的存在, 这个输出将包含有较大的噪声, 均值滤波器在零偏补偿模块之后完成噪声的过滤。

$$a(j) = \frac{1}{M} \sum_{i=j}^{j+M-1} a(i) \quad j = T_0, \dots, T_{end} - M \quad (4.5)$$

式中 $a(j)$ —— x 方向或 y 方向的加速度值;

M —— 可以调整滤波强度的参数

于是我们可以根据 x 方向和 y 方向的加速度值, 画出 10 个阿拉伯数字的加速度值曲线, 如图 4-6 所示。

4.3.2 特征提取

特征提取是模式识别过程中最重要的一部分, 对基于加速度计的字符识别也是一样。在基于视觉 (vision-based) 的字符识别中, 已经有很多成熟的特征提取算法, 尽管它们都是在空域内而不是加速度阈内完成, 我们还是可以借鉴其中的一些方法。在文献[9]中, 一种基于特征点的识别方法被提出, 也就提取出加速度值的最大点和最小点, 接着用 BN 算法进行字符识别, 这种算法我们定义为极值算法 EP (Extremem Points)。在本节中, 轨迹方向算法 TO (Trajectory Orientation) 和曲线特征算法 CF (Curve Feature)

也将依次被介绍。

4.3.2.1 轨迹方向算法 TO

尽管没有通过两次积分将加速度值转化成空域的值，我们还是可以直接通过加速度信息画出加速度阈的轨迹曲线。于是，x 轴就是 x 方向的加速度值，y 轴就是 y 方向的加速度值，如图所示，图 4-6 被重新绘制成图 4-7。

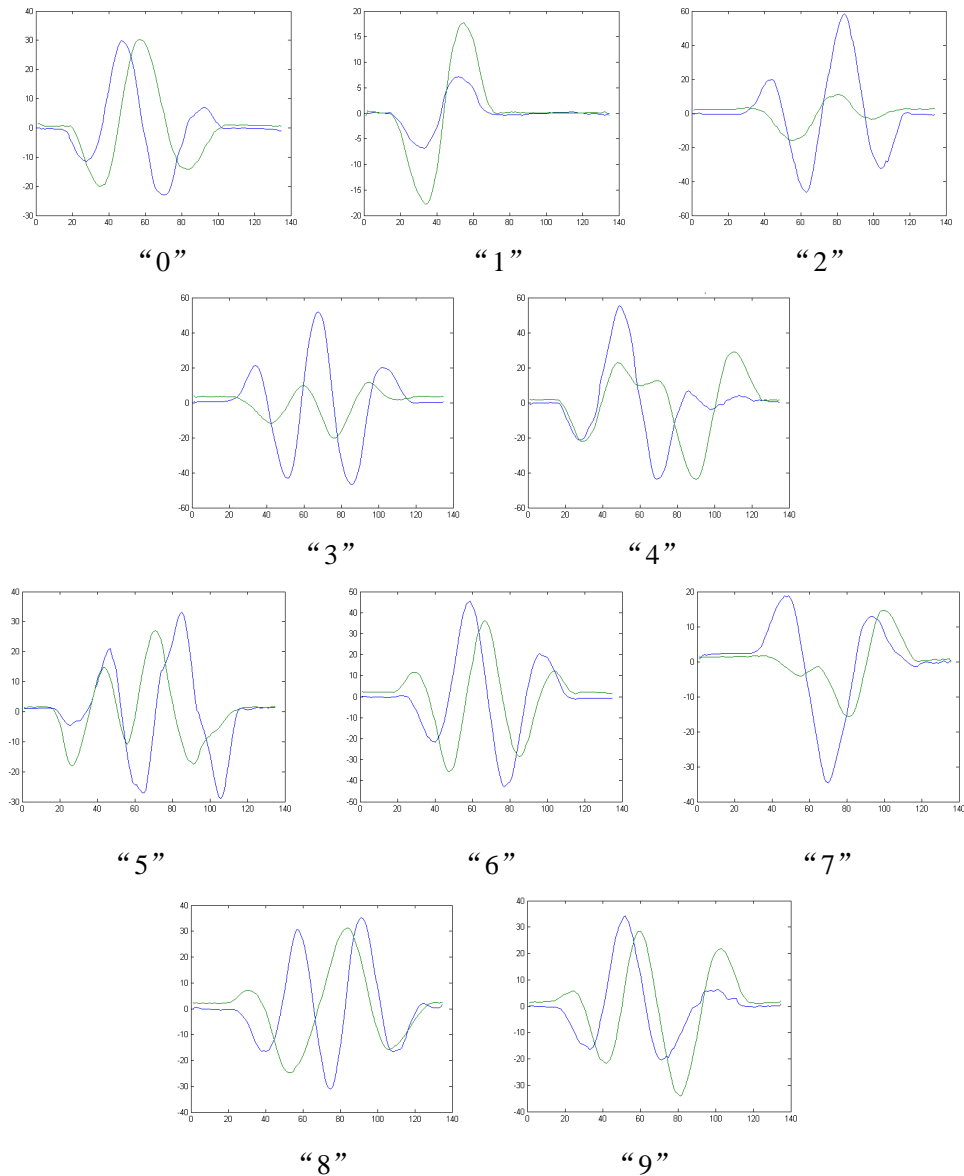


图 4-6 在 x 方向和 y 方向的加速度曲线

Fig. 4-6 Acceleration Curves in x and y Direction

为了有效地提取方向信息，二维空间被分割成几部分。理论上有无穷多种分割方法，但如下的一些限制条件使我们最终选择了如图 4-8 所示的分割方式。

1. 由于加速度计采样率的限制，空间分割的数量不能太大
2. 为了不损失角度信息，分割的数量也不能太小
3. 分割方法必须反映出曲线的特征（如图 4-7 中包含很多环形）
4. 考虑到加速度计的输出值域，分割应该在一个方形区域内完成

