摘　要

随着当前电子设备技术向数字化、网络化和智能化方向的发展，监控设备也在不断地更新换代。在集中了多媒体技术、数字图像处理及远程网络传输等最新技术之后，数字化网络化监控系统正在逐步取代传统的模拟监控系统，代表了监控系统的发展潮流。而数字网络监控设备的经济效益和社会效益巨大，正在被越来越多的石油、勘探等其它部门所使用。迄今为止，它已经发展到了第三代，统称为IP-CAMERA（网络摄像机）。相比前两代而言，数字化使得采集到的数据易于存储和传输，网络化使得监控区域不受物理连接的环境所限制，智能化使得系统易于分析及操作。

但是在图像清晰度方面，面对恶劣的环境时，当前已经存在的低等价位的监控设备仍然无法满足日益增长的高清需求，所以本论文在低成本的基础上，更加着重于对 IP-CAMERA 的高清方面的设计与实现。本论文在介绍 IP-CAMERA 相关的构架及关键技术的基础上,在 YLP2440 开发板的相关平台上,设计了高清 IP-CAMERA 的视频数据传输模块，实现了数据的网络传输，实验表明达到了预期的实验结果。文中以 YLP2440 开发板为例，仔细分析了高清 IP-CAMERA 作为一个嵌入式 Linux 系统所具备的功能及特点，在设计上采用了主机/目标机的开发模式，在服务器端实现了服务器/客户端的网络传输模式，在客户端实现了基本的 Visual C++响应界面。其中运用的设计方法及技术手段等在技术上是可行的，对高清 IP-CAMERA 的研究也有一定的利用价值。

关键词：

高清 IP-CAMERA；嵌入式系统；Linux；数据传输；套接字

Abstract

With the the reform and development of the digitization, networking and intelligent direction in the electronic devices area in current society, the monitoring equipment are also constantly updating. After increasing the latest technology of the multimedia, digital image processing and remote network, the digital network surveillance system is gradually replacing the traditional analog surveillance system, which represents the development trend of monitoring system. It means huge economic and social benefits, which is being used in the oil, prospect and many other areas. So far, it has developed to the third generation. Such devices are commonly referred as IP-CAMERA. Compared to the previous two generations, its digital feature brings easier storage and transmission ways. Its network feature brings no physical connection which would be limited by the environment any more. Its intelligent feature brings easier analyse process and operation.

But in the bad environment, the resolution of the monitoring devices doesn't meet the increased requirements any longer currently. That's what this paper attentions most. In this paper, the related architecture and key technologies of IP-CAMERA is introduced, and then a video data transmission module of the high-resolution IP-CAMERA is designed, and finally the experimental results are showed successfully. The text uses YLP2440 development board as an example. And it analyses the functions and characteristics of high-resolution IP-CAMERA as an embedded Linux system carefully. It adopts the host / target developing model in the whole, a server / client mode in the server side to achieve of the network transfer, and the basic Visual C ++ interface in the client side to response. In conclusion, the design methods and technical means in this paper have certain degree to the technical feasibility study, and also have a certain value to the high-resolution IP-CAMERA.

Key words:

High-resolution IP-CAMERA ;Embedded system ;Linux ;Data transmission ;Socket

独创性声明

本人声明所呈交的学位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，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本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和汇编本学位论文。

本论文属于　保密□，　在_____年解密后适用本授权书。

　　不保密□。

（请在以上方框内打“√”）

学位论文作者签名：　　指导教师签名：

日期：　　年　月　日　　日期：　　年　月　日

1 绪论

1.1 IP-CAMERA 监控技术的发展

随着数字化和网络化时代的到来，越来越多的设备都在进行数字化和网络化的革命。监控设备在经过这些技术的洗礼之后，到现在总共经历了三代的发展。第一代是模拟时代。此时的模拟监控设备一般由模拟摄像头和采集压缩卡组成，它传输和存储的都是模拟的数据，且它的采用还受到距离的限制，因为它的连接是通过物理线来连接的，而且只能在固定的地方进行监控。最原始的 CCTV 采用的设备 VCR（Video Cassette Recorder，磁带录像机）就是这种应用。第二代是半数字化半网络化时代，其传输的数据已经转化为了数字数据，而且其存储设备也已转化为硬盘之类可存储大量数据的装置，也开始加入了可扩展为网络存储的网络接口。其中有代表性的有 DVR（digital video recorder，数字视频录像机）和 NVR（Network digital video recorder，网络数字视频录像机）[1]。第三代是网络化时代，其网络化功能十分强，统称为 IP-CAMERA（网络摄像机）。数字化使得采集到的数据易于存储和传输，网络化使得监控区域不受物理连接的环境所限制，智能化使得系统易于分析及操作。IP-CAMERA 监控系统从一开始就是针对在网络环境下使用而设计的，因此它克服了前面两代监控系统无法直接通过强大的因特网来获取数字视频信息的缺点。于是，用户可以通过连入网络中的任何一台电脑来监控实时的视频信息[2]。

所谓 IP-CAMERA，泛指运用因特网来进行远距离传输检测信息的监控设备。它是一个单独工作的系统，即并不需要通用计算机来进行辅助工作，只要连接好了网络，就可以随时随地对任意监控点进行监控。数字化、网络化、智能化是目前 IP-CAMERA 监控系统的三大发展方向，而数字化是网络化的前提，网络化又是智能化的基础。所以，IP-CAMERA 监控发展的最大两个特点就是数字化和网络化[2]。数字化可以使得数据易于存储和传输，而且当前数字化技术已经很成熟了。而网络化在很大程度上简易地扩展了可监控区域，且几乎共享了整个网络系统的硬件和软件资源[3]。随着当前因特网的迅速发展，正如上面所说的，将网络运用于监控行业

的技术已经得到了广泛的应用。当前市场中，单个摄像头的数字网络监控技术已经颇于成熟，但含有多个摄像头、多个监控点、高清等特点的监控器还在不断的研究与更新中。本文基于高清这一特点，对运用因特网的监控设备的系统结构及其中运用的关键技术进行了深入的研究，并且对其传输模块进行了设计与实现。

1.2 基于高清技术的 IP-CAMERA 监控系统现况

大家都知道，在数字视频显示中，分辨率为 1280*720 的 720p 标准和 1920*1080 的 1080p 标准均是准高清标准。其中的“ p ”表示每行扫描时，分别有 720 和 1080 条可见扫描线。据研究，当前还只有 60 英寸以上的数字电视显示屏可以实现这个分辨率，且现在已把这个标准定义为目前数字电视的顶级显示模式[4]。基于高清的考虑，高清 IP-CAMERA 监控系统提供了视频监控的数字信源，对信源数据进行了高清编码和封装，通过以太网或因特网来高效传输，从而实现高清的特点。

现今在国内外市场上，主要推出的是数字控制的模拟视频监控和数字视频监控两种产品，而今正处在将这两种系统混合应用后的数字系统的阶段。目前市场上的数字视频监控系统分为两种，一种是以数字录像设备为主要部分的，另一种是以嵌入式视频 Web 服务器为主要部分的。前一种采用了数字化技术来处理数据，使得数据便于存储和传输，同时采用了大容量的存储介质，使得一次性存储的数据大大地增加。后者是专用性比较高的嵌入式系统，体积小，可以实现即插即用，使用起来特别方便，而且稳定性很好，不需要人进行坚守管理。另外也可以直接连接到标准的以太网，利用网络来传输数据，消除了传输距离的概念[5]。当前的监控设备已经完全实现了数字化的变革，而且网络化也已经开始融入到了应用中，但总体来讲，在清晰度要求比较高的场合中，即使是第三代的 IP-CAMERA 监控系统的解决方案也仍有待进一步地完善。

总体来说，高清 IP-CAMERA 监控系统具有下列特点：高清晰度的影像、高效率影像压缩、直接连接标准以太网络、远程访问和控制能力、灵活扩展功能和分布式网络结构。高清 IP-CAMERA 监控系统融合了传统视频技术和先进的 IT 技术，这些技术都是经过了长时间的实践检验，并在今后很长一段时间内保持行业领先的

地位，这一切都确保高清 IP-CAMERA 监控系统的技术先进性，为创造巨大的经济价值和社会价值带来澎湃的动力。

1.3 论文的主要研究工作

本文的研究内容主要就是介绍高清 IP-CAMERA 系统，尤其是其服务器端的各项基本特征及相关知识点，且设计并实现基本的高速数据传输模块。本论文中所研究的高清 IP-CAMERA 监控系统将选用 CMOS 图像采集器、高效率压缩编码芯片、扩展的高速传输网络接口等相关技术。服务器端的微处理器采用嵌入式 Linux 操作系统，客户端采用 Microsoft Visual C++ 6.0 的界面模式进行接收。前几章分析基础知识，后面再设计并实现数据传输模块：第一章，描述 IP-CAMERA 及其监控技术的发展概况；第二章，IP-CAMERA 服务器端所应用到的技术要点，如图像的采集、数据的压缩编码、嵌入式系统的开发模式，服务器/客户端的通讯模式，多线程编程模式等；第三章，总体描述 IP-CAMERA 网络结构及服务器端的软硬件架构；第四章，基于深圳市优龙科技有限公司推出的 YLP2440 开发板，选择相应的外加软硬件，来进行视频数据传输模块的设计；第五章，高清 IP-CAMERA 视频数据传输系统的实现结果的展示，即建立好服务器端及客户端的发送接收模块后，连通网络，进入服务器端超级终端，运行传输模块的应用程序，在客户端操作界面进行相应操作；第六章，对此系统的研究与实现进行总结，分析其优缺点及可改进性，展望今后它的发展。总体说来，本论文以研究为主，设计实现为辅，从基础上对嵌入式高清 IP-CAMERA 进行了较深入的了解。

2 IP-CAMERA 服务器端的关键技术

2.1 图像的采集

在 IP-CAMERA 服务器端，监控的来源是通过图像采集器采集数据后得到的，之后这些数据经过压缩编码发送到客户端。可见，如果这个源头的图像都不够清晰的话，那后面客户端收到的图像就更不够清晰了。所以，图像的采集是 IP-CAMERA 服务器端的首要关键技术。

视频采集，实际上就是将传感器采集到的视频信号转换成 PC 机可以识别的数字格式[6]。IP-CAMERA 硬件模块的第一部分是图像采集器，作为视频的来源依据，它占据着极其重要的位置。这类采集器一般有两种：CCD 和 CMOS。而相比起来，前者的成像质量较好，只是制造成本较高。在相同分辨率下，CMOS 的价格比 CCD 要便宜得多，且相对于 CCD 而言，其耗电量小，体积也小。目前，新的 CMOS 器件品种逐渐开始接近 CCD 的成像质量，已经开始有逐渐取代 CCD 的趋势，并有希望在不久的将来成为主流的感光器[7]。

作为视频来源的 CMOS 摄像头，本论文中设计的系统采用的是 CMOS 型的感光芯片，最大能够输出高清 1080p（1920*1080）图像阵列，其最大可以达到 30fps 的输出量，能够满足高清 IP-Camera 实时监控的要求。

2.2 数据的压缩编码

从采集器采集到的数据经过压缩编码后再传输的话，可以将传输的数据量大大减少，易于传输，且可保证数据的准确性。可见，数据的压缩编码模块也是高清 IP-CAMERA 的一项关键技术。在接收端解码恢复后，需要得到尽量和原来压缩前一样视觉效果的数据，可以采用标准化的压缩方式。压缩一般可分为两种：1.无损压缩，即压缩后能够完全恢复，它是可逆的；2.有损压缩，即压缩后不能够完全恢复，这一过程是不可逆的[8]。

虽然一般压缩时会希望进行无损压缩，但无损压缩的压缩比一般都不够高。本

文中设计的系统选用的是 JPEG 压缩标准。尽管 JPEG 压缩是一种有损压缩，但是其损失的部分只是计算机色彩中的高频信息部分，这一部分是人眼不敏感的部分。这样的话，在减少了数据信息的情形下，却不影响人们对它视觉上的效果。在人眼的观察下，同样可以实现高清的目标。JPEG 在压缩编码时，除了去除视觉上的多余信息即空间冗余度之外，还对数据本身的多余信息即结构（静态）冗余度进行了清除，从而极大地减少了需要处理的数据量，整体上达到数据压缩的目的。而压缩的实质就是采用另一种方式来表达原来的数据，即对数据所表示的信息量进行重新编码，故压缩和编码实质上是分不开的[8]。本论文设计的系统中的图像格式转换部分实质上是将 RGB 转换成 YUV。图像压缩编码模块是将从采集器模块传输过来的原始图像数据（YUV 采用的 4:2:2）进行 JPEG 压缩编码，生成标准 JPEG 格式的数据。由于从采集器获得的原始数据量很大，不适于通过网络实时传输，本系统采用压缩比 1：40 以上的高效压缩算法对图像信息进行压缩。由于采用硬件编码和压缩技术，大大提高了编码效率[9]。

另外，MPEG-4 和 H.264 压缩编码是表示视频信息的最新标准。虽然这两种标准都涉及到了对视频数据的压缩，但是前者更注重于灵活性，而后者更注重于有效性和可靠性。MPEG-4 的灵活性主要表现在它能够处理多种类型的视频数据，如可以处理静止的图像、混合的图像及合成的图像等。而 H.264 的有效性和可靠性主要表现在优秀的编码效率性能和传输性能上[10]。

2.3 嵌入式 Linux 系统的开发

2.3.1 嵌入式 Linux 系统

IP-CAMERA 系统的服务器端是一个嵌入式系统，而且本文中设计的系统选用的操作系统是 Linux 操作系统。所谓一个嵌入式系统（Embedded System），是指能够在特定环境下独立完成某些特定功能的一个计算机硬件和软件的集合体。而说一个系统是一个嵌入式 Linux 系统，是说它是一个基于 Linux 内核的嵌入式系统。即是说，嵌入式 Linux 系统是利用 Linux 操作系统自身的许多特点，把它应用到嵌入式系统里而形成的[11]。

嵌入式系统是嵌入式计算机系统的简称，简单地说，嵌入式系统就是嵌入到目标体系中的专用计算机系统。具体地讲，嵌入式系统是指以应用为中心，以计算机技术为基础，并且软硬件可裁剪，适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。但是嵌入式系统与嵌入式设备却并不是一样的，后者比前者的范围更为广泛些，可以说嵌入式设备是内部包含有嵌入式系统的设备。比如手机就是一个嵌入式设备，而手机内部的单片机芯片才是一个嵌入式系统。随着计算机的不断微型化、通用化、嵌入式化的发展趋势，可以猜想到嵌入式系统将成为后 PC 时代的主宰。

嵌入式系统的 3 个基本要素是嵌入性、专用性与计算机系统性。嵌入式系统是一种应用系统，它至少包含一个基于可编程的微控制器、微处理器或数字信号处理芯片的微型计算机，却让使用该系统的人一般意识不到该系统是基于计算机的。比如说日常使用的微波炉就是一个嵌入式系统，却几乎没有人会觉得它是基于计算机来进行工作的。因为微波炉就是面向给放入其中的物品进行加热这个特定任务而设计的，专用性很强。这样的话，形成的嵌入式系统就具有多样性了。不同的嵌入式系统运用于不同的环境中，更专业更高效地完成相应的任务。环顾四周，每当看到内含微处理器的设备，多半会发现它又是一个嵌入式系统。比如常见的有商店的自动柜员机、MP3 播放器和打印机等[12]。

其实，Linux 原本与嵌入式系统毫无关系，如今却成为大家优选的操作系统。用 Linux 作为嵌入式系统的操作系统，有许多方面的优势。首先，Linux 的源代码是自由开发的，是完全公开的，也是完全免费的，可以直接从网络上下载到。其次，Linux 与 Internet 是同步发展起来的，涵盖了 UNIX 的所有特征的同时，还融合了许多其他操作系统的功能[13]。再次，Linux 操作系统具有模块化与结构化、容易修改、可扩充、可配置、可预测、错误恢复能力、长期运行能力的特性。另外，Linux 是一个多用户、多任务的操作系统，支持多种文件系统，符合 POSIX 1003.1 标准（最小 UNIX 操作系统接口的标准定义），具有较好的可移植性，支持多平台和多处理器，全面支持 TCP/IP 网络协议。总体来说，Linux 简短精悍，适应于多种 CPU 和多种硬件平台，是一个跨平台的系统。而且性能稳定，裁剪性很好，开发和使用都

