青岛科技大学

硕士学位论文

基于DM355的嵌入式网络视频监控系统设计

姓名：刘继超

申请学位级别：硕士

专业：控制理论与控制工程

指导教师：刘云

20090612

基于 DM355 的嵌入式网络视频监控系统设计

摘要

随着电子，计算机，通信和自动化技术的发展，使嵌入式网络视频监控系统成为现阶段安防系统的发展趋势。本文通过将多媒体技术，嵌入式技术和网络传输技术融合到视频监控系统领域并结合青岛市科技局项目城市污染源远程监控系统的实际需求，提出一种采用 DM355 达芬奇视频处理芯片和 MPEG4 图像压缩格式，开发成本相对较低的嵌入式视频监控系统的解决方案。

本文从嵌入式网络视频监控系统的发展现状入手，详细介绍了嵌入式视频监控系统技术基础和核心技术，对现阶段采用的嵌入式网络监控系统设计方案进行比较分析。在硬件电路设计中，大量采用了性能较好的去耦和旁路电容来加强系统的稳定性，并为了调试方便设计了 JTAG 接口。软件设计中选取嵌入式操作系统 Monta Vista Linux，并对其内核进行裁剪编译，使它和硬件系统更好的结合，最终成功移植到 DM355 硬件平台。为了提高 CPU 的利用率，采用了多线程技术，并且设计了按同一顺序访问对象的方法以防止死锁发生。同时采用 Log4plus 的日志记录机制，提高了代码的开发和调试效率。在视频采集和网络传输过程中通过共享内存的技术保证了视频传输的稳定性和流畅性，并采用自适应算法实时调整媒体速率，解决了流媒体速率收敛问题。最后经过合理测试并根据测试结果得出本系统的软硬件设计都达到了预期要求。

关键词：DM355 视频监控 嵌入式 Linux 图像采集

ABSTRACT

With the development of electronics, computer, communication and the automation technology, the embedded video monitoring system becomes the trend of present security system. The paper merges the multimedia technology, embedded technology and network transmission technology into the field of video surveillance system and combine with the actual demand of Qingdao Technology Bureau source remote monitoring system, put forward a solution of embedded video monitoring system which uses the DM355 video processing chip and the MPEG4 video compression format, and the cost of development is relative lower.

This article firstly introduces the current situation of embedded network video surveillance system and analysis the basic technology and core technologyof embedded video monitoring system in detail, compares the design scheme of embedded network monitoring system at the present stage, concludes that the four present programs can't meet the requirements, so we give a design scheme of embedded network video monitoring system which is suitable for the requirements of the project. To enhance the stability of the system, design a lot of decoupling and bypass capacitors in hardware circuit. And the JTAG debug interface is designed for convenient debugging. Select a real-time embedded operating system, Monta Vista Linux, for the real-time requirements. Modify and compile the operating system kernel to make it suitable with the hardware systems and transplant the kernel to the DM355 hardware platform successfully. In order to improve CPU utilization, use of multi-threading technology, and design a same sequential access method to prevent the deadlock. Improve the code efficiency of the development and debugging with the Log4plus mechanisms. Ensure the stability and smooth of video transmission through the shared memory technology in the video acquisition and network transmission process. To solve the problem of the convergence rate of streaming media with the adaptive algorithm which can adjust the media rate timely. Finally, after a reasonable test, the test results show that the hardware and software design of the system achieved the expected requirements.

Keywords: DM355 Video surveillance Embedded Linux Image collection

独创性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含本人已用于其他学位申请的论文或成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：刘继超　　日期：2009 年 4 月 20 日

关于论文使用授权的说明

本学位论文作者完全了解青岛科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权学校可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人离校后发表或使用学位论文或与该论文直接相关的学术论文或成果时，署名单位仍然为青岛科技大学。(保密的学位论文在解密后适用本授权书)

本学位论文属于：

保密 □，在　　年解密后适用于本声明。

不保密☑。

(请在以上方框内打“√”)

本人签名：刘继超　　日期：2009 年 4 月 20 日

导师签名：　　日期：2009 年 4 月 28 日

1 前言

1.1 课题的研究背景

本课题的来源是青岛市科技局攻关项目“城市远程污染源图像信息融合系统的研究”。本项目主要任务是借助于视频通信和信息融合技术，将现场采集到的污染源传感器数据和视频数据进行分析处理，为高层判决提供详细可靠的信息。但是本课题只是研究其中的视频监控部分。

随着人们安全意识的不断提高，数字视频监控[1]这一行之有效地监控方式被广泛的应用起来。嵌入式技术的不断发展，使产品不断智能化集成化。计算机网络技术的不断发展为网络视频监控提供了可能性。

数字视频是一种对可视化数据采用数字表现方式而非模拟表现方式进行工作的视频系统。数字视频终端设备采用数字视频格式来采集、处理、保存及播放视频，这些设备包括数码相机、监控摄像机、高级医疗成像设备、便携式视频播放器、无线手持终端、汽车信息娱乐产品、机顶盒以及其他流媒体应用。因此，数字视频技术无疑将重塑整个电子行业的面貌。当然，数字视频技术也正在使我们的视频体验、传输以及交互方式发生着深刻的变化，开始进入汽车、计算机、移动电话及网络等领域。过去，工程师们在实施数字视频时选择非常有限，硬连线以及基于 ASIC（附剥的方案总是限制着器件的用途、功能，以及它们的自适应性;虽然专用器件的灵活性稍高于 ASIC，但是，面对日新月异的多媒体标准与应用，它们的效用仍然很有限;而且缺少具有足够性能、成本足够低、灵活性足够高的数字视频开发平台）。为了解决这些难题，德州仪器公司提供了一种很好的解决方案，即基于达芬奇[2](DaVinei)技术及其产品如 TMS320DM355，简称 DM355)，以简化数字视频创新。包括两个基于数字信号处理器(DSP) 的片上系统(SoC) 以及多媒体编解码器、应用编程接口 (APD、框架与开发工具等。达芬奇(DaVinci)技术充分满足众多新兴的数字视频创新产品对实时视频的需求。这些应用领域包括: 视频安全监控系统、IP 机顶盒、视频会议、车载信息娱乐系统、便携式媒体以及数码像机等。达芬奇(DaVinci)数字媒体技术平台 DM355 具有极其丰富的板级硬件资源及外围设备接口，它几乎集成了目前业界嵌入式开发平台应具有的各类元器件。

因此从数字视频监控市场需求和青岛市科技局攻关项目要求的考虑，引出了本课题，提出了基于 DM355 的嵌入式网络监控系统的解决方案。

1.2 国内外视频监控系统发展动态

随着多媒体技术和通信技术的的发展，视频技术已经成为现在研究的热点，视频监控系统是视频技术的一个重要的应用。视频监控一直是人们关注的应用技术热点之一，它以其方便、直观、信息内容丰富而被广泛应用于许多场景。在电子技术与通信技术的发展过程中，图像监控[3]系统的技术水平，直接反映了不同阶段电子与通讯的技术发展状况。所以视频监控系统的发展大致经历了三个阶段[4]：

1. 在 20 世纪 90 年代初以前，主要是以模拟设备为主的闭路电视监控系统，称为第一代模拟监控系统。随着摄像机、电视机的出现，原始的图像监视系统就已诞生。它被广泛应用于保安、生产管理等场合。本地图像监控系统主要由摄象机、视频矩阵、监视器、录象机等组成，由视频线、控制线缆等连接。模拟监控系统一般采用模拟方式信号传输，采用视频电缆（少数采用光纤），传输距离不能太远，主要应用于小范围内的监控，如大楼监控等。监控图像一般只能在监控中心查看。

2. 在 90 年代中期，随着计算机处理能力的提高和视频技术的发展，出现的基于 PC 机的多媒体控制台系统称为第二代数字化本地视频监控系统。这种系统利用计算机的高速数据处理能力进行视频的采集和处理，利用显示器的高分辨率实现图像的多画面显示，从而大大的提高了图像质量。数字视频压缩编码技术的日益成熟，PC 的不断发展，为基于 PC 的多媒体监控系统的发展创造了条件。这种监控系统一般采用下面的结构：在远端监控现场，有若干个摄像机、各种检测、报警探头与数据设备，通过各自的传输线路，汇接融合到多媒体监控终端上，多媒体监控终端可以是一台 PC 机，也可以是专用的工业机箱组成的多媒体监控终端。除了处理各种信息和完成本地所要求的各种功能外，系统利用视频压缩卡和通信接口卡，通过通信网络，将这些信息传到一个或多个监控中心。基于 PC 的多媒体监控系统功能比较强，但稳定性不够好；功耗高；需要有人值守；同时，软件的开放性不好。

3. 在 90 年代末期，随着网络带宽、计算机处理能力和存储容量的快速提高，以及各种实用视频处理技术的研究出现，视频监控进入了全数字化的网络时代，称为第三代远程数字视频监控系统。近几年随着远程监控系统被越来越多的应用于各个领域，对视频监控系统的要求也越来越高：因此操作简单、数字化、实时可靠、经济实用、多功能的视频监控系统的开发和设计正受到越来越多的人的瞩目。基于嵌入式技术的网络化视频监控系统也应运而生。嵌入式系统被称为是以应用为中心、以计算机技术为基础、软件硬件可裁剪、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。嵌入式系统以其本身体积小，实时性高，稳定性好，支持以太网等优点，成为工控领域的新热点。基于嵌入式技术的网络化视频监控系统有效地将嵌入式技术和视频技术结合在一起，可以较好地解决基于 PC 的视频监控系统存在的问题。基于嵌入式技术的网络化视频监控[5] 主要的原理是：在前端监控现场采用嵌入式技术。摄像机传送来的视频信号和麦克风采集的音频信号数字化后由高效芯片压缩，通过组播传送方式传到网络上。用户可以直接用监控终端软件通过网络收看到监控现场的图像，被授权用户还可以控制摄像机云台镜头的动作或对系统参数进行配置操作。嵌入式数字视频监控系统是新近崛起的以数字视频压缩技术为核心的新型视频监控系统，是计算机网络技术、数字视频处理技术和嵌入式技术有机结合的高科技产品。因此，它受到了学术界、产业界和其他使用部门的高度重视[6][7]。

从国外运营商发展情况看，家庭综合通信、娱乐及综合信息服务、家庭安防、智能家居及远程控制是家庭信息化产品的四个主要方向，家庭视频监控则是家庭安防的核心产品。

国外运营商在家庭视频监控方面发展较快，以 at&t、英国电信、法国电信、西班牙电信、NTT、KT、SK 电讯为代表的运营商已经基于手机、PC 和摄像头，通过移动和无线网络，提供了家庭视频监控业务，有的还提供了附加的 IPTV、VoIP 等服务。

视频监控发展侧重于网络摄像机，英国电信（BT）将数字家庭网关称作 HomeHub，通过该网关为用户提供信息、通信、娱乐的综合服务，强调易用性，也就是即插即用和丰富的连接，还强调完整性，通过多样化的家庭终端提供多样化的服务，支持各种智能监控设备的数据接口和物理接口，在视频监控上更侧重于网络摄像机的方案。法国电信也有相应的动作，它以 Live 为统一品牌，推出了 LivePhone、LiveMusic 等设备，将 LiveBox 通过 ADSL 接入到互联网，然后通过数字家庭内的无线接入方式接入各种监控设备和智能家电。法国电信将视频监控称为 LiveZoom，采用的是 AXIS206W 无线网络摄像机，内嵌 Web 服务器，功能非常丰富，被誉为世界上最小的无线网络摄像机。西班牙电信的家庭网络业务遵循“通信——多媒体——安全、本地和远程管理——自动控制”的路线，在业务提供上具有带宽充足、可支持多个家庭子网的特点。西班牙电信采用 OSGI 的基于 WebService 的方案对家庭内各个设备实现基于 Web 形式的监控，比较侧重于无线网络摄像机监控方案。它的视频监控对任何业务都有充足的带宽提供，支持多个家庭子网，满足家庭中不同业务之间 QoS 的需要；通过 Wi-Fi 网关实现家庭网络与接口网络的互联，尽量使用无线，在屋内的任何地方都可以接入各种服务，对室内设备可以远程管理和维护，远程监控设备也多采用无线的 Wi-Fi 协议接入

网关。西班牙电信的视频监控方案被业界称为充满艺术气息的无线智能化数字家庭解决方案。

不仅仅是视频监控，美国 at&t 结合它在固定电话网、移动电话网、有线电视网、互联网等网络基础设施上的优势，与杰尔系统的 TrueONE 解决方案相结合，为用户提供电信级的手机或 PC 上的视频监控应用，以及 IPTV、VoIP 等大数据量应用，将数字家庭的智能监控服务送达到远端的 PC 或手机上，支持无线和有线的视频监控方案。日本的 NTTdocomo 利用自身在移动服务上的优势，推出了 i-mode 运营模式，可以直接通过手机或 PDA 远程控制家中的智能家电，在远程监控上采用了“M2M 远程控制”的理念，实现人和机器、机器和机器之间的远程控制。NTT 推出了可通过因特网或 FOMA 手机控制的 HC-1000 监视摄像头，该设备采用 300 万像素 CMOS 感光器，2 倍数码变焦，以太网接口，支持 802.11a/b/g，可拍摄 QVGA 画质的视频；内建“主动防御”系统，可发出刺耳的警报声吓跑盗贼，甚至还有麦克风/扬声器双向通信系统，方便用户与盗贼谈判。NTT-Neomeit（NTT 西日本的子公司）还推出了通过 PDA 或手机远程控制家电和接收监控报警信息的服务，使用起来非常方便。除此以外，还实现了生物认证，通过生物认证研发了自动门识别系统，人们站在安装于入口处的摄像机前，只需约 1 秒钟的时间，如果确认来人为公寓居民，大门就会打开，非常方便。韩国的 KT 和 SK 电讯在数字家庭的远程监控和报警上大量采用了无线传感网络中的普适计算技术（UbiquitousComputing），采用 Zigbee 的无线传输协议对家中的智能监控设备进行管理和监控，并支持通过 PC 或手机对数字家庭设备远程监控信息的获取和基于 IP 的视频监控点播，用户点播时直接由数字家庭网关提供视频的处理。KT 将数字家庭监控及联动报警系统称为 Nespot 系统，该系统立足于“控制与防止”，将有线与无线网络结合于一体。采用 Nespot 系统，不论用户在家还是在外，都可通过微型监视摄像头、安装在门上的传感器、煤气泄漏传感器等，将家庭状况实时传到用户的电脑、手机或 PDA 上。用户也可以远程遥控开灯，使家里看上去有人在。紧急情况下，用户还可以呼叫急救中心对家中的病人进行救助。

综合视频监控将成发展趋势中国电信（全球眼）和原中国网通已经在有条件的家庭发展远程视频监控，如宽视界视频监控业务前期已在中国网通的山西、山东、吉林、浙江、河南等省公司推出，受到了客户的普遍欢迎，目前宽视界业务发展迅速，2008 年上半年，仅湖北网通“宽视界一神眼”业务就发展了 500 多户用户。随着家庭信息化的迅猛发展，用户对家庭安防的需求表现得越来越强烈，宽视界将会保持迅速的发展。中国移动也在发展以“手眼通”为代表的手机彩信监控，基于 3G 的视频监控也在测试和试商用中。由于数据和信号流庞大，对网

络依赖性强，需要家庭部署家庭网关和摄像头等，因此家庭视频监控研发和推广难度还比较大，目前视频监控主要应用在行业领域和家庭紧密联系的小区，家庭监控应用得还比较少。在平安城市、平安家庭政策的引导下，伴随着家庭网络环境不断改善，用户消费能力提升，安防意识增长，独立式居住家庭增多，国内家庭监控将开始兴起。在我国全业务运营已经到来，运营商必然会给用户提供包括有线和无线在内的多样化技术监控手段，通过业务融合，给用户带来无所不在的监控服务。监控实现了对家庭安全的发现和预防，也是对家庭关爱的体现。因此，开发技术含量高，适合各种 IP 网络环境的多媒体视频传输系统是大势所趋[8]。

1.3 本文主要研究内容

本文首先对于嵌入式视频监控系统的总体设计方案进行分析。然后分析了嵌入式视频监控系统硬件平台和软件系统。提出一种基于 DM355 的成本相对较低的高清视频监控方案。介绍了硬件平台，及软件系统。软件设计中加入了 Log4plus 的日志控制机制，以便保障程序调试的效率和维护。

第一章是文章的前言，介绍了本课题的研究背景，国内外研究动态

第二章介绍了嵌入式监控系统的技术基础。主要包括：嵌入式技术，多媒体编码技术和网络传输技术。

第三章分析了嵌入式视频监控系统的设计。从常用的设计方案入手，分析了这些方案的优缺点，并选择了适合本课题要求的设计方案。

第四章介绍了开发的嵌入式网络监控系统的硬件平台，并介绍了其中的部分电路设计。分析了电源模块中的去耦和旁路问题。

第五章介绍了开发的应用软件平台。完成了 Linux、Uboot 的编译和烧写。

第六章介绍了开发的应用软件中线程。包括主线程，控制线程，视频压缩采集等线程。

第七章对于嵌入式网络监控系统的关键技术进行研究。在传输 MPEG4 视频时采用自适应控制方法，并且对线程间的交互做了深入分析。

第八章对整体系统的性能进行了测试，能够达到预期效果。

第九章对全文进行了总结，并对以后的研究进行了展望。

3 嵌入式视频监控系统设计分析

本章首先对嵌入式系统进行了简要的介绍，接着从嵌入式系统，嵌入式网络视频监控系统应用环境、嵌入式网络视频监控系统硬件系统技术进行分析，并根据系统实际需求作出嵌入式视频监控系统总体设计方案选择。

3.1 嵌入式系统简介

嵌入式系统近些年是很热的话题，那到底什么是嵌入式系统呢？嵌入式系统是以应用为中心，以计算机技术为基础，软硬件可裁剪，适应于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。简言之，嵌入式系统就是一个硬件和软件的集合体，硬件包括嵌入式处理器、存储器、外设器件、输入输出端口等；软件部分包括操作系统和应用程序。一般而言，嵌入式系统所需要的硬件会因用途不同而有显著的差异，相应的软件也会因为硬件的不同而不同。

嵌入式系统应具有很高的可靠性，要求在恶劣的环境或突然断电的情况下也仍然能够正常工作，同时还应具有很高的实时性，在复杂的应用中具有实时处理能力，其软件代码固化在只读存储器或闪存中，相对 PC，嵌入式系统要求质量和可靠性更高。

嵌入式系统的软件最为强调的是系统整合以及友善的用户界面。随着网络与无线通信的广泛应用，软件的组件化越来越重要。嵌入式系统设计的目的在于满足某种特殊功能，其软件需要稳定、高效且程序代码尽可能短。这里所说的软件主要指嵌入式操作系统及其应用软件。当软件功能丰富、结构复杂时，就迫切需要有一个屏蔽底层硬件、能够进行实时多任务并行工作、功能强大的操作系统来支持。与 PC 的操作系统比较，嵌入式操作系统并不要求全能，但必须能够根据系统的具体情况而有效地发挥硬件的性能。

嵌入式系统的开发包括硬件和软件方面的设计，因此开发过程可以软件和硬件并行可发，但对于开发团队的协作要求很高；也可以硬件开发完后开发软件，但影响开发进度，但团队协作要求低，对于开发周期长的产品可以考虑。所以嵌入式系统产品必须综合考虑，其开发过程一般包括需求分析、详细设计、设计实现、系统测试等阶段。

需求分析阶段是指在项目开始时了解项目的情况，明确用户需要做什么的问题。这一阶段主要包括：分析用户的需求、初步确定硬件和软件、检查需求分析

的结果、确定项目的约束条件、概要设计等几个方面。当然也可以根据系统的具体复杂情况简化步骤。

详细设计阶段是通过需求分析的结果，设计出满足用户需求的嵌入式系统产品，这一阶段主要包括：审查需求分析获得的资料、进行体系结构设计、硬件和软件功能的划分、硬件和软件的设计、检查设计等过程。

设计实现阶段是指在前面两个阶段所做工作的基础上，进行产品实现工作。这一阶段一般来主要包括：选择开发平台，如选择处理器和硬件部件进行硬件实现；选择操作系统和编程语言等进行软件开发；还有开发文档的撰写等。

测试阶段根据开发方法不同而采取不同的测试方法。一般先把硬件平台测好，可以正常工作。再把硬件与软件的联合测试，测试时一般先进行软硬件单元测试、在进行整体测试，最后还有性能等测试。

3.2 嵌入式网络视频监控系统概述

3.2.1 嵌入式视频监控系统应用场景

摄像头采集视频数据通过接口进入嵌入式处理器的内部总线，然后经过压缩处理，将图像通过嵌入式视频监控系统的以太网接口传输图像给服务器端。远程客户端可以通过 Internet 或无线网络访问视频监控服务器。系统的应用场景如图 3-1 所示。

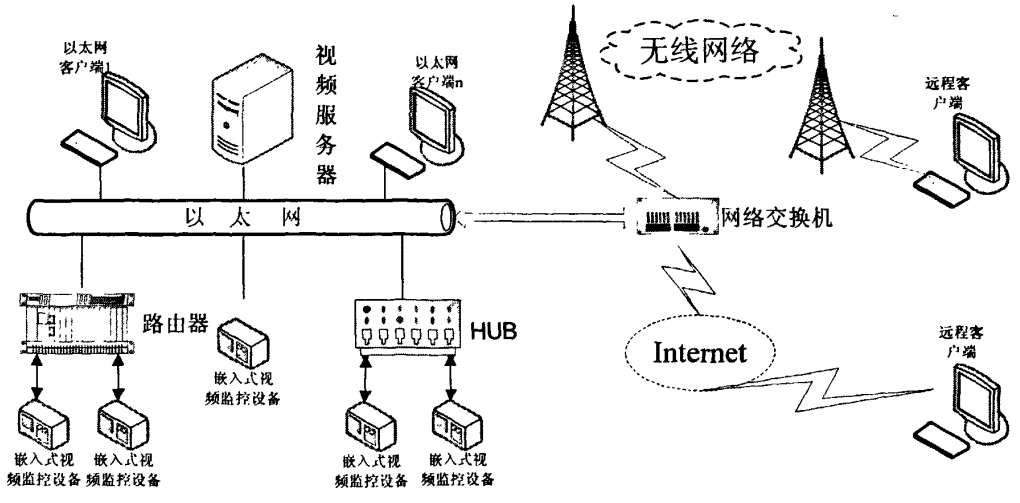

图 3-1 嵌入式视频监控系统应用场景图

Fig 3-1 application scene of the embedded video monitoring system

3.2.2 嵌入式视频监控系统需求分析

嵌入式视频监控系统需要满足以下基本功能：

1. 图像质量可定制。也就是说图像分辨率和每秒图像帧数可调。
2. 在最大分辨率情况，要求图像传输速率能够在 15 帧/秒左右。
3. 能够远程观看监控视频。客户端通过以太网或无线网络登陆视频服务器端观看视频。可以是 Web 客户端也可以是专门的客户端软件
4. 系统稳定可靠。