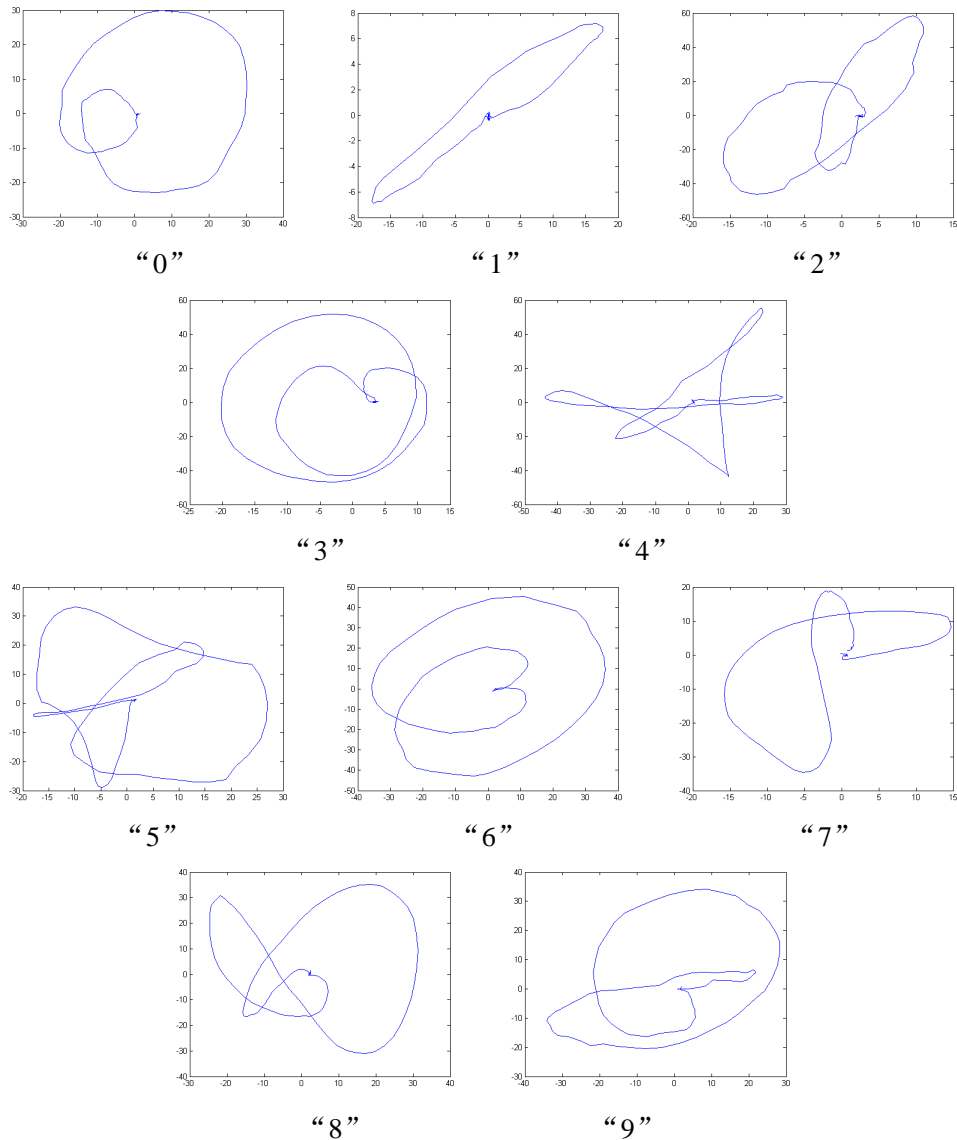


图 4-7：阿拉伯数字在加速度域的轨迹曲线

Fig. 4-7 Trajectory Curve of Arabic Numerals in Acceleration Domain

根据加速度域的轨迹曲线，就可以提取出一系列的数字，这些数字包含了极坐标中的角度信息。同时，注意到提取出的数字有大量的重复，在接下来的部分，我们还将证明这些重复是否对字符的识别有意义。针对是否包含连续的、重复的数字，隐马尔科夫模型（HMM）可以有左右型和全连接型两种。关于 HMM 的问题将在接下来的章节中介绍。

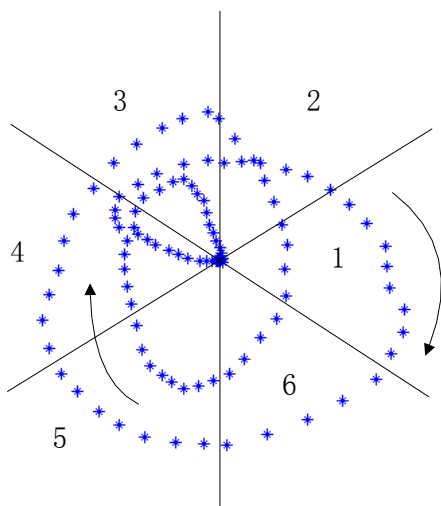


图 4-8 根据轨迹方向的特征提取算法

Fig. 4-8 Feature Extraction Algorithm Based on Trajectory

4.3.2.2 极值点算法 EP

加速度曲线中的最大值和最小值点也包含着大量的信息，这些信息也可以用来识别字符。以图 4-9 为例，从图 4-9-a 中可以提取出图 4-9-b 中的极值点。不同的极值点顺序就可以表示字符的特性。如图 4-10，用一个有限状态机 FSM（Finite State Machine）来完成极值状态的追踪。可以定义状态 1 为 x 轴的极大值，状态 2 为 x 轴的极小值。这时的状态机是不能从一个状态跳回到原来的状态，这种 FSM 被定义为左右型（LR）HMM。对于任何一个状态都可以跳转到任何一个状态（包含原状态）的 HMM，我们定义为全连接型（FC）HMM，接下来我们证明在基于加速度的字符识别领域 FC 的性能要优于 LR。