很容易。且 Linux 内核的结构在网络方面是非常完整的，Linux 对网络中最常用的 TCP/IP 协议具有最完备的支持。因此，在本文中的嵌入式系统中使用 Linux 操作系统是头等选择。

2.3.2 主机/目标板开发模式

嵌入式系统的开发如果只是局限于嵌入式系统本身，那么它的难度就很大了。毕竟当前 PC 机中最常用的操作系统并不是 Linux，于是就产生了在熟悉的 Windows 操作系统的 PC 机中去开发不是很熟悉的 Linux 操作系统的嵌入式计算机系统的开发思想，即主机/目标机开发架构。目前，常用的主机/目标板开发架构有三种：连接式设置架构、可抽换存储装置设置架构和独立式设置架构。实际上，每一个系统的开发过程中都涉及到了多种架构，或者是随时间的改变进行架构的转变，具体的分配由需求和开发方法而定[14]。本论文开发的类型运用到了所有这三种设置，随着开发过程的向前推进，系统逐渐趋向独立化。

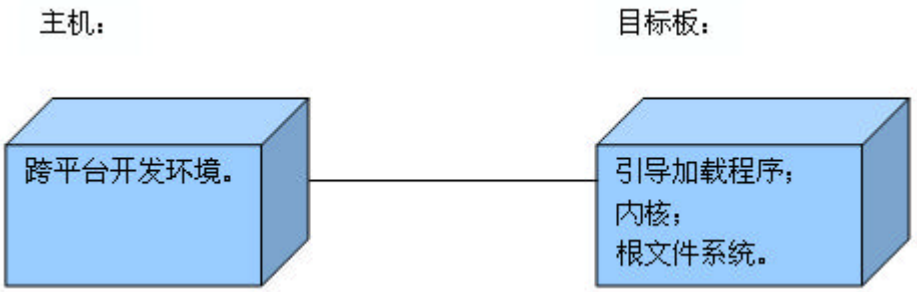

图 2.1 主机/目标板的连接式设置架构

第一种设置是连接式设置架构，如图 2.1 所示。在这种设置中，开发的全过程中，主机和目标板都会一直被一根串口线或是一根网线连接在一起。这种设置中，主机和目标板之间并未用到实际的硬件存储装置。既然连接一直都成立，数据都是通过连接来进行传送的。其中，开发时主机必须包含跨平台开发环境，目标机必须包含对应的引导启动程序、操作系统内核、根文件系统基本部分。这种设置中还可以用一种远程传送数据的方法来简化目标板的开发，即利用 NFS（Net File System，网络文件系统）或 TFTP（Trivial File Transfer Protocol，简单文件传输协议）来进行根文件系统的安装或者是内核的下载，这样的好处在于不需要不断在主机与目标机

之间传送数据，更不需要在目标板中使用存储介质。这种设置是最常用的架构，它也可用于进行调试，只是基于效率等方面的考虑，一般的设置均是采用网线连接起来进行调试。使得连接与调试分工，各尽其擅长的职务。比如本论文所描述的系统在开发的时候就是连接了网线与串口线两根连接线，前者用于启动引导程序、内核及文件系统的下载，后者用于调试。

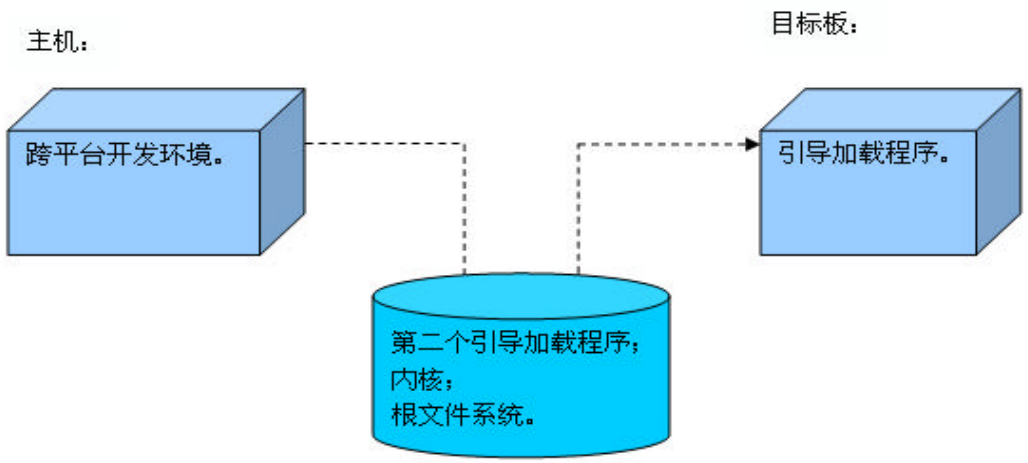

图 2.2 主机/目标板的可抽换存储装置设置架构

第二种设置是可抽换存储装置的设置架构，如图 2.2 所示。这种设置中，主机和目标板之间采用可抽换存储媒体来进行数据连接，那么它们之间就并不需要物理连接了。但跟第一种设置仍然有相同点，即主机上也必须有跨平台开发环境，只是目标板上只有最起码的引导加载程序。其它的引导加载程序、内核以及根文件系统都在可抽换存储装置上。系统启动开发时，必须先由主机向存储媒体送入第二个引导加载程序、内核及根文件系统，再由目标板中的最起码的引导加载程序来加载启动。实际上，并非所有的目标板中都会带有永久性存储装置。于是采用可抽换存储装置将更加灵活便捷。具体操作时，是先将可抽换存储装置连接上主机，从主机中拷入所需数据文件，然后将它插到目标板相应的插座中，目标板就可以调用了。这种设置比较稳定，当目标板出现状况时，其上的可抽换存储装置仍然是没有受到破坏的，只要不需要重新拷入数据文件，目标板就可以直接使用这个可抽换存储装置进行相应的操作[15]。

第三种设置是独立式设置架构，由图 2.3 所示。这种设置只要有目标板就可以

了，它内部涵盖了所有的引导加载程序、内核、完整的根文件系统及固有的开发环境。当然，但这只是在开发后期才会存在的状态。它不需要额外的任何跨平台开发环境等软件，也不需要另外的存储媒体，其内部模块已经具备了所有相应的功能。它所有所需的数据文件及开发环境的镜像均已经形成在自身的系统中，不需要传送工作就可以正常启动及工作。

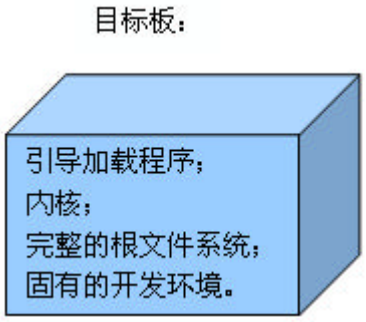

图 2.3 主机/目标板的的独立式设置架构

无论采用第一种还是第二种设置架构，主机和目标板之间进行数据传输的接口都有三种形式：串行接口、网络接口和特殊的调试硬件接口。这几种接口本质上的区别并不大，串行接口在操作上相对比较方便，但网络接口在传输速率上更胜一筹，而特殊的调试硬件接口需要在特殊的硬件调试下才能够运用[16]。本论文涉及到得系统开发的时候用到了前两种接口，串行接口主要用于观看超级终端，网络接口主要用于传输数据。更何况，本论文所选用的开发板的引导启动代码 u-boot 中并不包含直接进行串口传输的 rz（接收）和 sz（发送）命令，即无法直接通过串行接口来传输数据。当开发任务基本完成后，就可以采用第三种设置模式，不依靠 PC 机，也可以启动系统及运行程序，从而检测及完善独立体相应功能的实现。

2.4 网络通信的技术支持

大家都知道，linux 具有强大的网络功能，它很好地利用了当前最强大的 TCP/IP 协议来进行网络上的传输。故此处对 TCP/IP 协议相关知识也作一个简单的介绍。

2.4.1 OSI 参考模型与 TCP/IP

OSI（Open System Interconnection，开放式系统互连）参考模型是 ISO（International Organizations for Standardization，国际标准化组织）提出的网络互连模型，如图 2.4 所示，共分为七层。OSI 参考模型是依据系统的功能进行层次划分的，这样就可以以一种框架的形式来协调和组织各层协议的制定。这种分层的思想有利于将功能复杂的系统分解成较小的领域单独进行研究，从而更有利于问题的解决。总体来说，在这七层中，第一层到第四层总称数据流层，用来管理硬件。从第五层到第七层总称应用层，用来控制软件。即从 OSI 参考模型看，第五层会话层之下各层是面向网络通信的，会话层之上各层是面向应用的[17]。

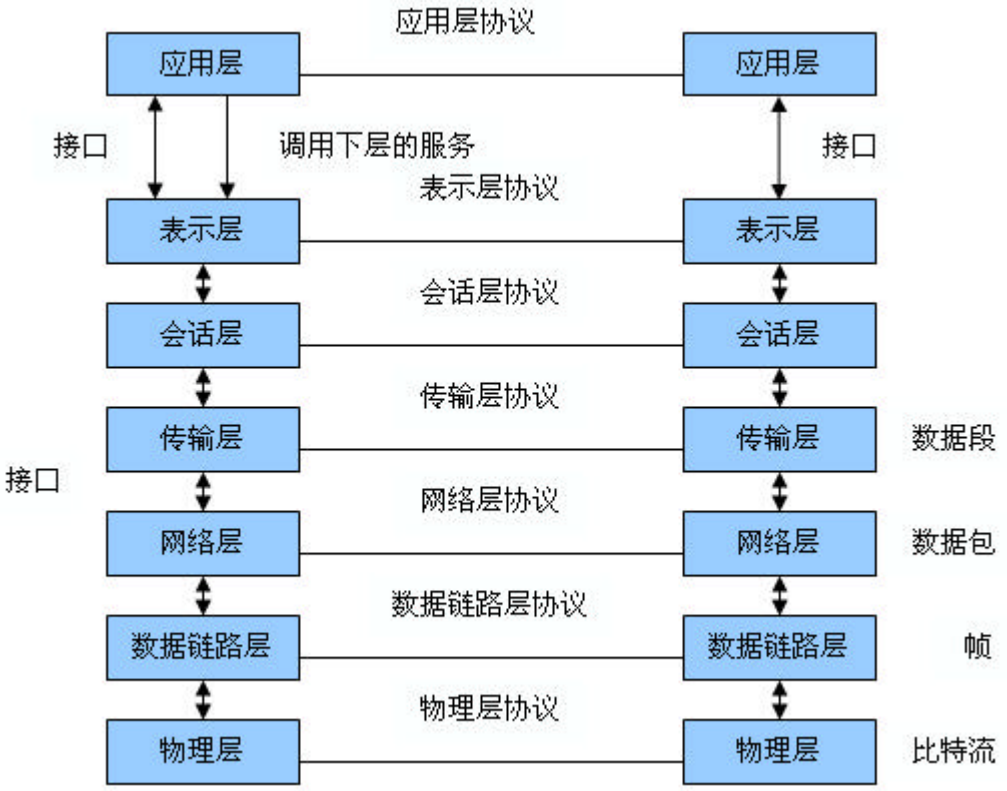

图2.4 OSI模型的网络体系

其中，物理层是 OSI 参考模型的最底层，其主要功能为利用物理传输介质为数据链路层提供物理连接，以便透明地传送二进制比特流。物理层协议是各种网络设备进行互连时必须遵守的低层协议，它对数据链路层屏蔽了物理传输介质的特性，

从而实现了对高层协议最大的透明性。在这一层，数据的单位称为比特（bit）。数据链路层是 OSI 参考模型的第二层（从下往上），它介于物理层与网络层之间。其主要功能是将一条原始的、有差错的物理线路变为对网络层无差错的数据链路。即将数据进行分帧操作，并处理流控制、指定其拓扑结构并提供硬件寻址等。为了实现这些功能，数据链路层必须执行链路管理、帧传输、流量控制、差错控制等。在这一层，数据的单位称为帧（frame）。第三层网络层的主要任务是路由选择，即通过寻址来确定分组从发生端节点到目的端节点之间的连接路径。具体说来，网络层实体在确定好下一个将要发送的网络节点后，将从传输层接收到的数据加上网络层包头，通过数据链路层的服务将包发送到那个节点。在这一层，数据的单位称为数据包（packet）。第四层传输层是 OSI 参考模型中从下往上第一个端到端、即主机到主机的层次，它实现点对点间的无差错数据传输及流量控制问题，从会话层接收数据，并根据网络层的具体需要，将上层数据分段并提供端到端的、可靠的或不可靠的传输。在这一层，数据的单位称为数据段（segment）[18]。

第五层会话层的主要功能是向会话的实体之间提供会话组织和同步服务，对数据的传送提供控制和管理，以达到协调会话过程、为表示层实体提供更好的服务。它允许在不同主机之间建立、管理和结束会话。由于利用了传输层提供的服务，可以使得两个会话实体之间进行透明的、可靠的数据传输，而不需要考虑一些繁琐的网络通信细节。第六层表示层的主要功能是对上层数据或信息进行变换以保证一个主机应用层信息可以被另一个主机的应用程序理解。表示层提供两类服务：相互通信的应用进程间交换信息的表示方法与表示连接服务。第七层应用层是最高层，是为操作系统或为了应用程序提供访问网络服务的接口。它负责整个网络应用程序一起很好地工作，是让最有意义的信息传过的地方。如电子邮件、数据库等都是利用应用层来传送信息的。总体来讲，应用层是面向不同的需求，实现不同的功能[19]。

TCP/IP（Transmission Control Protocol/Internet Protocol，传输控制协议/互联网络协议）并非仅仅是常见的 TCP 协议和 IP 协议，它跟 OSI 模型一样，是解决网络互连问题的一套体系。只是它被提出的时间要相对晚些，而且在现今社会中更常用。这两者之间有很多相似的地方，如图 2.5 所示。在 TCP/IP 体系中，最高层中有 FTP、

Telnet、SMTP 和 DNS 域名系统等，它们与 OSI 模型的应用层和表示层两个层次相对应。传输层有两种：TCP（传输控制协议）和 UDP（用户数据报协议），负责实现点对点之间的通信，且分别提供可靠地面向连接的点对点通信服务和无连接的不可靠的服务。网络层主要使用的是 IP 协议。而 ICMP 是 IP 差错控制协议、RIP 和 OSPF 分别为距离向量选路和链路状态选路算法的路由协议。数据链路层包括以太网 Ethernet 和令牌环 Token 等等[20]。目前市面上一般都使用 TCP/IP 协议，很大程度上是因为它可以用在各种各样的信道和底层协议之上，即屏蔽了具体的底层协议，用户只需要了解在应用层上的操作即可。

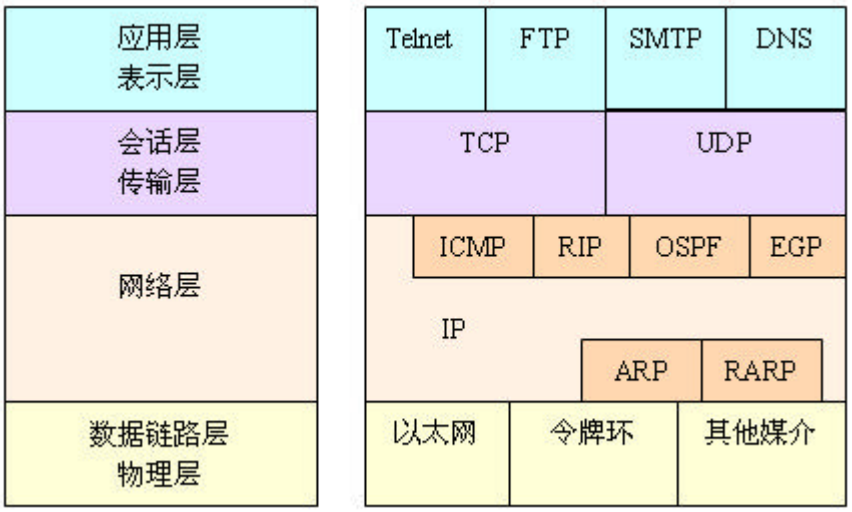

图2.5 TCP/IP协议栈及其与OSI模型的对应

TCP/IP 协议的传输层根据不同的要求分为两种传输协议方式：TCP（传输控制协议）和 UDP（用户数据报协议）。其中，TCP（Transfer Control Protocol）实现的是可靠的数据传输，而 UDP（User Datagram Protocol）实现的是高速但非可靠的数据传输。TCP 是面向连接的、端对端的可靠的流传输方式。说其是面向连接的，是因为它在进行数据传输之前会首先在进行通信的设备之间保持一个虚拟连接。而 UDP 是面向无连接的高效率的传输方式。它是建立在 IP 协议基础上的。它不能保证接收到得数据包的正确性及顺序性。但是也就是因为它的不可靠性与不进行重传性，使得它的时延较短，即传输速度比较快。很明显，TCP 协议用于当数据传输的性能要求更完整、更可靠和可控制的情况。而 UDP 协议用于更加强调传输性能而