3.3 嵌入式视频监控系统硬件设计方案分析

3.3.1 典型嵌入式视频监控系统硬件设计方案分析

目前的嵌入式视频监控设计大多采取方案主要有以下几种：

视频采集芯片+DSP 处理器方案

该方案的实现是采用视频采集 A/D 芯片完成前端的图像基本处理，把模拟信号转化为数字信号，把变换后的视频数字信号送到 DSP 进行存储、MPEG1/2/4 或 H.264 等格式的图像压缩、网络传输等。该方案中采用的视频采集芯片可以是 TI 公司的 TVP5150、Philips 公司的 SAA7111A/7113/7114A/7115 系列等。DSP 处理器可采用 TI 公司的 TMS320DM642，ADI 公司的 Blackfin 系列。文献[31]提出了基于 TMS320DM642 的嵌入式网络视频服务器的方案。这种方案的典型的原理框图如图 2-2 所示。

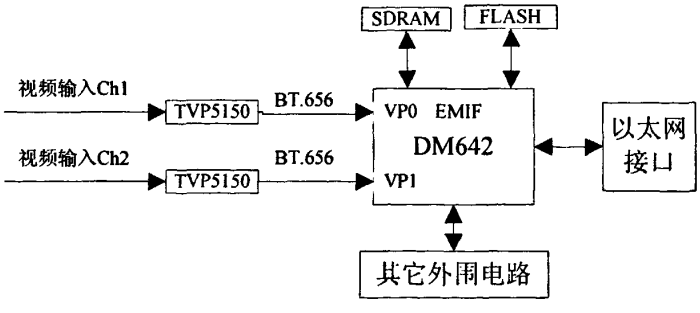

图 3-2 第一种方案的典型原理框图

Fig 3-2 the first typical diagram

视频采集芯片+DSP 处理器+嵌入式处理器

该方案与上一种方案相比，增加了一个嵌入式处理器。同样使用视频采集芯片来完成图像的数字化处理。嵌入式处理器用来控制整个系统，包括视频采集芯

片、DSP 处理器及一些外围设备，DSP 处理器用来实现各种格式的图像压缩（软件压缩）。此方案中，嵌入式处理器一般选择 32 位的处理器，比如 Xscale、ARM、MIPS 处理器等。DSP 处理器则可以与上一种方案相同，也可选择比上一种方案功能稍弱一点的芯片。文献[32] 提出一种采用 Analog Devices 公司的 BF533 芯片实现 MPEG4 标准的图像压缩，以 SUMSUNG 公司的 S3C2410 作为控制的嵌入式视频监控系统。该方案的典型原理框图如图 3-3 所示：

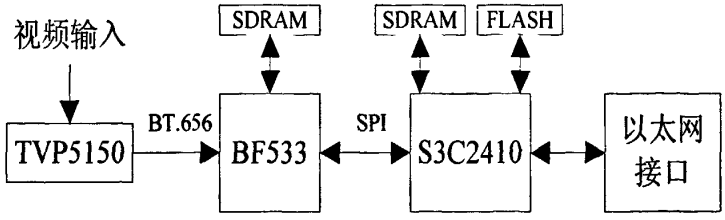

图 3-3 第二种方案的典型原理框图

Fig 3-3 the second typical diagram

视频采集芯片+视频压缩芯片+嵌入式处理器方案

此方案在前端使用一个视频采集芯片，经过变换后的视频数字信号由视频压缩 ASIC 芯片负责图像的编码压缩（硬件压缩），编码方案有 H.264、MPEG4 等，嵌入式处理器负责图像的存储、传输等系统控制[33]。该方案中常用的 MPEG4 码芯片有 INTIME 公司 IME6400、PentaMicro 公司的 AT2021/AT2041/AT2042/AT2043 等。嵌入式处理器同前面方案一样一般也采用 32 位的处理器。这种方案的典型原理框图如图 2-4 所示。

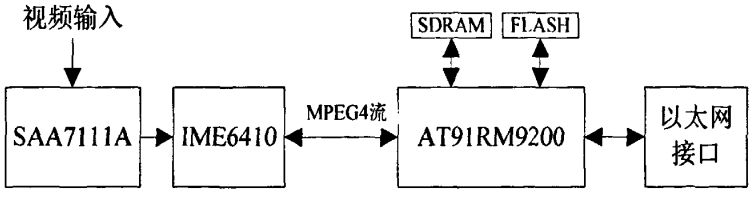

图 3-4 第三种方案的典型原理框图

Fig 3-4 the third typical diagram

USB 摄像头+嵌入式处理器的方案

文献[34]所提到的方案适合要求低成本，对图像质量要求不是很高的应用场景，由于摄像头集成了数据采集压缩，尽管图像质量没有前面三种的图像质量高，可是省去了前面的视频采集芯片和视频压缩所需的硬件资源，大幅度的降低了成本，而且使用方便。这种方案的一种典型原理框图如图 3-5 所示。

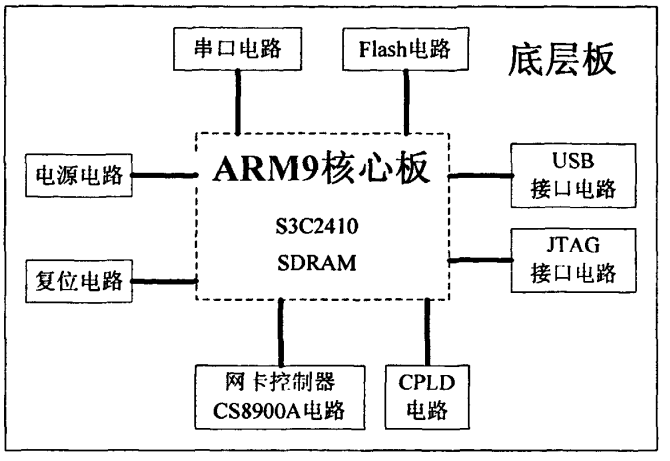

图 3-5 第四种方案的典型原理框图

Fig 3-5 the fourth typical diagram

从技术分析角度，前面两种方案都采用了 DSP 处理器，第二种方案增加了一个嵌入式处理器，可以在上面运行操作系统，增加了系统的灵活性。DSP 处理器对视频进行软件压缩，算法灵活性好，画面质量优异，而且系统升级维护比较容易，但存在系统开发周期长，开发难度大，成本高等缺点。第三种方案采用视频压缩芯片来实现硬件视频压缩，开发难度较前两种方案小，成本较前两种方案低，缺点是算法灵活性小，画面质量一般，系统升级困难。第四种方案采用了 USB 摄像头+嵌入式处理期的方案控制了成本，但仅适用于对图像质量要求不高的应用场景。

3.3.2 嵌入式视频监控系统硬件总体方案设计

前面三种嵌入式视频监控方案成本相对较高，是因为 DSP 处理器和视频压缩芯片的价格比较高，而且有些芯片还会对每件产品收取版权费用。第四种虽然控制了成本，但仅能应用于对图像质量要求不高的场景。所以针对我们的需求分析，既要控制成本，又对于图像质量要求不能太差，本系统的设计方案决定采用的视频采集芯片+嵌入式处理器+视频协处理器也就是视频采集芯片+DM355 芯片的方案，由于 DM355 集成了 MPEG4 协处理器，所以省去了成本高的 DSP，降低了成本，而且使用软件编程可以控制图像的输出大小，使用方便。系统的硬件总体设计方案如图 3-6 所示。

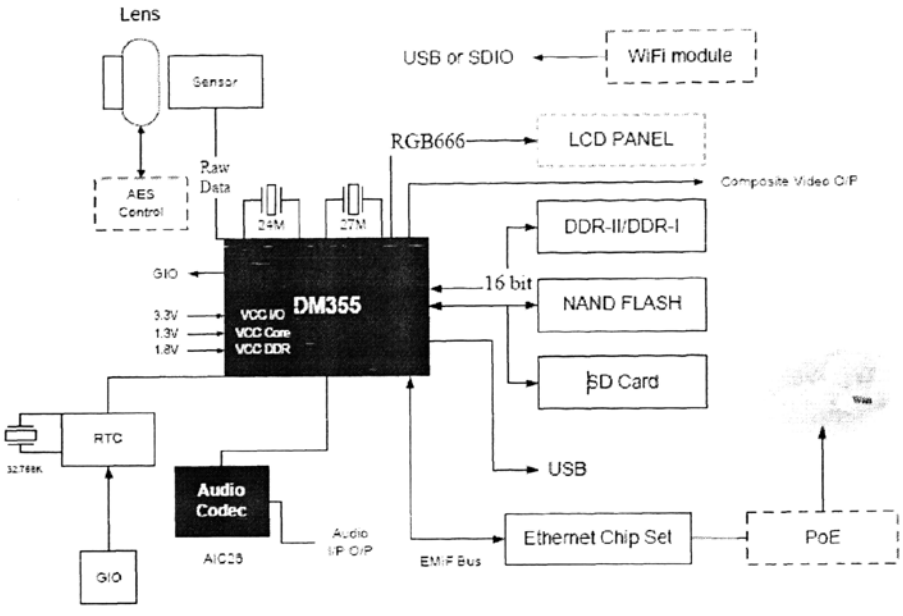

图 3-6 嵌入式视频监控系统硬件设计原理图

Fig 3-6 the hardware design of embedded video monitoring system

3.4 嵌入式视频监控系统软件方案设计

3.4.1 嵌入式视频监控系统软件总体方案设计

嵌入式监控系统的软件方案大体是由服务器端软件和客户端软件组成。服务器端软件一般有两中方案，一种是在 Windows PC 下运行的服务器端软件，一种是在终端 Linux 系统下运行的服务器端软件。本系统采用的是 PC 作为视频服务器[35]。

嵌入式视频监控系统要实现的功能较多，首先要实现摄像头的视频采集处理，所以要求操作系统支持常用摄像头驱动；其次设备需要接入网络，所以操作系统必须很好的支持 TCP/IP 等网络协议；再者视频传输要保证实时性，所以对于操作系统要选取实时操作系统，而且软件的编写要充分考虑线程间的通信和编码效率。相当关键的一点是需要整体开发成本。根据这些需求且根据前面各种嵌入式操作系统的比较综合考虑，本系统选择嵌入式 Linux 操作系统。

确定嵌入式 Linux 操作系统后，我们的主要工作是搭建嵌入式软件开发环境，和内核的裁剪烧制等工作。为下一步视频采集，压缩和发送做好准备，最后是基于 PC 的视频客户端软件的设计[36]。嵌入式视频监控系统软件结构如图 2-6 所示。

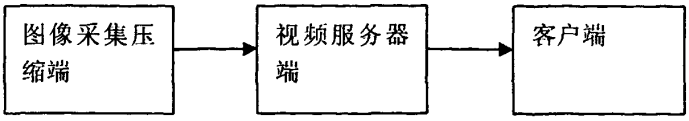

图 3-7 嵌入式视频监控系统软件结构框图

Fig 3-7 the software design of embedded video monitoring system

3.5 本章小结

本章首先介绍了嵌入式系统和嵌入式监控系统的应用场景案，其次分析现阶段常用的嵌入式视频监控系统硬件方案和软件系统方案，由于前三中方案成本过高，第四种方案虽然控制了成本，但图像清晰度不够，满足不了现在监控系统对于图像的要求所以提出了一种新的嵌入式视频系统解决方案。

4 嵌入式视频监控系统硬件设计

嵌入式系统的硬件平台是嵌入式系统开发的基础，一切应用程序的开发都是基于硬件平台的。因此，硬件平台的搭建在嵌入式系统的开发中起着至关重要的作用。

4.1 硬件总体框图

通过上一章的分析，本系统采用了如下的硬件结构：

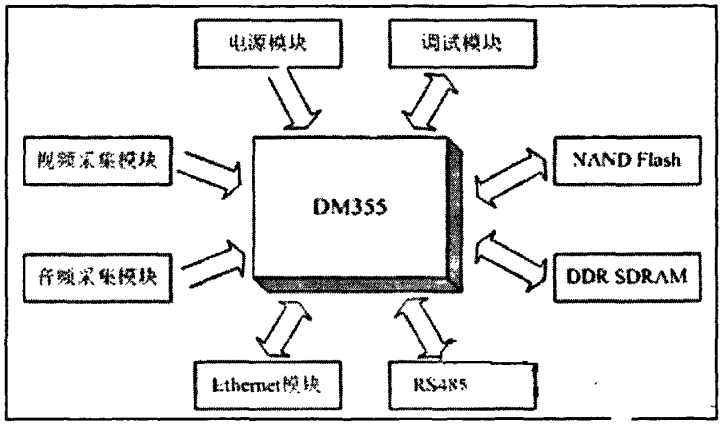

图 4-1 嵌入式视频监控系统硬件件结构框图

Fig 4-1 the hardware design of embedded video monitoring system

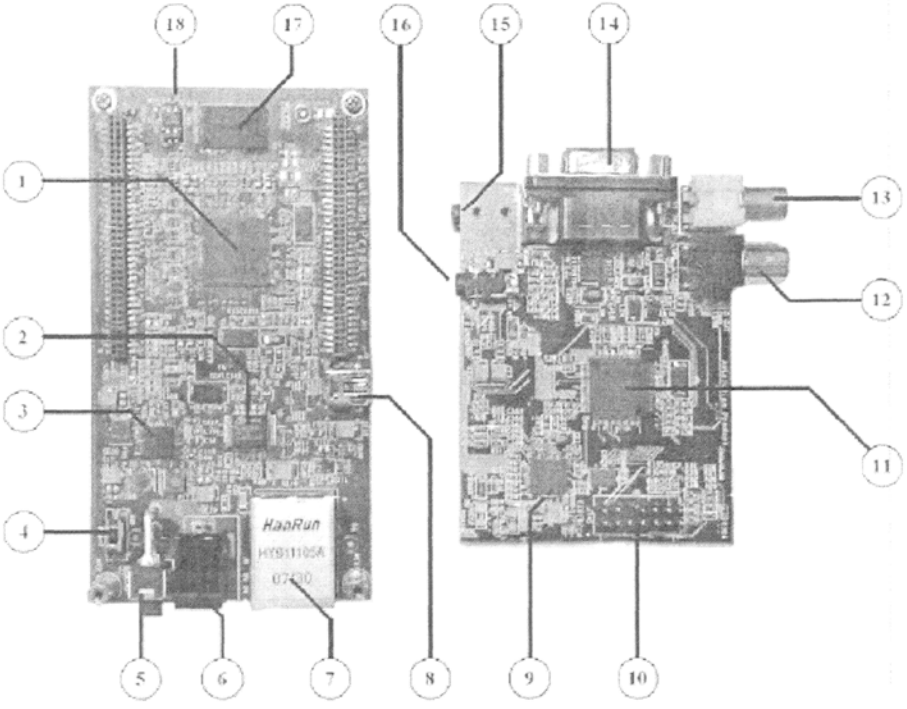

图 4-2 主板和子板的俯视图

Fig 4-2 the planform of mainboard and childboard

本系统的硬件部分采用了核心板加底层板的设计方式。

表 4-1 板卡主要部件名称表

Table4-1 the name of the main board components

1	TMS320DM355 处理器，270 主频
2	DM9000A 网络芯片，10/100M 自适应
3	电源芯片
4	复位按钮
5	电源开关
6	电源插座
7	RJ-45 以太网插槽
8	USB miniAB 接口
9	AIC33 音频接口芯片
10	JTAG 接口
11	TVP5146 视频接口芯片
12	RCA 复合视频输出
13	RCA 复合视频输入

14	DB-9 串口
15	麦克风输入
16	耳机输出
17	DDR2-533 颗粒（128Mbyte）
18	启动没事配置拨码开关
19	Nand 芯片(512Mbyte)，在主板背面
20	MicroSD 卡插槽，在主板背面

4.2 部分功能模块介绍

4.2.1 TMS320DM355 处理器

2007 年 9 月份左右的时候，德州仪器（TI）推出针对便携式高清(HD) 视频产品市场的最新达芬奇 (DaVinci) 处理器——TMS320DM355，其具备 ARM 主机控制与全套开发工具。该产品不仅能够实现高清视频（MPEG-4-JPEG）性能，而且其电池使用寿命也是当前其它高清产品的两倍[37]。

DM355 处理器由集成的视频处理子系统、MPEG-4-JPEG 协处理器(MJCP)、ARM926EJ-S 内核以及多种外设组成，针对数码相机、IP 摄像机、数码相框以及婴儿视频监护器等应用。

该处理器可提供 216MHz 或 270MHz 的时钟速率，因而能够实现可扩展的产品系列。开发人员能够重复利用强大稳建的达芬奇技术系列 IP ，或利用 ARM 处理器丰富的开放源代码资源加速开发。此外，DM355 还可应用在如医学成像、超低成本数码摄像机以及便携式测试设备等大众化产品。

通过集成视频/影像协处理器，针对高清视频精心优化的 DM355 不仅价格便宜，同时还实现极低功耗，并获得极高性能。该 MJCP 能够以 720p 格式与每秒 30 帧的速度提供高清 MPEG-4 SP 编解码功能，以及每秒 5 千万像素的速度提供 JPEG 编解码功能。所有达芬奇器件均集成了视频处理子系统，并在子系统的硬件中高度集成了预览引擎、柱状图(histogram)、图像缩放工具以及屏幕视控系统。该 MJCP 提供了相当于 400MHz 的 DSP 来实现高清视频，同时视频处理子系统执行的任务也等同于 DSP 约 240MHz 的性能。这样，MJCP 与视频处理子系统结合起来就能提供相当于 640MHz 的 DSP 处理性能，高达 270MHz 的 ARM 处理能力仍可实现产品差异化。此外，DM355 还包含一套精心挑选的外设，如高速 USB OTG 2.0。集成型 10 位数模转换器与视频编码器还能为开发人员节约多达 2 美元的材料成本以及使用分立部件所需的相关制造与设计成本。尽管 DM355 处理器的价格低廉，但其实际拥有的价值远远超过售价，因为该产品包含符合生产要求

的高请 MPEG-4 与 JPEG 编解码器，且无需向 TI 支付许可费或版税。

采用 DM355 构建的系统将拥有超长电池使用寿命，相当于现有便携式高清系统寿命的两倍。根据应用的不同，DM355 在高清 MPEG-4 编码过程中的功耗约为 400mW，而待机功耗仅为 1mW。举例来说，这意味着消费者在视频模式下使用基于 DM355 的数码相机时，只需用两节 AA 电池即可录制 80 分钟的高清视频。

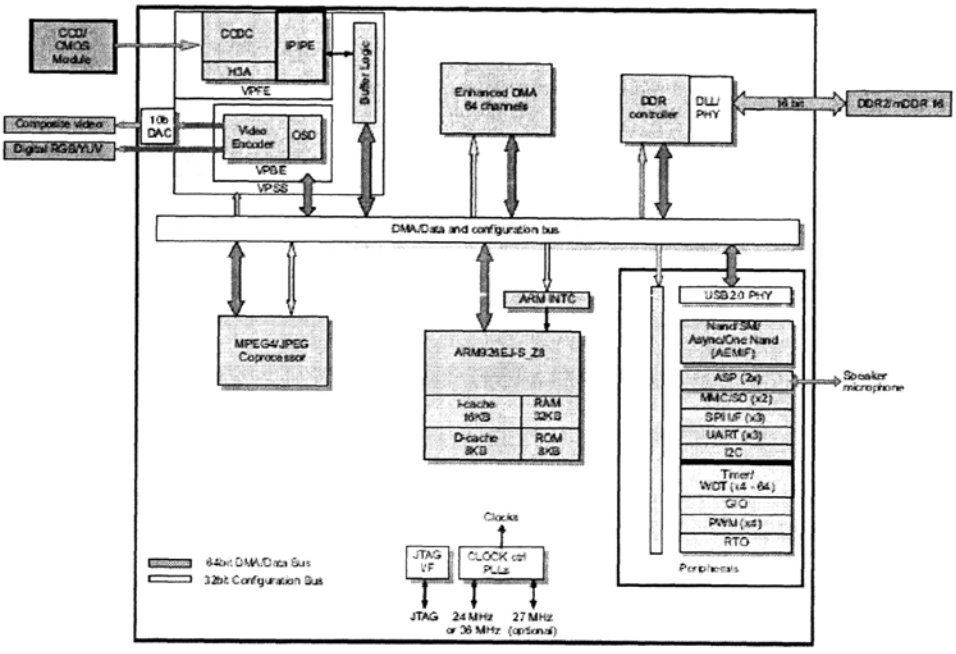

图4-3 TMS320DM355处理器功能模块图

Fig.4-3 the function module of the TMS320DM355 processor

4.2.2 电源管理模块

电源设计在嵌入式系统中有着非常重要的位置[37]。因为他直接关系到系统的能量提供。在 DM355 系统中采用电源和休眠控制器 PSC 来控制电源和时钟的开/关或重启。PSC 的结构图如所示。

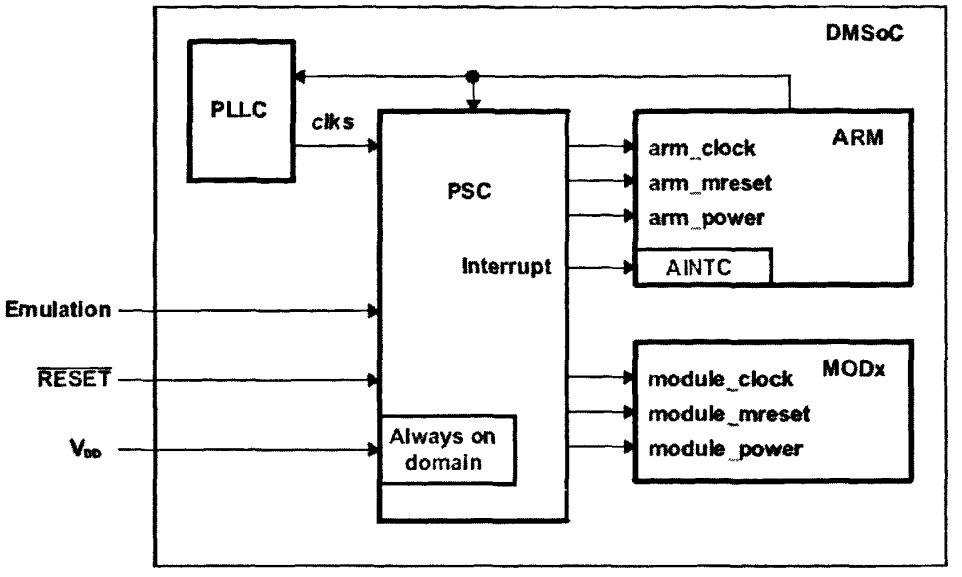

图4-4 DM355电源和休眠控制模块

Fig.4-4 DM355 Power and Sleep Controller (PSC)

为了确保机械特性的稳定性，通过查阅芯片手册，DM355 需要下面所列的电源供电。

Power-On:

1. Power on 1.3V: CV_{DD}， $V_{DDA\_PLL1/2}$，V_{DDD13_USB}， V_{DDA13_USB}

2. Power on 1.8V: V_{DD_DDR}， V_{DDA18_DAC}

3. Power on 3.3V: D_{VDD}， V_{DDA33_DDRDLL}， V_{DDA33_USB}， $V_{DDA33\_USB\_PLL}$，V_{DD_VIN}， V_{DD_VOUT}

Power-Off:

1. Power off 3.3 V: D_{VDD}， V_{DDA33_DDRDLL}， V_{DDA33_USB}， $V_{DDA33\_USB\_PLL}$，V_{DD_VIN}， V_{DD_VOUT}

2. Power off 1.8 V: V_{DD_DDR}， V_{DDA18_DAC}

3. Power off 1.3 V: C_{VDD}， $V_{DDA\_PLL1/2}$， V_{DDD13_USB}， V_{DDA13_USB}

为了使能量在传递时尽可能的降低由于自感应和电抗所引起的损耗，所以再设计电源电路时，核心器件和 I/O 的供电应该尽可能的靠近 DM355。另外，为了获取 DM355 更高性能的应用，核心版分层给核心器件，I/O 和地供电。

本系统设计中大量用到了电容的去耦和旁路，去耦和旁路可以防止能量从一个电路传到另一个电路，进而提供配电系统的质量。去耦是克服由数字电路切换逻辑引起的物理上和时间上限制的手段，旁路则是提供一个通道来转移不想要的

能量。而且尽可能的靠近 DM355 的电源引脚，最大的有效距离是 1.25cm。所使用的最小的电容是 D402，因为它有较好的性能。对于电容容量的选取也是十分重要的。小的旁路电容（大约 560pF）被放置在最靠近电源引脚的位置。中等的旁路电容放置（220 nF）在第二靠近电源引脚的位置。为了大量的去耦更大些的电容放置在更远的位置。大约在 100uF 的大电容放置在最远的位置，但是仍要尽可能的靠近电源引脚。

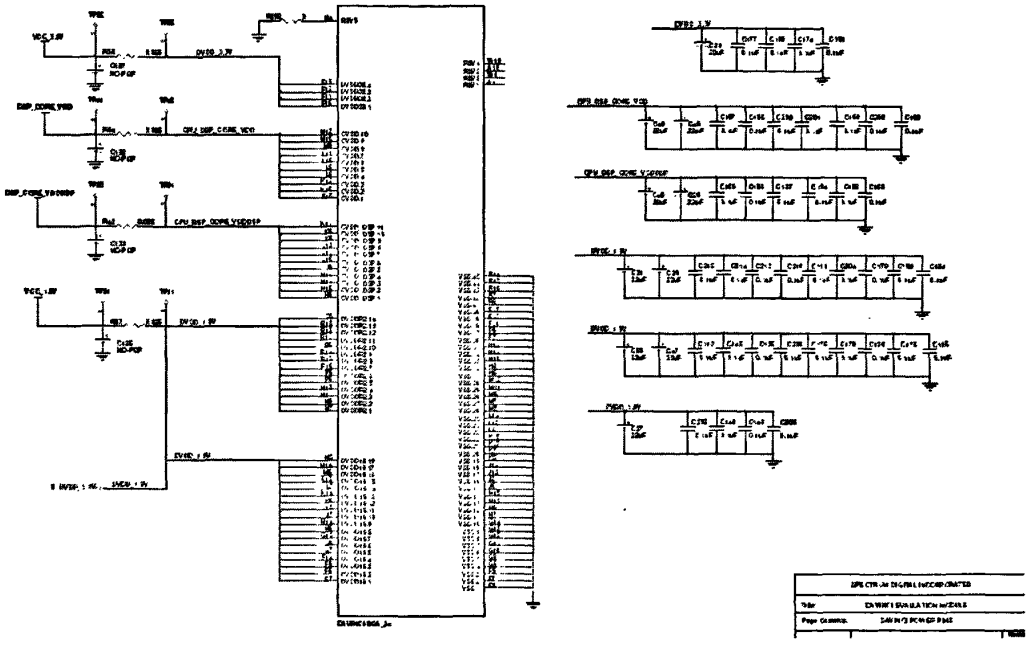

图4-4 电源电路

Fig.4-4 Power circuit

4.2.3 NAND/SRAM flash 模块

本文采用了 Samsung 的 K9K1208Q0C Nand Flash，片内寻址采用 26 位地址形式。对 NAND 存储芯片进行操作，必须通过设置 NAND 专用控制寄存器(NANDFCR)及状态寄存器(NANDFSR)才能完成。所以，不能对 NAND Flash 进行总线操作。在对 NAND 的基本原理、存储结构及操作方法等了解以后，在 Windows 环境下借助 CCS3.3 集成开发环境及 XDS560 仿真器实现了将在 DDR2 中的一段存储区中的数据(如 Bootloader，ulmage)写到 NAND Flash 存储空间中，

并能够被 UBL 读到 DDR2 中运行。

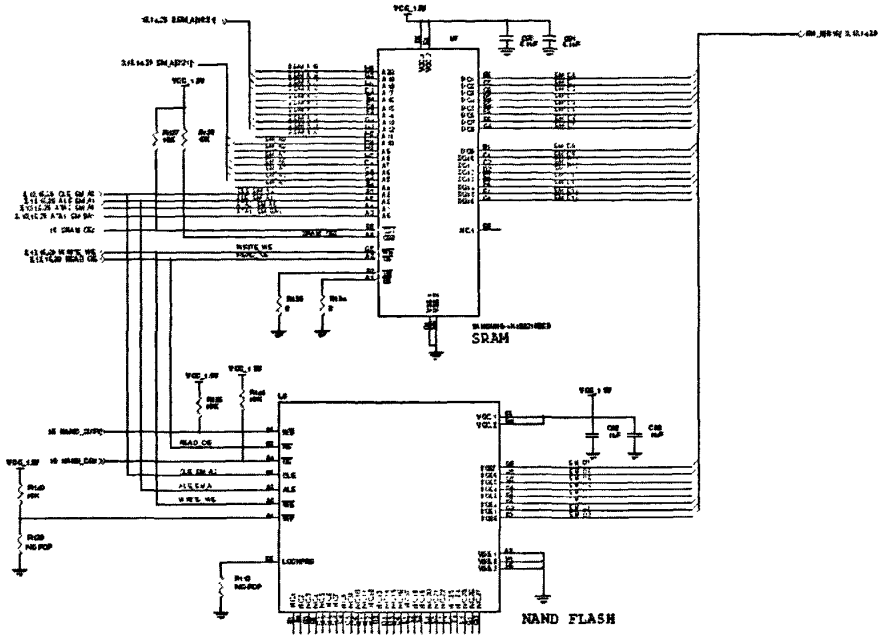

图4-5 NAND FLASH/SRAM 电路

Fig.4-5 NAND FLASH/SRAM circuit

4.2.4 底层板 JTAG 调试模块

JTAG 嵌入式调试技术在嵌入式系统中被广泛的应用，它在芯片内部封装了专门的测试电路 TAP (Test Acess Port，测试访问端口)，通过专用的 JTAG 测试工具对内部节点进行测试。JTAG (Joint Test Action Group，联合测试行动小组) 是 1995 年制定的国际标准测试协议，主要用于检测 PCB 和 IC 芯片。本系统的 JTAG 电路设计如图所示

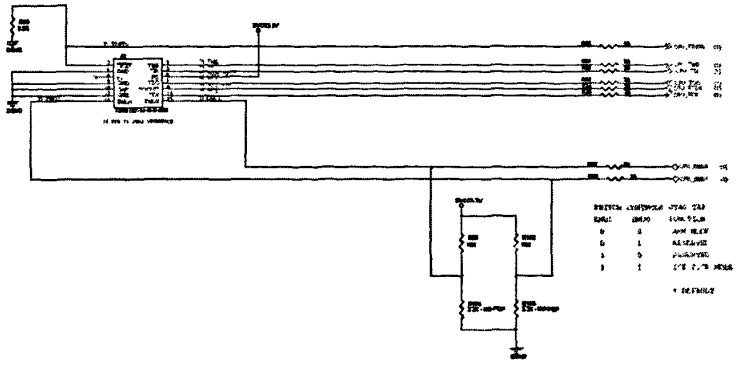

图 4-6 JTAG接口电路

Fig 4-6 JTAG interface circuit

4.2.5 底层板模拟视频输入/输出模块

本系统采用 TVP5146 视频接口芯片，此芯片是一种高质量的数字视频解码器。它支持 RGB 和 YpbPr 信号的模拟信号到数字信号的转换，也支持 NTSC，PAL 和复合视频信号的解码。TVP5146 包括四个 10bit 30-MSPs A/D 转换器。它支持的输出格式有：20-bit 4:2:2 YcbCr or 10-bit 4:2:2 YcbCr。

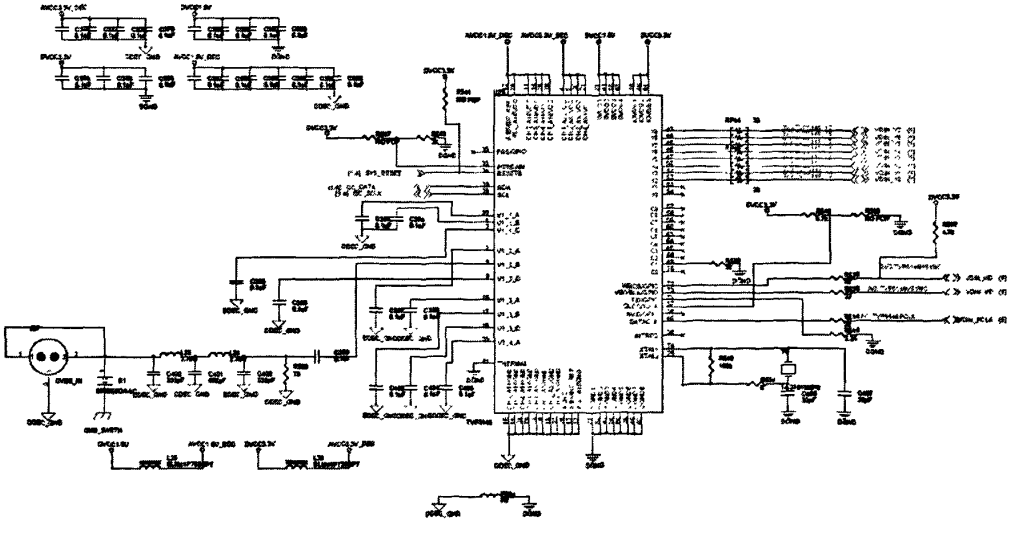

图 4-7 视频模拟输入电路

Fig 4-7 video analog input circuit

并能够被 UBL 读到 DDR2 中运行。

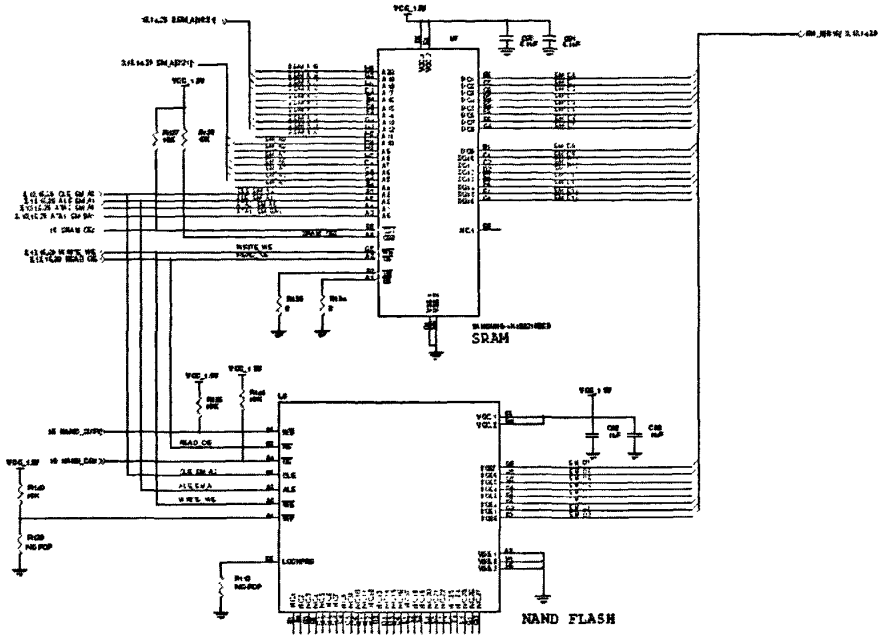

图4-5 NAND FLASH/SRAM 电路

Fig.4-5 NAND FLASH/SRAM circuit

4.2.4 底层板 JTAG 调试模块

JTAG 嵌入式调试技术在嵌入式系统中被广泛的应用，它在芯片内部封装了专门的测试电路 TAP (Test Acess Port，测试访问端口)，通过专用的 JTAG 测试工具对内部节点进行测试。JTAG (Joint Test Action Group，联合测试行动小组) 是 1995 年制定的国际标准测试协议，主要用于检测 PCB 和 IC 芯片。本系统的 JTAG 电路设计如图所示

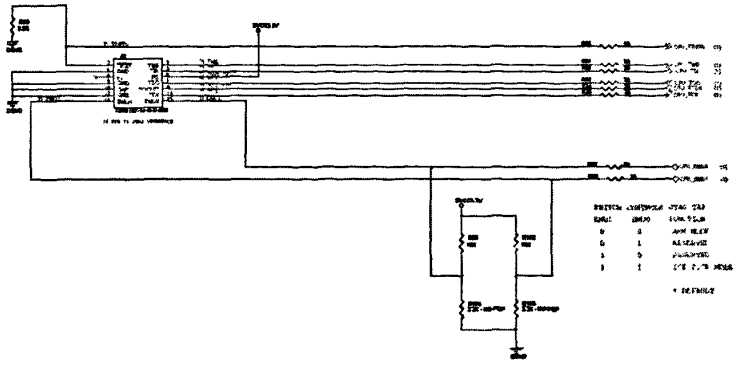

图 4-6 JTAG接口电路

Fig 4-6 JTAG interface circuit

4.2.5 底层板模拟视频输入/输出模块

本系统采用 TVP5146 视频接口芯片，此芯片是一种高质量的数字视频解码器。它支持 RGB 和 YpbPr 信号的模拟信号到数字信号的转换，也支持 NTSC，PAL 和复合视频信号的解码。TVP5146 包括四个 10bit 30-MSPs A/D 转换器。它支持的输出格式有：20-bit 4:2:2 YcbCr or 10-bit 4:2:2 YcbCr。

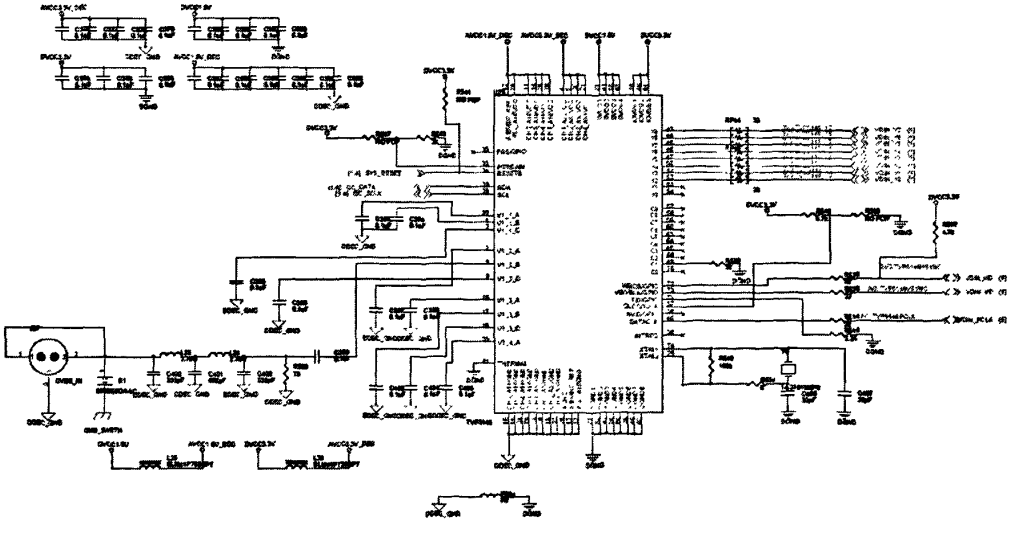

图 4-7 视频模拟输入电路

Fig 4-7 video analog input circuit

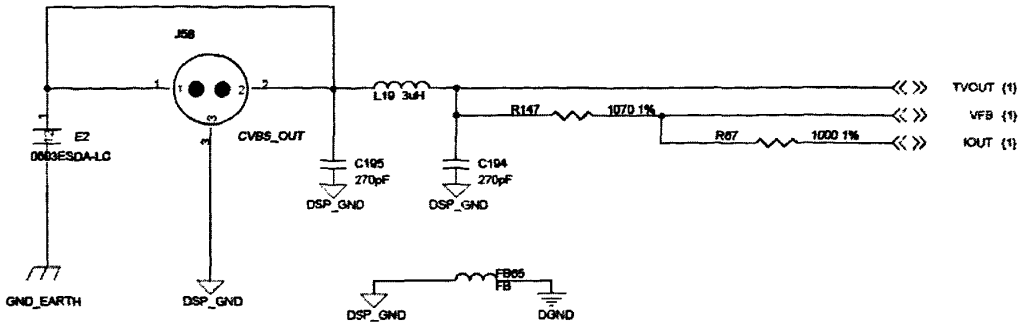

图 4-7 视频模拟输出电路

Fig 4-7 video analog output circuit

4.2.6 底层板音频处理模块

本系统使用的音频接口芯片是德州仪器的 TLV320AIC33。它是一款低噪声的立体声音频解码器。采用集成音响处理技术，从而使小型扬声器也能实现出色的音效。低功耗的编码器能够进一步延长电池的使用效率。而且 AIC33 基于寄存器实现广泛的功率控制，使用 3.3V 的模拟电源供电既能实现 448kHz 的立体声回放，且功耗仅为 14mW。此外，其引脚实现多路复用功能。由于消除了外部晶振，因而进一步降低了成本并简化了设计。

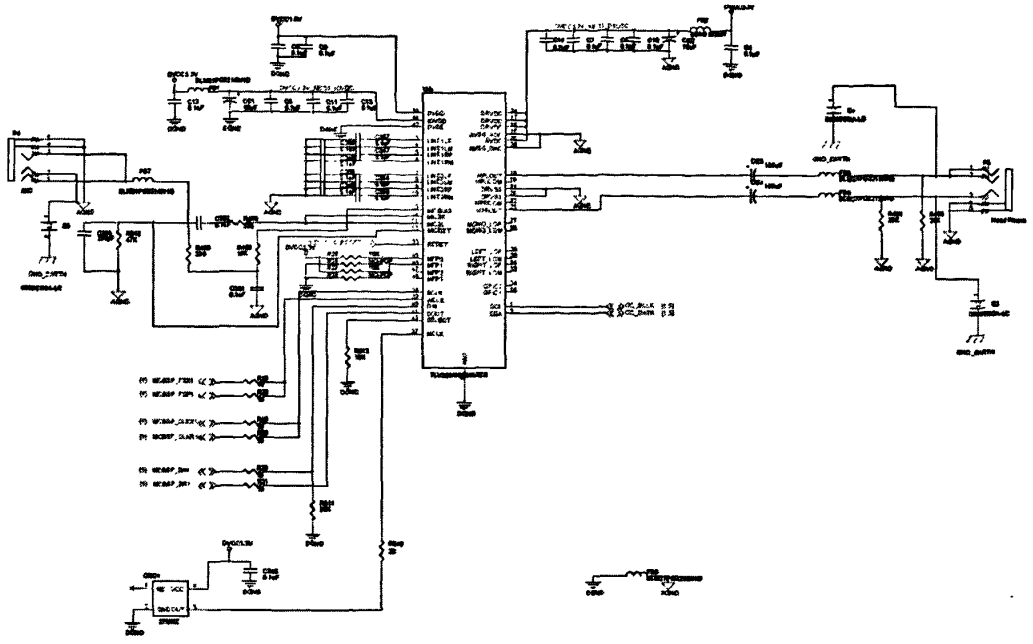

图 4-7 音频处理电路

Fig 4-7 audio disposal circuit

4.3 本章小结

硬件平台是软件开发的基础，本章首先介绍了本系统的硬件的结构，并且分析了核心部件的性能，介绍了硬件系统中的部分模块设计。包括电源模块，NAND/SRAM 模块和底板的调试模块，模拟输入输出模块，音频处理模块。

5 嵌入式视频监控系统软件平台

嵌入式系统中硬件平台搭建好只是完成了开发的第一步，如果要在此硬件平台开发产品必须要搭建软件平台，也就是最接近硬件的程序。要求软件平台必须和硬件平台相适应。下面将介绍软件平台的搭建过程。

5.1 烧写 NAND flash

步骤如下：

1. 用 CCS3.3 和仿真器连接 TechvDM355 板卡，上电并下载 Nand_programmer.out 文件，点击运行。
2. 期间会要求用户输入 ublDM355-nand.bin 的文件路径，比如 D:\dm355\ublDM355-nand.bin;
3. 等待烧写，然后接着输入 ubootfixip.bin 文件，要全路径;
4. 看 CCS 的输出窗口，完成后会有提示，形如：

```
Enter the UBOOT File Name
D:\DM355Projects\board_utilities\u-boot-1.2.0-dm355_evm_new.bin
U-Boot: Number of pages: 63
U-Boot Descriptor copy starts now
BlockNum: 8
NANDWriteBlock done on block 8
UBOOT descriptor is successfully programmed,  BlockNum: 8
UBOOT programming starts now pages 63,  blockNum 9
UBOOT programming is completed
NAND flash programming is completed
```

断掉 CCS 的连接，按下板卡的复位键，即可看到串口有信息输出。第一次上电 Uboot 会检测 Flash 信息，所以请稍等 10 秒。

5.2 烧写 UBOOT

此部分与烧写 Linux 内核非常相似，都是用的 Uboot 命令。

1. 编译修改的 Uboot，得到 u-boot-1.2.0-dm355_evm.bin 文件，把它拷贝到/tftpboot

目录。请确保上位机的 tftp 服务已经开启！

2. 把上述的 bin 文件下到 DDR 内存中，在 Windows 终端敲入如下命令：

```
tftp 80800000 u-boot-1.2.0-dm355_evm.bin
```

稍等即可看到：

```
DM355 EVM # tftp 80800000 u-boot-1.2.0-dm355_evm.bin
TFTP from server 192.168.1.147; our IP address is 192.168.1.222
Filename 'u-boot-1.2.0-dm355_evm.bin'.
Load address: 0x80800000
Loading: T ##############################
done
Bytes transferred = 128720 (1f6d0 hex)
DM355 EVM #
```

已连接 0:38:4 | VT52 | 115200 8-N-1 | SCROLL | CAPS | NUM | 捕 | 打印

图 5-1 U-BOOT 烧写成功

Fig 5-1 U-BOOT programmer

可见，下载完毕，共 128720 个字节！

3. 继续在终端输入：

```
nand erase 140000 20000
```

等待 Flash 擦除完毕。

4. 最后在终端输入烧写命令：

```
nand write 80800000 140000 20000
```

5. 按下板卡复位键，即可看到效果！

5.3 烧写 Linux 内核

1. 把新内核下载到Flash中。我们通过Uboot的TFTP功能来下载编译好的Linux内核。/tftpboot 目录是 Linux 自己创建的，目的是给 TFTP 服务用。烧写的基本原理是，Uboot 通过网线连接到 Linux 的 TFTP 服务器，然后读取给定的文件名（uImage-dm355）到 DDR 中，最后通过 Uboot 命令“nand”把 DDR 的数据烧写到 Flash。

2. 首先确认 Linux 的 TFTP 服务是否开通，按照如下步骤进行：

点击“服务”（Services），出来对话框，查看 TFTP 即可，没有运行就勾上让其运行.