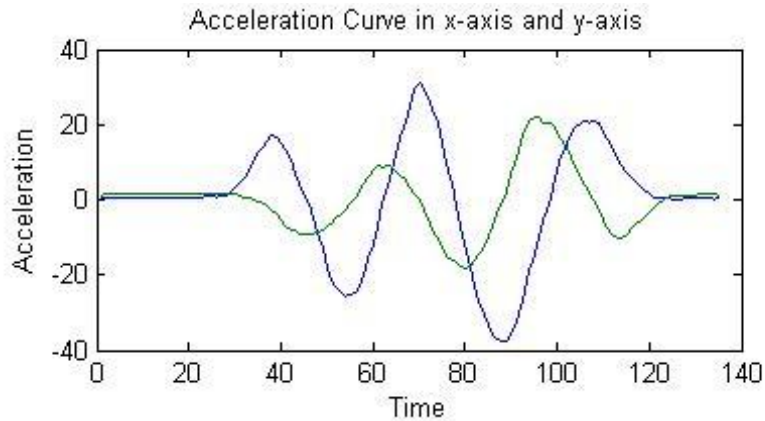
4.3.2.3 曲线特征算法 CF

对于在加速度域的 online 类型的字符识别，任何两个相邻的点的位置都包含了大量的信息，这些信息也可以用来对字符进行识别。如图 4-11 所

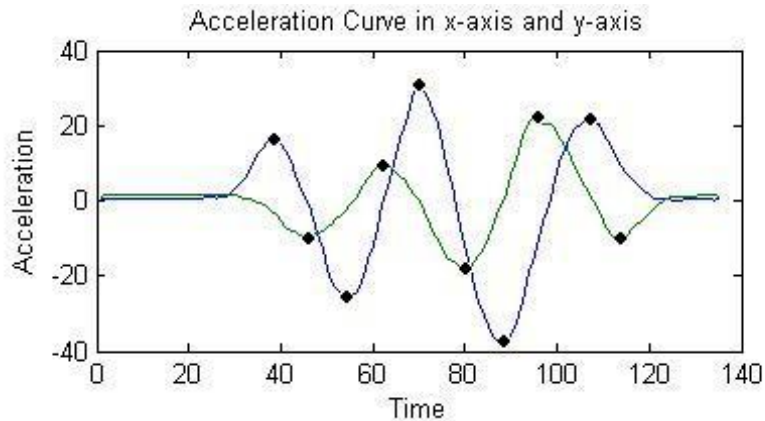
示，每个点的坐标都表示在确定时间的加速度值，根据这些值可以计算出加速度向量的绝对值和方向。这个加速度向量可以看做加速度点之间的加速度向量或者理解为加速度值的即时变化量。

4.3.3 离散隐马尔科夫模型

这里我们用 HMMSim 来简化运算量，在文献 xx 中，这个工具软件的作者详细介绍了开发 HMMSim 的动机和使用方法。



a) 原始加速度曲线



b) 经过极值点提取的加速度曲线

图4-9 极值点特征提取算法

Fig. 4.9 Feature Extraction Algorithm Based on Extremum Points

4.3.3.1 隐马尔科夫模型（HMM）

马尔科夫模型是一种分析随机过程的数学模型。在这类数学模型中，可

以通过马尔科夫模型，根据一系列的概率分析随机过程的结果。该模型是可以通过训练得到的，同时因为后台运行的随机过程是不可见的，我们称之为隐马尔科夫模型。对于 HMM 的研究的挑战在于，如何通过可观察到的参数决定模型参数。一旦模型参数被决定，HMM 便可以用来分析随机过程，在模式识别领域已经被广泛应用。隐马尔科夫模型可以被看做最简单的贝叶斯网络模型，所以 BN 的成熟理论可以直接用在 HMM 的分析上。

隐马尔科夫是一系列通过转换参数连接的状态的集合，每一个状态都有一组唯一的可见符号组成，这些可见的信息对应着系统的物理输出。HMM 可以简单的表示为如下式子。

$$\lambda = (A, B, \pi) \quad (4.6)$$

式中 A ——矩阵 $\{a_{ij}\}$ ，状态转换概率分布；

a_{ij} ——从 S_i 到 S_j 的转换概率；

B ——矩阵 $b_i(x)$ ，状态 S_i 的可见符号概率分布；

π ——初始化状态概率分布向量

A 是一个 $N \times N$ 的矩阵，B 是一个 $N \times M$ 的矩阵， π 是一个 $1 \times N$ 的向量。这些矩阵和向量最初都要根据经验进行初始化。这里 N 代表模型的状态数，M 代表可观察状态的数量。HMM 可以分为 FC (Fully Connected) model、Left-Right model 和 Left-Right Banded model。在本文中采用第一种模型来初始化 HMM 模型，这里的 N 和 M 都被设定为 10，A、B、 π 中的所有单元都被设定为 0.1，这些初始值并不精确，但是通过大量的迭代，它们将趋近一个很好的输出。

4.3.3.2 分类器

HMM 主要解决三类问题：

1. 我们有大量的已经训练好的 HMM 模型，这些模型分别描述了不同的系统，当我们手里重新得到一个可观察序列时，问题就是找到最可能产生这个序列的 HMM。
2. 根据一系列的大量的可观察序列，找到最可能的状态序列，通过可见状态推测隐藏的不可见序列。
3. 通过观察序列产生 HMM 模型。