非传输的完整性和可靠性的情况[20]。虽然是不可靠地传输，但其在现实生活中的应用还是很多的。如可以忽略部分信息的音视频应用，运用的就是 UDP 协议。两种协议在传输层都占据着十分重要的地位。

2.4.2 服务器端/客户端通讯模式

在网络通讯中，套接字起着很大的作用。既然 IP-CAMERA 涉及到了网络传输，自然少不了要用到套接字这个关键技术。套接字 Socket，英文原意就是“孔”或“插座”，在网络通讯中取后者，且理解起来类似于电话插座。抽象地讲，套接字实质上提供的是进程间通信的端点。因为在两个进程进行通信之前，两边都必须创建一个端点，否则是无法连接起来进行通信的。就像打电话之前，两边都需要有电话机一样。套接字是面向客户/服务器模型而设计的，针对客户和服务器程序会分别提供不同的套接字系统调用。每一个完整的套接字都要有一个相关描述，其具体形式为：{协议，本地地址，本地端口，远程地址，远程端口}。每一个套接字在被使用时都有一个唯一的套接字号，由操作系统统一进行分配。套接字号的使用，巧妙地解决了随机的用户进程之间的通讯连接问题，就像打电话时所需知道对方的唯一性质的电话号码一样[21]。

本系统软件结构采用的就是服务器端/客户端的网络通讯模式，即由客户端主动向服务器端提出请求，服务器对请求做出确认响应并执行相应的任务，建立连接后计算机用户就可在客户端监控各个远程被检测点，服务器根据用户要求向当前设备发出控制命令，从而实现远程监控。

服务器端/客户端的网络通讯模式是 TCP/IP 网络中两个进程间通讯最常用的模式，其两个最大的特点是非对等作用和完全异步的通信，且在操作的过程中采用的是主动请求方式。首先服务器端要启动，打开主通信通道，在某端口地址上监听是否有客户的请求到来。接收到请求时产生一线程去处理，然后继续监听是否有其它客户的请求，继续产生线程来分开处理。而客户端是打开通道后，向服务器端发送请求，接到应答时继续发送请求或是关闭通道。此时，客户端对服务器端是多对一的，且可并行地得到响应[22]。

本论文中描述系统中的套接字采用的是面向连接的工作流程，其一般调用流程如图 2.6 所示。在服务器端和客户端通讯之前，首先建立了一条虚拟的物理链接。且具体的套接字通讯流程为：服务器端先启动，通过调用函数 socket ( )创建一个套接字，然后调用函数 bind( )将先前创建的套接字和本地网络地址绑定在一起，再调用函数 listen( )使套接字作好连接准备，开始监听客户端的请求，并规定其请求对应的长度，最后在监听到请求时调用函数 accept( )来接收连接。而客户端在建立套接字后可用函数 connect( )和服务器建立连接。当连接建立时，客户端和服务器端之间就可以通过调用函数 read( )和函数 write( )来发送和接收数据。当数据传送结束后，双方调用 close()关闭套接字。

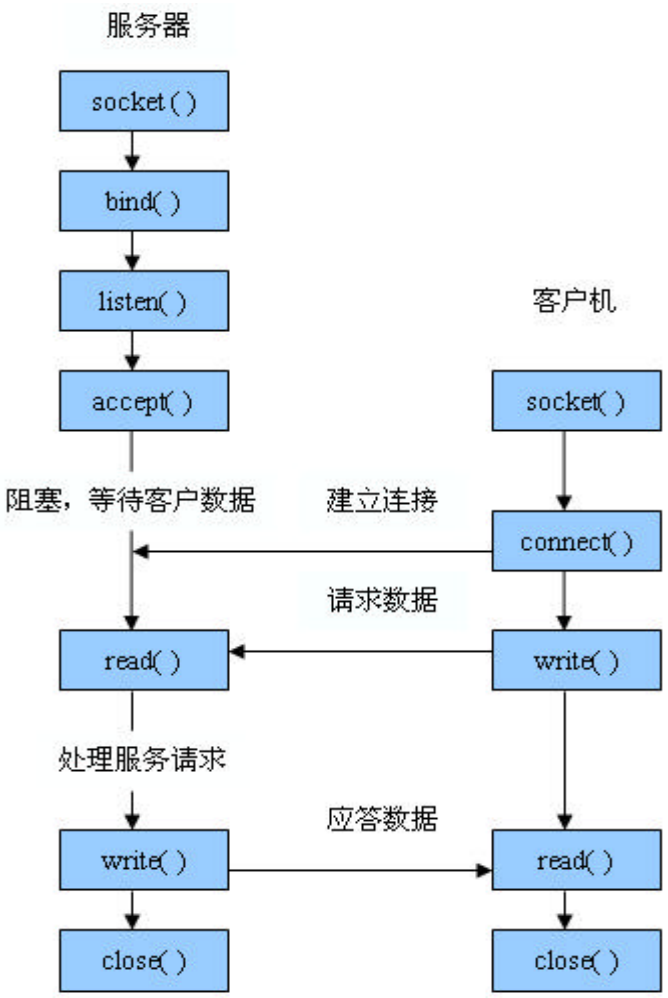

图 2.6 面向连接的套接字的工作流程

除了这种通讯模式（C/S）外，还有一种模式也比较常用：浏览器/服务器（B/S）

模式。日常中的网页浏览都是这种通讯模式的应用。但是用户对浏览的部分几乎无法进行更改等操作，不适合于此处带交互性质的系统。

2.5 多线程编程

大家都知道，进程是程序的一次执行。而线程则可以理解为进程中执行的一段程序片段。在一个多任务环境中，线程运行在进程空间内，同一进程所产生的线程共享同一内存空间，且同一线程中的两段代码可以同时执行[23]。

基于以上这些，如果运用多线程进行并发编程，可高效率地实现所要实现的功能。如本系统中服务器端的传输应用程序，即是采用的多线程并发编程。主线程先运行继而进入监听状态后，一检测到客户端 A 的请求，服务器端立马产生一个 A 线程去处理 A 客户端的请求。而此时若主线程又检测到另一客户端 B 的请求时，服务器端又可立马产生一个 B 线程去处理 B 客户端的请求。于是，A 线程负责处理 A 客户端的请求的时候，B 线程负责处理 B 客户端的请求，互不干扰。同理，此时有再多的客户端发来请求都可以得到及时且有序的响应。从而实现本系统中，多个客户端同时操作一个被监测点的视频监控。

2.6 本章小结

本章对本文中的系统服务器端运用到的一些基本的关键技术进行了详细的介绍和分析。鉴于该系统服务器端为嵌入式 Linux 系统，首先对嵌入式 Linux 系统基本概念进行了介绍，且针对本系统服务器端的开发模式及开发工具也进行了介绍。网络通信是该系统的一个重要的应用，而它的基础就是套接字的应用，该系统服务器端主要应用的是数据流套接字，实现可靠地传输。而这是基于既成的因特网实现的，这就又涉及到了基本的 TCP/IP 协议。在开发的过程中，一直采用的是服务器/客户端模式，这也是实现完该系统后应用中的直接体现。同时为了高效率地实现检测功能，该系统采用了多线程编程，使得服务器端由并发响应客户端的功能。

3 IP-CAMERA 服务器端的软硬件构架

3.1 IP-CAMERA 的总体网络构架

如图 3.1 所示为 IP-CAMERA 服务器端及其响应的客户端的总体系统架构，其中网络占了非常重要的中间连接位置，可见其在 IP-CAMERA 应用中的重要性。左端的多个 IP-CAMERA 在局域网中由存储器及缓冲服务器来控制，组成一个网络，然后连接到因特网上，再由因特网分别与右端的多个 PC 客户端相连。而事实上，其中的因特网运用的就是 TCP/IP 协议。TCP/IP 协议实现了两个系统之间的数据传输问题，且不需要考虑两个系统的版本及具体的硬件配置的差异。而且它也可运用于 UNIX 系统，以至 Linux 系统[24]。

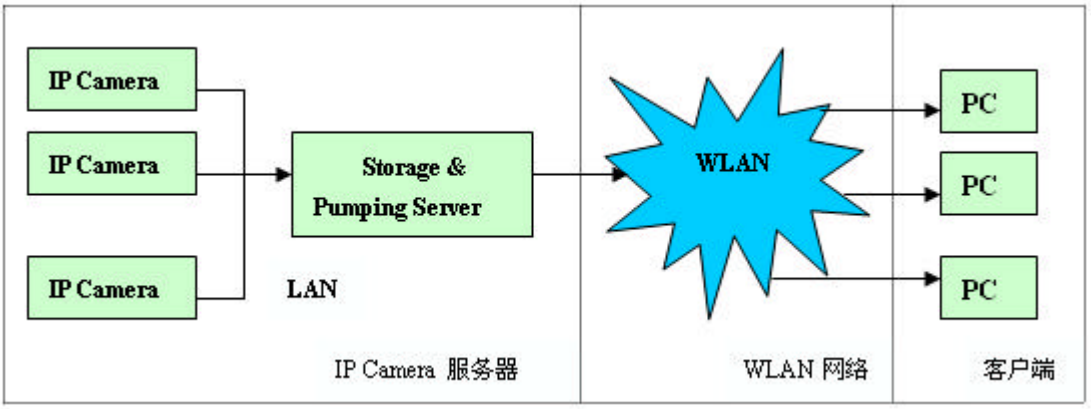

图 3.1 IP-CAMERA 的总体系统架构

TCP/IP 协议的操作隐藏了底层的操作，且在传输中提供了通用的框架，它的实施是独立于操作系统和计算机平台的。而 Linux 的网络功能是基于 BSD Unix 的 Socket 机制的。在 Linux 内核网络模块中已经搭建好网络协议接口、设备接口、设备功能驱动和媒介层的四层框架，在设计 Linux 下的硬件的网络驱动程序时，主要工作就是根据具体的硬件设备向它的高层提供服务。Linux 下的外围设备根据驱动的类型来分可分为三种：字符设备，块设备和网络设备[25]。为了屏蔽网络协议底层的多样性，Linux 将所有的底层硬件设备出入数据的地方统一为接口的概念。所有

的网络硬件在访问的时候都通过接口来访问，从而一致化上层发来的命令，抽象化掉种类繁多复杂的命令，对其进行响应。可见，Linux 系统的潜力是无限的。它在操作上，也是将各种设备均统一为文件，对文件的操作就是读写等有限的操作。在对外的网络操作上，又统一为接口，对接口的操作无异于发送接收等操作，从而使得开发得到了大大地简化[26]。

在客户端，跟大多数 Visual C++界面程序一样，此处 Windows 下客户端的传输程序很好地利用了 MFC 中封装 WinSock API 的两个类：CAsyncSocket 类和 CSocket 类[27]。前者是产生非阻塞的套接字的类，即即使在没有完成操作的情况下，在一定的时间里也会返回操作。而后者是前者的派生类，它能够利用非阻塞的套接字阻塞地实现任务，不受少量数据丢失的影响[28]。

相对地说，Windows 与 Linux 之间的网络通讯就是通过套接字 Socket 来搭建起来的，即这两个系统的套接字在网络上是相通的。在服务器/客户端的通讯模式下，服务器端与客户端的发送接收命令是相对的。只有有了一端的监听过程，接收到了另一方的请求命令，才有监听到请求发出响应的过程；只有有了一端的发送过程，才会有另一端的接收过程[29]。可见在定义模式上，Windows 和 Linux 定义的套接字结构体也会是相通的。否则服务器端在监听到客户端的请求时，就无法正确地通过网络获取到客户端的套接字信息，就无法正常地进行请求的响应[30]。

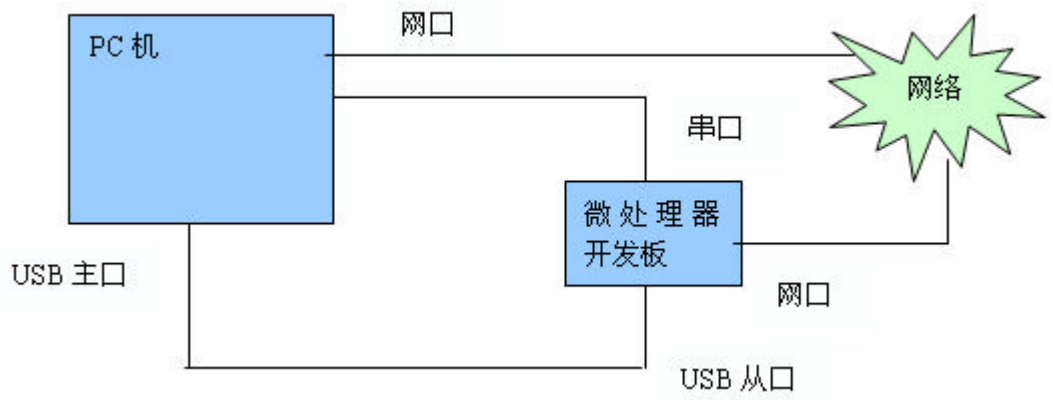

图3.2 系统开发时的总体环境模拟图

IP-CAMERA 服务器开发时的总体环境模拟图如图 3.2 所示，要解决它们之间的通讯问题，一般采用一根串口线或是 USB 线或是网线就可以连接起来。另外本

论文中设计的系统是通过网络来传输数据的，故无论是 PC 机，还是开发板都要分别用网线连接到因特网上。

3.2 IP-CAMERA 服务器端的硬件架构

IP-CAMERA 服务器端硬件主要由四个模块构成：CCD/CMOS 图像采集模块、压缩编码模块、微处理器控制模块和网络传输模块。如图 3.3 所示。

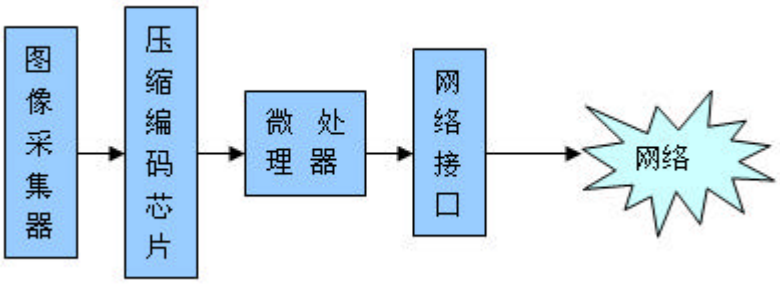

图 3.3 IP-CAMERA 服务器端基本硬件架构

本论文中所研究的高清 IP-CAMERA 监控系统将选用 ARM 微处理器、CMOS 图像采集器、高效率压缩编码芯片、扩展的高速传输网络接口等相关技术。若将 IP-CAMERA 服务器端各模块更加细化，可如图 3.4 所示。

图中左上角模块为视频 Video 模块，当图像采集器 CMOS 获取到数字图像数据后，首先送往 CODEC 压缩编码芯片进行数据压缩，然后通过模块接口送到 ARM 微处理器。在 ARM 中，可以看到有 RAM（Ramdom Access Memory，随机存储器）、ROM（Read Only Memory，只读存储器）、FLASH（闪存）、I/O（输入/输出设备）等等。ARM 是一款功能强大的微处理器，它将会利用 TCP/IP 协议，将接收到的压缩数据通过其网络接口发送出去，是整个传输模块中最重要的部分[31]。在图 3.4 下方，网络接口模块的右边还有串行接口模块和 USB 接口模块。这在开发的过程中有可能运用得到，或者是充当开发板的显示器用，或者是用于数据的传输。而图上方视频 Video 模块右边的音频 Audio 模块和云台控制模块，是为了整个 IP-CAMERA 系统更加方便于用户而设计的。

