## 摘要

1. 研究背景及意义

随着生活水平的逐步提高，以及安全意识的增强，人们对安全监控的需求更加强烈，各个公共场所到处可见摄像头的身影，比如学校、街道、小区、公交站台、火车站等。可以说安全监控已经与我们的生活息息相关了，而传统的视频监控仅仅可以进行单一的视频监控，已经无法满足如今多样化的需求。视频监控拥有着巨大的市场潜力，在如此强烈的需求刺激下，视频监控也得到了快速的发展，视频监控已经发展成了一个融合多种技术的领域，比如数字视频、网络通信、嵌入式技、图像处理技术、人工智能技术等等。近年来，由于国家对人工智能技术的重视，以及各高校，各大公司对人工智能的投入，促使人工智能的发展进入了一个爆炸时期，人工智能技术相比过去有了较大发展。人工智能赋予了机器新的生命，机器不再是以前的单单完成一些简单的特定工作，如今，机器可以实现模拟人类的复杂行为，在某些方面像人类一样智能。比如谷歌研发的围棋博弈机器人alphago接连击败世界围棋高手。人工智能技术的发展俨然已经到达了一个新的高度。将人工智能技术与视频监控相结合，使视频监控更加的智能。智能视频监控系统利用人工智能技术、计算机视觉技术对视频数据中的每帧图像进行分析、处理，在没有人为干预的情况下，自动对图像进行目标定位、识别、提取等操作，并对提取的目标利用人工智能算法进行相关业务分析，当判定为异常情况时可以及时报警通知工作人员进行危机处理，在这个过程中要最大限度地降低误报率和漏报率。与此同时，嵌入式平台由于其成本低，方便部署，体量小等特点已经逐步的进入了普通家庭。在科学技术快速发展的今天，视频监控也更加的智能、人性、方便、通用。

进入21世纪，社会高速发展，人们的生活节奏也愈来愈快，大部分人在家中的时间很少，当主人不在家中时，家中的财产存在极大的安全隐患，人们希望不在家中时也可以随时了解家中的一举一动，当家中发生异常情况时，可以及时的得知。

在此背景下，本文设计了一款家用智能监控系统，本系统基于嵌入式linux平台，采用嵌入式arm9处理器，实现对视频信息的实时采集和传输，成本较低。同时移植了嵌入式web服务器，用户只需通过浏览器便可方便地观看到监控画面，满足了用户随时掌握家中状况的需求，并且没有pc和手机等设备的限制，简单实用。将采集到的视频信息传输至入侵检测模块服务器进行处理和存储，当发现异常情况时，会及时报警和通知用户，体现了智能化和人性化，同时也保障了用户的财产安全。

1. 现状和发展趋势

视频监控在国内发展也不过短短几十年，但在其他科学技术发展的带动下，视频监控也在这短短的几十年飞速发展。视频监控主要经历了以下发展阶段：模拟视频监控系统、数字视频监控系统、网络化数字视频监控系统、网络智能化数字视频监控系统。

在九十年代初期的模拟视频监控系统，还是采用模拟信号传输，视频监控系统主要由摄像头、视频矩阵、监视器等组成，这种监控系统使用模拟传输方式，传输距离短，可靠性不高，抗干扰能力也不强，由于当时互联网在中国并没有被广泛应用，所以那时的监控系统还不能并入internet。

在九十年代中期的数字视频监控系统已经发展成了数字信号处理和传输，随着计算机技术和图像处理技术的高速发展，视频采集速度和质量也大幅提高，以及采用高分辨率的监控显示器，组成了新一代的数字监控系统。数字监控系统可以对采集到的数字图像信号进行处理，比如运用数字图像处理技术对视频信号进行处理，可以让系统更加智能。但是限于图像采集的单片机以及处理图像的计算机的处理能力，当时的系统还是有一定的局限。