对于特定的可观察序列，用 HMMSim 我们可以很方便的找到概率最大

化的 HMM 参数模型，这也就是问题 3，同时还能估计出特定可见状态发生的概率，也就是问题 1。HMM 的训练和检测流程如图 4-12 所示。

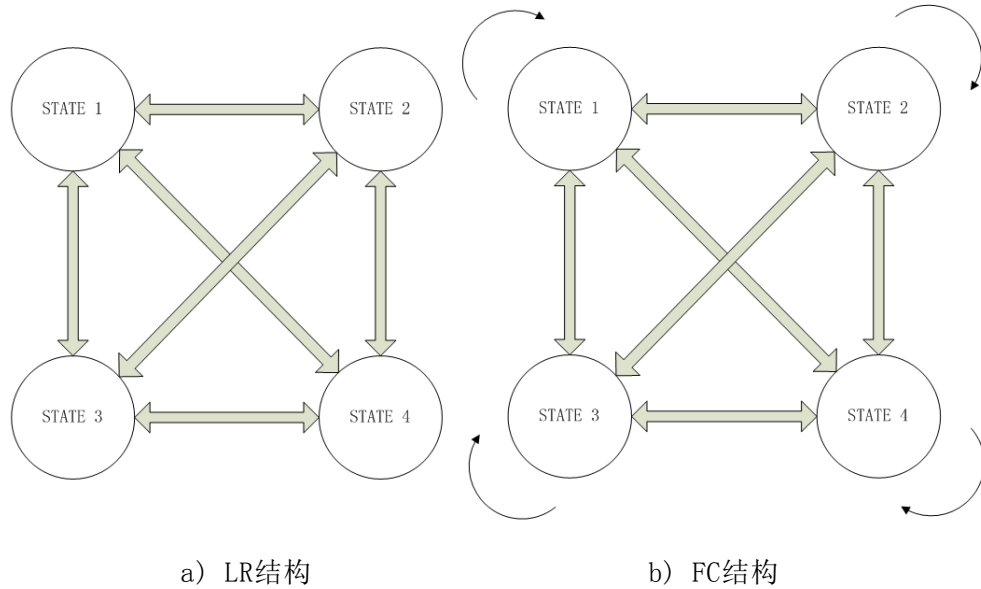


图4-10 两种结构的极值点状态机
Fig. 4-10 Two Structures of Extremum State Machine

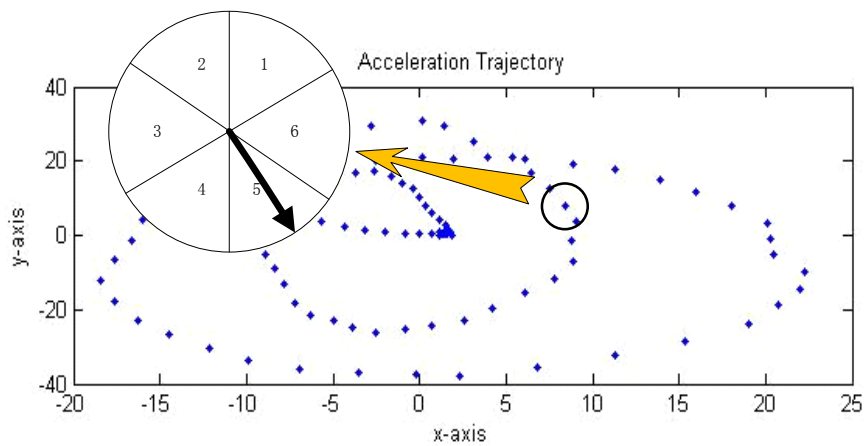


图 4-11 曲线特征算法
Fig. 4-11 Algorithm of Curve Feature

4.4 实验结果

在上一节中已经介绍了 TO 算法、EP 算法和 CF 算法。根据同样的顺序

再来分别比较一下他们对于 LR 和 FC 结构的 HMM 的性能。首先测试了 LR 结构的 HMM 模型，这里所有的数字重复都被省略掉。这样特征数字就只能从一个状态跳到另一个，而不能再跳回到原来的状态。这些没有重复的数字直接用在分类器中。用两种不同的 HMM 结构，用 TO 算法得到表 4-1 和表 4-2，横轴表示用户输入的数字，纵轴表示系统识别出的字符。

表 4-1 使用 TO 算法、LR 结构 HMM 的识别统计

Table 4-1 Recognition results by HMM with TO algorithm and LR structure

Out	In	0	1	2	3	4	5	6	7	8	9
0		20		7	11						
1			19								
2				12							
3		1		2	10						3
4			1			20					
5			1				20				
6								18			
7									20		
8										21	
9											18
None						1	1		1		

在使用 LR 结构的 HMM 时，数字“2”和“3”识别的结论大约半数是不正确的。对于用户来说，这么低的正确率是不能接受的。当使用 FC 结构的 HMM 时，对每一个阿拉伯数字，输出正确率都有了较大幅度的提高。

对于 EP 算法的实现，在 x 方向和 y 方向都只有两种特征点，所以量化数量为 4。如果 z 方向没有被忽略，量化数量将会是 6，但根据前面的分析，我们知道这个量化数量的增加并不会包含更多的信息。与 TO 算法相似，我们也测试了 EP 算法和 CF 算法，实验结果将同 TO 算法一同表示。

为了更好的表示这三种算法的性能，这里定义三个衡量识别率和正确率的标准。

$$RR(N) = \frac{R(N)}{T(N)} \quad (4.7)$$

$$CR(N) = \frac{C(N)}{R(N)} \quad (4.8)$$

$$TCR(N) = \frac{C(N)}{T(N)} \quad (4.9)$$

式中 RR ——Recognition Rate，表示识别率，不区分识别结果正确与否；

CR ——Correct Rate，表示正确率，是在识别出来的字符中统计的；

TCT ——Total Correct Rate，总的正确率，在所有用户输入中统计；

R ——所有识别出来的字符数目；

C ——表示识别出来，并且正确的数目；

T ——所有输入的字符的数目；

N ——我们要检测的字符数，也就是从 0 到 9 的十个数字

表 4-2 使用 TO 算法、FC 结构 HMM 的识别统计

Table 4-2 Recognition results by HMM with TO algorithm and FC structure

Out	In	0	1	2	3	4	5	6	7	8	9
0		21									
1			19								
2				20							
3				1	21						
4			1			19		1			1
5			1				19				
6								20			
7									21		
8										21	
9								2			17
None						2	2	1			3

之所以要定义 RR、CR 和 TCR 是因为在不同的情况下需要用不同的标准来衡量一个系统的优劣。对于希望系统识别的字符有很高的正确率的用户，他们就希望得到 CR 很高的系统，而这种系统很有可能识别率 RR 并不高。根据一系列的实验结果，绘制出一个描述识别算法和 HMM 结构的六种方法的柱状图，如图 4-13 所示。

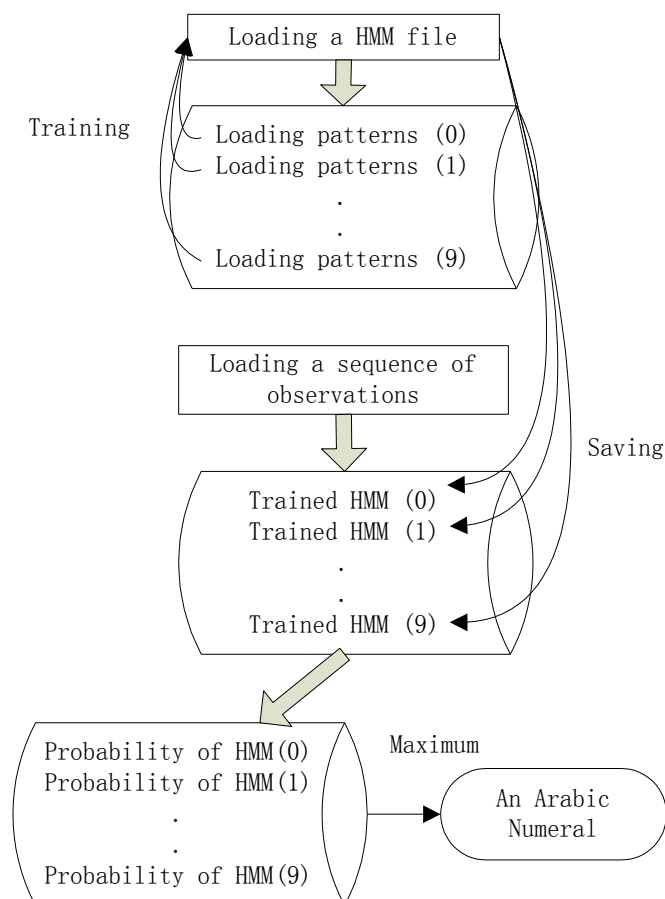


图 4-12 HMM 训练与检测流程

Fig. 4-12 Flow Chart of HMM Training and Detecting

从图中可以看出一些数字，如“0”和“8”，更容易被识别，因为他们的加速度轨迹中包含更多的特征信息。在使用 EP 算法时，“1”的 TCR 远比其他数字低很多，这主要是因为该数字的极值点很不容易判断。总体来说奇数列的识别率低于其右侧，这也就意味着 FC 结构的 HMM 性能要优于 LR 结构。另外一个结论就是 TO 算法和 CF 算法要比 EP 算法有更高的识别率。根据实验数据列出表 4-3。