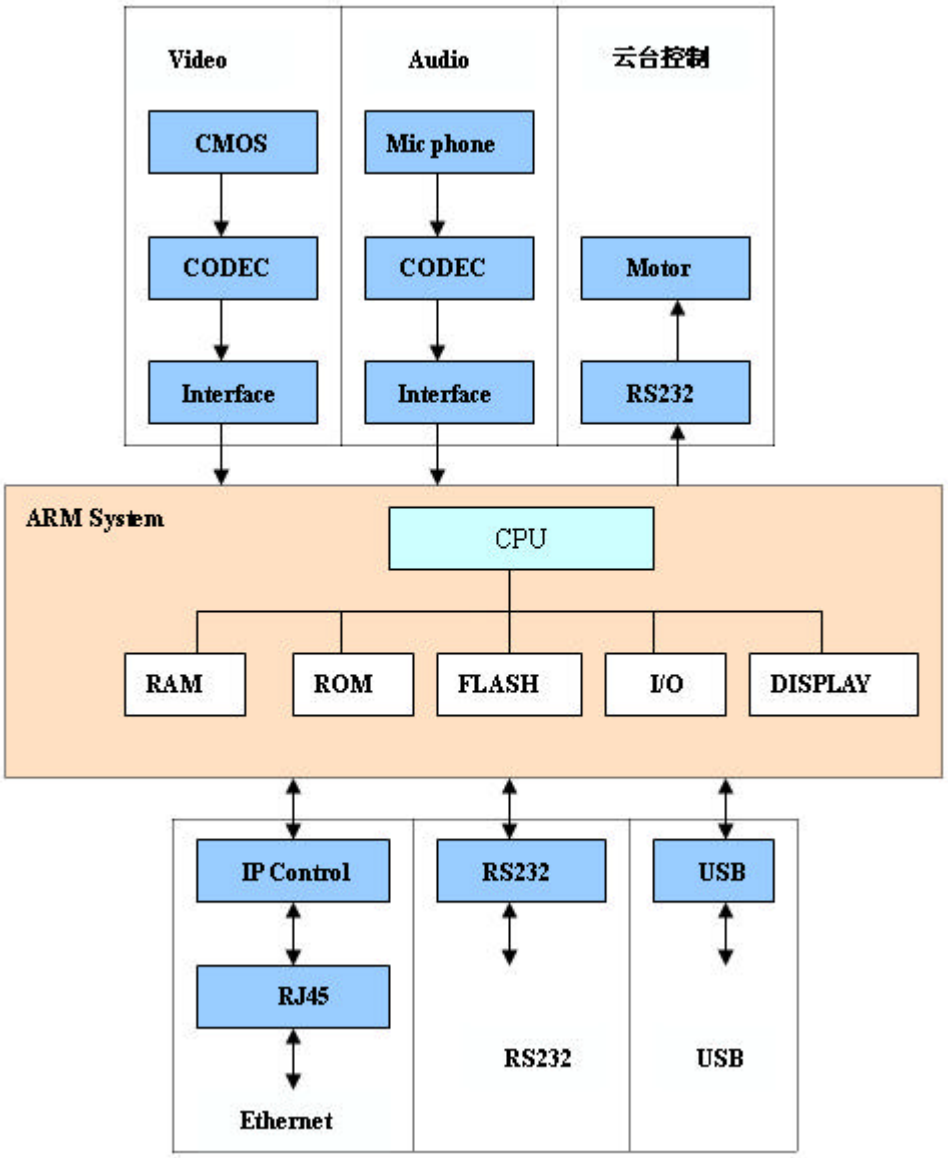

图 3.4 IP-CAMERA 服务器端完整硬件框架结构

其中，起主导控制作用的 ARM，它是一类嵌入式微处理器，体积小、功耗小、成本低，却性能高。如今，ARM 微处理器及其技术的应用几乎已经深入到全球的各个领域。当前已上市的 ARM 微处理器有 ARM7、ARM9、ARM9E、ARM10E、SecurCore 和 StrongARM[32]。其中，从 ARM9E 开始往后，都是综合处理器，性能虽强大，但针对性高，设计与要求也高。基于价格和性能上的比较，在此只考虑 ARM 7 和 ARM 9。ARM 7 和 ARM 9 均是低功耗的 32 位 RISC 处理器，其中 ARM 7 最适合于对价位和功耗要求较高的消费类应用，主要应用于工业控制和移动电话

等领域。而 ARM9 在高性能和低功耗特性方面更占据优势，处理能力能够达到 ARM7 的两倍以上，主要应用在无线设备和机械类安全系统[33]。所以此处选择 ARM9 将更加高效地完成任务。ARM9 本身有 4 个主要特点：1. ARM 9 采用的是 5 级流水线，即在一个时钟周期内可以同时处理 5 条指令，而 ARM 7 只能处理 3 条，这样 ARM 9 的处理能力很明显要强大得多；2. ARM 9 采用的是哈佛结构，此结构具有分离的访问总线，即可以并行地进行取地址和取数据操作。相对的，ARM 7 采用的冯·诺依曼结构的地址和数据的存储空间却是公用的，其处理速度自然要慢得多；3. ARM9 加入了高速缓存（Cache）和写缓存（Write Buffer）功能，这样在处理器本身的处理速度比较快的时候，存储器的访问速度也不会对它造成什么影响，毕竟一般的存储器的访问速度都要比处理器的速度要慢得多；4. ARM9 支持 MMU（内存管理单元），这样真正地对内存实行了保护，即使在一个进程运行失败的情况下也不会影响到其他的进程，大大地增强了处理器的稳定性和可靠性[34]。

3.3 IP-CAMERA 服务器端的软件架构

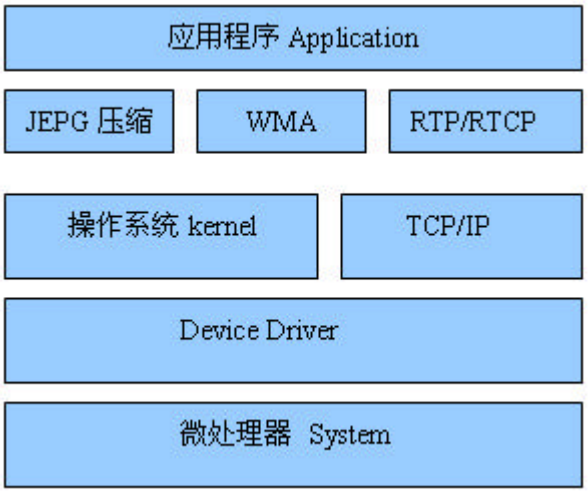

图 3.5 IP-CAMERA 软件架构

IP Camera 系统的软件架构如图 3.5 所示。IP Camera 系统的软件主要有外设驱动程序, TCP/IP，操作系统 Kernel，RTP/RTCP，音频/视频编解码库和应用软件等构成。最底层微处理器 System 是系统中选用的处理器硬件，Device Driver 是与微处

理器 System 相连接的一些外围设备的驱动，操作系统 kernel 是基于微处理器 System 的操作系统核心，TCP/IP、JPEG 压缩、WMA、RTP/RTCP 是此处理器安装上相应的操作系统之后一般所带有的一些基本应用，而应用程序 Application 则是开发过程中新加入的一些应用软件。本论文中主要开发的是传输模块，属于最高层的应用程序的开发。

3.3.1 开发环境的选择

如果要在一台电脑上安装两个或两个以上的操作系统，若不用虚拟机，最常用的方法是在同一个硬盘上装多个操作系统。但这个方法不够安全，因为硬盘对于操作系统而言是非常重要的，出状况的时候几个操作系统可能都会乱套。虚拟机 VMware 实际上只是一种应用软件，顾名思义，它的功能就是在一片硬盘上，虚构出一个计算机出来，这个虚构的计算机也有自己虚拟的内存与硬盘等，也可以安装任意类型的操作系统，并且进行相应的任何操作。在这个虚拟机上，可以安装 Windows、Linux 等真实的操作系统，及其相应的各种应用程序。所以，运用虚拟机软件可以在一台电脑上模拟出几台 PC，且每台 PC 可以运行单独的操作系统而互不干扰，可以实现一台电脑并行运行几个操作系统，这几个操作系统可以分别运行不同的应用程序，同时也都可以连入网络。只是，虚拟机毕竟是将两台以上的电脑的任务集中在一台电脑上，所以对硬件的要求比较高，主要是对 CPU、硬盘和内存的要求。故此处本文中的系统开发时选择的是虚拟机平台[35]。

选择了嵌入式操作系统之后，需要在 Linux 机上建立处理器的交叉编译环境，编译和烧写引导启动程序，然后建立 Linux 调试环境，添加 Linux 应用程序和驱动程序的步骤等。由于使用的是嵌入式 Linux 操作系统，那么嵌入式系统需要的基本软件有：文件系统、内核和引导启动程序。这些软件都是使用虚拟机 Linux 操作系统下安装的交叉编译器（于每一个处理器而言，封装的交叉编译器工具包不完全一样，但原理上都是一样的）来编译的。除了在 Windows 操作系统虚拟平台 VMware 中安装 Linux，然后在这个虚拟 Linux 操作系统中安装交叉编译器外，还可以在 Windows 下的模拟 Linux 环境 cgywin（是一个在 Windows 平台上运行的 Unix 模拟

环境）中进行安装。由于本系统考虑可以使用Windows下的烧写软件来烧写FLASH，将烧写 FLASH 与调试引导启动程序的调试环境建立一个操作系统平台下，这样调试、烧写就不用不断切换操作系统，所以更要选用虚拟机来建立开发调试环境[36]。

本文中的系统在开发时采用的是主机/目标机开发模式，这种开发模式中最重要的是在主机中及主机与目标机之间设置交叉开发环境。由于主机为熟悉且易于操作的开发环境，故在其上进行开发，然后直接将开发出来的代码及程序运用于目标机即可。对于目标机这个相对特殊的处理器，必须在开发的主机上安装跨平台开发工具链[37]。

一般而言，为了对任何目标板进行应用程序的交叉开发，需要将各种二进制工具程序集成进工具链，其中包括如 C 编译器（gcc）和 C 链接库（glibc）等。你可以到免费的镜像网站上下载 GNU 跨平台开发工具链的各个组件，包括 binutils 包、gcc 包和 glibclinuxthreads 包。或者是直接通过购买开发板时获得的光盘软件中获得，而且一般的开发板都由厂家提供了更适合其开发板的跨平台开发工具链版本。GNU 实质上就是一种工具，是集成了许多小开发工具的总称。它将被看成一个安装包，需要像一个应用程序一样被安装进 Linux 操作系统。安装成功后，调用其中的命令即可完成相应的开发及调试工作[38]。常用应用程序的编译命令为 gcc，常用应用程序的调试命令为 gdb，编译 Linux 内核命令为 make menuconfig，编译文件系统的命令为 mkcramfsimage 或是 mkyaffsimage（针对于生成的文件系统的不同类型而定）等。YLP2440 开发板中提供的是 arm-linux-gcc-3.4.1.tar.gz 和 cross- 3.3.2.tar.bz2 工具包。把它们解压安装到 Linux 的根目录“/”下即可。

3.3.2 服务器端传输模块框架结构

众所周知，在 IP-CAMERA 中，最重要的是要实现视频数据的监视。那么，在开发此系统的过程中，最重要的模块就属服务器端的视频数据传输模块。数据传输模块的主要功能是让嵌入式系统实现通过网络，响应从指定 PC 机发来的接收、发送的任何格式数据包的命令。服务器端传输模块在整个软件架构中属于最上层外加的应用程序，它的基本框架结构如图 3.6 所示。

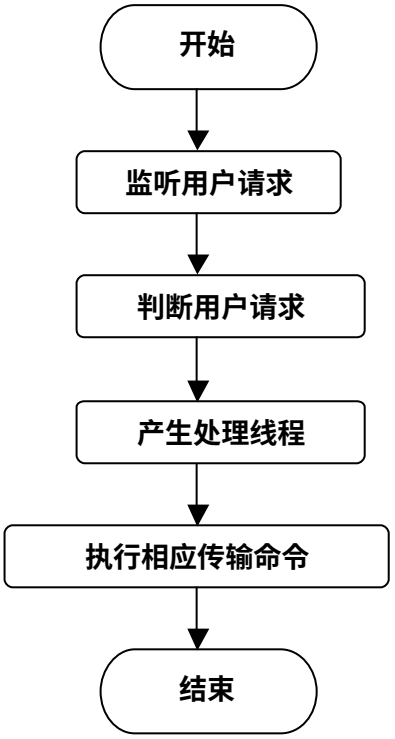

图 3.6 **服务器端传输模块框架结构**

3.4 本章小结

本章首先对 IP-CAMERA **系统服务器端的网络架构及软硬件架构进行了分析，并针对图像采集器、压缩编码方式及** ARM **微处理器的选择进行了简要的介绍。最后对服务器端传输模块的框架结构进行了简要的分析，勾勒出整个** IP-CAMERA **的软硬件框架。**

4 高清 IP-CAMERA 服务器端软件的设计

4.1 系统软件的设计原则

首先，整个系统软件主要是遵从模块化的设计原则。而模块化设计最大的好处在于可以在系统的开发过程之中或是之后简便地加入或是裁剪掉一些模块，以对系统进行增加或是修改某些模块功能。其次，系统软件采用了可扩展化的设计原则。在各模块设计的过程中，预备一些可扩展接口，可以使得在之后的开发过程中随时扩展新的功能。最后，系统软件采用的可重复开发的设计原则。在利用关键模块进行相应的扩展后，可以开发出新的相应功能的应用系统，这样避免了重复开发的麻烦，使系统具有良好的可重用性[39]。

而且各个模块均可以分开由不同的人员来进行开发，这样开发人员在开发某一模块时可以先实现本模块，开发成功后再与其它的已实现的模块一起进行检验整个系统的功能可行性。各个模块之间具有相对的独立性，因此在开发的过程中，需要对其它的模块特别是相邻接的模块有一定的了解。从总体上看，包括服务器端和客户端来说，IP-CAMERA 监控系统就分为五大模块：网络通信模块、音视频应用模块、系统配置模块、用户登陆与管理模块、云台与镜头控制模块。从开发角度来看，可分为 IP-CAMERA 服务器端模块和 PC 客户端模块两部分。而 IP-CAMERA 服务器端模块又可细分为图像采集压缩模块、网络通信模块和传输接口模块。PC 客户端一般由通用 PC 机组成，在 Windows 操作系统下完成图像网络传输、解码、存储与显示。各模块由项目组的成员分开开发，使得软件细化而又有条理。而本人负责的主要是服务器端的通信模块的软件设计。

4.2 服务器端软件的设计

如图 4.1 所示，嵌入式系统的软件由 4 大部分组成：root, kernel, ramdisk, application。此处设计都要考虑到。其中，广义上讲，ramdisk 包括 Device Driver 和 OS。但此处要说明的是，一般而言，驱动是加进 kernel 的，但有时驱动又可以作为

Application 放入。要使系统中的微处理器正常运作，并实现相应功能的应用程序，这 4 大部分缺一不可。前一项是后一项的基础，后一项的运作是前一项积累起来的应用[40]。这里，目标机即为服务器端，即服务器为一嵌入式系统。

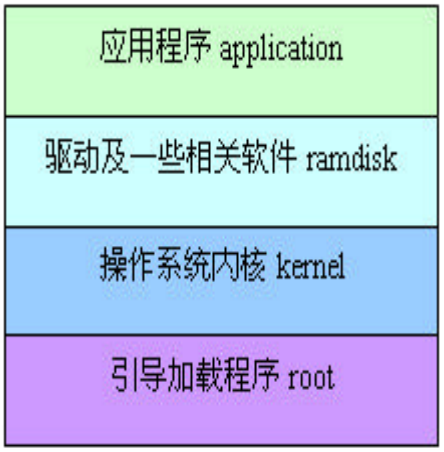

图 4.1 嵌入式系统软件基本架构

可见，服务器端的软件总体上可分为两大类，一类是微处理器正常进入系统工作时所需的基本软件，另一类是新加入功能对应的应用程序软件的设计[41]。本文中设计的系统开发过程中，对第一类基本的软件仅仅是进行了重新了解烧入，而对于第二类软件则是完全自己编写代码进行设计实现。