在九十年代末期，随着计算机硬件技术的发展以及互联网的普及，视频监控也发展成了网络数字化视频监控。网络数字化视频监控利用互联网，采用视频流的传输方式在网络中传输视频信息。

步入21世纪，由于深度学习、机器学习等人工智能算法的发展，视频监控在智能识别等人工智能方向的发展更进一步，视频监控更加的智能化，识别精度与速度也随着新的优秀的算法的不断提出和计算机硬件技术的发展而提高，智能视频监控在医疗、交通、民用、军用等各大领域大放异彩。

在国外，基于嵌入式web的视频监控的发展更加趋于无线化和智能化，国内在人工智能相关领域的发展还是相对比较滞后的，各大高校近几年才纷纷开始开设人工智能这一研究方向，光这一点就比国外晚了很多年。在智能视频理解方面，国外是一直处于领先地位的，例如美国IBM公司早在2006年即发售了一款安全软件S3（Smart Surveillance System）,已成功实现了远程的智能化监控功能[1]。此外，还有三星SNO-6084RP、安讯士AXIS系列产品等。这些产品大都拥有强大的功能、优越的性能，但却属于中高端产品，价格昂贵，普通用户是无法接受的，因此在性价比上不适合于家庭室内这种小型场所。

国内在相关方面的研究开展虽然相对较晚，在智能化、网络化方面与国外相比有一些差距，但也是紧跟国际研究步伐。例如国内的海康威视、大华等公司的产品都有着不错的性能，但其应用领域还是集中在银行、大型超市、交通等行业场所，随着数字化家庭网络的逐渐普及，个人场所的应用有很大的发展空间[2]。

发展趋势

随着互联网、嵌入式及人工智能等技术的不断发展，视频监控系统已经发展成基于高性能的嵌入式设备，打造更加快速、便捷、智能的监控物联网络。嵌入式系统软硬件可裁剪，适用于对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统，根据摩尔定律，硬件的处理性能越来越高，而价格则会更低，这对于嵌入式的发展是极其有利的。与此同时，国际上的优秀的人工智能算法不断被提出，改进，其准确度和运行速度都有了很大提高。近几年，我国又全面覆盖无线4G网络，而新的5G网络的建设也已经箭在弦上，无线网络的覆盖率和传输速率大幅提高。这些技术的快速发展有效的加快了嵌入式视频监控的更新换代，同时也奠定了远程监控技术的未来发展方向：

1. 嵌入式化

嵌入式系统是以应用为中心，以计算机技术为基础，并且软硬件可剪裁，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【4】。可以为远程视频监控打造满足不同用户需求配置的专用嵌入式系统，机动、高效。

1. 物联网化

将视频监控与无处不在的互联网相结合，实现物联网化，这样用户就可以随时随地的观看监控画面。

1. 智能化

智能视频监控指利用智能行为识别算法识别和检测监控画面中的运动物体，进而实现相应的报警功能。

之前，监控系统还只是存在于大型公共场所，比如学校、交通、车站等地。但是，近几年，随着人们安全意识的增强和监控系统成本的下降，监控系统也逐渐推广到小型的个人场所，对于人们生命财产安全起到了极其重要的作用。嵌入式远程视频监控蕴藏着巨大的应用前景和商业价值，同时，也需要很长的发展之路去走。

3.课题研究内容

本文设计一种基于嵌入式web的智能监控系统。采用嵌入式web服务器实现视频的bs传输，使用移动目标检测算法来检测目标，并实现报警。主要有以下部分：基于arm9的硬件平台，采用linux嵌入式操作系统，uvc摄像头通过usb与arm相连，采集视频是基于linux的v4l2驱动框架，将采集到的图像信息压缩成jpeg格式经过web服务器传输给浏览器。用户只需打开浏览器访问相应的url便可观看到监控画面。同时我们对采集的图像经过移动物体检测算法的检测，来实现报警功能。