表4-3 特征提取算法与HMM结构组合的性能统计

Table 4-3 Performance statistics of the combinations of feature extraction algorithms and HMM structures

	LR-TO	FC-TO	LR-EP	FC-EP	LR-CF	FC-CF
<i>RR</i>	98.57	96.19	95.24	98.57	90.01	99.05
<i>CR</i>	86.19	98.02	87.40	89.00	93.22	95.67
<i>TCR</i>	84.76	94.29	82.86	87.62	83.81	94.72

正如上面的分析，评价一个字符识别系统，CR 和 TCR 都是重要的衡量标准。其一，LR-TO 算法的 CR 是最高的，接近于 100%，所以这种组合可以看做是可靠地也是可行的。其二，尽管 FC-CF 的 CR 低于 FC-TO，前者的 TCR 全远高于任何一种其他的组合，所以这种组合也是一种较优的选择。

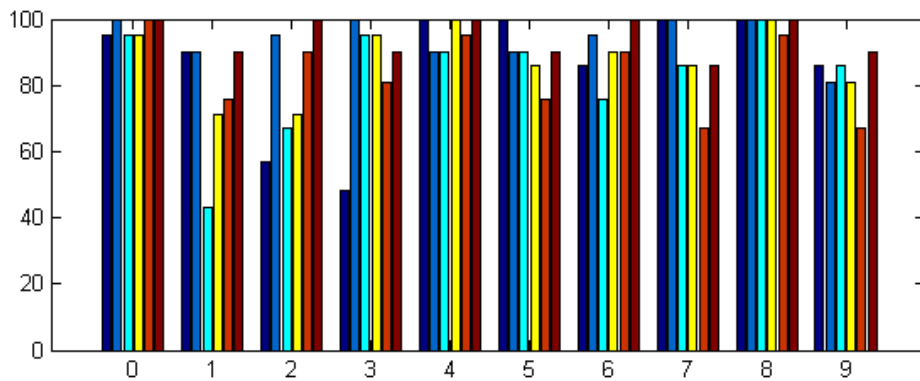


图 4-13 识别算法和 HMM 结构的六种组合的识别率

从左到右依次是：LR-TO、FC-TO、LR-EP、FC-EP、LR-CF、FC-CF

Fig. 4-13 Recognition Rates of the Six Combinations of Recognition Algorithms and HMM Structures

From Left to Right are LR-TO、FC-TO、LR-EP、FC-EP、LR-CF、FC-CF

4.5 本章小结

基于高性能的三轴加速度计，建立了一个阿拉伯数字识别系统，识别过程在加速度域完成，这样既降低了系统的复杂的，也避免了积分过程中带来的误差，最后用离散 HMM 模型进行分类。从实验结果看出，系统工作稳定、正常，识别率高。

结论

论文首先讨论了家庭网关的发展现状，根据目前市场上的家庭网关的不足，提出了基于 IBM STB04500 多媒体处理芯片的多媒体家庭网关 MHG。参考 Redwood5 评价板，实验室自行设计了 MHG 硬件平台，并且基于此平台开发了相应的软件平台和部分应用软件。同时为 MHG 平台设计了基于加速度传感器的新型无线控制器，用户通过该控制器可以彻底摆脱复杂按键的束缚，该无线控制器操作简单、体积小，市场应用前景好。简言之，围绕 MHG 平台，主要完成了：

1. MHG 硬件平台的设计与调试
2. MHG 平台的 OpenBios 编写，以及嵌入式 Linux 系统内核的裁剪
3. 为 MHG 平台设计新型的无线控制器
4. 在基于惯性器件的模式识别中，讨论了三种特征识别方案的优劣

以上工作的完成体现了 MHG 平台界面友好、操作简便、应用广泛、易于扩展及后续开发等优点，该平台有很好的应用前景。

参考文献

- 1 Rejean Plamondon, Sargur N. Srihari, On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey IEEE, Transaction on Pattern Analysis and Machine Intelligence, VOL 22, NO.1, January 2000
- 2 adidas, <http://www.micoach.com/>
- 3 Nike, <http://nikeplus.nike.com/nikeplus/>
- 4 Sung-Do Choi, Alexander S. Lee, and Soo-Young Lee, On-Line Handwritten Character Recognition with 3D Accelerometer Proceeding of the 2006 IEEE International Conference on Information Acquisition, August 20-23, 2006, Weihai, Shandong, China
- 5 IBM, Redwood5 Data Sheet
- 6 李强, 多媒体家庭网关软件平台设计与实现, 清华大学工学硕士学位论文, 2007.1
- 7 IBM, STB04500 Data Sheet
- 8 柴潇毅, “多媒体家庭网关设计与实现”, 清华大学工学硕士学位论文, 2006.5
- 9 SPANSION, Am29LV320D Data Sheet
- 10 SHARP, BS2F7VZ0624 Data Sheet
- 11 IBM, RISCWatch Debugger User's Manual
- 12 张坤, MHG 应用环境及其编解码库的设计与实现, 清华大学工学硕士毕业论文, 2007.1
- 13 Mahmoud Elmezain, Ayoub Al-Hamadi, Gesture Recognition for Alphabets from Hand Motion Trajectory Using Hidden Markov Models. 2007 IEEE International Symposium on Signal Processing and Information Technology, 2007
- 14 Jian Kang Wu, Liang Dong and Wendong Xiao, Real-Time Physical Activity Classification and Tracking using Wearble Sensors, 2007 6th International Conference on Information Communication and Signal Processing, Singapore, pp. 1-6, Dec.2007
- 15 Jin He, Huaming Li and Jindong Tan, Real-time Daily Activity Classification with Wireless Sensor Networks using Hidden Markov Model, Proceedings of the 29th