3. 打开 Windows 超级终端，给板卡上电，切换到 Uboot 状态，输入如下命令：

```
tftp 80700000 uImage-dm355
```

这句话的意思就是，Uboot 把/tftpboot 目录中的 uImage-dm355 文件下载到 0x80700000 的 DDR 内存中。

4. 然后就出来如下信息：

```
DM355 EVM # tftp 80700000 uImage-dm355
TFTP from server 192.168.1.147; our IP address is 192.168.1.222
Filename 'uImage-dm355'.
Load address: 0x80700000
Loading: checksum bad
checksum bad
T ################################################################################
         ################################################################################
         ################################################################################
         ################################################################################
         ##########################################################
done
Bytes transferred = 1585316 (1830a4 hex)
DM355 EVM # _
```

图 5-2 Linux 下载成功

Fig.5-2 downloads Linux

说明已经下载完毕！

5. 接着在终端输入如下命令：

```
nand erase 400000 200000
```

这句话的意思是，擦除 NANDFlash 偏移地址为 0x400000 的区域，并且长度为 0x200000。

6. 等待片刻即可看到超级终端返回的信息：

7. 最后输入如下命令：

```
nand write 0x80700000 0x400000 0x200000
```

这句话的意思是，把地址为 0x80700000 的 DDR 中的数据（刚才通过 TFTP 下的 Linux 内核），烧写到偏移地址为 0x400000 的 NANDFlash 中，长度为

0x200000。

8. 这时在 uboot 下直接输入“boot”命令，即可进入 Linux 环境中！。

5.4 编译新的 Linux 内核

1. 以 userfordm355 用户登陆，在用户目录的 workdir 下建立文件夹，名为 lsp，即：**/home/userfordm355/workdir/lsp**
2. 把目录 /opt/mv_pro_4.0.1/montavista/pro/devkit/lsp/ti-davinci 拷贝到第一步所建目录，可以用直观的方法操作，也可以用命令行：

```
[userfordm355@localhost ~]$ cd~
[userfordm355@localhost ~]$ mkdir –p workdir/lsp
[userfordm355@localhost ~]$ cp –R /opt/mv_pro_4.0.1/montavista/pro/devkit/lsp/ti-davinci .
```

3. 切换到 ti-davinci 目录，并打入如下命令进行默认的内核配置：

```
[userfordm355@localhost ti-davinci]$ ls
arch                          fs       localversion  REPORTING-BUGS
COPYING                       include  MAINTAINERS   scripts
CREDITS                       init     Makefile      security
crypto                        ipc      mm            sound
DEV_LSP_01_20_00_004_1.txt    kernel   mvl_patches   System.map
Documentation                 ktools   net           usr
drivers                       lib      README        vmlinux
[userfordm355@localhost ti-davinci]$ make ARCH=arm CROSS_COMPILE=arm_v
5t_le- davinci_dm355_evm_defconfig
Shell
```

图 5-3 默认内核配置命令

Fig.5-3 default kernel config command

稍等出来如下界面：

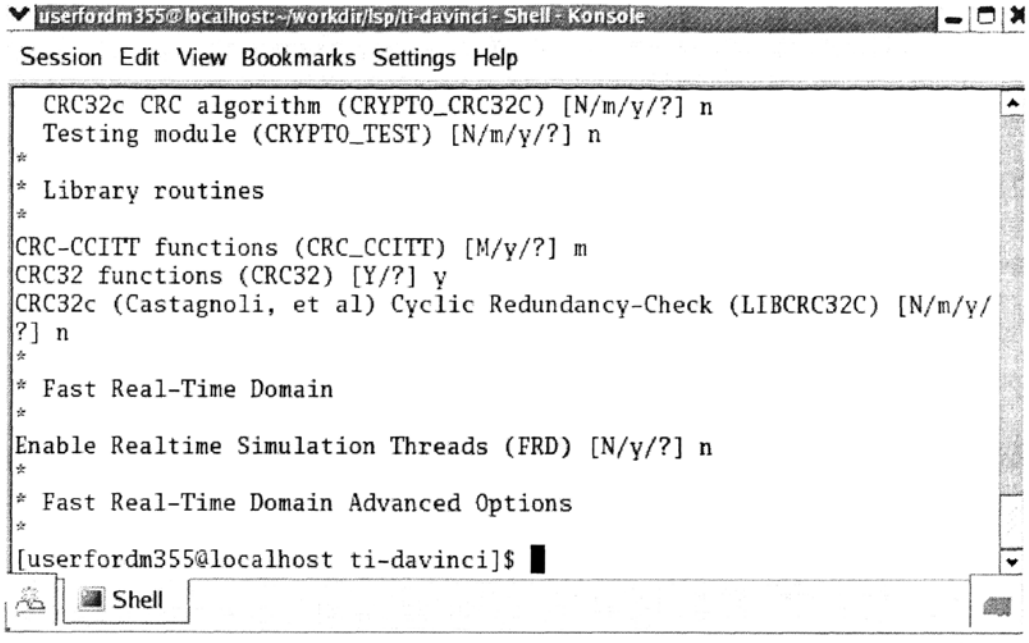

图 5-4 配置界面

Fig.5-4 default config

如果需要修改内核，请在终端输入 make menuconfig，会出来如下文本配置界面，

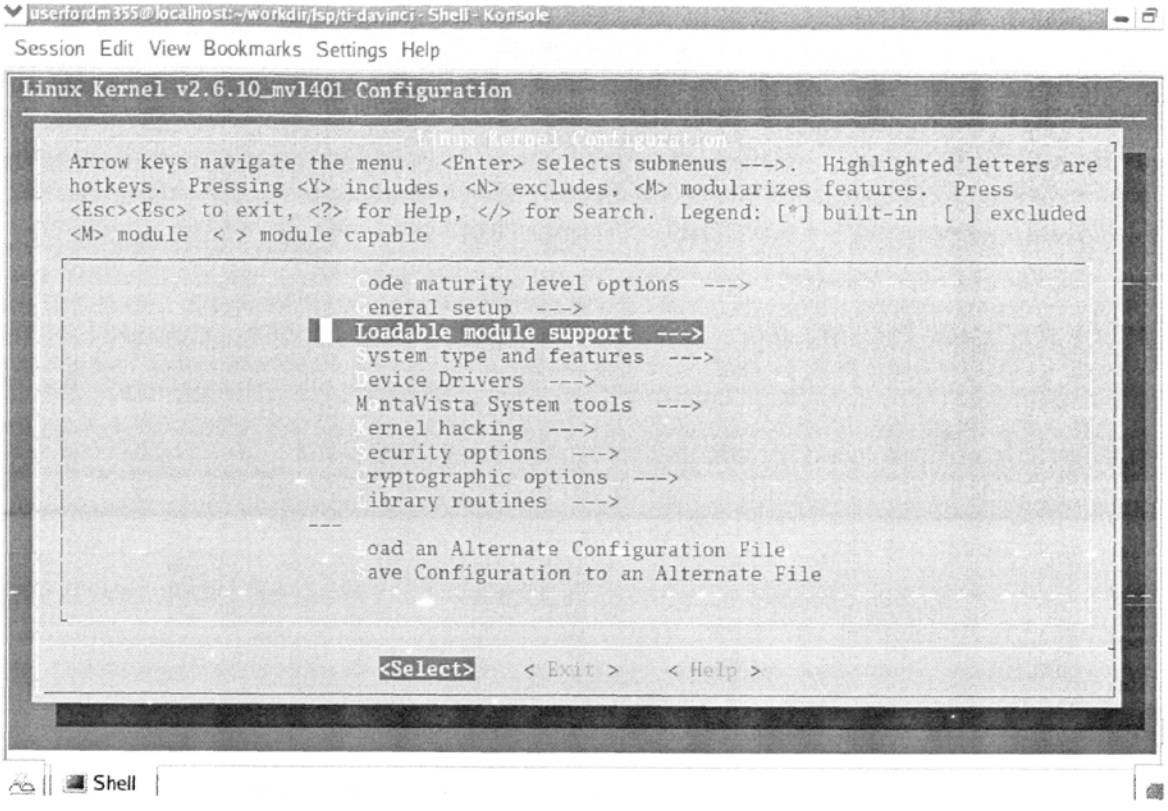

图 5-5 内核配置界面

Fig.5-5 kernel config

退出上图所示界面，按“Tab”键选择<Exit>，然后回车；

我们使用默认的内核设置，需要输入如下命令：

```
[userfordm355@localhost ti-davinci]$ make ARCH=arm CROSS_COMPILE=arm_v5t_le-checksetconfig
```

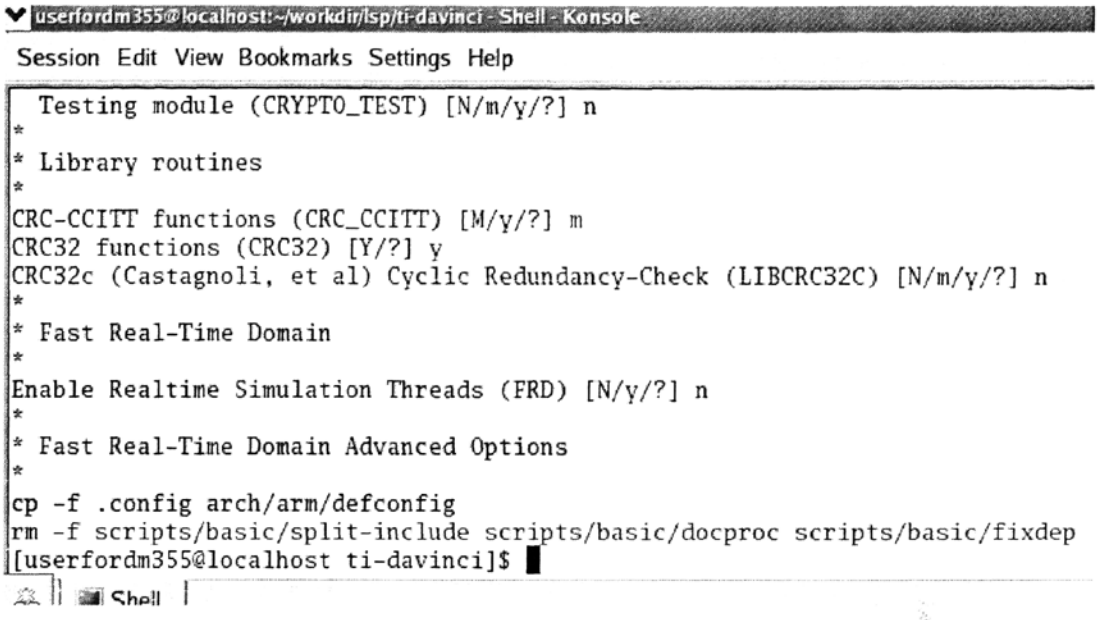

图 5-6 内核配置完成界面

Fig.5-6 complete the kernel config

4. 至此，编译内核的准备工作已经完毕，执行如下命令便可以开始编译内核：

```
[userfordm355@localhost ti-davinci]$ make ARCH=arm CROSS_COMPILE=arm_v5t_le-uImage
```

稍等（视电脑速度，有时候需要多达 30 分钟）即可出来如下界面：

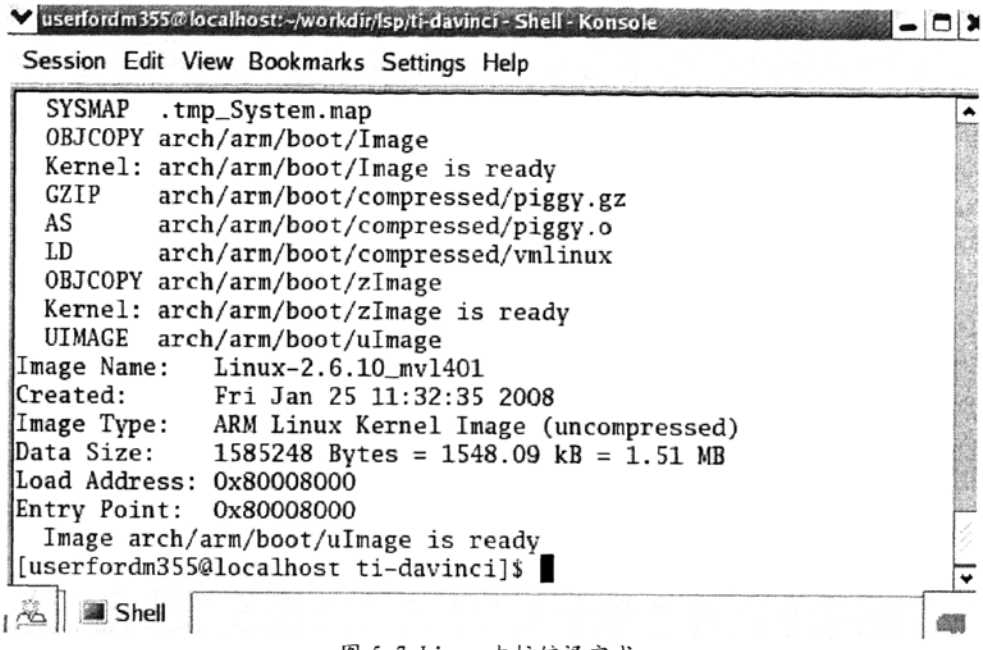

图 5-7 Linux 内核编译完成

Fig.5-7 Linux kernel compile complete

可见，uImage 已经编译好了，此文件不带任何后缀名。

5.5 编译 Uboot

编译 Uboot 非常简单，修改完毕后，只需要切换到 Uboot 目录下，打开终端并按顺序输入如下命令即可：

make distclean

make dm355_evm_config

make

看起来像如下界面：

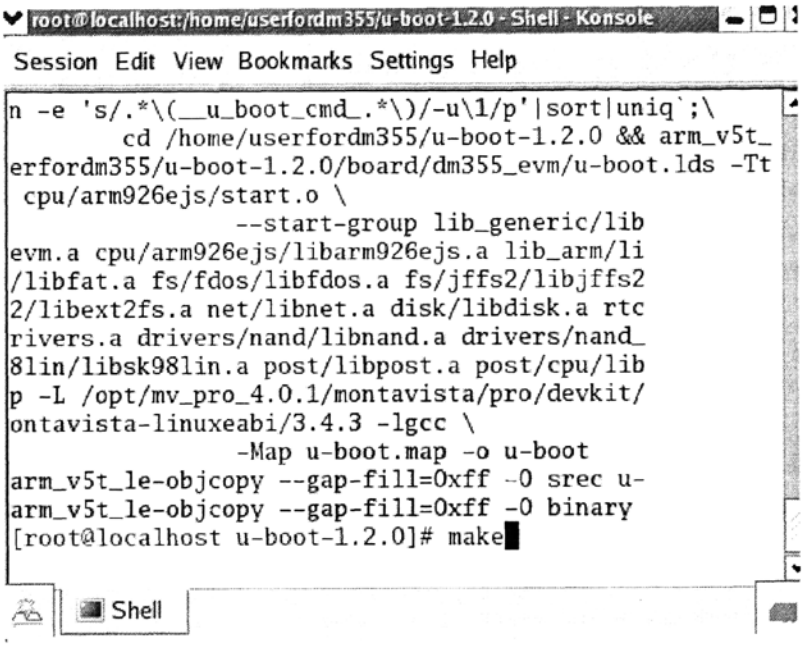

图 5-8 编译 U-BOOT

Fig.5-8 compiles U-BOOT

5.6 操作系统

5.6.1 操作系统介绍

嵌入式操作系统可以说是是嵌入式系统的“大脑”。嵌入式操作系统的出现是大大推动了嵌入式的发展。目前嵌入式操作系统[30]已经从简单走向成熟，主要有 VxWorks、QNX、PalmOS、Windows CE、Hopen OS 和嵌入式 Linux 等。

Windows CE 内核较小，能作为一种嵌入式操作系统应用到工业控制等领域。其优点在于便携性、提供对微处理器的选择以及非强行的电源管理功能。内置的标准通信能力使 Windows CE 能够访问 Internet 并收发 Email 或浏览器 Web。除此之外，Windows CE 特有的与 Windows 类似的用户界面使最终用户易于使用。Windows CE 的缺点是速度慢、效率低、价格偏高、开发应用程序相对较难。

3Com 公司的 Palm OS 在掌上电脑和 PDA 市场上独占其霸主地位，它有开放的操作系统应用程序接口，开发商可根据需要自行开发所需的应用程序。

QNX 是由加拿大 QSSL 公司开发的分布式实时操作系统，它由微内核和一组共同操作的进程组成，具有高度的伸缩性，可灵活地剪裁，最小配置只占用几十

KB 内存。因此，可以广泛地嵌入到智能机器、智能仪器仪表、机顶盒、通讯设备、PDA 等应用中去。

Hopen OS 是凯思集团自主研制开发的嵌入式操作系统，由一个体积很小的内核及一些可以根据需要进行定制的系统模块组成。其核心 Hopen kernel 一般为 1OKB 左右大小，占用空间小，并具有实时、多任务、多线程的系统特征。

VxWorks 是 WindRiver 公司开发的实时操作系统，它支持各种工业标准，包括 POSIX、ANSI C 和 TCP/IP 网络协议。VxWorks 运行系统的核心是一个高效率的微内核，该微内核支持各种实时功能，包括快速多任务处理、中断支持、抢占式和轮转式调度。微内核设计减轻了系统负载并可快速响应外部事件。目前在全世界装有 VxWorks 系统的智能设备数以百万计，其应用范围遍及互联网、电信和数据通信、数字影像、汽车、火控、导航与制导、通信和情报、声纳与雷达、空间与导弹系统、模拟和测试等众多领域。

Linux 是个与生俱来的网络操作系统，成熟而且稳定。Linux 是源代码开放软件，任何人都可以修改它，或者用它开发自己的产品。Linux 系统是可以定制的，系统内核目前已经可以做得很小。Linux 作为一种可剪裁的软件平台系统，是发展未来嵌入设备产品的绝佳资源。

嵌入式 Linux 与标准 Linux 的一个重要区别是嵌入式 Linux 与硬件芯片的紧密结合。这是一个不可逾越的难点，也是嵌入式 Linux 技术的关键之处。它不是一个纯软件的 Linux 系统，而比一般操作系统更加接近于硬件。嵌入式 Linux 的进一步发展，逐步地具备了嵌入式 RTOS 的一切特征：实时性及与嵌入式处理器的紧密结合。嵌入式 Linux 的另一特点是：代码的开放性。代码的开放性是与后 PC 时代的智能设备的多样性相适应的。代码的开放性主要体现在源代码可获得上，Linux 代码开发就像是“集市式”开发，任意选择并按自己的意愿整合出新的产品[31]。

5.6.2 选取操作系统

由于嵌入式操作系统与嵌入式系统密不可分，而且它是嵌入式系统的一个十分重要的组成部分。嵌入式操作系统的选择主要从一下几个方面考虑：

1. 操作系统的硬件支持，包括操作系统是否支持目标硬件平台，基于该选择的操作系统上开发的嵌入式应用软件是否具有很好的移植性；
2. 开发工具的支持程度，包括在线仿真器、编译器、汇编器、连接器、调试器等能否支持操作系统；
3. 应用需求，包括操作系统的性能、兼容性、技术支持等。

所以根据本嵌入式视频监控系统要求图像传输实时性，可靠性，稳定性，而且网络适应性等要求。所以选择了嵌入式实时操作系统 Monta Vista Linux 。

5.7 本章小结

本章介绍了嵌入式操作系统的选取，和在此操作系统上的开发平台搭建。包括 Nand flash 的烧写，Uboot 和 linux 内核的编译和烧写。完成了嵌入式开发应用程序的基础工作。即在此平台上可以完成应用程序的开发与调试。

6 嵌入式视频监控系统应用程序开发

本系统基于 SIP 协 SIP 服务器和各个终端组成，本次开发中 SIP 服务器的架设有两种方案：一是在 Windows 主机上运行 miniSipServer，二是在 Linux 主机上运行 openser。这两种方案任选其一即可。SIP 服务器的主要功能是维护各个可视电话终端的用户名、密码信息，提供注册、认证、定位服务……

架设好 SIP 服务器后，给各个终端分配好用户名，预设密码，然后每次可视电话开启就会在服务器上进行注册，服务器会把该用户名和当前的网络地址信息进行绑定。之后终端就可以接收或发起会话请求。

当一个终端发起会话请求并获得对方同意后，两个可视电话就开始视频和语音的通信。之后可视电话要不断进行以下工作：视频采集、编码、发送；视频接收、解码、显示；语音采集、转换、发送；语音接收、转换、播放。视频编解码采用 MPEG-4 标准，语音把双声道输入转化为单声道再进行传输，DSP 上运行的是 DSP/BIOS 实时操作系统，ARM 上运行的是 montavista linux。视频的 MPEG-4 编解码由 DSP 实现，其他部分包括视频语音数据的采集播放，网络通讯都是通过 ARM 上基于 Linux 的应用程序完成，程序用 C 语音编写，达芬奇平台自带各个外设驱动，包括摄像机、LCD、语音采集芯片 AIC33，网卡等。Montavista 公司提供 Linux 系统内核包括各个驱动的更新。

用户可通过 LCD 得到系统提示及当前会话信息等，并使用红外遥控器进行控制和输入，当视频会话建立起来后，通过 OSD（On Screen Display）在视频图象前端显示当前会话信息等。

程序运行后，提示用户输入自己的用户名和密码，注册成功后，可以选择等待呼叫或者发起会话请求，如果选择发起会话请求，则提示输入对方用户名，然后等待对方回应，如果对方拒绝会话，则结束，如果对方接受请求，则双方建立起视频语音通信，直到一方结束。

6.1 应用程序开发

在主程序里设置全局变量，并创建网络应用线程 uaThread，然后进入控制线程，对系统进行控制，相关信息在 LCD 上进行显示。

在 ua thread 中…

在 video thread 中会创建负责视频播放的 display thread…

整个系统一共有 4 个循环缓冲队列，分别由以下 4 对线程共享：video thread 和 video send thread、video receive thread 和 display thread、mic thread 和 speech send thread、speech receive thread 和 speech thread。上述 4 对线程中，前者对缓冲区进行写操作，后者对缓冲期进行读操作，缓冲区队列的可读、可写分别由 1 个信号量进行控制，这 4 对信号量也是分别由上述 4 对线程共享的。