4.2.1 服务器端所需各基本软件的设计

图 4.2 所示是本系统的服务器端软件的总体框架，以及各模块在整个服务器端所处的位置、实现的功能和最后的可执行代码形式。

首先，代码资料中的 bootloader 是此处 ARM9 的 root，实现对其上的各芯片的初始化及引导内核、操作系统的功能。实际上，bootloader 和上一节中提到的 root 一样，就是常说的引导启动程序 u-boot，只是总称不一样而已。对于开发板而言，这一部分是不可或缺的，一般由开发板商家提供，不用更改，此处也不做变化。它主要有两个作用：1.负责初始化处理器及相关部件，这一部分相当于 Windows 下的 BIOS；2.它负责引导操作系统。引导的方式也有两种：a.在连上 PC 机的情况下，使用 tftp 服务器进行调试，这时根文件系统均是从 PC 机上直接挂载下来的；b.在不

连接 PC 机的情况下，从 flash 的另一区域（不同于 u-boot 的区域）将文件系统引导到内存里进行。

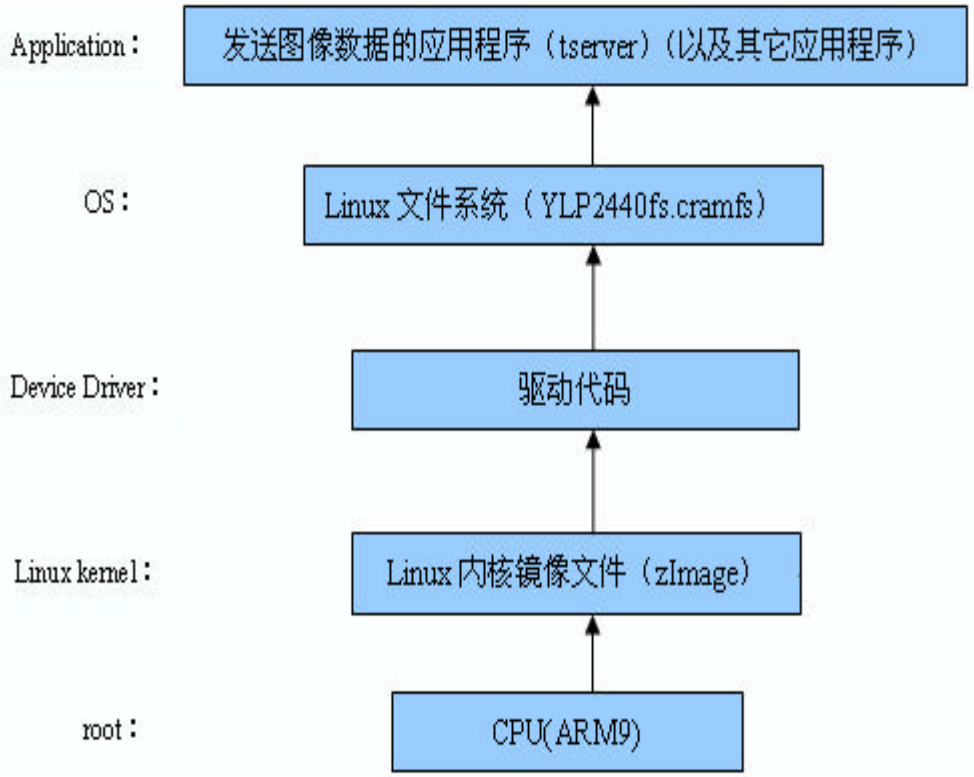

图 4.2 高清 IP-CAMERA 服务器端软件总体框架

然后，ARM9 就会引入 Linux 内核镜像文件，作为要引入的操作系统内核与各芯片硬件的配置。接着，如果新加入了硬件芯片，还需要 Linux 内核与此新硬件芯片进行配置，这就是由该硬件芯片的驱动来完成的。而内核 kernel 的镜像文件，在不外加驱动及不需用另行的裁剪的情形下，它一般采用开发板自带的内核镜像即可。若想自己重新编译，也可用 make menuconfig 这个图形化的模式来编译。

再后来，引入真正的操作系统各个文件，此处引入的 Linux 操作系统基本功能文件。最后，基于基本功能的操作系统，加入应用程序，就可以实现先前没有的功能。各部分模块一起正常工作的话，服务器端就可以正常启动其操作系统，收到命令后，将摄像头采集的数据压缩后通过网络发送到客户端，并能够接收客户端的命令请求做出相应的反应。本文主要研究的是在具有基本硬件的条件下，正常运用基

本工作软件后，发送图像数据的应用程序 tserver 的开发及其应用。所以，对于文件系统，由于其中加入了新的应用程序，是需要重新编译的。Linux 支持的文件系统格式主要有 CRAMFS（Compressed ROM File System，只读压缩文件系统）和 JFFS（Journalling Flash File System，日志闪存文件系统）两种。其中前者是只读的，在启动时装载较快，可用来存储动态连接库、应用程序、初始化脚本等；后者可进行读写操作[42]。因为此处 YLP2440 开发板默认支持的是 cramfs 类文件系统，本文编译烧写的也就是 cramfs 类型的文件系统。mkcramfs 即为相应的命令工具，它的安装包在开发板自带的光盘里或是网络上都可以得到。直接将其压缩包解压到 Linux 系统中的工具目录下，就可以使用了。一般文件系统的重新编译是由于新增了应用程序，需要把新增应用程序也整合到文件系统的镜像文件中，这样就可以作为文件系统的一部分载入系统[43]。

4.2.2 服务器端传输模块应用程序的设计

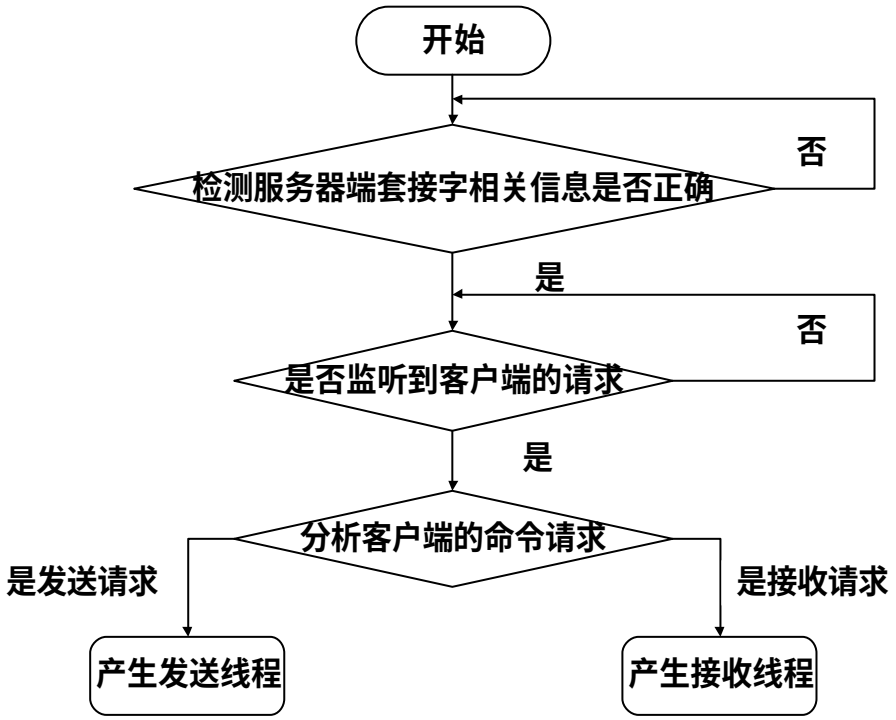

图 4.3 服务器端总代码流程图

服务器端传输模块要实现的功能为：收到请求命令后，将摄像头采集的数据压

缩后通过网络发送到客户端，并能够接收客户端的命令请求做出相应的反应。对于服务器端传输模块应用程序实现的具体代码，此处作一个简要的分析，其总体流程图如图 4.3 所示。

服务器端的代码是要在虚拟机中的 Linux 操作系统下编译成 ARM 板上可执行的二进制代码的，它的源代码为最常用的 C 代码。由于此处 Linux 操作系统是在虚拟机中设置的，可以直接在 Windows 下编写代码，再放入与虚拟机下 Linux 共享的目录下进行编译[44]。服务器端传输模块的应用程序一开始运行，就应该建立套接字，绑定本地信息，然后监听客户端的请求。监听到客户端的发送/接收请求时，分别产生发送/接收线程。

产生主线程后，最基本的也有两种命令的处理。即传输模块主要分两部分：发送数据模块和接收数据模块。从本质上将，服务器/客户端模式中的服务器端与客户端只是一个相对的概念。即服务器端与客户端是可以互换位置的。就像打电话的过程中，主叫方与被叫方在挂断电话后，被叫方也可以主动发出请求，成功建立通信后，从而变为主叫方。所以此处若只是把嵌入式系统称为服务器端的话，它既可以向被称为客户端的 PC 机端发送数据，也可以从 PC 机端接收数据，从而实现两者之间的相互通信[45]。服务器端的发送线程流程图和接收线程流程图分别如图 4.4 和 4.5 所示。

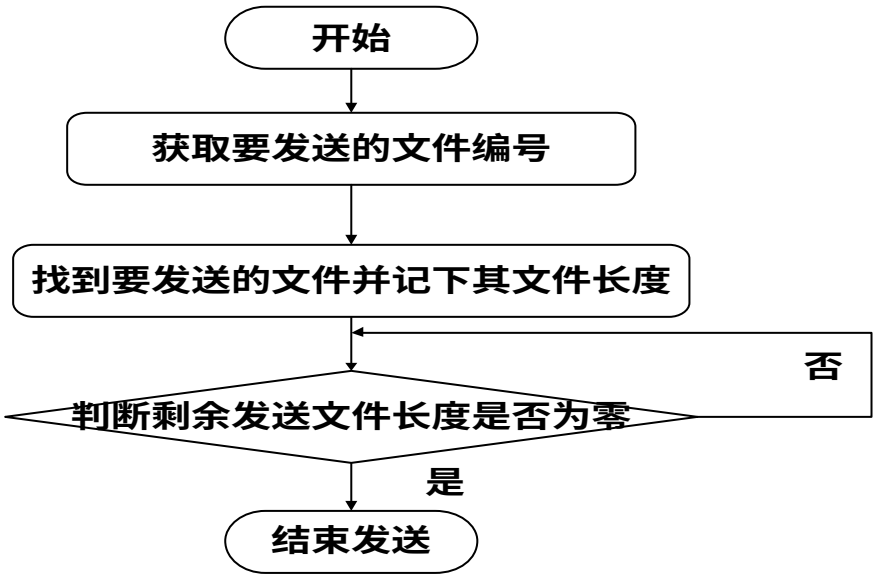

图 4.4 服务器端发送线程流程图

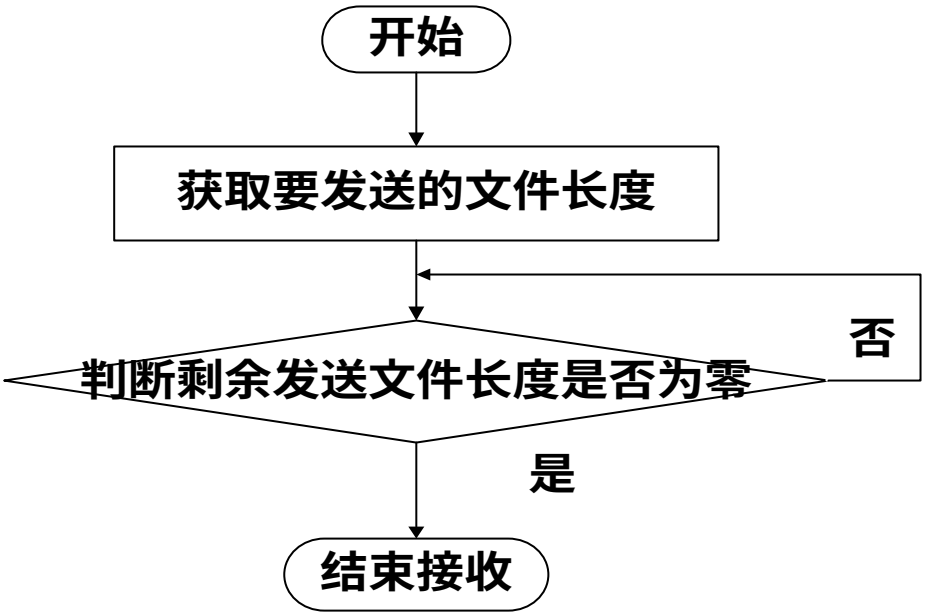

图 4.5 服务器端接收线程流程图

4.3 客户端软件的设计

先前说到本论文中的系统选用的是虚拟机上的 Linux 操作系统开发环境。因为虚拟机是作为一个应用程序装在 Windows 操作系统上的，而开发板上的处理器中运行的操作系统为开发出的 Linux，故从硬件角度来看，开发的时候需要解决的是 Windows（PC 机）端与 Linux（ARM 开发板）之间的通讯问题。

客户端的代码是在 Windows 下的 Microsoft Visual C++6.0 套件下开发的。Microsoft Visual C++是在进行界面编程中最受欢迎的开发工具，其利用的基础开发包 MFC 封装了很多有用的类，包括最常用的对话框等。编写好代码后，生成批处理文件，最后打包安装在任意 Windows 下的 PC 机端，就可以作为一个客户端来监控服务器端了。

本论文中除了基本的对话框外，还引用了数据库的处理，进行对授权的客户端的确认登陆处理。既然上面提到服务器端与客户端是相互的，那么此处，客户端的传输模块无疑也是由发送模块和接收模块两部分组成的。从整体来讲，客户端从正确登陆对话框后，就转入传输模块对话框，若客户需要主动发送数据到服务器端，则启动其发送对话框，相反则启动接收对话框。

简单地说，客户端传输模块就是一个 Visual C++的设计界面。基于对话框类 CDialog，设计客户端用户易于操作的许多对话框界面链接。基本框架结构如图 4.6 所示。开始运行客户端传输程序时，跳出身份验证对话框，输入正确信息后，跳出相应的连接服务器及发出命令请求的对话框，最后跳出发送/接收文件操作对话框。其间会有一些操作成功或失败的提示对话框。在熟悉 VC 的情况下，客户端的传输模块还是很简单就可以实现的。只是在发送接收文件的过程中，会涉及到网络发送与接收，这就要求熟悉 Windows 下套接字的相关知识。

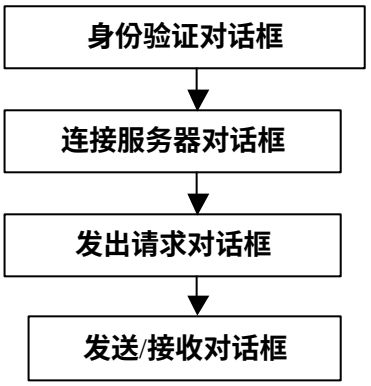

图 4.6 客户端总体框架结构图

4.4 本章小结

本文主要是研究设计高清 IP-CAMERA 服务器端传输模块的软件部分的，此章开始着重于设计实现的软件部分了。首先对整个服务器端传输模块的系统软件进行设计原则的分析，然后对服务器端的基本软件和应用程序的设计进行了详细地分析，之后对需要响应服务器端的客户端作了简要的设计分析，从而形成一个完整的系统的设计。

5 高清 IP-CAMERA 视频数据传输的实现

5.1 开发环境的建立

5.1.1 硬件环境

虽然 Windows 和 Linux 是两个操作机制完全不同的操作系统，但是在通信方面还是有许多通处的。鉴于采用的 ARM 开发板没有配液晶屏，本文中系统使用串口线在 Windows 上设置其超级终端来充当 ARM 的显示器窗口。由于这些接口资源只能被相应硬件独享，所以再接一个 USB 线来实现数据下载的功能。传输数据时利用的是网络，故同时要将 PC 机和 ARM 板分别用网线连接到网络上。另外，如果需要重新烧入 bootloader，还可从 PC 机并行口通过 JTAG 调试板接线接上 ARM 的并行口来进行烧写[46]。相应开发的实物图如图 5.1 所示。