- Annual International Conference of the IEEE EMBS, Lyon, France, August 23-26, 2007
- 16 Meng Chen, Bufu Huang and Yangsheng Xu, Human Abnormal Gait Modeling via Hidden Markov Model, Proceedings of the 2007 International Conference on Information Acquisition, Jeju City, Korea, July 9-11, 2007
 - 17 Eun-Seok Choi, Won-Chul Bang, Beatbox Music Phone: Gesture-based Interactive Mobile Phone using a Tri-axis Accelerometer, IEEE International Conference on Industrial Technology, pp. 97-102, Dec. 14-17, 2005
 - 18 J. F. Velez, A. Sanchez and A. B. Moreno, HMMSim: An Interactive Simulation Tool for Teaching Discrete Hidden Markov Models, 4th International Symposium on Computer Science Education, Vigo, Spain, November 22, 2002
 - 19 Sushmita Mitra and Tinku Acharya, Gesture Recognition: A Survey, IEEE Transactions on System, Man, and Cybernetics, Vol.37, NO.3, May 2007
 - 20 N. Liu, B. C. Lovell and P. J. Kootsookos, Model Structure Selection and Training Algorithms for a HMM Gesture Recognition System, International Workshop in Frontiers of Handwriting Recognition, pp. 100-106, October 2004
 - 21 VMMantyla, Discrete hidden Markov models with application to isolated user-dependent hand gesture recognition, Espoo 2001. Technical Research Centre of Finland, VTT Publications 449. 104p
 - 22 Renado De Mori, Spoken language understanding: a survey, 2007 IEEE Workshop on Automatic Speech Recognition and Understanding, 9-13 Dec. 2007
 - 23 Hiromichi Fujisawa, Forty years of research in character and document recognition-an industrial perspective, Pattern Recognition 41 (2008) 2435-2446
 - 24 Henry S. Baird, Daniel Lopresti, Robust Document Image Understanding Technologies, HDP'04, ACM, Washington, DC, USA, November 12, 2004
 - 25 Sung-Jung Cho, Jong Koo Oh and Won-Chul Bang, Magic Wand: A Hand-Drawn Gesture Input Device in 3-D Space with Inertial Sensors, Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition, 2004
 - 26 Guanglie Zhang and Guangyi Shi, Towards an Ubiquitous wireless Digital Writing Instrument Using MEMS Motion Sensing Technology, Proceedings of the 2005

- IEEE/ASME International Conference on Advanced Intelligent Mechatronics Monterey, USA, 24-28 July, 2005
- 27 Sanna Kallio, Huha Kela and Jani Mantyjarvi, Online Gesture Recognition System for Mobile Interaction, IEEE International Conference on System, Man and Cybernetics, pp. 2070-2076, Vol. 3, 5-8 Oct. 2003
- 28 闫明, 段发阶. 曲线的特征识别方法研究. 传感技术学报. 2006 年 6 月, 19 卷, 第三期: 724-727
- 29 张剑飞. 贝叶斯网络学习方法和算法研究. 东北师范大学工学硕士学位论文. 2005 年 5 月: 27-32
- 30 宫秀军. 贝叶斯学习理论及其应用研究. 中科院工学博士学位论文. 2002 年 6 月: 24-34
- 31 P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. Proceedings of the International Conference on Machine Learning, ACM Press, New York, NY, USA, 2004
- 32 B. Argall, B. Browning, and M. Veloso. Learning by demonstration with critique from a human teacher. In Second Annual Conference on Human-Robot Interaction, 2007
- 33 K. S. Barber, A. Goel, and C. Martin. Dynamic adaptive autonomy in multi-agent systems. Journal of Experimental & Theoretical Artificial Intelligence, 12:129-147, 2000
- 34 D. C. Bentivegna, A. Ude, C. G. Atkeson, and G. Cheng. Learning to act from observation and practice. International Journal of Humanoid Robotics, 1(4), 2004
- 35 Aires, S.B.K.; Freitas, C.O.A.; Bortolozzi, F. Sabourin, R.; Perceptual Zoning for Handwriting Character Recognition, In Proceedings of 12th Conference of IGS, p.178-182, 2005
- 36 Duin, R.P.W., Pekalska, E., Tax, D.M.J. The characterization of classification problems by classifier disagreements. In Proceedings of International Conference on Pattern Recognition, Cambridge - UK, p.140-143, 2004
- 37 Freitas, C.O.A., Bortolozzi, F. Sabourin, R. Study of Perceptual Similarity Between Different Lexicons, International Journal on Pattern Recognition and Artificial Intelligence, IJPRAI, 18(7): 1321-1338, 2004

- 38 Kapp, M.N., Freitas, C. and Sabourin, R. Handwritten Brazilian Month Recognition: An Analysis of Two NN Architectures and a Rejection Mechanism, In Proceedings of 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9), p. 209-214, 2004
- 39 Kittler, J. Hatef, M. Duin, R.P.W. Matas, J. On Combining Classifiers. IEEE Trans. On Pattern Analysis and Machine Intelligence, 20(3):226-239, 1998
- 40 A. Teller and M. Veloso. PADO: A new learning architecture for object recognition. In K. Ikeuchi and M. Veloso, editors, Symbolic Visual Learning, pages 77–112. Oxford Press, New York, 1997
- 41 H. Ishida, S. Yanadume, T. Takahashi, I. Ide, Y. Mekada, H. Murase, Recognition of low-resolution characters by a generative learning method, Proceedings of the 1st International Workshop on Camera- Based Document Analysis and Recognition, Seoul, Korea, 2005, pp. 45–51
- 42 刘阳. MHG 中无线控制平台和数字版权保护方案的设计与实现. 清华大学工学硕士学位论文. 2007 年 7 月: 12-21
- 43 Cherkassky, Vladimir. Multiple regression estimation for motion analysis and segmentation. IEEE International Conference on Neural Networks, 2004, 4: 2547-255
- 44 Sang-Bum Kim. A new method of parameter estimation for multinomial naïve bayes text classifiers. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 2002, 391-392