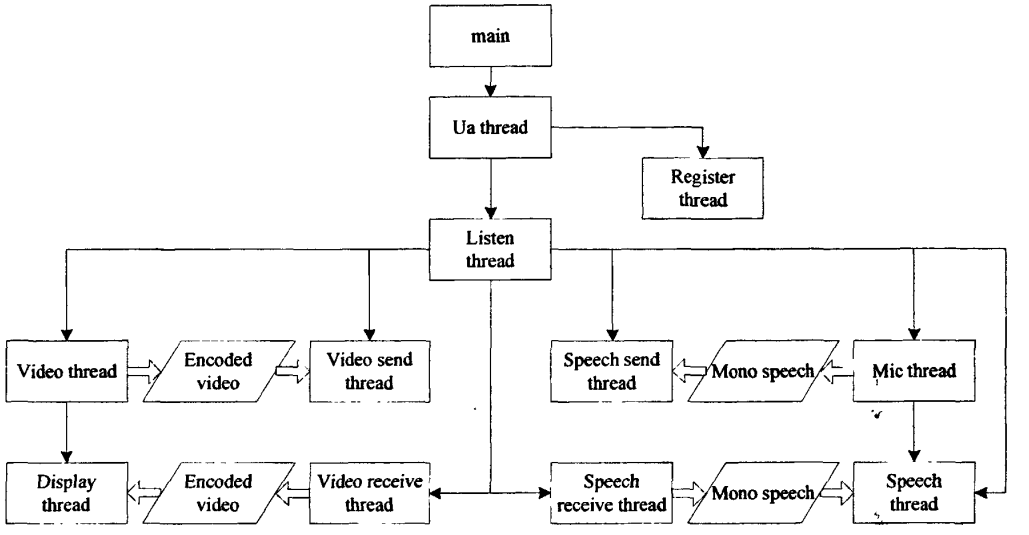

图6-1 视频监控板级程序线程示意图

Fig6-1 the relation of the threads

主函数里各个线程初始化的同步是通过自带的 Rendezvous 模块提供，它用 POSIX 条件变量配合互斥锁实现此功能。每个线程完成初始化之后，Rendezvous 计数器减一，直到所有线程均初始化完毕，Rendezvous 通知所有线程解锁，进入主循环。(多线程程序的开发效率问题)

6.1.1 主线程

main thread 的任务是初始化全局参数，创建负责会话建立和网络传输的 ua thread，探测视频采集格式（PAL 或 NTSC），启动 Codec engine，创建视频语音采集播放线程并设置相应优先级，然后进入红外控制状态。(采集格式的确定，要在 encodedecode 程序中进行选择，如果是用 5146 的系统，那么 PAL 和 NTSC 的采集格式都是支持的)

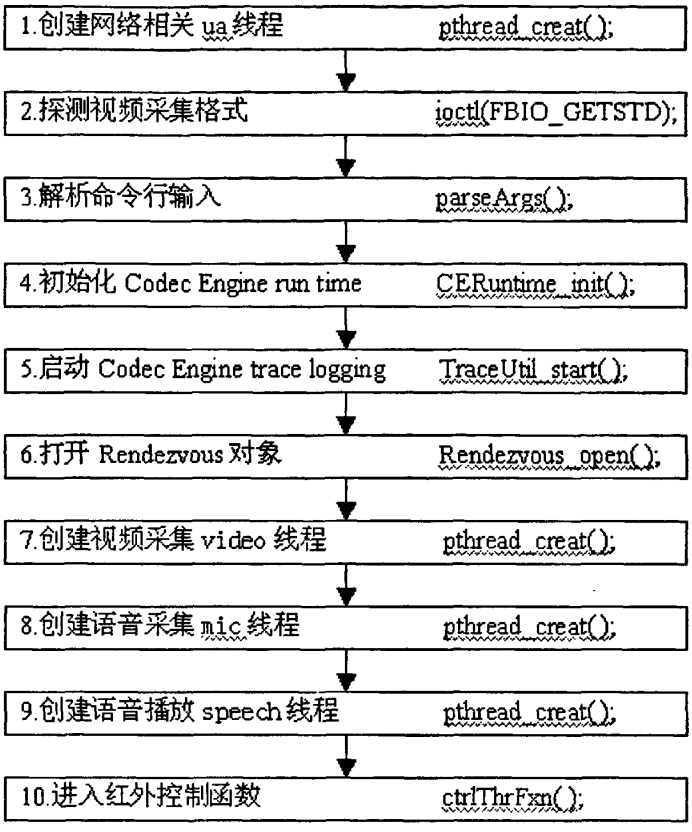

图6-2 主线程流程图

Fig.6-2 main thread process

在启动开发板之前，需要根据摄像机的制式把 S3 的开关 10 拨到相应位置，如果使用的是 NTSC 的摄像机，则把开关 10 拨到 OFF，如果使用的是 PAL 制的，则拨到 ON，然后在 u-boot 参数 bootargs 中添加“video=dm355fb:output=pal”。（这个是确定显示格式的）

首先启动负责 SIP 会话初始的 ua 线程，这时 ua 线程会进行用户注册以及 RTP 传输缓冲区的初始化，进行监听或根据用户输入发起会话。然后通过调用 FBIO_GETSTD 这个基于 frame buffer 显示设备驱动的 ioctl 接口，探测 S3 的开关 10，获得当前使用的制式。parseArgs 函数对运行程序时的命令行参数进行解析并设置相关全局变量。之后启动 Codec Engine 和它的 TraceUtil 模块以便跟踪。Rendezvous 对象的作用是同步各个线程的初始化，打开这个对象并制定需要同步的线程数之后，就创建 video、mic、speech 线程，各个线程初始化完毕，则 Rendezvous 通知上述需要同步的各线程可以进入主循环，主线程调用 ctrlThrFxn，进入控制状态，此时主线程成为控制线程。

6.1.2 控制线程

控制线程的作用是随时接收来自用户的输入，并在 LCD 上显示系统、会话的实时状态信息，提示用户进行相应操作。

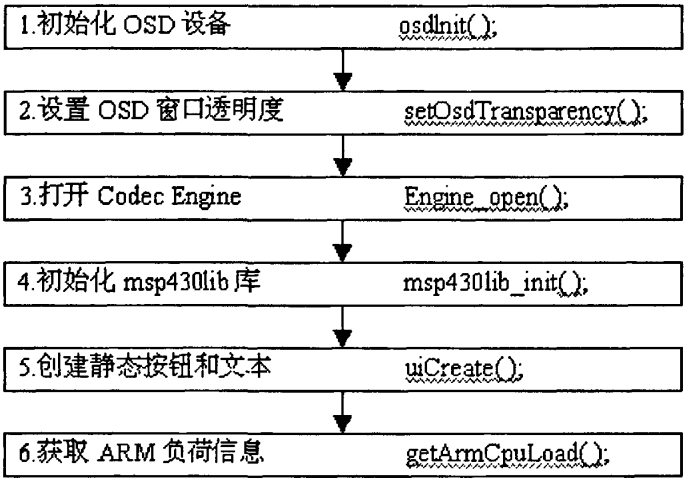

图 6-3 控制线程流程图

Fig.6-3 control thread process

在程序运行命令行参数中打开键盘控制接口，在键盘输入时，会定时调用 getKbdCommand()函数检测是否有键盘输入。 如果得到键盘输入，则调用 keyAction()函数对输入进行识别，并采取相应操作。程序不断循环检测是否有键盘输入，每次检测并完成相应任务，到下一次探测之前，线程都会 usleep 一个恒定的间隔，这个间隔是在 ui.h 头文件中定义的 REMOTECONTROLLATENCY，单位 us。（相当于对键盘等的扫描是采用等时间间隔的扫描模式，然后扫描到后就类似进入了中断执行）

除了接收键盘的的输入，控制线程还负责动态更新 OSD 上的文字图形信息。在达芬奇平台上，OSD 窗口（通过/dev/fb/0 进行访问）是在视频窗口（通过/dev/fb/3 进行访问）的前端。OSD 窗口的透明度是通过属性窗口（通过/dev/fb/2 进行访问）进行设置的，在属性窗口中，每个象素的透明度由半字节（4 bit）表示，它的取值范围从 0（完全透明）到 7（完全不透明）。控制线程调用 setOsdTransparency() 函数对 OSD 的透明度进行设置，默认透明度为 5。

在 OSD 上绘图以及显示文字的时候，控制线程用到了 simplewidget 库。首先 osdInit()函数对 OSD 设备进行初始化，uiCreate() 创建静态文字和按钮。之后控制线程基本以一秒为间隔调用 drawDynamicData() 更新动态信息。这些动态信息

是从其他线程采集到的，要用互斥锁进行保护，在头文件 encodedecode.h 中定义了对这些变量进行安全访问的内嵌函数。

ARM 和 DSP 的当前负荷分别由 getArmCpuLoad ()和 Engine_getCpuLoad()函数进行计算。其他的动态信息还包括比特率、视频帧率、程序运行时间等。OSD 窗口有两个缓冲区，在显示其中被称为 display buffer 的缓冲区的数据的同时，动态信息被写入另一个被成为 work buffer 的缓冲区。当 work buffer 中的数据准备好之后，调用 FBIOPAN_DISPLAY 这个 ioctl 接口，把 work buffer 中的数据交换到 display buffer 中，然后调用 FBIO_WAITFORVSYNC 这个 ioctl 接口等待下一次垂直同步信号（NTSC 制式 29.97Hz，PAL 制式 25Hz）

6.1.3 视频压缩采集

视频采集编码线程负责从视频采集设备获取图象缓存，并使用 MPEG-4 编码算法对图象进行压缩编码，然后通知 RTP 视频发送线程进行发送。另外，视频解码播放线程会在视频采集编码线程中被创建。

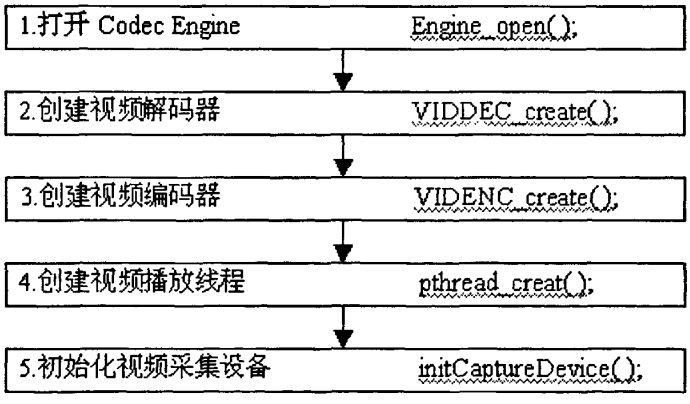

图 6-4 视频采集编码流程图

Fig. 6-4 capture video flow

如上图所示，视频采集编码线程的初始化步骤如下：

1. 使用 Engine_open()创建一个 Codec Engine 实例。这个函数会返回一个句柄，之后就用这个句柄初始化算法实例。所有使用同一个 engine 的线程都需要各自的句柄，通过这个句柄来对 engine 进行访问并不是线程安全的。

2. 调用 videoDecodeAlgCreate()函数创建解码器，本 demo 使用 MPEG-4 解码算法。在 videoDecodeAlgCreate()函数中，设置好解码器参数，然后调用 VIDDEC_create()创建解码算法实例。重要提示：由于现有的 H.264 编解码器初始

化的一个 bug，在创建编码器和创建解码器之间不可以调用 VIDDEC_control()或 VIDENC_control()。在 video.c 文件开头定义了 ALGO_INIT_WORKAROUND 预处理变量，这样程序将使用最坏情况下的编码缓冲区大小。正常情况下，编码缓冲区大小是给 VIDDEC_control()函数传递 XDM_GETBUFINFO 命令，向解码器查询得知的（代码中 #ifndef ALGO_INIT_WORKAROUND 后的部分）。在解决这个 bug 之前，为了设置编码器比特率，不得不牺牲动态获取编码缓冲区大小值的特性。

3. 调用 videoEncodeAlgCreate()函数创建编码器，本 demo 使用 H.264 编码算法。在 videoEncodeAlgCreate()函数中，设置好编码器参数，然后调用 VIDENC_create()创建编码算法实例。根据当前系统的 PAL 或 NTSC 制式，设置 targetFrameRate 和 refFrameRate 参数。这些参数对于算法是否能满足命令行中指定的期望比特率来说很重要（算法需要了解现实世界时钟）。如果用户在命令行中制定了一个小于零的比特率（默认-1），则通过把 rateControlPreset 参数设为 IVIDEO_NONE 值选择可变比特率（相对于设为 IVIDEO_LOW_DELAY 的恒定比特率）。最后，通过调用 VIDENC_control()函数，并传递 XDM_SETPARAMS 命令，设置动态视频编码器参数。

4. 在设定线程属性之后，创建视频播放 POSIX 线程。

5. 调用 initCaptureDevice()函数初始化视频采集设备。视频采集设备驱动是一个 Video 4 Linux 2 (v4l2)设备驱动。首先通过 VIDIOC_S_INPUT 这个 ioctl 设置自选输入连接器（合成或 s 端子），用 VIDIOC_QUERYCAP 这个 ioctl 来核实视频采集设备的实际性能。然后视频采集设备会自动探测制式（NTSC 或 PAL），并与系统启动时通过 u-boot 传入 Linux 内核的命令行参数中的视频显示制式进行对比确认。分辨率被设为 D1 格式，通过调用 VIDIOC_S_FMT ioctl，设备会把两个交积场组合到一帧（VL42_FIELD_INTERLACED）里。调用 VIDIOC_S_CROP ioctl，告知视频采集设备把 D1 格式的图象裁减成命令行输入的分辨率格式。调用 VIDIOC_REQBUFS ioctl 在视频采集驱动中分配 3 个视频采集缓冲区，并用 mmap()把它们映射到用户空间进程中。最后使用 VIDIOC_STREAMON ioctl 启动视频采集。

至此，视频采集编码线程初始化完成，使用 Rendezvous 模块实现与其他线程的同步，等待其他线程初始化完成后，再等待网络传输的 RTP 视频发送线程发出缓冲区可写信号量，之后就进入视频采集编码主循环，把编码视频数据写入与视频发送线程共享的视频发送缓冲区中。

6.1.4 网络应用线程组

在主线程中创建网络应用线程，网络应用包括会话初始（采用会话初始协议 SIP），视频、语音会话的创建等。

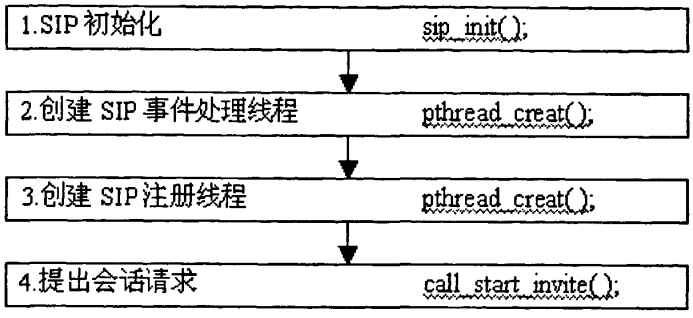

图 6-5 网络应用线程流程图

Fig.6-5 network thread flow

如果用户需要提出会话请求，则转入 call_start_invite 函数。

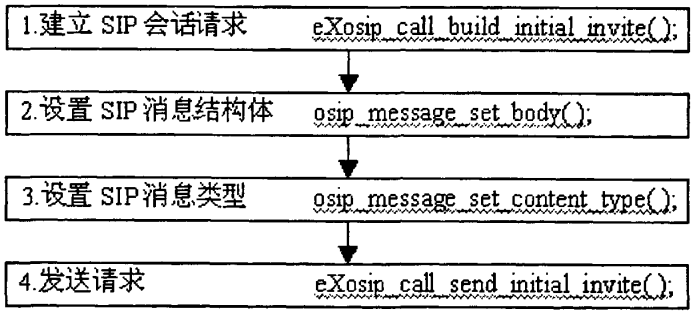

图 6-6 提出会话请求函数流程图

Fig.6-6 invite call function flow

6.1.5 SIP 事件处理线程

在网络应用线程中创建 SIP 事件处理线程，它将循环检测有没有 SIP 事件发生并采取适当回应。

SIP 事件处理包括：

收到注册成功信息：打印屏幕提示信息。

收到注册失败信息：打印屏幕提示信息。

收到会话请求信息：转入会话请求处理函数。

收到会话请求接受信息：转入会话请求接受处理函数。

收到会话确认信息：转入会话确认处理函数。

收到会话结束信息：转入会话结束处理函数。

其他未知信息：打印屏幕提示信息。

在会话请求接受处理函数 call_answered_handler() 或会话确认处理函数 call_ack_handler ()中，会启动负责图象采集的 video thread，负责语音采集的 mic thread，负责语音播放的 speech thread，以及相应的视频接收线程 rtp_video_send_thread、视频发送线程 rtp_video_receive_thread、语音发送线程 rtp_audio_send_thread，和语音接收线程 rtp_audio_receive_thread。

会话请求处理函数 call_accept_invite_handler ()

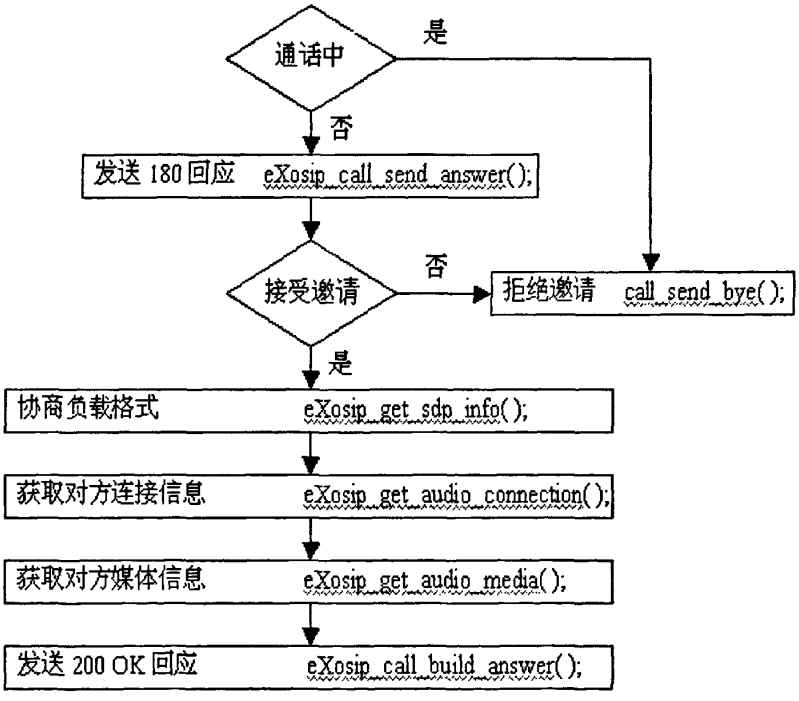

图 6-7 会话请求处理函数流程图

Fig.6-7 disposal invite function flow

会话请求接受处理函数 call_answered_handler ()

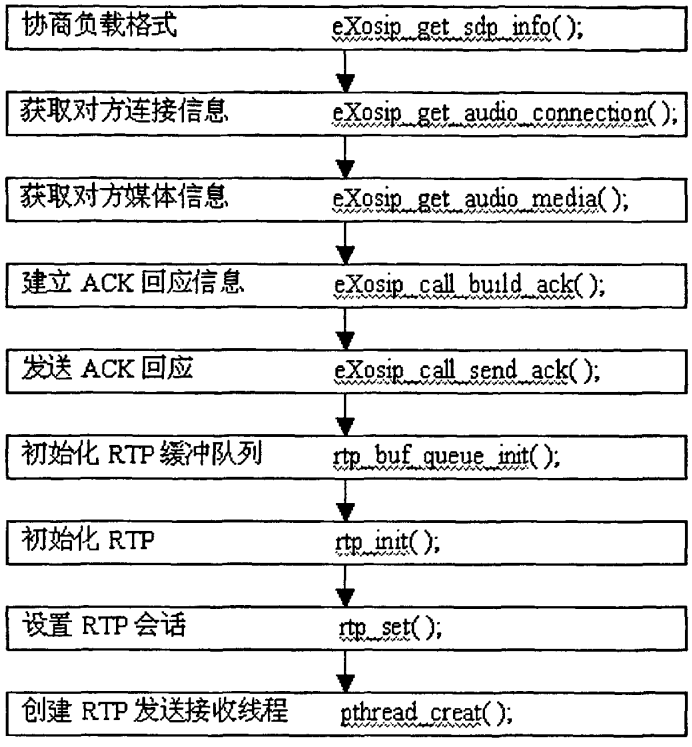

图 6-8 会话请求接受处理函数流程图

Fig.6-8 accept the invite the call function flow

其中，初始化 RTP 会话的 rtp_init()函数流程如下图：

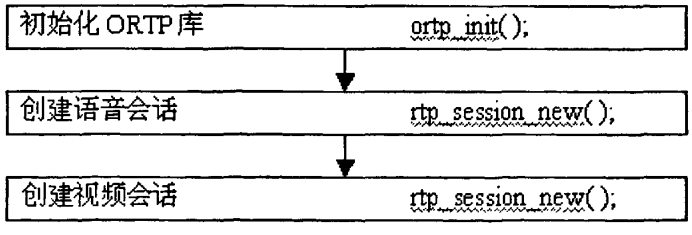

图 6-9 初始化RTP会话函数流程图

Fig.6-9 initializes RTP session function flow

会话确认处理函数 call_ack_handler ()

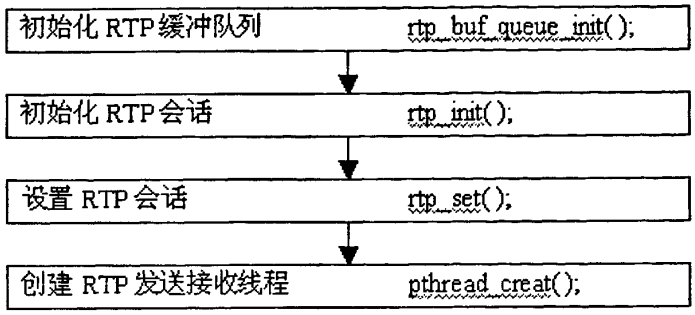

图 6-10 会话确认处理函数流程图

Fig.6-10 accept session solution function flow

会话结束处理函数 call_close_handler ()

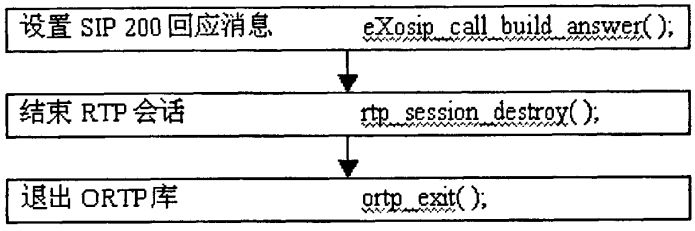

图 6-11 会话结束处理函数流程图

Fig.6-11 terminal session solution function flow

6.1.5 SIP 注册线程

在网络应用线程中设置（可包括接收用户输入信息）本地 SIP 用户、SIP 服务器等相关信息，并把其作为参数创建 SIP 注册线程。在 SIP 注册线程中，根据入口参数生成用户注册认证信息，并向 SIP 服务器提交注册请求。之后进入主循环定时向 SIP 服务器提交注册信息更新。