图5.1 开发总体环境实物图

本论文系统中选用的是深圳市优龙科技有限公司推出的 YLP2440 开发板，硬件框架如图 5.2 所示。其 CPU 是基于 ARM9 的体系结构，型号为 S3C2440A，主频 400MHz。其外围硬件主要有一个 64MB 的 NAND Flash，一个 64MB 的 SDRAM，一个 USB 主设备，一个 USB 从设备，一片音频编解码芯片，2 片缓冲芯片，两个波特率高达 115200bps 的五线异步串行口，一个 10/100M 网口，一个标准 JTAG 接口，一个 12V 电源，另有红外模块接口、标准 SD Card 接口、Camera 接口、IDE 硬盘接口、LCD 接口及用户扩展接口等支持多种功能的接口。

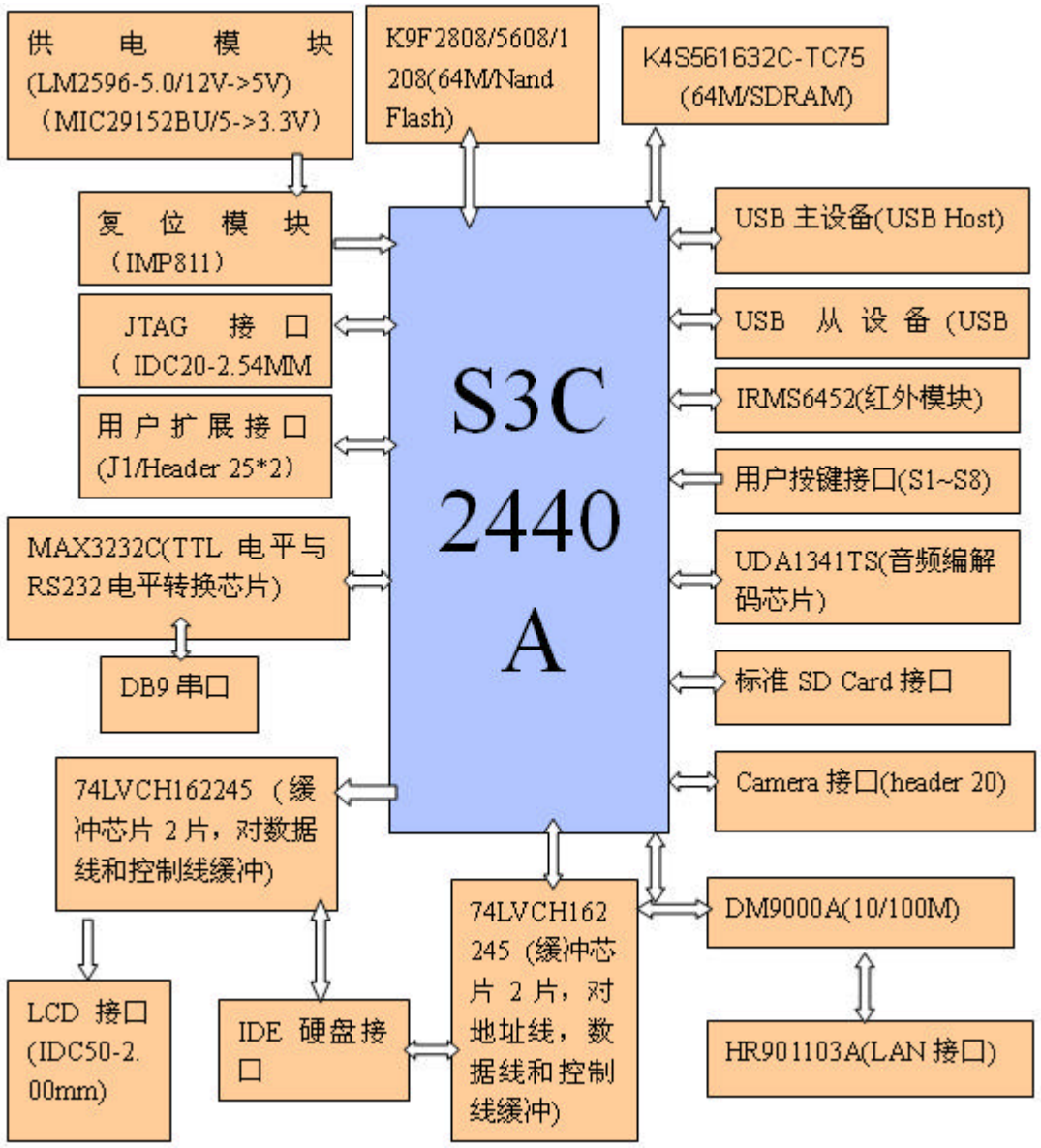

图 5.2 YLP2440 开发板主要硬件框架图

5.1.2 软件环境

由于开发时采用的是宿主机/目标机的模式，即由宿主机（PC 机）开发出在目标机（ARM 板）上运行的二进制代码文件，然后把这些二进制文件烧入目标板中由目标机去运行，达到嵌入式系统独立完成任务的结果。因为 PC 机的 CPU 和 ARM 板的 CPU 之间还是存在比较大的区别的，所以它们能够运行的二进制文件是无法通用的。这时就需要交叉编译环境，它是宿主机上的一组软件工具，经过它编译（而不是直接由宿主机上自带的工具编译）的二进制代码文件可以由目标机来正常执行。故此开发过程中，建立交叉编译环境还是很重要的。一般的开发板出厂时，会自带些特许它使用的交叉编译环境工具包，直接作为一个应用软件安装到 Linux 操作系统中即可。比如最简单的编译单独应用程序的命令 gcc 是宿主机上的，由它编译的二进制文件只能在 PC 机上运行，若是由 ARM 板来运行，它根本就无法执行。而由 arm-linux-gcc 命令编译的二进制文件则只能由相应的 ARM 板来执行，若在 PC 机下执行，则它就永远执行不出来，就像这个执行过程死掉了一样，只能中断退回命令符端。

此处 YLP2440 的编译工具包安装命令为：#tar xvzf arm-linux-gcc-3.4.1.tgz –C /。直接安装到 Linux 操作系统的工具命令目录下即可调用。应用程序、Linux 内核的编译工具在这些包中都带有，唯有文件系统的编译工具需要根据不同的文件系统类型来单独安装到工具目录下。

在 PC 上，系统使用的软件工具有虚拟机上的 Linux、DNW 工具，有时为达到清晰显示系统启动效果，还会引入 Windows XP 自带的超级终端。其中，Linux 选用的是 Red hat9.0，很经典的版本，其内核为 2.4.20。

对于开发板自身操作系统的启动，有一些开发板如经典的 QQ2440 有两种启动模式，一种是从非易失性的 NOR FLASH 启动，另一种是断电数据会丢失但价格相对便宜的 NAND FLASH 启动。一般前一种是在 NOR FLASH 中烧入了 BIOS 及操作菜单代码，即使没有烧入操作系统，开发板也可以进入此项，并通过选择菜单选项来进行要做的任何操作，比如重新烧入内核或文件系统、设置环境变量、启动操作系统等。后一种模式就直接进入操作系统。两种启动模式一般是通过跳线座来实

现，将跳线帽接到一边是以第一种模式启动，接到另一边是以第二种模式启动。但是由于 YLP2440 只有一种 FLASH，且是 NAND FLASH，则它的启动模式只有一种。系统上电后，直接是从 bootloader 的菜单功能界面进入，根据选项来选择操作。其实菜单只是对各个操作的封装，这样对初学者而言更易于操作，实际上都是将相应的代码加载到具体的 SDRAM 硬件地址，让系统去运行。对于 YLP2440 开发板，跳线模式是默认的短接状态。系统上电后，按任意键就进入菜单选项，不按键，就直接进入操作系统。

5.2 实验步骤

5.2.1 引导程序、内核、文件系统的编译与烧写

在 Linux 操作系统环境下，上面那个 gcc 与 arm-linux-gcc 只是独立应用程序的编译命令。若应用程序的代码需要多个源文件和头文件组成，则需要加入 Makefile 文件进行编译配置。Linux 内核的编译是使用的 make menuconfig 及 make zImage，前者主要是配置，后者才是实质上的编译过程。这些命令都是在当前命令符位于 Linux 内核源码包内的文件夹中时才能使用的，如果编译成功的话，最后会在当前目录下的 arch/arm/boot 下得到 Linux 内核压缩镜像 zImage。由于 YLP2440 开发板默认使用的是 cramfs 类型的文件系统，所以其 Linux 文件系统编译时使用的是 mkcramfs 命令（这个工具是单独安装到 Linux 上的一个应用程序）。这个命令是在当前命令符位于文件系统源码包所在的文件夹中时运用的，编译成功后，会在当前文件夹下生成以.cramfs 为后缀的文件系统镜像文件。

在与开发环境断开的情况下，开发板能够独立地执行相应的应用程序。应用程序的编译需要用 arm-linux-gcc 命令，且由于本文编写的代码中涉及到线程问题，需要在编译时加入参数-pthread，即编译命令为：#arm-linux-gcc –o tserver –pthread tserver.c。此处 tserver 即为可在 ARM9 开发板上运行的服务器端二进制代码名，tserver.c 为 C 源代码名。应用程序编译完成后，将可在 ARM 下执行的二进制文件拷贝到文件系统 fs 源代码所在文件夹下的/sbin 目录下。因为本论文涉及到的服务器端代码涉及到了网络编程，需要对开发板的 IP 地址进行统一化，故可在文件系

统源代码中作适当地更改。可将/usr/etc/rc.local 中的 Eth0 改为 192.168.1.106，作为开发板的 IP 地址。然后将命令符操作进入此源代码相应的文件夹中，运行 mkcramfs 命令，即可在当前目录下生成文件系统镜像文件。如若文件系统源代码所在文件夹命名为 ylp2440fs，则编译此文件系统时的命令为 #mkcramfs ylp2440fs ylp2440.cramfs，编译完成后，就会在当前目录下生成 ylp2440.cramfs。

而最终是要把从宿主机上开发的代码文件转到目标机上运行的，这个转的过程通常叫做烧写。具体来说，就是从宿主机的硬盘将一些软件文件拷入到目标机的 flash 中，再通过一些设置，使得目标机在接到启动系统命令时启动系统、接到运行某应用程序时运行这个应用程序等操作得到实现。烧写的过程有许多种，常见的有通过串口、网口、USB 口，甚至是 JTAG 调试板连接的并口。对于本论文运用的 YLP2440，它主要使用的是后两者。而要烧入的软件有三个：bootloader、Linux 内核及文件系统。鉴于对于开发板的启动顺序，首先需要烧入 bootloader，这样开发板才能正常初始化，才能在检测到所需基本硬件都正常且可以在其上运行操作系统及其应用程序后，才会去引导 Linux 内核和文件系统，再进入系统。当然，无论是 YLP2440 的 flash 中根本就没有任何软件，还是先前的 bootloader 因为某种原因出了问题（但一般都不会），都可以来烧写 bootloader。说这么多，就感觉到 flash 中根本就没有任何软件时，bootloader 的烧入要重要且麻烦得多。此时它的烧入必须通过 YLP2440 开发板自带的 JTAG 调试板来烧入，而若以前烧写过、只是重新烧写的话，仍然可以使用后面 Linux 内核及文件系统烧写的方式来烧写。用 JTAG 调试板来烧写 bootloader 的时候短接帽 JP1（使得系统从 NAND FLASH 中启动的模式选择接帽）可以拔掉。

YLP2440 开发板的文件都是直接从 Windows 下烧入的。即从虚拟机与 Windows 共享的文件夹中将要烧入的文件拷入到特定的硬盘上，然后在大家都十分熟悉的 Windows 下操作即可。安装好 SJF2440 工具（运用 JTAG 调试板来烧写的 Windows 端软件）及相关驱动后，就可以直接根据跳出来的对话框来选择要烧入的文件。操作完毕，即可在断掉电源的情况下，接上短接帽 JP1，连上串口线，设置好超级终端。然后接上开发板电源，就可以在超级终端看到 bootloader 在开发板中正常启动

了。至于内核和文件系统的烧写就比 bootloader 简单的多了。

bootloader 烧写完成后，一启动开发板，就会发现相应的一些菜单选项。其中就有下载、写至 flash、启动系统等功能选项。YLP2440 默认选用 USB 线来下载，连接好 USB 线后，超级终端会显示 USB 连接成功。在 Windows 端安装好 DNW（在 Windows 端运用 USB 线来下载的执行软件界面），其上显示 USB 连接成功后（第一次连接开发板时需要安装相应驱动），即可直接通过界面上的 USB 发送按钮来完成。下载后，菜单会进入下一级，选择文件所在的分区号，即可让开发板识别下载的是 boot、Linux 内核，还是文件系统，分别送入相应的分区。这些操作都是在 Windows 下执行的，在 DNW 端从 PC 上选择要烧写的文件来发送，在超级终端中选择接收模式及烧写目标路径。

5.2.2 开发后服务器端的实验步骤

在服务器端启动前，在硬件上，首先接上开发板的短接帽 JP1，连好串口线和 USB 线分别至 PC 机的串口和 USB 口；在软件上，分别设置好超级终端、安装好 DNW 及安装 USB 驱动。其中，超级终端中应设定 COM1、波特率 115200、无硬件流。当然，把 DNW 当作超级终端仅仅作为开发板的显示器的时候，其设置与 XP 自带的超级终端设置是一样的。而 USB 驱动是凡第一次将类似于 ARM9 的开发板连到 PC 机上时均需要安装，直接在第一次插入开发板时，Windows XP 跳出的找到新的硬件向导对话框中引入其 USB 驱动即可。

以上设置好后，连接好电源，如果系统正常，就可以在超级终端中（也可以用 DNW，具体根据个人习惯而定，以下不再赘述）看到系统的启动界面。如图 5.3 所示即为超级终端下看到的开发板 YLP2440 的嵌入式 Linux 操作系统正常启动的部分过程。

若要重新烧入 bootloader、内核或文件系统，在开发板上电或复位时敲入任意键，就可以直接进入 bootloader 功能界面菜单，进行操作。若直接进入系统，可以接上电源后直接让它跑进系统或者进入 bootloader 菜单界面选择 5.boot the system 进入操作系统。

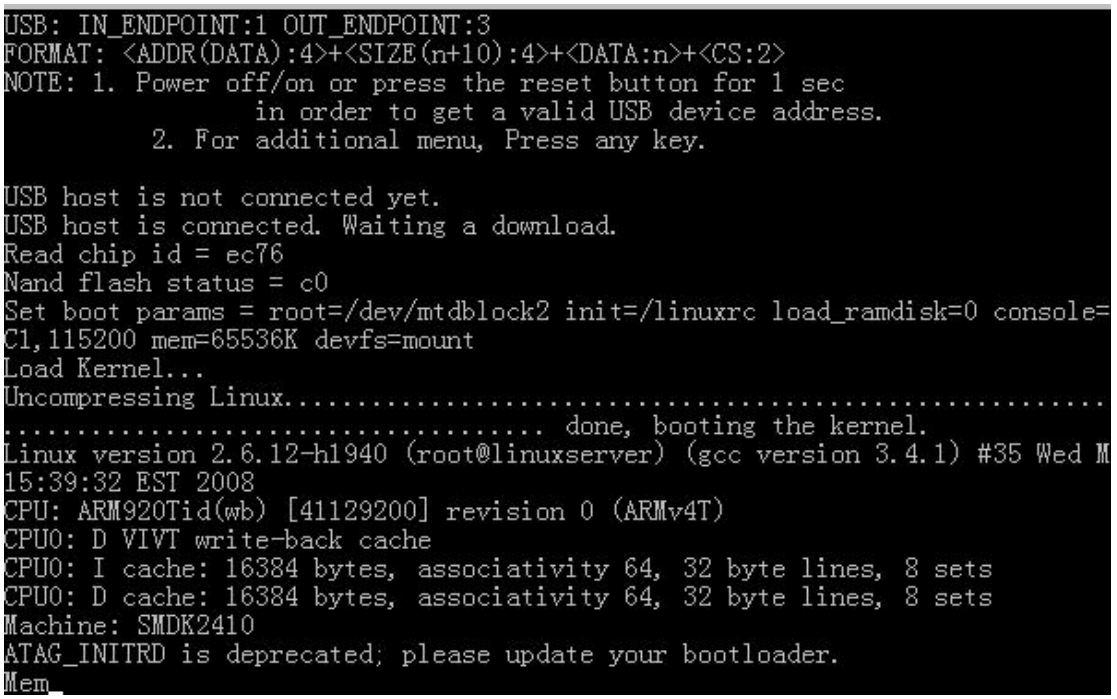

图 5.3 **嵌入式** Linux **启动界面**

进入系统成功后，可以从超级终端看到和整个 Linux **操作系统的运行终端中一样的命令符#，表示系统已经准备好接受操作了。此时我们可以在**/sbin **目录下发现自己后加进去的** tserver **应用命令，如图** 5.4 **所示。**