附录 1

```
unload all
print "Resetting STB chip"
halt on
reset sys
stop
halt off
#-----
# INIT VIDEO MULTIPLEXING REGISTERS
#-----
write cic0_sccr 0xc94bedee
write cic0_muxo 0x00000000
write cic0_muxod 0x1b000000
write CIC0_cr 0x108000e8
write cic0_vcr 0x00431700
write cic0_sel3 0x00000000
#-----
# INIT EBIU BANK REGISTERS
#-----
write EBIU0_BRCRH0 0x00000000
write EBIU0_BRCRH1 0x00000000
write EBIU0_BRCRH2 0x40000000
write EBIU0_BRCRH3 0x00000000
write EBIU0_BRCRH4 0x00000000
write EBIU0_BRCRH5 0x00000000
write EBIU0_BRCRH6 0x00000000
write EBIU0_BRCRH7 0x00000000
write EBIU0_BRCR0 0xFC58BFFE # set for 16 bit
#write EBIU0_BRCR0 0xFE18BFFE # set for 16 bit
write EBIU0_BRCR1 0xff00bffe # External Connector
write EBIU0_BRCR2 0x207CD0BE # Xilinx registers
write EBIU0_BRCR3 0xff00bffe
write EBIU0_BRCR4 0xff00bffe
write EBIU0_BRCR5 0xff00bffe
write EBIU0_BRCR6 0xff00bffe
write EBIU0_BRCR7 0xff00bffe
#-----
# INIT GPIOs
#-----
write dmem 0x40060000 0x00000000
write dmem 0x40060004 0x00000000 # ThreeState Control
write dmem 0x40060008 0x00000001 # Output Select 0-31
write dmem 0x4006000c 0x00000000 # Output Select 32-63
write dmem 0x40060010 0x00000000 # ThreeState Select 0-31
write dmem 0x40060014 0x00000001 # ThreeState Select 32-63
```

```

write dmem 0x40060018 0x00000000
write dmem 0x40060030 0x00000000
write dmem 0x40060034 0x00000000
write dmem 0x40060038 0x00000000
write dmem 0x4006003C 0x00000000
write dmem 0x40060040 0x00000000
write dmem 0x40060044 0x00000000
#-----
# INIT XILINX 1 Registers
#-----
#writeth dmem 0x72040000 0x0000
#writeth dmem 0x72040002 0x0002
#writeth dmem 0x72040006 0x0600
#writeth dmem 0x72040008 0x0000
#writeth dmem 0x7204000A 0x0001
#writeth dmem 0x7204000E 0x0000
#-----
# INIT HSMC 0 and 1 for SDRAM
#-----
write icbs0_amap0    0x00000000
write icbs0_amap1    0x00000000
write sdram0_besr    0x00000000
write sdram0_br0     0x00004000
write sdram0_br1     0x02004000
write sdram0_br2     0x00000000
write sdram0_br3     0x00000000
write sdram0_cr0     0x01f00000
write sdram0_cr1     0x01f00000
write sdram0_cr2     0x00000000
write sdram0_cr3     0x00000000
write sdram0_cr0     0x01f08000
write sdram1_besr    0x00000000
write sdram1_br0     0x20004000
write sdram1_br1     0x22004000
write sdram1_br2     0x00000000
write sdram1_br3     0x00000000
write sdram1_cr0     0x01f00000
write sdram1_cr1     0x01f00000
write sdram1_cr2     0x00000000
write sdram1_cr3     0x00000000
write sdram1_cr0     0x01f08000
#-----
# Set up some machine state registers.
#-----
write sgr 0
write dcwr 0
write esr 0
write tcr 0
write msr 0

```

```

write dbr 0xffffffff
write ebiu0_bsr 0xffffffff
write xer 0

#Initializing serial port
print "Initializing Serial Port"
writeb dmem 0x40040003 0x80
writeb dmem 0x40040000 0x80
writeb dmem 0x40040001 0x00
writeb dmem 0x40040003 0x03
writeb dmem 0x40040002 0x00
writeb dmem 0x40040004 0x03
readb 0x40040005
readb 0x40040000
writeb dmem 0x40040007 0x00
writeb dmem 0x40040001 0x00
writeb dmem 0x40040000 0x41
readb 0x40040005
writeb dmem 0x40040000 0x42
readb 0x40040005
writeb dmem 0x40040000 0x43
readb 0x40040005
writeb dmem 0x40040000 0x44
readb 0x40040005
writeb dmem 0x40040000 0x45
readb 0x40040005
writeb dmem 0x40040000 0x46
readb 0x40040005
writeb dmem 0x40040000 0x47
readb 0x40040005
writeb dmem 0x40040000 0x0d
readb 0x40040005
writeb dmem 0x40040000 0x0a
readb 0x40040005
#-----
# Set up some machine state registers.
#-----
write sgr 0
write dcwr 0
write esr 0
write tcr 0
write msr 0
write dbr 0xffffffff
write ebiu0_bsr 0xffffffff
write xer 0

print "Loading BIOS image into SDRAM"
load binary e:\Stb\RISCWatch\bios_release\mhg\openbios.bin 0xA0010000
#load binary .\openbios.bin 0xA0010000

```

```
print "Loading Flash update code into SDRAM"
load file e:\Stb\RISCWatch\bios_release\mhg\flprog
#load file .\flprog

#Start of program flprog.s linked at entry label
write iar 0xA0000000
bp clear all

# Flprog will branch to here and end
# this address will not change when flprog.s is modified.
bp set ihw 0xA0000004

print "Programming Flash..."

run
if (r0 == 0)
    print "Done: Flash program passed"
elseif (r0 == -1)
    print "Error: Unable to correctly read manufacturer and device id"
elseif (r0 == -2)
    print "Error: Flash program failed"
else
    print "Error: bad value in R0"
endif
end
```

附录 2

```
#include "../include/config.h"
#include "../m4/arch.inc"
#include "../include/platform.h"

//Uses 16 bit acesses.

#define FAST_PROG

.globl _entry
_entry:
// Branch to Start and begin
    b        start

// branch to here when completed
// This address will not change (rw_flash.cmd will not need modifications)
done:
    b        done
start:
// Two chips
    addis    r2,0,0
#ifdef FAST_PROG
    ori      r2,r2,0x0001          // only the last device
    mtctr    r2
    // only second chip
    addis    r3,0, ((FLASH_START>>16) + (FLASH_DEV_SIZE>>16) *
(FLASH_NUM_DEVS-1))
#else
    ori      r2,r2, FLASH_NUM_DEVS // number of devices
    mtctr    r2
//set R3 to flash base of 1st chip
    addis    r3,0, FLASH_START@h
#endif
    lis      r10, FLASH_DEV_SIZE@h

erase_chip:
//check access to flash
    addis    r0,0,0xf0
    ori      r0,r0,0xf0
    sth      r0,0x0000(r3)
    addis    r0,0,0xaa
    ori      r0,r0,0xaa
    sth      r0,0x0aaa(r3)
    addis    r0,0,0x55
    ori      r0,r0,0x55
```

```

    sth    r0,0x0554(r3)
    addis  r0,0,0x90
    ori    r0,r0,0x90
    sth    r0,0x0aaa(r3)
    lhz    r5,0x0000(r3)
    lhz    r6,0x0002(r3)
    addis  r0,0,0xf0
    ori    r0,r0,0xf0
    sth    r0,0x0000(r3)

//compare
    addis  r0,0,0x0000
    ori    r0,r0,0x0001
    cmpw   cr0,r5,r0
    bne     device_err
    addis  r0,0,0x0000
    ori    r0,r0,FLASH_DEV_STRING
    cmpw   cr0,r6,r0
    beq     device_ok

device_err:
    addi    r0,0,-1
    bl      done

device_ok:
//erase whole chip
    addis  r0,0,0xaa
    ori    r0,r0,0xaa
    sth    r0,0x0aaa(r3)
    addis  r0,0,0x55
    ori    r0,r0,0x55
    sth    r0,0x0554(r3)
    addis  r0,0,0x80
    ori    r0,r0,0x80
    sth    r0,0x0aaa(r3)
    addis  r0,0,0xaa
    ori    r0,r0,0xaa
    sth    r0,0x0aaa(r3)
    addis  r0,0,0x55
    ori    r0,r0,0x55
    sth    r0,0x0554(r3)
    addis  r0,0,0x10
    ori    r0,r0,0x10
    sth    r0,0x0aaa(r3)

erase_loop:
    lwz    r0,0x00000000(r3)
    cmpwi  cr0,r0, -1 // 0xffff
    bne     erase_loop