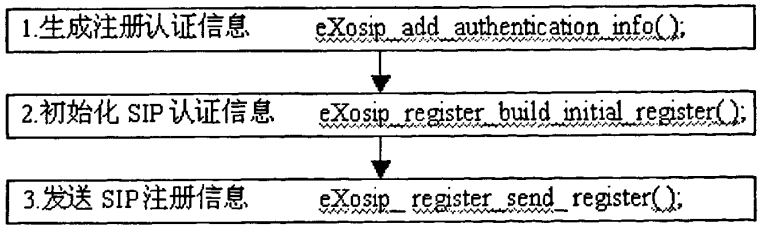

图 6-12 SIP注册线程流程图

Fig.6-12 SIP register thread flow

6.1.6 RTP 相关线程

(1) RTP 视频发送线程

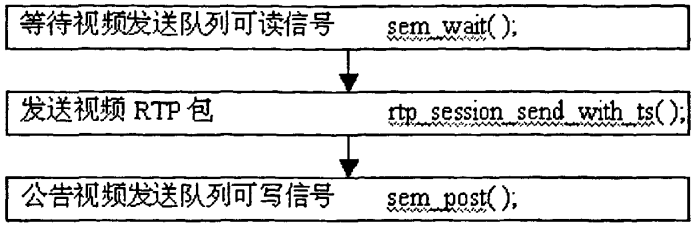

图 6-13 RTP视频发送线程主循环流程

Fig.6-13 RTP send video thread flow

(2) RTP 视频接收线程

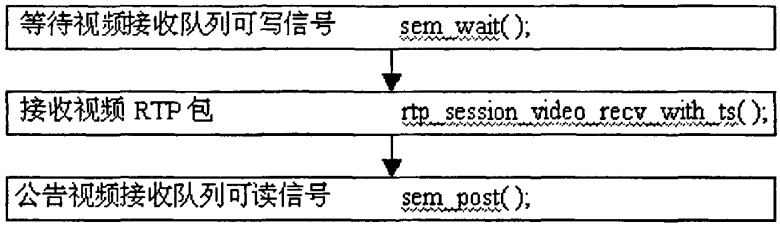

图 6-14 视频接收线程流程

Fig.6-14 video receive thread flow

其中，rtp_session_video_recv_with_ts()函数流程如下图：

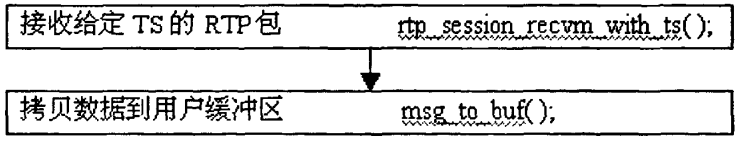

图 6-15 视频接收函数流程

Fig.6-15 video receive function flow

(3) RTP 语音发送线程

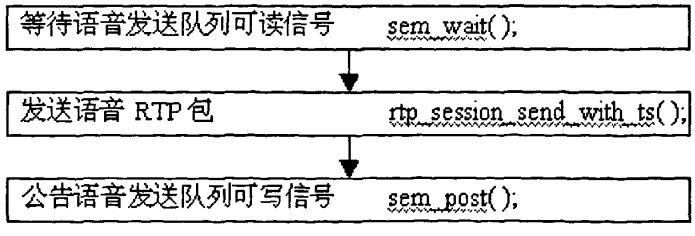

图6-16 RTP语音发送线程主循环流程

Fig.6-16 RTP audio send thread flow

(4) RTP 语音接收线程

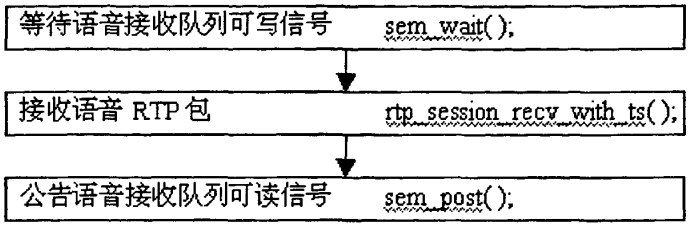

图6-17 RTP语音接收线程主循环流

Fig.6-17 RTP audio receive thread flow

6.2 本章小结

本章首先分析了 SIP 服务器的架设方案，虽然 Linux 主机作为 SIP 服务器的宿主机有较好的稳定性，但是考虑到如果界面的友好性和可操作性，还是选取了在 Windows 主机上运行 Minisipserver 这种方案。在线程的设计中采用了多线程的设计思路，且加入了视频和音频的缓冲区，来保证视频和音频的流畅性，稳定性。

7 嵌入式视频监控系统关键技术研究

7.1 基于 UDP/RTP/IP 的媒体网络传输

视频流媒体在网络媒体传输时常受到网络时延，带宽限制和高丢包率等外因的影响。从而使 Internet 实时多媒体应用的服务质量变差，最常见的表现是出于延时丢包引起的图像马赛克问题。

本系统采用了基于 UDP 协议的应用层解决方案，利用丢包率近似反映网络的性能指标。通过使用 RTCP 建立的反馈通道，采用自适应算法实时调整媒体速率，解决了流媒体速率收敛问题。即充分发挥 UDP 协议的良好实时性的优势，又实现了多媒体传输 Qos 的自适应控制。

7.1.1 码率收敛问题的提出和解决

有记忆系统及自相似，多级分形的网络模型

首先建立如下网络流量数学模型：

假设 X={x_t : t =0，1，2，...}是网络中的协方差平稳过程，其均值为μ，方差为σ^2；其自相关函数为 r(k)~$k^{-\beta}$ L(t)，其中$k \to \infty, 0 < \beta < 1$，L(t)在无穷大处缓慢变化后可简化为常数 L。

传统的研究认为，网络业务模型是基于 Poisson 或 Markov 过程，这样的网络没有或只具有简单的短时相关特性(SRD)，其长时相关性为零，即 $\lim_{n\to\infty} \sum_{k=1}^{n} r(k) < \infty$。但现在越来越多的研究表明，很多网络业务(特别是变比特率视频业务通信量)具有长时相关性(LRD)，也称约瑟夫效应(Joseph efect)[40]，即网络的协方差平稳过程{x_t}，自相关函数 r(k) 不收敛，即 $\lim_{n\to\infty} \sum_{k=1}^{n} r(k) = \infty$。此时可近似地认为网络表现出自相似的特性(自相似是长相关的一种)，但是距离较远的数据相关性是不可忽略的。例如，如果某一区间内的平均流量值比较大，则下一区间内的平均流量值较大的可能性也较大，反之亦然。也就是说网络是有记忆的实时变化体系。未来的状态值不但与当前值有关，还与所有历史值有关。利用这些历史信息，借助模型，能对未来网络状况做出合理的预测。在通信领域，人们已经提出了许多统计预测模型和自适应模型。针对网络 MAC 层协议的特点，丢包率和多媒体业务的特征，我们选择了自相似、多级分形的模型，同时根据实

际情况进行了相应改进和简化，以此来预测并跟踪网络系统可用带宽，相应地调整更新媒体流速率，实现网络传输速率的实时自适应控制。

7.1.2 自适应算法和最优滤波理论

自相似的网络是有记忆的系统，在一定时间内有关联性。借鉴自适应滤波器，建立一个带宽预测和速率调整的模型。定义代价函数 J(n)，使此代价函数依照某个统计意义下准则的最小化，可以实现该准则下的最优的自适应算法。本文中 J(n) 使用了最小均方误差(MMSE)准则。考虑第 i 个 RTCP 包反馈媒体速率为 $R_{history}(i)$，若 $R_{history}(i)$为 n 维的历史速率向量，W(i)为与之对应的 n 维权重系数向量，则预测该时间段内网络媒体速率为:

$$\bar{R}(i)=f\left(R_{history}(i),W(i)\right) \tag{1}$$

定义估计误差

$$e(i)=R_{history}(i)-\bar{R}(i) \tag{2}$$

定义梯度算符∇_k，根据 MMSE 准则推导得到与“正交性原理”及其引理相对应公式:

$$J(i)=E\left\{e(i)\right|^2\}=E\left\{e(i)e^*(i)\right\} \tag{3}$$

$$\nabla_k J(i)=\frac{\partial J(i)}{\delta w_k(i)}=0 \quad k=0，1，2... \tag{4}$$

$$-2E\left\{R_{history}(i-k)e^*(i)\right\}=0 \tag{5}$$

$$\begin{cases} E\left\{R_{history}(i-k)e^*_{opt}(i)\right\}=0 \\ E\left\{\bar{R}(i-k)e^*_{opt}(i)\right\}=0 \end{cases} \tag{6}$$

其中，已 e*()为 e()的共轭转置。

自适应算法越接近满足式(6)的条件，则自适应性能越好。借助 LMS 算法和 RLS 算法分别建模以逼近这个条件，通过进一步的推导，对应得到如下两种算法过程，自适应梯度算法和自适应高斯--牛顿算法。

(1) 自适应梯度算法。

LMS 算法

初始：W(0)=0:

更新：i=1，2...

$$\bar{R}(i) = W^n(i-1)R_{history}(i)$$
$$e(i) = R_{current} - R(i)$$
$$W(i) = W(i-1) + \mu(i)R_{bistory}(i)e^*(i)$$

(2) 自适应高斯--牛顿算法

RLS 算法

初始：w(0)=0；P(0)=δ^{-1}I　W(0)=0；

更新：i=1，2...

$$\begin{cases} \bar{R}(i) = W^n(i-1)R_{history}(i) \\ e(i) = R_{current} - \bar{R}(i) \end{cases}$$

$$\begin{cases} k(i) = \dfrac{P(i-1)R_{history}(i)}{\lambda - R^H_{history}(i)P(i-1)R_{history}(i)} \\ P(i) = \dfrac{1}{\lambda}\left[P(i-1) - k(i)R^H_{history}(i)P(i-1)\right] \end{cases}$$

$$W(i) = W(i-1) + k(i)e^*(i)$$

研究表明，在变化比较慢时，采用 LMS 算法可以更好的自适应跟踪系统变化。因此本系统的最后的模型基础就是 LMS 的自适应算法。

7.1.3 速率控制策略分析

由服务器端在线实时调整的下一时刻速率是既和当前速率又和历史速率相关(如下图所示)。变量函数关系推导如下：

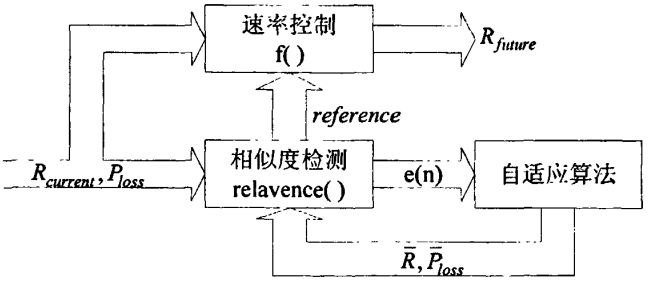

图7-1 速率控制图及变量函数关系推导

Fig.7-1 the relation of variable function

因为

$$QR_{future} \leftarrow f\left(R_{current}, \bar{P}_{loss}, reference\right) \tag{7}$$

又

$$Qreference \leftarrow relavence\left(\bar{R}, R_{current}, \bar{P}_{loss}, P_{loss}\right) \tag{8}$$

而

$$\begin{cases} \bar{R} \leftarrow g_R\left(R_{current}, R_{history}(i)\right) \\ \bar{P}_{loss} \leftarrow g_\rho\left(P_{loss}, P_{history}(i)\right) \end{cases} \quad i=1\ldots n \tag{9}$$

所以

$$R_{future} \leftarrow f'\left(R_{current}, R_{history}(i), P_{loss}, P_{history}(i)\right) \tag{10}$$

其中 R_{future} 为服务器端控制的流媒体播放速率，$R_{current}$，P_{loss} 分别为 RTCP 反馈得到的客户端媒体速率和网络丢包率，$\bar{R}$，$\bar{P}_{loss}$ 为按自适应算法更新的预测媒体速率和网络丢包率。

图 7-1 中，相似度检测的一组输入 $R_{current}$，P_{loss} 为服务器通过 RTCP 接收获得的反馈信息，另一组 $\bar{R}$，$\bar{P}_{loss}$ 为由自适应算法计算得到的预测值。传递函数 relavence()为相似度函数，它将两组输入进行相似性的比较。输出中 reference 为当前状态与历史状态之间的相似性标志，用作速率更新函数 f()的输入参数，e()为误差函数，用来驱动自适应算法实时调整，是算法收敛。

速率控制的输入为 reference 和 $R_{current}$，P_{loss}。传递函数 f()位速率更新函数，在自相似、多级分形的网络模型下，根据不同输入参数使用相应的公式。速率控制的输出为服务器端的更新速率 R_{future}。很多文献都对网络传输带宽的预测和控制函数模型进行了研究，一般都会基于快升降策略进行优化和改进，类似的，本方案采用的具体算法如下：

$$R_{future} = R_{current} + \Delta R \tag{11}$$

$$\Delta R = \begin{cases} -R_{current} \min\left(\Delta R_{down}, R_{down\_\max}\right) P_{loss} > 0 \\ \max\left(\Delta R_{up}, R_{up\_\min}\right) P_{loss} < 0 \end{cases} \tag{12}$$

$$\Delta R_{down} = \begin{cases} r_{down} \bar{P}_{loss} \text{ 非收敛状态} \\ \left(R_{current} + \bar{R}\right)/2 \text{收敛状态} \end{cases} \tag{13}$$

$$\Delta R_{up} = r_{up}\left(1 - \frac{R_{current}}{\bar{R}}\right) \tag{14}$$

式中 ΔR_{up} 为实习调整的上升速率改变步长，采用类似于对数函数的曲线上升策略，使最终速率逐步稳定收敛；ΔR_{down} 为实时调整的下降速率改变步长，由相

似性标志 reference 决定。

若系统与前一时间段相似，处于收敛态，则降幅逐步减半，是码率振幅减小趋于稳定收敛

$$\Delta R_{down}(t+1) \approx \frac{1}{2}\Delta R_{down}(t) \tag{15}$$

如果网路状况恶化，则使下切步长与丢包率正相关，速率更新为式(16)，这样求得的切换值可以很好的追踪网络环境变化，尽快环节拥塞，改善媒体传输的质量

$$R_{future} = R_{current}\left(1 - r_{down}\bar{P}_{loss}\right) \tag{16}$$

式(13)中 r_{down} 和式(14)中 r_{up} 分别是下降和上升速率控制因子，其数值选定将影响系统算法的性能。

在自相似网络的环境下，短时间尺度的反馈控制难以反映网络流量长期变化趋势，利用适时调整的 r_{up}，r_{down} 可以来弥补这个缺陷，增强端节点对自相似网络可用带宽的跟踪能力。R_{up_min}，R_{down_max} 为速率上、下限经验值，有助于算法的优化和系统稳健性维护。

7.1.4 系统采取的方案

图中是服务器端采用的自适应码率控制方案和公式，

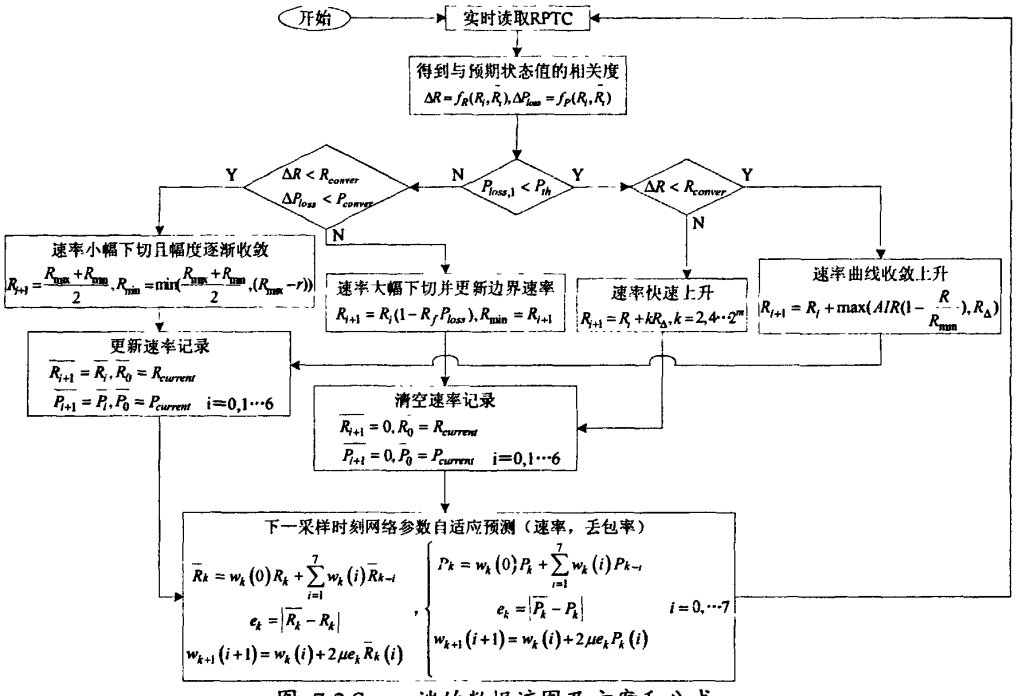

图 7-2 Server端的数据流图及方案和公式

Fig.7-2 data stream and formula

方案解释如下：

1. 获得网络参数和速率切换的判断

服务器端每隔 0.2s 查询 RTCP 包，该包是利用 RTCP 链路从客户端反馈过来的。通过查询，服务器端得到实时的网络参数 $R_{current}$，R_{loss}，这些参数是在客户端被提取并封装在 RTCP 包里。随后服务器端判断丢包率 Rloss，超过丢包阈值 Pth 时控制使速率降下来，反之则使速率上升。P_{th} 的值根据传输业务要求的 QoS 和网络实时环境影响来预先设置，本方案预设为 0。

2. 相似性的判断

服务器端通过相关度函数 $f_R()$， $f_P()$ ，对由 RTCP 实时得到的网络参数 $R_{current}$， P_{loss} 和利用历史数据预测的 $\bar{R}$， $\bar{P}_{loss}$ 来进行计算，求得其相关度，以进行相似比较来判断速率是否处于相似态

$$\begin{cases} \Delta R = f_R\left(R_i, \bar{R}_i\right) \\ \Delta P = f_P\left(R_i, \bar{R}_i\right) \end{cases} \tag{17}$$

本方案中，作为一种易于实现的简化，可以认为满足式(18)时可以判断系统当前处于相似态

$$\begin{cases} \Delta R = \left| R_i - \bar{R}_i \right| < R_{conver} \\ \Delta P_{loss} = \left| P_{lossi} - \bar{P}_{lossi} \right| < P_{conver} \end{cases} \tag{18}$$

R_{conver}， P_{conver} 为实验得出收敛判定标尺经验数据。

3. 速率切换控制

$$R_{futture} = R_{current}\left[1 - \min\left(\Delta R_{down}, R_{down\_\max}\right)\right] \tag{19}$$

根据所分析的速率控制策略，在相似状态下，速率下切调整将采用式(9)，降幅逐步减半，从而码率振幅减小趋于稳定收敛。否则速率下切调整采用式(10)，下切步长与丢包率正相关，这样速率切换的幅度将快速适应并改善网络状况。

4. 速率上升控制

$$R_{futture} = R_{current} + \max\left(\Delta R_{up}, R_{up\_\min}\right) \tag{20}$$

与速率下降控制时根据相似态和非相似态分别考虑的情况类似，若系统处于相似态，参考 TCP 协议改进的快降慢升策略，使用式(14)使速率缓慢上升，其变化类似于对数曲线。否则，使用式(21)，使速率较快上升并迅速跟踪到最大可用带宽，步进上升值 R_Δ 为

$$\Delta R_{up} = kR_\Delta, \quad k=2, 4 \ldots 2^m \tag{21}$$

5. 历史数据的更新或清空

不管速率上升还是下降，相似态使每一个数据都与未来的数据相关，被用做预测。作为一种简化，采用 8bit 移动窗，把最新的当前数据移入，窗内历史数据移位保持，历史数据移出

$$\begin{cases} \overline{R_{i+1}} = \overline{R_i}, \overline{R_0} = R_{current} \\ \overline{P_{i+1}} = \overline{P_i}, \overline{P_0} = P_{current} \end{cases} \quad i=0, 1, \ldots 6 \tag{22}$$

反之，一旦跳出相似态，所有历史信息要清空复位，以等待下一次相似态进行预测。

$$\begin{cases} \overline{R_{i+1}} = 0, \overline{R_0} = R_{current} \\ \overline{P_{i+1}} = 0, \overline{P_0} = P_{current} \end{cases} \quad i=0, 1, \ldots 6 \tag{23}$$

6. 自适应预测

根据上面章节所述，采用 LMS 自适应算法在低速率的多媒体传输中可以得到好的跟踪效果，根据原始公式推导可得公式(24)(25)，以用来预测未来数据。

$$\begin{cases} \overline{R_k} = w_k(0)R_k + \sum_{i=1}^{7} w_k(i)\overline{R_{k-i}}(i) \\ e_k = \left|\overline{R_k} - R_k\right| \\ w_{k+1}(i+1) = w_k(i) + 2\mu e_k \overline{R_k}(i) \end{cases} \quad \text{i=0，1，2...7} \tag{24}$$

$$\begin{cases} \overline{P_k} = w_k(0)P_k + \sum_{i=1}^{7} w_k(i)\overline{P_{k-i}}(i) \\ e_k = \left|\overline{P_k} - P_k\right| \\ w_{k+1}(i+1) = w_k(i) + 2\mu e_k \overline{P_k}(i) \end{cases} \quad \text{i=0，1，2...7} \tag{25}$$

其中 $w_k(i)$ 为当前权重向量，$w_{k+1}(i+1)$ 为下一时刻权重向量，e_k 为误差，$\overline{R_k}$ 为预测当前速率，$\overline{P_k}$ 为预测当前丢包率，$\overline{R_k}(i)$，$\overline{P_k}(i)$ 为当前历史数据向量。

7.2 视频和音频线程研究分析

应用程序中采用的线程很多，只有各个线程都能稳定良好工作，且线程之间可以稳定交互通信，软件系统才能稳定工作。应用程序运行时必须合理的分配内存空间，防止内存泄露导致系统崩溃。所以本章研究关键技术线程间的交互和内存管理以及各缓冲区的设置。