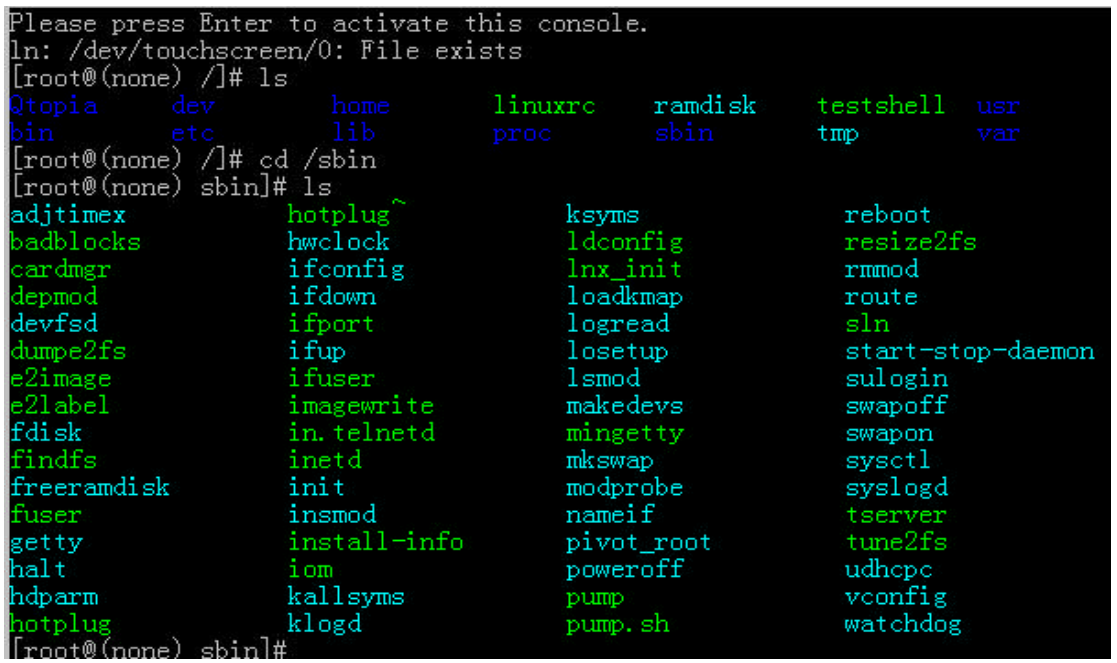

图 5.4 **正常进入嵌入式系统后的界面**

5.2.3 开发后客户端的实验步骤

服务器端启动后，在客户端（即任意联网的 Windows 操作系统的 PC 机端）安装客户端程序 client，就可以直接运行拷入的 VC 6.0 相应可执行文件，进入客户端身份验证界面，如图 5.5 所示。

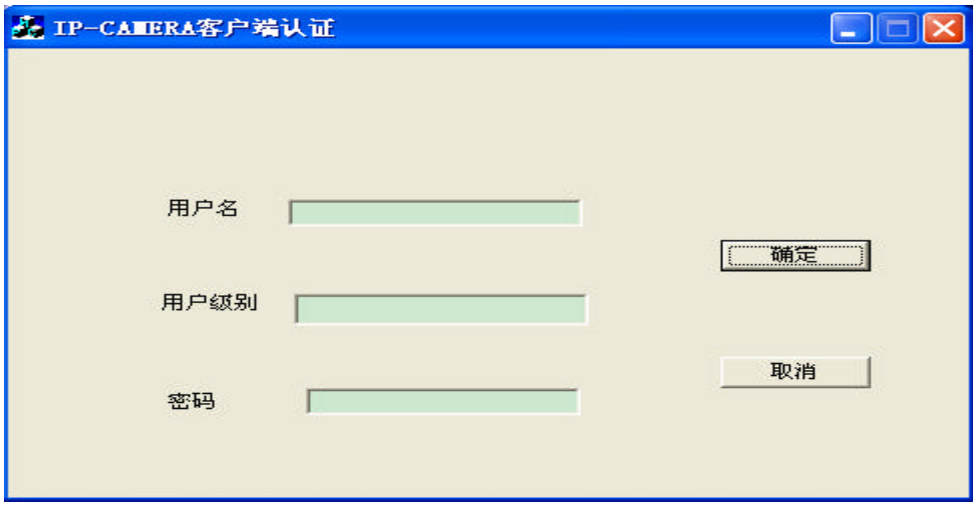

图 5.5 客户端身份验证界面

系统确认客户端授权信息正确后再进入操作界面，如图 5.6 所示。

图 5.6 用户登陆成功界面

成功登陆后，点击确定即可进入监控界面。此处以管理员监控界面为例，如图 5.7 所示。首先输入要检测的服务器的 IP 地址（此点在以后的开发过程中，可以将 IP 地址与服务器所在地点或特征等其它让人易于记忆的关键词绑定，只要输入关键

词，系统自动引入其 IP 地址，然后进行连接）。然后在接收文件栏或是发送文件栏进行操作。图 5.8 即为客户端接收文件选择保存路径及重命名的操作界面。

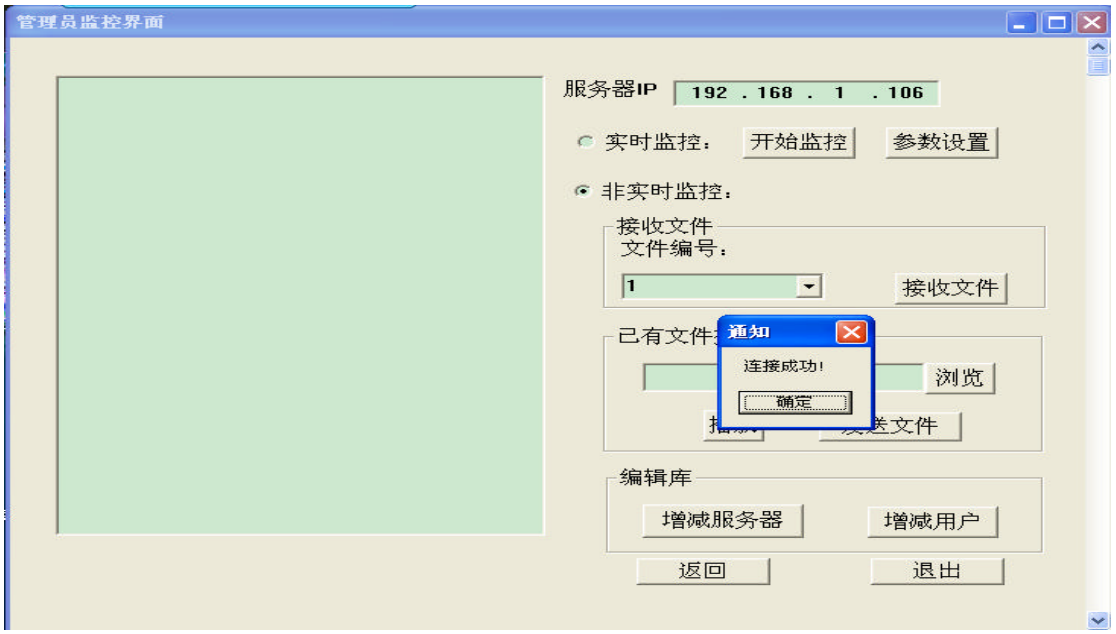

图 5.7 管理员客户成功发出接收文件编号为 1 的请求

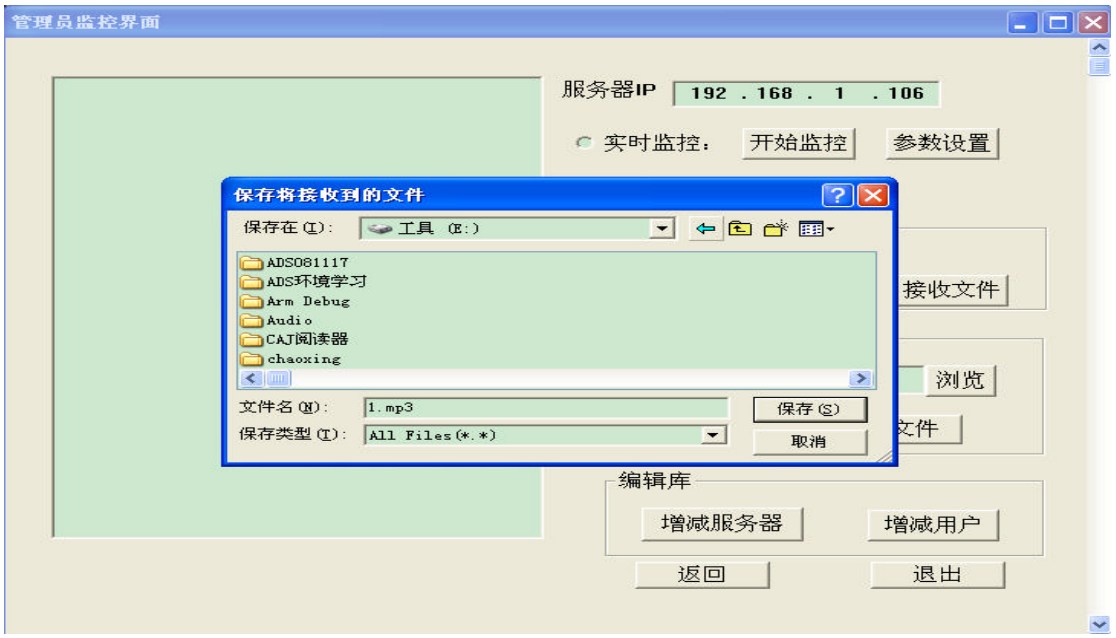

图 5.8 客户端接收文件时的选择路径及重命名操作界面

5.3 实验结果

5.3.1 服务器端的实验结果

此时运行服务器端的应用程序命令 tserver，服务器端即启动了，开始监听客户端的请求信息。当接收到客户端的请求时，首先判断是需要服务器端接收数据还是发送数据，然后再调用相应的线程来分别作出处理。图 5.9 所示为服务器端传输模块应用程序运行成功时的界面。前面一个操作是响应一个客户端（对应 IP 地址为 192.168.1.130）的发送请求，后面一个操作是响应另一个客户端（对应 IP 地址为 192.168.1.116）的接收请求，且这服务器端是并行处理这两个客户端的请求的。

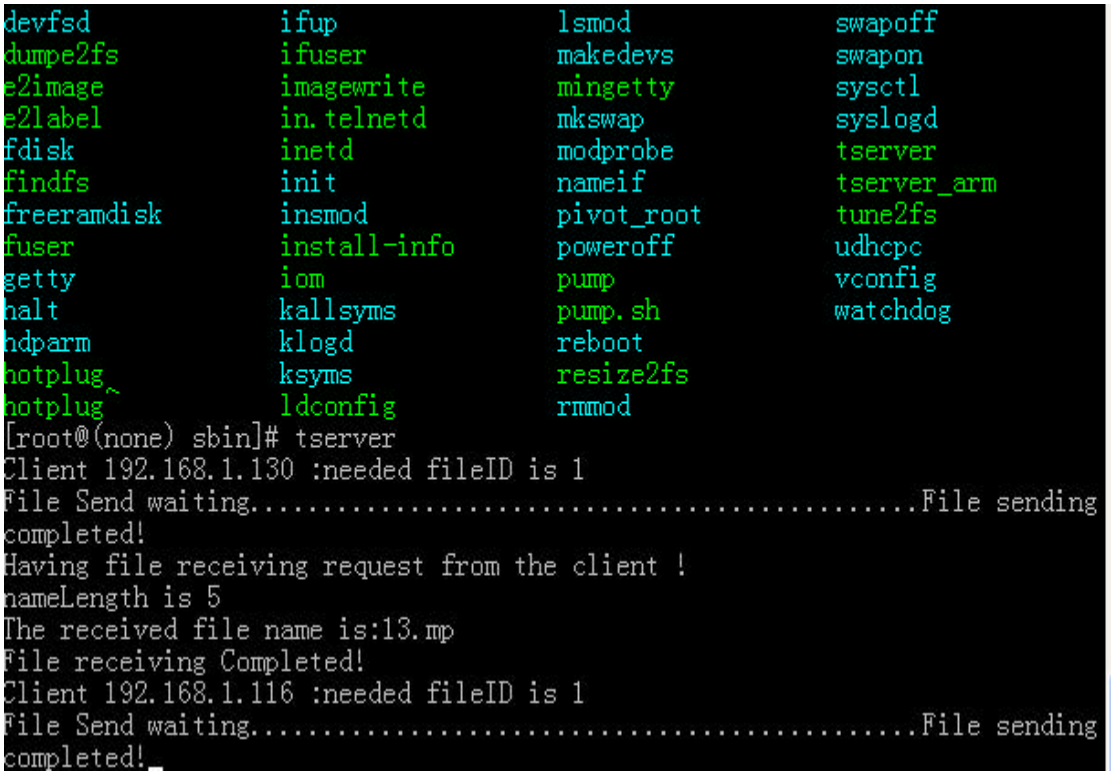

图 5.9 运行服务器成功界面

5.3.2 客户端的实验结果

注意，由于嵌入式系统在开发后通常就已成形。一般的操作都是从客户端发出的。也即只有客户端才是用户，才是主动请求端。当客户端需要某些文件时，向服务器端发出请求，服务器端响应成功之后发送出相应的文件。当客户端需要上传一

些文件到服务器端存储时，也要先向服务器端发出请求，服务器端响应成功之后开始接收相应的文件。如图 5.10 **和图** 5.11 **分别为客户端接收和发送文件成功的界面。**