```



```

//set R3 to flash base of 2nd chip
    add    r3, r3, r10
    bdnz   erase_chip

//set counter to 10000 half words
    addis  r6,0,0x0001
    ori    r6,r6,0x0000
    mtctr  r6

//ram address to copy from is 0xa0010000 (r4)
//rom address to write to is 0xfffe0000 (r5)
//rom base address of 2nd chip 0xffe00000 (r3)
    addis  r3,0,0xffe0
    addis  r4,0,0xA001
    addis  r5,0,0xfffe

    bl     program_flash
#ifndef FAST_PROG

//set counter to 00012 half words - length of vpd_string / 2
    addis  r6,0,0x0000
    ori    r6,r6,0x0010
    mtctr  r6
//ram address to copy from is vpd_string (r4)
//rom address to write to is 0xffc00000 (r5)
//rom base address of 1st chip 0xffc00000 (r3)
    lis    r3, VPD_ADDR@h
    ori    r3,r3, VPD_ADDR@l

    lis    r4, vpd_string@h
    ori    r4,r4, vpd_string@l
    or     r5, r3, r3
    bl     program_flash
#endif
    addi   r0,0,0
    bl     done
program_flash:
    mflr   r31
//program_flash
top_of_loop:
    addis  r0,0,0xaa
    ori    r0,r0,0xaa
    sth    r0,0x0aaa(r3)
    addis  r0,0,0x55
    ori    r0,r0,0x55
    sth    r0,0x0554(r3)

```

```

    addis    r0,0,0xa0
    ori      r0,r0,0xa0
    sth      r0,0x0aaa(r3)
    lhz      r0,0(r4)
    sth      r0,0(r5)
prog_loop:
    lhz      r6,0(r5)
    xor      r7,r6,r0
    andi.    r7,r7,0x80
    beq      prog_low_done
    andi.    r6,r6,0x20
    beq      prog_loop
    lhz      r6,0(r5)
    xor      r7,r6,r0
    andi.    r7,r7,0x80
    bne      prog_fail
prog_low_done:
    lhz      r6,0(r5)
    xor      r7,r6,r0
    andis.   r7,r7,0x80
    beq      prog_loop_done
    andis.   r6,r6,0x20
    beq      prog_low_done
    lhz      r6,0(r5)
    xor      r7,r6,r0
    andis.   r7,r7,0x80
    beq      prog_loop_done
prog_fail:
    addi     r0,0,-2
    bl       done
prog_loop_done:
    addi     r4,r4,2
    addi     r5,r5,2
    bdnz     top_of_loop
prog_done:
    mtlr     r31
    blr

vpd_string:
    .byte    "*VPD"
    .byte    0x00,0x00
    .byte    "*NA"
    .byte    0x00,0x06
    .byte    0xff,0xff,0xff,0xff,0xff,0xff
    .byte    "*END"
    .byte    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
    .byte    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00

```

攻读学位期间发表的学术论文

- 1 Liu Zhao, Yuan Zhijie, Xu Weichang, Zhang Shiqi, Song Xiang and Huang Qibo, “Gesture Recognition Device”, Journal of Global Electronics China, Vol 4, 2007 (中文)
- 2 Shiqi Zhang, Chun Yuan, Yan Zhang, “Self-defined Gesture Recognition on Keyless Handheld Devices Using MEMS 3D Accelerometer”, IEEE Int. Conf. on Natural Computing, Jinan, China, 20 Oct. 2008 (EI & ISTP)
- 3 Shiqi Zhang, Chun Yuan, Yan Zhang, “Handwritten Character Recognition Using Orientation Quantization Based on 3D Accelerometer”, ACM Int. Workshop on Human Control of Ubiquitous Systems”, Dublin, Ireland, 20-25 Jun. 2008 (ACM & EI)

攻读学位期间已提交的学术论文

- 1 Shiqi Zhang, Chun Yuan, Yan Zhang, “Comparison of Feature Extraction Methods for Handwritten Character Recognition Based on Acceleration”, Int. Journal of Pattern Recognition and Artificial Intelligence (已提交)

哈尔滨工业大学硕士学位论文原创性声明

本人郑重声明：此处所提交的硕士学位论文《 》，是本人在导师指导下，在哈尔滨工业大学攻读硕士学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

作者签字

日期： 年 月 日

哈尔滨工业大学硕士学位论文使用授权书

《 》系本人在哈尔滨工业大学攻读硕士学位期间在导师指导下完成的硕士学位论文。本论文的研究成果归哈尔滨工业大学所有，本论文的研究内容不得以其它单位的名义发表。本人完全了解哈尔滨工业大学关于保存、使用学位论文的规定，同意学校保留并向有关部门送交论文的复印件和电子版本，允许论文被查阅和借阅。本人授权哈尔滨工业大学，可以采用影印、缩印或其他复制手段保存论文，可以公布论文的全部或部分内容。

保密 ☐，在 年解密后适用本授权书。

本学位论文属于

不保密 ☐。

（请在以上相应方框内打“√”）

作者签名：

日期： 年 月 日

导师签名：

日期： 年 月 日

致谢

感谢我的导师张岩教授。张老师工作兢兢业业、带学生认真负责，对每个学生都能做到因材施教、循循善诱，并且在紧张的工作之余还常组织同学到户外活动，节日里也能和学生一起度过，让我们在异乡感受到家的温暖。衷心感谢张老师这两年来对我的教诲和帮助。

感谢我在清华大学交流期间的指导老师袁春副教授，袁老师平易近人、诲人不倦，为我的研究提供了很好的平台，每周例会上的悉心指导也给了我莫大的帮助。深深地感谢袁老师的指导和帮助。

感谢清华大学的钟玉琢教授，钟老师德高望重，让我高山仰止。做人、做学问，我都从钟老师身上学会了很多。

感谢清华大学的吴志勇老师在发表论文方面的指导，感谢张峤同学在我发文章时给予的帮助。

感谢我的父母和她，你们的支持和鼓励是我前进中最大的动力！

最后，向所有在我硕士期间给予我关心和帮助的亲人、师长和朋友们表示深深的感谢！