7.2.1 线程间的交互分析

视频采集编码与发送线程之间的交互

视频采集编码线程和 RTP 视频发送线程共享一个循环缓冲区队列，主要通过一对读写信号量实现两线程间的交互。

循环缓冲区内存分配由 ua 线程调用 rtp_buf_init()函数完成，这个函数完成了视频、语音的发送、接收共 4 个缓冲区队列的初始化和内存分配。对于视频发送、接收线程，由于缓冲区内数据要分别作为视频编码器的输出和视频解码器的输入，而 DSP 上运行的 DSP/BIOS 操作系统是不带 MMU（内存管理单元）的，直接对内存的物理地址进行操作，所以在分配内存空间的时候要用 Memory_contigAlloc()分配物理连续的内存，指定默认对齐模式。每节点的内存大小考虑最坏情况下的编码缓冲区大小，对于 D1 分辨率的 H.264 编码，此值为 1658880 字节。

开始时，循环缓冲区队列是空的，下图为循环缓冲区队列长度为 3 时的示意图。这时 RTP 视频发送线程中的读指针 rPtr 和视频采集编码线程中的写指针 wPtr

重合，可写信号量 sem_w 初始值为 3，可读信号量 sem_r 初始值为 0。

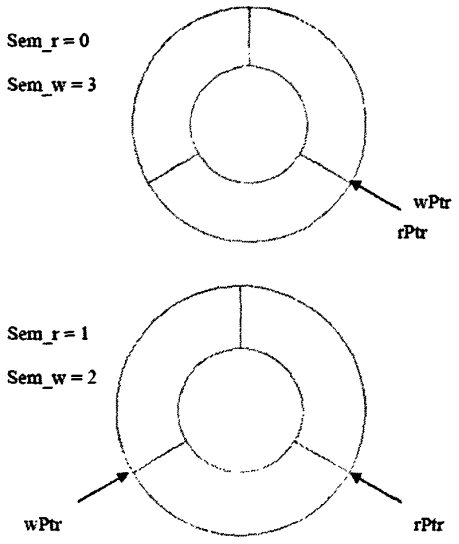

图 7-3 缓冲区示意图

Fig.7-3 circulation buffer hint

视频采集编码线程开始采集视频数据并进行压缩编码，在把压缩后的视频数据写入循环缓冲区队列的一个节点并设置好此节点已用长度 used_len 后，使用 sem_post 把可读信号量加 1，然后进入下一轮的阻塞等待直到可写信号量大于零。可读信号量 sem_r 加 1 后，满足大于零的条件，此时阻塞在 sem_wait 上的 RTP 视频发送线程就把可读信号量减 1 后退出阻塞状态，线程接着读取此时读指针 rPtr 指向的地址，把此地址中的数据通过封装成 RTP 格式后发送出去，负载大小为视频采集编码线程所设的 used_len。发送完毕，用 sem_post()把可写信号量加 1，进入下一轮阻塞等待直到可读信号量大于零。

视频接收与解码播放线程之间的交互

视频接收与解码播放线程之间的交互与 2.7 中所描述的视频采集编码与发送线程之间的交互类似。缓冲区队列长度可能有所区别。另外，考虑 IP 包尺寸的最大值 65535 字节以及一个视频帧压缩编码后大小的最坏情况，demo 中把接收缓冲队列节点连续内存块大小设为 40000 字节。

语音采集与发送线程之间的交互

语音采集与发送线程之间的交互与前面所描述的视频采集编码与发送线程之间的交互类似。区别在于缓冲队列长度、每节点缓冲内存块大小以及对连续性和对齐的要求不同。语音数据不需要经过编码解码的过程，分配内存块的时候只需要用 Linux 下标准内存分配函数 malloc()即可，不需要像视频发送线程那样分

配物理连续内存块。而且每次读取语音采样数据的大小都是固定的INPUTBUFSIZE，把双声道数据转化为单声道数据之后大小缩小一半，也是固定的(RAWBUFSIZE)，把此设为语音发送缓冲队列节点内存块大小。

语音接收与播放线程之间的交互

语音接收与播放线程之间的交互与 2.7 中所描述的视频采集编码与发送线程之间的交互类似。区别在于缓冲队列长度、每节点缓冲内存块大小以及对连续性和对齐的要求不同。语音数据不需要经过编码解码的过程，只需要把接收到的单声道数据转化为伪双声道数据，再写入声卡设备。

7.2.2 内存管理

DM355EVM 内存共有 256M，前 120M 分配给 ARM 上的 Linux 操作系统，在 u-boot 的 bootargs 中设置 mem=120M；接着的 8M 是 ARM 上 Linux 和 DSP 共享的空间，后 128M 归协处理器使用。

共享内存的设置需要调用 cmem 模块，在程序运行前分配好空间以及所需连续内存块大小，满足应用程序需要，在本 demo 中根据视频缓冲区和语音缓冲区的需要，修改“pools=”部分，配置以下脚本启动 cmem.ko 模块以及 dsplink.ko 模块：

```
# insert cmemk,  tell it to occupy physical 120MB-128MB,  create
insmod  cmemk.ko  phys_start=0x87800000  phys_end=0x88000000  pools=3x1660000 ,
5x40000,  3x700000,  30x1280
# insert dsplinkk,  tell it that DSP's DDR is at physical 250MB-254MB
insmod dsplinkk.ko ddr_start=0x8fa00000 ddr_size=0x400000
# make /dev/dsplink
rm -f /dev/dsplink
mknod /dev/dsplink c `awk "\\$2==\"dsplink\" {print \\$1}" /proc/devices` 0
```

7.2.3 缓冲区设置

视频缓冲区设置

视频缓冲区包括视频发送缓冲区和视频接收缓冲区。每个缓冲区都是一个由物理连续内存块组成的单向链表组成，每个链表中节点大小一致。

视频发送缓冲区队列节点作为视频编码器的输出缓冲区，所需大小由解码器根据图象分辨率指定。对于 NTSC D1(760x480)来说，最坏（大）情况是 1658880 字节。由于此队列节点大小是整个程序中最大的，考虑到在网络情况比较好的情

况下，发送速度比编码速度快，所以队列长度可以适当减小，后来经过实测队列使用率，把视频发送队列节点数设为 2。

接收缓冲区队列节点大小需要考虑的是编码后每帧图象的最大尺寸，程序中设为 40000 字节，队列长度需要考虑视频传输的抖动和时延，实时流媒体的时延和连贯性是一对矛盾。增加接收缓冲区长度可以改善流媒体的连贯性，但可能增加端到端的时延。而视频对于实时性的要求比语音低，比如 1 秒的时延也是可以接受的，可适当增加视频接收缓冲区长度。比如可以把视频接收队列节点数设为 15，大概可以缓存 1 秒的视频数据。

语音缓冲区设置

语音缓冲区同样包括语音发送缓冲区和语音接收缓冲区。Demo 暂时没有对语音数据进行编码，但考虑到将来的扩展，把语音缓冲区也放在共享内存中，每个节点也是物理连续的内存块。

在不对语音数据进行压缩的情况下，采样频率一定，发送语音数据包的频率一定，包大小是常数。比如现在我们每 40ms 发送一个语音包，采用 8000Hz 采样频率，把双声道数据转化为单声道数据后每个数据包大小为 640 字节。所以语音发送和接收缓冲队列节点大小都是 640 字节。

对于语音实时通讯来说，时延需要控制在 400ms 内，语音接收缓冲队列节点数相应设为 10。

oRTP 缓冲区

在 ortp 库中，可以设置抖动参数 jitter，比如在程序中把视频会话和语音会话的 jitter 设为 40，则 rtp 库在接收到数据后并不马上提交给上层应用，而是放在自身缓冲区中，根据各个数据包的时间戳 ts 以及频率计算出已缓存的数据包的持续时间，只有当收到的数据包达到 40ms 后，才提交给上层应用，这样可以在一定程序上改善媒体连贯性。

7.3 本章小结

本章对于嵌入式网络的监控系统的关键技术进行了研究，分析了在基于网络传输控制协议 UDP/RTP/IP 的自适应控制的方法。分析了多线程之间的交互问题，和程序开发中的缓冲区设置问题。通过以上分析，找到了合理分配 DM355 内存的方法，提高了应用程序的效率。

8 系统测试及性能评价

8.1 系统调试

系统调试时将客户端 PC 机和开发板通过调试串口、以太网接口建立起连接，然后设置具体参数，将开发板和 PC 机以及虚拟主机连系到一起，使他们能够正常运行。为了更好的调试，在 PC 机座位监控器，开发板运行监控程序和被调试程序。PC 机通过调试器与开发板监控器建立通信连接，相互间的通信遵循远程调试协议。监控程序包含基本功能的启动代码，并完成必要的硬件初始化，等待 PC 机命令。被调试程序通过监控程序下载到开发板，就可以开始调试。监控器方式操作简单易行，功能强大，不需要专门的调试硬件，使用面广，能提高调试的效率，缩短产品的开发周期，降低开发成本。正因为以上原因，监控器方式监控器方式才能够广泛应用于嵌入式系统的开发中。本系统开发板软件调试工作正是采用这种方式。

系统的测试流程如下：远程现场模块在监控现场通过摄像头采集视频图像流，网络视频监控终端对原始的视频数据进行数模转换、压缩、MPEG-4 视频编码处理，经过编码后的视频码流发送到视频服务器或者直接通过 WEB 服务器响应客户端请求。DM355 的板级 pal_no430_DM355_Camera 程序通过 RTP 协议传输完整的视频音频流。客户端通过连接到 SIP_Server(miniSipServer)注册，然后接受 RTP 数据。

详细步骤：

主机串口连接开发板，IP 地址为 192.168.1.246，波特率 57600，启动 miniSipServer。

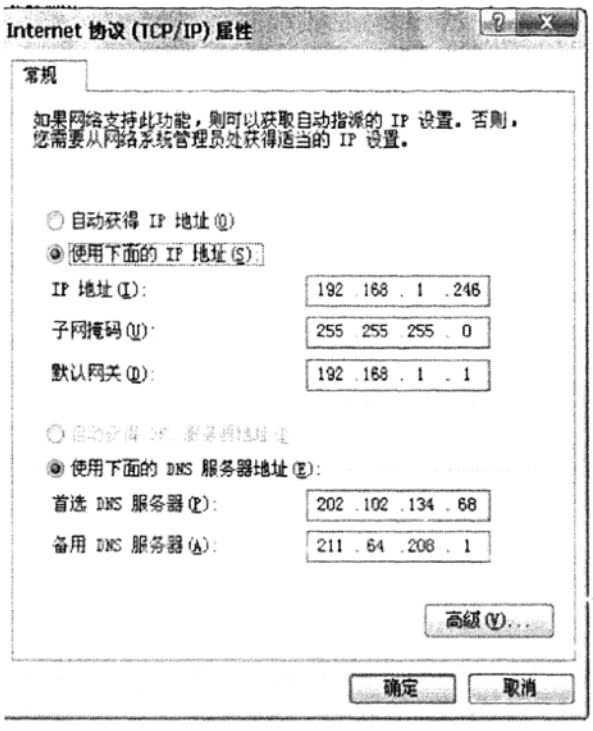

图 8-1 网络配置

Fig.8-1 network config

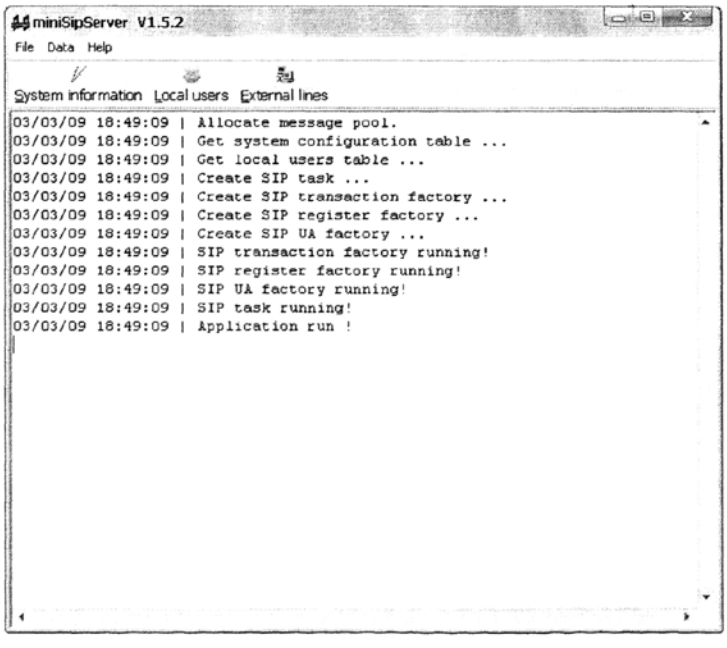

图 8-2 启动 Minisipserver 示意图

Fig.8-2 startup Minisipserver hint

主机安装虚拟机，运行 Fedora8 的 IP 地址设为：192.168.1.55。

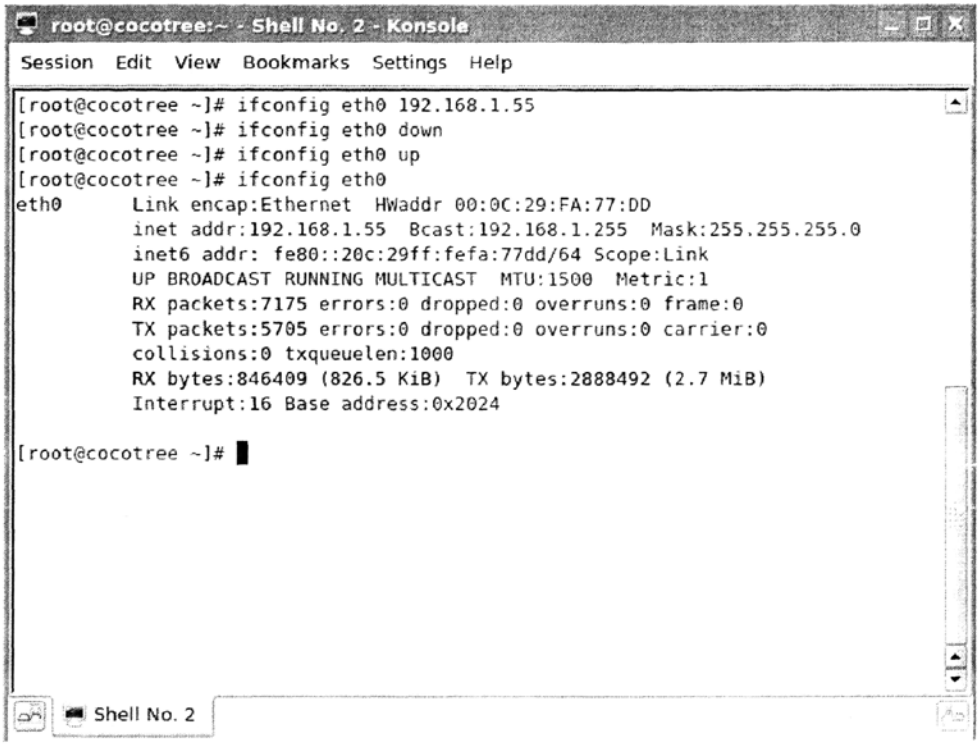

图 8-3 虚拟主机控制台示意图

Fig.8-3 simulate console hint

客户机 IP 设为 192.168.1.6。

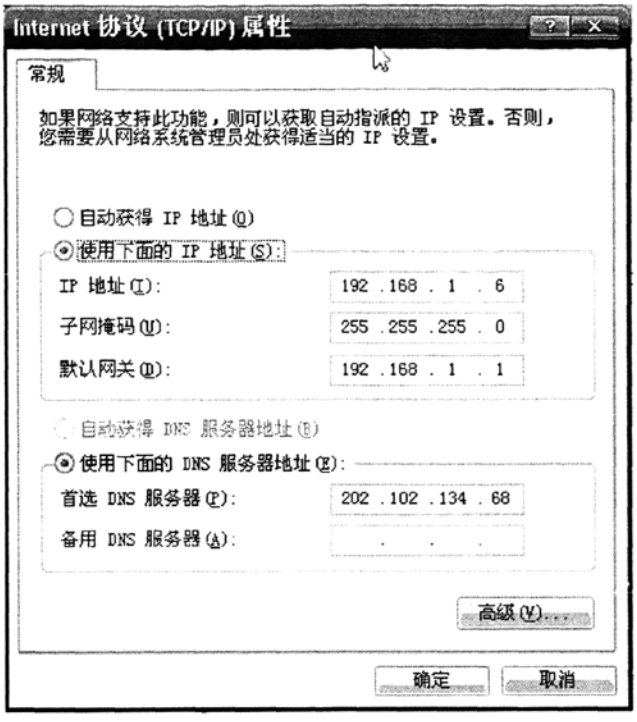

图 8-4 客户机网络配置

Fig.8-4 client network config

在主机的 Fedora 控制台下输入./onebuffer.sh 分配 9 个，总计 5MB 的地址池。

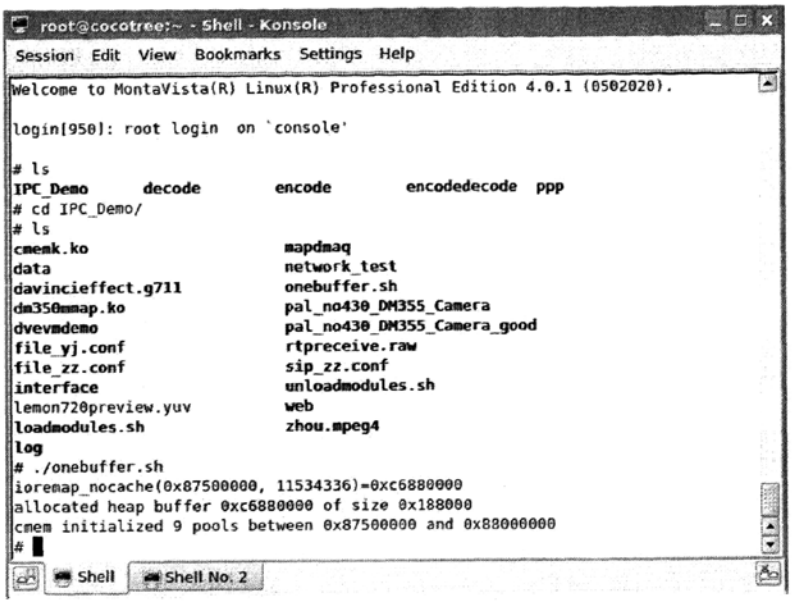

图 8-5 缓冲区线程执行示意图

Fig.8-5 buffer thread execute

客户端 MyPlayer 点“登录”连接 SIP 服务端，然后在主机 Fedora 控制台下执行./pal_no430_DM355_Camera。

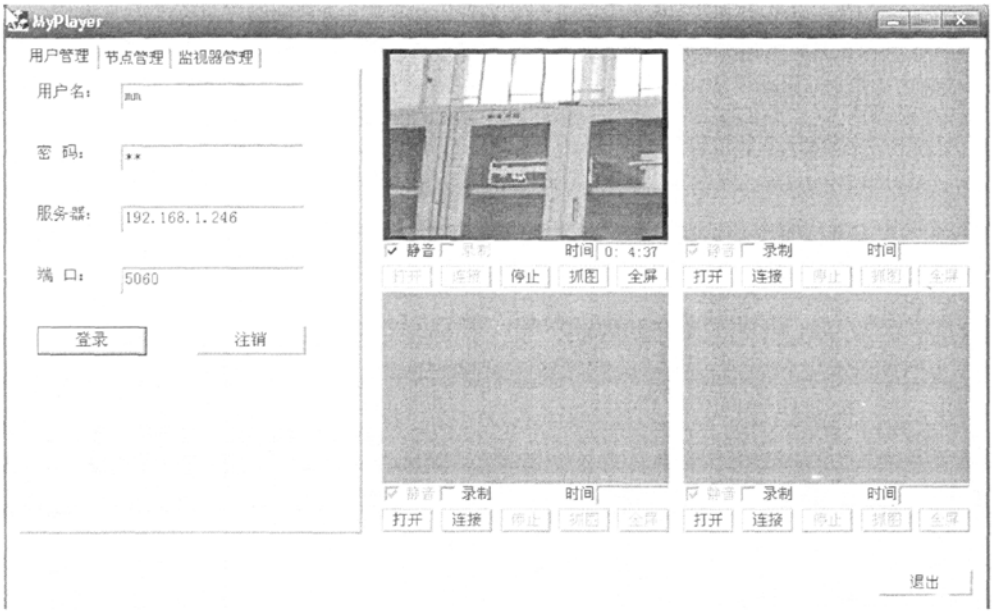

图 8-6 客户端图像

Fig.8-6 client software

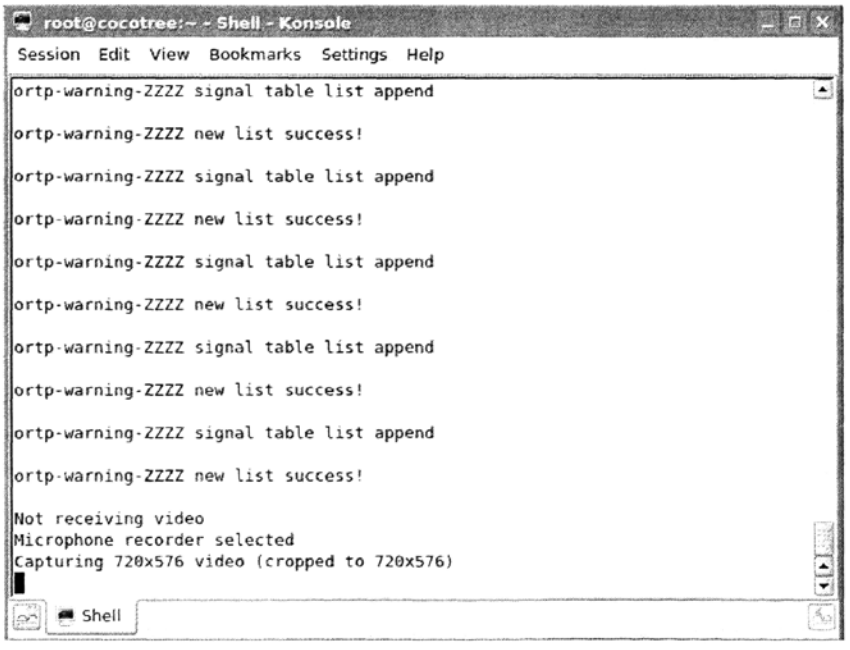

图 8-7 控制台显示视频参数信息

Fig.8-7 video parameter display

8.2 性能数据分析

测试的软硬件环境如下表所示：

表 8-1 测试环境

Table 8-1 testing environment

硬件	网络视频监控终端
	PC 配置—1.CPU：P4 2.4G；RAM：256M；显存：64M； 2.CPU：P4 3.0G；RAM：512M；显存：128M
播放控件	Media Player
音频格式	G.711
网络环境	局域网

为了测试系统的基本功能和性能，搭建了实验环境系统中，在局域网环境下，分别对 PAL 和 NTSC 的 CIF 和 D1 幅面大小进行传输和测试，

测试结果如下表所示：

表6-2 性能参数

Tab 6-2 performance parameter

	1	2	3	平均值
PAL	帧率：1(25 帧/秒)，流类型：video only，更改 VBR，CBR 值取得平均值为视频延时。单位：ms			
352*288	CBR=400 I/P 帧间隔：16	VBR=8 I/P 帧间隔：32	CBR=1200 I/P 帧间隔：32	
	176	208	317	233.67
704*576	CBR=1200 I/P 帧间隔：32	VBR=8 I/P 帧间隔：32	CBR=1500 I/P 帧间隔：32	
	336	298	308	314
NTSC	帧率：1(30 帧/秒)，流类型：video only，更改 VBR，CBR 值取得平均值为视频延时。单位：ms			
320*240	CBR=400 I/P 帧间隔：16	VBR=8 I/P 帧间隔：32	CBR=1200 I/P 帧间隔：32	
	151	121	123	131.67
640*480	CBR=1200 I/P 帧间隔：32	VBR=8 I/P 帧间隔：32	CBR=1500 I/P 帧间隔：32	
	189	346	770	435

从测试结果可以看出，网络视频监控终端压缩后的数据能在局域网内以小于 500ms 的延迟传输，肉眼感觉不到延迟与抖动的情况。

8.3 本章小结

本章主要介绍嵌入式网络监视系统的调试，从衡量嵌入式监控系统的性能参数进行测试，最后给出相应测试环境的测试数据，从数据结果可以看出本系统的软硬件设计都达到了预期要求。

9 总结

9.1 本文完成的任务

本文的主要工作是基于 DM355 的嵌入式视频网络监控系统的总体设计。完成了以下几项工作：

1. 分析了视频监控系统的现状，并针对青岛市科技局项目城市污染源远程监控系统的具体要求改进了以前的设计方案，提出基于 DM355 的视频监控方案。
2. 完成硬件的底层板设计。
3. 实现了嵌入式 Linux 操作系统移植到 DM355 硬件平台的全过程以及嵌入式开发环境的搭建，编译 UBOOT 内核烧写 NAND flash 和 UBOOT.
4. 设计了视频采集，压缩和传输程序模块，设计了客户端程序。程序设计多线程技术并在视频采集和网络传输过程运用线程交互和共享内存技术，进行了防止进程死锁的设计，提高了系统运行效率。
5. 完成了系统的调试与测试工作，给出测试数据。

9.2 今后的研究方向

本嵌入式网络视频监控系统实现了视频采集、视频压缩，网络传输、视频解码、视频播放等功能，系统功能和稳定性达到了预期设计目标。但由于受开发周期短，成本控制等影响，系统稳定性和处理速度应该还可以提高。

1. 核心板加底层板的复合结构，虽然节省了成本，但也给设计带来一些困难。例如走线不是很方便，这样会增加信号间的干扰。
2. 采用的协处理器，算法只能固定在 MPEG4， 不能灵活变换，而且现在使用效果最好的视频压缩算法是 H.264。所以在以后如果条件允许会采用 DM6446 等高端的 DSP 处理器，来处理视频图像，但相对的也会增加算法的开发成本。
3. 对于如果没有网络的环境，就需要无线视频传输功能。由于现在无线网络受带宽影响，传输图像质量不能太高。现在已有的无线视频监控基本都是基于 CDMA 和 GRPS 网络。随着 3G 网络的不断发展清晰的无线视频监控将会得到更好的发展。

因此以后将会进行基于 3G 网络的无线视频监控系统研究，并且丰富视频压缩格式。对与软件的开发坚强管理，引入软件开发管理系统对于多人开发的软件进行统一管理。

参考文献

[1] 刘富强，卢赤班.数字视频监控系统及其应用[J]，工矿自动化，2003年第3期:31-33.