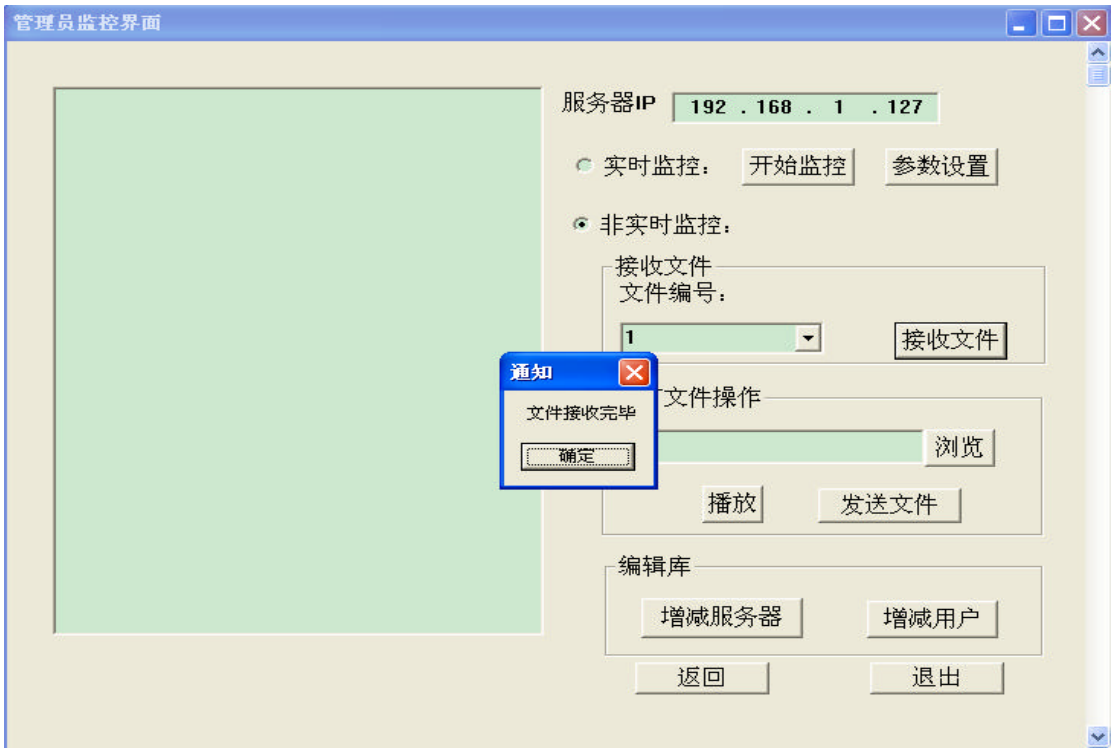

图 5.10 **客户端文件接收完毕**

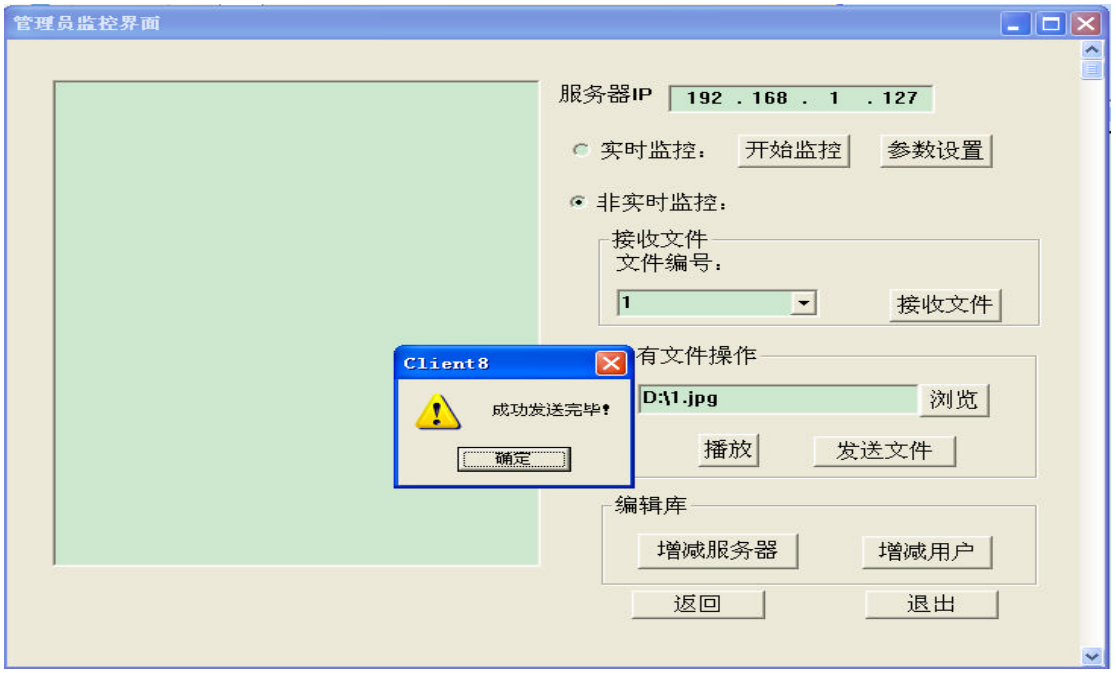

图 5.11 **客户端成功发送完毕**

鉴于多方面的原因，目前此客户端的操作界面中仍有一些模块没有完全实现。比如编辑库这一栏，原本是用于管理员客户来编辑合法用户的，此时还没有实现。但论文基本上实现了数据传输模块，而且曾经实验过传输大小为上百兆的文件，传输的效率非常高，不仅速度快而且十分准确。

图 5.12 接收到的图像 1.jpeg 在 PC 机下的显示

而从服务器端传输到客户端的几张图片 1.jpeg，2.jpeg，3.jpeg 图像在 PC 机下的显示结果分别如图 5.12，5.13，5.14 所示。其中图 5.12 是在一个背景下采集到的，就是图 5.1 开发总体环境实物图中显示器中检测到的图像。而图 5.13 和 5.14 是同在另一背景下进行检测的，只是调焦和天气变化会对高清结果有一定的影响。从整体上看，还是达到了预期的较高清的效果。

图 5.13 接收到的图像 2.jpeg 在 PC 机下的显示

图 5.14 接收到的图像 3.jpeg 在 PC 机下的显示

5.4 本章小结

嵌入式系统的开发环境是比较特殊的，从刚接触到这个概念，到熟悉地对其进行开发是需要逐渐过度的。基于本人的一些学习过程，本章主要对开发 IP-CAMERA 这个嵌入式系统的开发环境进行了详细的介绍，包括硬件上的具体设置、软件包的安装，同时还仔细地描述了在交叉编译环境下对开发板所需的软件执行代码的烧写过程，如系统引导程序 bootloader 的烧写、内核 kernel 的烧写及文件系统 fs 的烧写。最后还对这个跨平台的开发环境进行了进一步地分析其相关知识点在理解上的加强，从而从整体上对整个系统的开发有个全面的理解。

鉴于数据传输模块在 IP-CAMERA 中的重要性，本章对数据传输模块进行了初步地实现。使得服务器端（开发板 YLP2440）与客户端（任意 PC 机）可以互相发送及接收数据。且鉴于服务器端采用了并发的服务模式，可以同时一对多地对客户端进行响应。本章先后详细地描述了服务器端和客户端的开发流程和具体的操作过程，从而实现了基本的传输模块功能，网络上数据的高速传输得到了成功地保障。

6 总结及展望

6.1 论文完成的工作

监控技术经过多年的发展，从原理、方法到系统的实现上都形成了一套比较完整的体系结构。随着网络的迅猛发展，网络摄像机 IP-CAMERA 逐渐成为监控技术中备受关注的研究对象，其功能实现逐渐向更加高清的方向发展。IP-CAMERA 的研究涉及到嵌入式系统、网络通讯等多种技术，是当前最热门的一项课题研究。高清 IP-CAMERA 要求可以检测到具体每个事物的详细特征，如行人的面部模样、车辆的车牌号码等。

本文对高清 IP-CAMERA 的高清方面考虑地比较多，就其涉及到的相关知识点进行了详细地分析，并成功实现了基本传输模块。具体说来，做到了以下几点。

1. 详尽分析了高清 IP-CAMERA 所属的嵌入式系统的特征、开发环境及流程。特别针对本文所选的嵌入式 Linux 系统、ARM9 微处理器的基本架构及基本开发流程进行了具体的开发过程介绍。包括交叉编译环境的建立、ARM9 微处理器的引导启动程序 bootloader 的烧入、嵌入式 Linux 的内核的编译及烧入、Linux 文件系统的编译及烧入流程、新增应用程序的编译及向文件系统的融入过程等。
2. 对服务器端/客户端的通讯模式进行了详细的介绍。结合本论文中的系统，体现了其非对等作用和完全异步的通信模式这两大特点。具体说来，其体现包括服务器对客户端的一对多的通信模式、服务器端的并行处理模式、服务器与客户端模块之间的独立性等。
3. 实现了 Windows 与 Linux 之间的跨平台通讯。包括两端系统硬件上网络接口的统一、Windows 端系统向 Linux 端系统发送与接收数据、Linux 端系统向 Windows 端系统发送与接收数据。
4. 实现并在一定程度上体现了 Linux 下强大的网络功能。其中应用的最主要的模块就是符合 TCP/IP 协议的套接字结构体，具体体现在服务器端的 C 代码

网络编程及客户端的 VC 代码网络界面编程中。

6.2 研究展望

虽然目前已有许多监控系统被运用于实际生活中，但是满意度高的只是一些对图像清晰度要求不是很高的场合。而对于那些对图像清晰度较高的场合，比如出入人口比较多、且易发生一些犯罪相关现象的场合，需要做更多的努力。鉴于高清监控的目标就是让监控人员能够更清楚地看到被检测对象，比如能够清晰识别每一个过往之人的面部特征、每一辆过往车辆的车牌号码、每一样事物的外形特征、每一件短暂时间内发生的事情的来龙去脉等，就会给检测带来很大程度的方便。如果在高清检测方面发展强大了，一方面，于警方而言，对已发非法案件的侦破提供了强有力的证据。另一方面，于社会而言，臻于成熟的强有力的监控技术将对社会上不良分子进行精神上的救愈，将大大地减少不良事件乃至犯罪案件的发生。可见，在当今纷繁复杂的社会中，高清监控将在其中占据着十分重要的应用地位。

随着应用上的推广，高清监控系统将在社会中承担着越来越重大的任务的同时，它的性能也需要得到不断的提高。鉴于本文的研究，主要有以下几个方向的展望。

1. 实现报警监控。即当检测到预设的非正常现象时，自动拉响报警器。这一项可用在防范恐怖袭击的状况下，使得检测人员在同时检测多个检测点时通过声音来判断即可，不用全部使用有限的人力观察力，使用起来更加方便，检测起来也将更加准确。
2. 对新增的授权用户信息进行实时数据库更新。目前联网的许多软件都已经实现了此项功能，这样的话，能够充分体现在因特网上新注册合法用户的实时生效性。只要新增加一个合法用户，在客户端运行其应用程序时，立马检测是否有数据库更新状况，更新之后再进行应用操作。使得用户在任何地方的 PC 上都可以及时地进行检测工作。
3. 新设敏感的红外检测功能。目前禽流感、非典、猪流感接踵而至，威胁着全世界人的健康。在这些状况下若使得检测系统实施对温度高低的检测，同时

实施相应的报警措施，就可以大大节省人力。

4. 在编码方面可以通过更加优化的算法来实现高效的压缩编码，比如利用高效性的 H.264 进行编码等；同时在智能方面的发展也可以更进一层楼，使得系统更加易于被用户使用和操作等。

致 谢

本文是在陈建文老师的悉心指导下完成的，从文献的查阅、论文的选题、撰写、修改、定稿，我的每一步都包含着陈老师的指导和关注。在做毕业设计的过程中，陈老师主动为我提供和推荐学术资料，还在自身工作繁忙的情况下，依然多次不辞辛苦的为我做耐心讲解和指导。在此谨向陈老师表示衷心的感谢和诚挚的敬意。

转瞬间，两年的研究生生活即将逝去，但这两年的生活很充实，也很开心，更学到了很多东西。这一切都离不开老师们的辛勤培养。是你们教授给我知识，让我掌握了生存的技能和本领；是你们教导我为人治学，令我养成了严谨的生活态度。同时也很感谢学院领导和系、辅导员对我无微不至的关怀和爱护，给我鼓励和帮助，教会我更好地做人做事。

感谢同一实验室的贺德华老师，在学习和研究开发上给予了我无私的指导和帮助。

感谢同实验室的王奕、张翠翠、黎杨梅、廖洲、董威、严建涛、吴文生、张洁、辛颖等同学，他们在我的科研开发过程中给予了我莫大的帮助。

感谢所有的老师、同学和家人，他们对我的关心、支持和帮助，将永远鼓舞和激励着我！

参考文献

[1]. 马昕. 视频监控系统的现状及发展趋势. 中国公共安全, 2004.12

[2]. 代亮，刘大成，王芹华，郑力. 远程多摄像机协同跟踪实现方法研究. 制造技术与机床，2004.9

[3]. 周培明. 嵌入式网络摄像机的研究. 天津工业大学硕士论文. 2008

[4]. Anders Laurin. Networked Video Surveillance and Compression Technology. Axis Commnications, 2002

[5]. 杨亚雄. 嵌入式 Linux 网络视频监控系统研究与实现. 武汉理工大学硕士论文. 2008

[6]. M. Greifenhagen, D. Comaniciu, H. Niemann, and V. Ramesh. Design, analysis and engineering of video monitoring systems: An approachand a case study. Proceeding of the IEEE, 2001,89(10)

[7]. Baer W G, Lally R W, An open-standard Smart Sensor Architecture and System for Industrial Automation. Oceana Sensor Technologies, 2000

[8]. （日）小野定康，铃木纯司著，叶明译. JPEG/MPEG2 技术. 北京：科学出版社，2004

[9]. TD-BD-JPEGH264IPCam User's Manual，Tokyo Electron Device Limited Corporation

[10]. （英）理查森（Richardson, I.E.G.）著，欧阳合，韩军译. H.264 和 MPEG-4 视频压缩：新一代多媒体的视频编码技术. 长沙：国防科技大学出版社. 2004

[11]. 刘亚珂. 基于 Web 的嵌入式网络监控系统的设计与实现. 郑州大学硕士学位论文，2006.5

[12]. Epplin J., Linux as an Embedded Operating System, Embedded Systems Programming, 1997(10)

[13]. 徐虹，何嘉，张钟澎著. 操作系统实验指导：基于 Linux 内核. 北京：清华大学出版社. 2004

[14]. （美）雅默著；韩存兵，龚波改编，构建嵌入式 Linux 系统. 北京：中国电力出版社，2004

[15]. 王学龙. 嵌入式 Linux 系统设计与应用. 北京：清华大学出版社，2001

[16]. Laurence T.Yang. Embedded software and systems. Berlin: Sringer, 2005

[17]. Szymansli J W. Embedded Internet Techno logy in Process Control Devices. Proto Portugal, WFCS-2000

[18]. M. Tim Jones. TCP/IP Application Layer Protocols for Embedded Systems. CHARLES RIVER MEDIA, 2002

[19]. 黄守君. 基于嵌入式 WEB 服务器的网络摄像机设计与实现. 广东工业大学硕士学位论文，2007.5

[20]. Jeremy Bentham. TCP/IP Lean Web Server for Embedded System. CMP Books, 2000

[21]. 林宇，郭凌云著. Linux 网络编程. 北京：人民邮电出版社. 2000

[22]. 李卓恒，瞿华等著. Linux 网络编程. 北京：机械工业出版社. 2000

[23]. 欧立奇，刘洋，段韬著. 程序员面试宝典. 北京：电子工业出版社. 2006

[24]. 李蕊. 基于嵌入式 Linux 的网络摄像系统研究和实现. 天津大学硕士学位论文，2006.1

[25]. （美）鲁宾尼（Rubini.A.）著；聊鸿斌等译. Linux 设备驱动程序. 北京：中国电力出版社. 2000

[26]. Alessandro Rubini and Jonathan Corbet, Linux Device Drivers, 2nd Edition, O'Reilly, 2002

[27]. 郭景锐. 嵌入式网络视频监控系统监控端软件设计与实现. 西南交通大学硕士学位论文，2006.3

[28]. 明日科技 宋坤，刘锐宁，马文强. Visula C++视频技术方案宝典. 北京：人民邮电出版社. 2008

[29]. 孙海民著. 精通 Windows Sockets 网络开发：基于 Visual C++实现. 北京：人民邮电出版社. 2008

[30]. （美）杰夫瑞（Jeffrey, J），（法）克里斯托夫（Christophe, N.）著；葛子昂，周靖，廖敏译. Windows 核心编程（第 5 版）. 北京：清华大学出版社. 2008

[31]. Andrew N. Solss, Dominic Symes, Chris Wright. ARM System Developer's Guide: Designing and Optimizing System Software. Beijing University of Aeronautics and Astronautics Press, 2005

[32]. ARM Limited. ARM Architecture Reference Manual, Arm DDI 0100E, U.K. 1999

[33]. Samsung Electronics Co.Ltd., S3C2410X 32-Bit RISC Microprocessor User's Manual, Rev. 1.2, 2003

[34]. 孙琼著. 嵌入式 Linux 应用程序开发详解. 北京：人民邮电出版社. 2006

[35]. 马喜廷，梦荣芳. IP 网络摄像机. 电视技术，2003.5

[36]. QQ2440 用户手册，广州友善之臂计算机科技有限公司

[37]. 怀石工作室著. LINUX 上的 C 编程. 北京：中国电力出版社. 2001

[38]. （美）庞特著；陈继辉等译. C 语言嵌入式系统开发. 北京：中国电力出版社，2003

[39]. 刘先虎. 网络摄像机语音编解码及网络通信与控制研究. 浙江大学硕士学位论文，2006.2

[40]. 孙天泽，袁文菊，张海峰. 嵌入式设计及 Linux 驱动开发指南——基于 ARM9 处理器. 北京：电子工业出版社. 2005

[41]. Anddrews Tanenbaum, Modern Operation System. New Jersey:PrenticeHall, 1999

[42]. 探矽工作室著. 嵌入式系统开发圣经. 北京：中国青年出版社. 2002

[43]. Michael K. Johnson, Erik W. Troan 著. Linux 应用程序开发. 北京：人民邮电出版社. 2006

[44]. 鸟哥著. 鸟哥的 Linux 私房菜——基础学习篇. 北京：科学出版社. 2005

[45]. 张炯著. Unix 网络编程实用技术与实例分析. 北京：清华大学出版社. 2002

[46]. YLP2440 使用手册，深圳市优龙科技有限公司