[2] 宋磊，方向忠.达芬奇技术的视频应用分析[J]，电视技术，2006年第9期:31-33.

[3] 张洁，何晓燕.基于ARM技术的远程图像监控系统设计[J]，自动化仪表，2006年第27卷第11期:5-10.

[4] 白木，周洁.数字视频监控系统[J]，广播电视信息，2002年第05期:35-40.

[5] 张跃进，谢昕.嵌入式网络数字视频监控系统的设计[J]，计算机工程与设计，2009年04期:805-807.

[6] 吴健新，李翠华.数字视频监控系统开发平台的设计与实现[J]，厦门大学学报：自然科学版，2006年45卷3期：352-354

[7] 刘欣，杨雪鹏，方加宝等.基于Internet的嵌入式监控系统设计[J].自动化技术与应用，2005，12:15-17.

[8] 马昕.视频监控系统的现状和今后发展趋势[J]，金卡工程，2005年总第3期:71-73.

[9] 张大波等.嵌入式系统原理[M].北京:机械工业出版社，2005.

[10] 辛建光等.嵌入式Web视频服务器的设计[J]，兵工自动化，2005，24(1):67-71.

[11] 何鹏举，陈明等.基于嵌入式Web服务器的远程视频监控系统[J]，测控技术，2004年，23卷第6期:62-63.

[12] 邹宪民，温惠英.数字监控系统在技防工程中的应用[J]，交通与计算机，2001年第6期:32-35.

[13] 胡国兵.嵌入式系统综述[J].南通职业大学学报.2004，18(4):29-32.

[14] 孙建恒.嵌入式系统应用研究及实例[J].微计算机信息.2004，20(6):65-66.

[15] 毛德操，胡希明.嵌入式系统采用公开源代码和StrongARM/XScale处理器[M].杭州:浙江大学出版社，2003:335-365.

[16] 朱珍民， 隋雪青.嵌入式实时操作系统及其应用开发[M].北京:北京邮电出版社，2006.

[17] Craig Hollabaugh.嵌入式Linux[M]，北京：电子工业出版社，2003.

[18][美] WayneWolf.嵌入式计算系统设计原理[M].北京:机械工业出版社，2002.229-240.

[19] 张克非 ，嵌入式实时操作系统分析[J]，计算机工程与设计，2005年第26卷第8期:26～29.

[20] 韩立宏，嵌入式实时操作系统性能测试方法[J]，指挥控制与仿真，2008年02期:98-101.

[21] 林福宗.多媒体技术基础[M].北京:清华大学出版社，2001，7:167-184.

[22] 余兆明，李晓飞，陈米春.MPEG标准及其应用[M].北京:北京邮电大学出版社，2002:251-267.

[23] 李树前.基于MPEG-4的网络视频监控技术研究与系统实现[D].南京航空航天大学，2007.

[24] Iain E.G.Richardson.H.264 and MPEG-4 Video Compression:Video Coding for Next-generation Multimedia.1st edition.USA:John Wiley&Sons，2003.85-98.

[25] Wiegand，T.Sullivan，G.J.Bjntegaard，G.et al.Overview of the H.264/AVC video coding standard.Circuits and Systems for Video Technology，IEEE Transactions on，2003，13(7):560-576.

[26] Min-Chi Tsai， Tian-Sheuan Chang， High Performance Context Adaptive Variable Length Coding Encoder for MPEG-4 AVC/H.264 Video Coding， Circuits and Systems， 2006. APCCAS 2006. IEEE Asia Pacific Conference on， 4-7 Dec.2006:586-589.

[27] Douglas E. Comer，David L.Stevens，TCP/IP网络互联技术[M].北京:清华大学出版社，2004.

[28] M.Tim Jones.TCP/IP Application Layer Protacols for Embedded systems.CHARLES RIVER MEDIA，2002.

[29] 章民融，徐亚锋，尤晋元等.RTP/RTCP协议在视频监控系统中的实现[J]，计算机应用与软件，2006年01期:79-81.

[30] 艾书华.基于嵌入式视频监控系统的IP组播技术研究[J]，计算机与现代化，2006，1:85-88.

[31] 肖政宏，韩秋凤，朱丽群.基于ARM和DSP的远程视频监控系统的设计与实现[J]，计算机工程与科学，2006，9，53-55.

[32] 陈耀武，伍鹏，汪乐宇.基于流媒体技术的网络化嵌入式视频监控系统[J]，工程设计学报，2004，11(2):57-61.

[33] 刘洁瑜，王黎明，钱培贤.基于MPEG2000嵌入式网络视频采集压缩系统的设计与实现[J].微电子学与计算机，2004，21(2):168-173.

[34] 卢少同.嵌入式网络视频监控系统设计[D].青岛科技大学，2008.

[35] 程德强，钱建生.数字视频监控服务器及其关键技术研究[J]，煤炭科学技术，2004年10期:43-46.

[36] Changwoo Jee，Shin，K.G.A DAVIC Video-on-Demand system based on the RTSP.Applications and the Internet，2001.Proceedings，2001，1:231-238.

[37] 顾海州，马双武.PCB电磁兼容技术——设计实践[M].北京:清华大学出版社，2004.40-62.

[38] 陈怡秋，陈晓光. 基于UDP的无线自相似环境下MPEG-4码率自适应控制[J]. 通信学报，2005，(05):17-23.

[39] 蒋建国，苏兆品，李援等. RTP/RTCP自适应流量控制算法[J]. 电子学报2006年09期:1659-1662.

[40] 张贤达. 现代信号处理(Modern Signal Processing, Second Edition)[M]. 北京：清华大学出版社，2003.

致谢

首先向我的导师刘云教授表示衷心的感谢和崇高的敬意。本论文是在导师刘云教授的悉心指导下完成的，刘老师在课题的选定、课题的指导、论文的撰写、经费的保证、实验条件的提供等方面给予了全面支持。在三年的研究生求学生涯中，刘老师的严格要求和悉心培养使我受益匪浅，不仅使我的理论和实践水平获得了较大的提高，而且在思想和认识方面的综合素质都获得了较大的进步。刘老师治学严谨、学识渊博，在学术方面一丝不苟，对科学具有敏锐的洞察力，这些都将成为我以后工作和学习的榜样。还要感谢刘老师为我们创造了良好的科研环境，让我有机会参与多个科研课题的研究开发，在科研水平得到提高的同时，也拥有了未来更多的发展机会和更广阔的发展空间。

感谢图像处理研究室的王传绪老师、张祥光老师、王剑锋老师，感谢他们在平时繁重的教学和工作之余，对我在学习、科研及生活上给予关心和指导。

感谢图像处理研究室甘志杰、张燕、卢少同、李作永、孙玉、李雪、孟岩、张鹏、孙芳同学的帮助，在他们的帮助下，课题得以圆满完成。

感谢我的家人，一直以来，他们无私的支持是我前进途中不可缺少的动力，感谢他们为我所做的一切，我会继续努力。

感谢所有曾经帮助过我的老师和同学。

希望所有人在今后的人生路上取得更大的成就。

攻读学位期间发表的学术论文目录

(1) 刘继超，刘云.基于NI采集卡的虚拟示波器的开发实现[J].《青岛科技大学学报》,2008年第29卷第4期。

(2) 刘继超，孙玉. Frequency estimation with sub-nyquist sampling based on virtual instrument[J].《全国第十五届信息论学术年会暨全国首届网络编码学术研讨会》上册,2008年7月

基于DM355的嵌入式网络视频监控系统设计

作者： 刘继超
学位授予单位： 青岛科技大学

相似文献(10条)

1.学位论文 柯飞雄 基于ARM9网络视频监控系统的设计与实现 2008

随着信息技术的飞速发展和网络技术的日益成熟，通过网络实现远程监控是视频采集技术的一个发展趋势。本文针对网络视频监控设备的实际应用需求，结合图像采集、动态Web技术和嵌入式系统三方面的新技术，设计了基于ARM9和Linux嵌入式网络视频监控系统。

系统的硬件方面主要以三星公司生产的ARM9处理器(S3C2410)为核心的嵌入式系统，并结合OV511芯片的USB摄像头。软件方面采用嵌入式Linux操作系统进行视频采集、压缩并通过ActiveX控件显示动态视频，通过嵌入式Web Server和CGI技术融合整个系统。最后将系统连接到公网，利用远程网络的PC机上Web浏览器登陆视频服务器对系统进行测试，并对测试结果进行了分析和比较。

本文作者成功利用了USB摄像头作为视频监控系统的终端，USB摄像头和嵌入式开发板的组合更加可以方便的实现中间数据的处理、保存和查询等功能。同时将动态Web技术应用到嵌入式网络监控系统中，利用Web浏览器实现了用户和嵌入式系统的交互。相关的工作为视频监控领域的研究提供了有价值的参考。

2.学位论文 李威 嵌入式网络视频监控系统的硬件设计与实现 2009

视频监控系统经历了第一代的本地模拟视频监控系统、第二代的基于PC机的远程视频监控系统，到现在的第三代基于嵌入式技术的网络视频监控系统。在国内，基于嵌入式技术的网络视频监控系统的研究才刚刚起步，还无法掌握核心芯片和嵌入式开发的关键技术，尚不能完全独立开发专业的硬件平台，产品性能稳定性、可靠性等方面还有待进步提高。因此通过对嵌入式网络视频监控系统作深入研究，开发具有自主知识产权的适应监控发展的网络化数字视频监控产品，对我国监控行业的发展具有很大的促进作用。

为了建立具有自主知识产权的第三代网络视频监控系统，本论文介绍了国内外视频监控系统的发展现状。通过比较第三代网络视频监控系统与第二代网络视频监控系统的异同，指出第三代网络视频监控系统的优缺点。接着详细阐述了硬件部分的系统实现，包括：配置电源电路，把SDRAM和FLASH连接到FIC8120，配置数模转换芯片SAA7113的外部电路，配置实时时钟Real time clock的外部电路和配置本地存储SD卡外部电路。通过对视频监控系统的研究，以集成了ARM9和MPEG4实时视频编解码的SoC芯片FIC8120为核心设计和实现了嵌入式网络视频监控。

第三代视频监控系统具有体积小，成本低，稳定性高等优点，为使携型视频监控系统提供了很好的解决方案。

3.期刊论文 罗青.易理告.LUO QING.YI LIGAO 嵌入式在小型网络视频服务器中的应用 -微计算机信息2008,24(8)

本文介绍了视频监控系统的发展现状及趋势,给出了一种基于嵌入式系统的小型网络视频服务器硬件实现:包括该网络视频监控系统的实现原理,以及各个功能模块的硬件设计.

4.学位论文 张文涯 基于嵌入式Linuz的网络视频监控系统设计与实现 2009

随着计算机网络技术、视频压缩技术以及嵌入式技术在近些年来的迅速发展，网络视频监控系统进入快速发展期，市场上出现了很多网络视频监控产品，以满足人们对安全的各种要求。采用嵌入式Web技术的网络视频监控系统，目前正成为网络视频监控系统的一个重要发展方向。用户通过浏览器或其他专门的客户端软件，就可远程访问监控摄像机，实现对现场的远程视频监控。

本课题主要研究内容是基于ARM9平台的嵌入式Linux网络视频监控系统的设计与实现，着重研究了网络视频监控系统中的视频采集传输和客户端GUI设计。在系统设计中采用B/S结构与专用客户端软件相结合，将嵌入式视频采集终端的功能模块和PC机客户端监控功能模块相分离，采用RTP/RTCP协议对视频数据进行网络传输。在嵌入式视频采集终端，对目前Linux下USB摄像头驱动的两种接口标准Video for-Linux和Video for Linux Two分别进行了研究，分别实现了基于这两种标准的驱动下的图像采集，并在最终的系统中采用了基于Video for Linux标准的驱动程序，完成了图像采集功能。考虑到硬件平台本身的制约和降低成本，对视频采集的数据采用软件压缩，将每一帧图像压缩成JPEG格式。通过在嵌入式视频采集终端上建立嵌入式Web服务器，使用户通过Web页面控制图像的采集传输工作。在PC机上采用Linux操作系统，使用Qt设计开发客户端视频监控软件，该软件可实现对多个数据源的监控，并可以对监控的图像数据进行抓图、AVI格式的视频文件保存、制定视频保存计划、相关信息的数据库管理功能，并可以通过RTP/RTCP协议与局域网内其他客户端软件进行视频共享。

5.学位论文 孙延均 基于BF533的网络视频监控系统设计与实现 2009

传统视频监控系统具有成本较高，需要铺设专用线路，无法联网，需要耗费大量的存储介质等缺点。随着视频处理和网络技术的日益成熟，网络视频监控系统被越来越多地应用到生产生活中，它使人们能够及时地获取被监控现场的环境状况，做出正确的决策。本文设计与实现了基于ADSP-BF533处理器的嵌入式网络视频监控系统，它具有成本低廉，安装灵活，功能丰富等特点。

通过分析网络视频监控系统三种解决方案的优缺点，综合考虑了成本、开发难度等因素，最终选择了BF533嵌入式DSP处理器作为本系统的基础平台架构，并在此基础之上对硬件电路所需的各种外围芯片和系统软件进行了选型。

本文对BF533处理器、SDRAM、FLASH、视频解码器、视频编码器、音频编解码器、以太网控制器、RS232驱动器/接收器、CPLD以及电源模块的设计进行了详细的描述，包括各模块之间的相互连接情况，设计中需要注意的事项等。

在将U-BOOT移植到特定开发板之前，需要先熟悉其源代码，本文依照U-BOOT的启动流程，对其源代码进行了比较深入的分析。编写了DM9000AE和SST39VF1601驱动程序，给出了U-BOOT移植的具体步骤。

为了检验系统硬件是否能正常工作，本文讨论了系统硬件各模块的测试方法，编写了具体的测试例程，给出了相应的测试结果。说明了编译U-BOOT的方法，以及设置U-BOOT从网络加载μClinux的方法。

6.学位论文 张贝 基于嵌入式系统的网络视频监控系统前端研究 2005

本文通过分析视频监控系统从模拟时代向嵌入式数字网络时代的发展，提出了应用DSP芯片进行视频图像压缩然后通过网络传送的解决方案。

采用数字视频技术虽然具有许多优越性，但也存在数据量大的缺点，如果不对数据进行有效的压缩，那么将不能在现有的网络条件下进行传输。本文对应用DSP进行JPEG图像压缩做了详细的论述。

论文共分为四章。第一章是绪论，对基于嵌入式系统的网络视频监控系统作了简单的介绍并给出了系统的构成情况。第二章讨论了图像压缩编码技术并分析了DSP做图像压缩的可行性。第三章对系统设计中所涉及的主要硬件单元做了详细的介绍。第四章是系统的软件分析单元，研究了JPEG压缩算法，其中详细分析了编码系统的结构和编码的各个步骤，给出了系统主程序的开发流程并给出了系统图像压缩的仿真结果。

7.期刊论文 汪庆年.李桂玲.元美玲 基于S3C2410网络视频监控系统的设计与实现 -安防科技2008,""(1)

本文介绍了一种基于S3C2410的嵌入式网络视频监控系统的设计方案.系统中以嵌入式Linux为操作系统,采取MPEG-4专用压缩编码芯片MPG440对采集到的数字视频信号进行压缩编码,生成MPEG-4视频码流,视频码流通过S3C2410主控制器外接的网络控制芯片传输到PC机,再通过内嵌的MPEG-4解压插件的IE浏览器来播放视频数据以及控制摄像机的监控状态.

8.学位论文 方元武 基于FIC8120的网络视频监控系统的设计与实现 2008

视频监控系统在家庭安防、工业生产和日常生活中已经得到了广泛应用,并发挥着重要的作用。随着视频压缩技术、流媒体技术、嵌入式技术和网络技术的发展,前端一体化、视频数据数字化、监控网络化、系统集成化已经成为视频监控系统公认的发展方向。

视频监控系统经历了三代的发展,第一代是本地模拟视频监控、第二代是基于PC的远程视频监控,第三代是基于嵌入式技术的网络视频监控。现在应用很成熟的主要是基于PC+视频采集卡的视频监控系统。由于PC机无法长期稳定工作,而且需要高昂的费用来购买PC机和视频采集卡,同时需要复杂的电缆

连接,因此本文在分析了视频监控系统的历史和发展现状的基础上,提出了基于高集成嵌入式SoC芯片的网络视频监控系统设计方案。与其它视频监控系统相比,基于高集成嵌入式SoC芯片的网络视频监控系统具有体积小、成本低、稳定性高、开发周期短、扩展性强等优点。

本文主要通过对视频压缩技术MPEG-4和嵌入式操作系统的研究,以集成了ARM9和MPEG4/JPEG实时视频编解码硬件引擎的SoC芯片FIC8120为核心设计和实现了嵌入式网络视频监控系统。该系统使用视频硬编解码芯片,简化了视频压缩算法的开发时间,并通过移植FFMPEG开放源代码项目到嵌入式系统中,利用SoC芯片上的MPEG4/JPEG实时视频编解码硬件引擎API函数来取代FFMPEG项目上原有的视频编解码算法,克服了FFMPEG不具有视频监控系统所需实时性的缺点,又利用了FFMPEG上的流媒体协议来实现视频流在网络上的实时播放。同时还移植了嵌入式Web服务器到嵌入式系统上,从而方便了对嵌入式网络视频监控系统的管理和配置。

该系统具有体积小、成本低、稳定性高、系统可扩展性强等优点,为基于网络的远程嵌入式视频监控系统提供了很好的解决方案。

9.学位论文 郑庆宁 基于DSP的嵌入式网络视频监控系统的研究及硬件设计 2007

嵌入式网络视频监控系统作为新一代的视频监控设备，集先进的视频压缩处理和以太网通信于一体。通过Internet，远程用户可以在任意时间、任意地点访问监控系统，进行实时视音频监控，接收报警信息，同时还可以对摄像机进行控制操作，对图像的质量进行调节。系统以嵌入式系统技术为基础，具有高度的稳定性和可靠性，在道路交通、银行、电信、智能家居等众多领域具有广泛的应用前景。

本文在深入进行功能需求分析的基础上，采用TI公司的高性能多媒体处理器TMS320DM64x为核心进行系统设计。该处理器具有强大的处理能力和丰富的外围接口，是嵌入式网络视频监控系统的理想解决方案。

本文首先介绍了视频监控系统的发展历史，分析了嵌入式网络视频监控系统的特点和关键技术；在仔细研究了嵌入式系统相关技术的基础上，分析比较了系统的解决方案并提出了本系统的设计方案；然后，对系统的主要功能模块，包括视频接口、音频接口、以太网通信、存储器扩展、RS485通信与报警输入、I<'2>C总线控制、启动与供电等，进行了详细的分析和设计；在完成了系统原理图设计后，针对高速系统设计中的信号完整性问题进行了深入的研究，采用HypedJynx工具和IBIS模型对本系统的关键部分进行了分析和仿真，并结合仿真进行了PCB的设计，从而构建了可靠的系统硬件平台；最后，对系统的软件开发环境和代码开发流程进行了介绍，并简要分析了软件功能模块的设计。

10.期刊论文 汪庆年.李桂勇.元美玲.WANG QINGNIANG.LI GUIYONG.YUAN MEILING 基于S3C2410网络视频监控系统的设计与实现 -微计算机信息2007,23(35)

本文介绍了一种基于S3C2410的嵌入式网络视频监控系统的设计方案.系统中以嵌入式Linux为操作系统,采取MPEG-4专用压缩编码芯片MPG440对采集到的数字视频信号进行压缩编码,生成MPEG-4视频码流,视频码流通过S3C2410主控制器外接的网络控制芯片传输到PC机,再通过内嵌的MPEG-4解压插件的IE浏览器来播放视频数据以及控制摄像机的监控状态.