课题研究内容主要有以下部分：

（1）

Arm开发板的硬件环境

（2）

嵌入式Linux操作系统的布置，包括uboot的移植，nand flash，nor flash，串口等驱动的移植

（3）v4l2摄像头数据采集的编程

（4）摄像头的移动目标检测以及移动目标的人体检测

（5）嵌入式web服务器，包括前端页面的制作等。以及对视频信息的传输。

4.章节结构与内容安排

根据课题研究内容，本论文共分为七个章节。

第一章:绪论。本章阐述了论文选题(智能视频监控系统)的背景和意义;

叙述了视频监控领域的相关概念、发展历史以及国内外研究现状;最后介

绍了所做工作以及论文的结构安排。

第二章远程视频监控系统总体设计

通过功能需求分析，设计了监控系统的整体架构和实施方案，在系统硬件设

计方案中完成硬件框架设计和设备选型，软件方案设计中主要是应用程序详细设

计和具体实现。

第三章嵌入式开发环境的构建

搭建嵌入式Linux系统的软硬件开发环境，通过建立的基于Ubuntu的交叉

编译环境，完成了嵌入式Linux2.6.30系统软件平台的构建，以及摄像头、无线网

卡驱动程序的移植。

第四章服务器端应用程序设计与实现

进行主程序设计，对采集模块、压缩模块、传输模块和运动目标检测模块进

行了详细设计。通过移植Thttpd服务器和MJPG-streamer流媒体服务器实现了基

于B/S结构的视频监控，利用OpenCV函数库实现了运动目标的检测功能。

第五章客户端网页设计与功能实现

介绍Web界面总体设计和涉及到的编程技术;详细阐述了登录界面和视频

监控界面的网页制作，以及登录验证和视频显示等功能的实现;

第六章系统测试与结果分析

搭建软硬件和网络测试环境，进行远程视频监控的功能测试，测试的功能项

包括登录验证、视频显示、邮件提醒等，以及对浏览器兼容性的测试，测试完毕

进行测试结果分析。

第七章总结与展望

对本课题所做的工作内容进行总结，指出系统仍需完善和改进的方面，并在

此基础上提出了课题和下一步的工作目标。

第二章 家用视频监控系统设计

2.1需求分析

本文设计的家用监控系统主要针对小型场所例如家庭室内、门店等这种场所，主要应用场景是当用户出门在外，家中无人或者家里有老人、小孩等需要监护的情况。由于主要应用于普通家庭室内，所以其开发成本一定要低，这样才能被大众家庭所接受，因此选用低成本、高性价比的嵌入式系统是非常有必要的。此外，由于是家庭在室内，用有线传输的话，需要对室内进行改装，费时费力，且影响美观，因此需要实现无线wifi传输，需要嵌入式系统支持无线wifi，这样便可与家用路由器很方便的实现组网。此外，还需要一定的智能化，能够对摄像头监控的区域进行运动目标识别，实现智能监控【】。根据家用智能监控系统的应用场景和需求，确定了监控系统需要满足以下几方面要求：

1. 视频数据的实时采集、传输、显示。

视频数据要实现实时的采集、传输，以满足用户观看即时监控画面的需求。这需要嵌入式系统可以实时采集监控画面，并且对传输带宽等都有一定的要求。

1. 数字化的视频采集

数字化的视频采集方便图像的压缩传输，方便图像的智能算法处理，且在传输时，抗干扰能力强，出错率低，有利于网络传输。

1. 无线网络传输

视频数据的传输需要接入互联网，用户在远程打开浏览器便能观看监控画面，因此设备要支持无线wifi传输。

1. 一定的智能性

对采集到的图像数据进行智能算法处理，比如运动目标识别，当发现异常情况时，自动报警，发邮件或短信提醒用户，从而实现智能化监控。

1. 一定的权限限制

当用户观看监控画面时，需要用户名密码正确登录才能观看，而对于用户名密码错误的访客拒绝其观看监控画面。

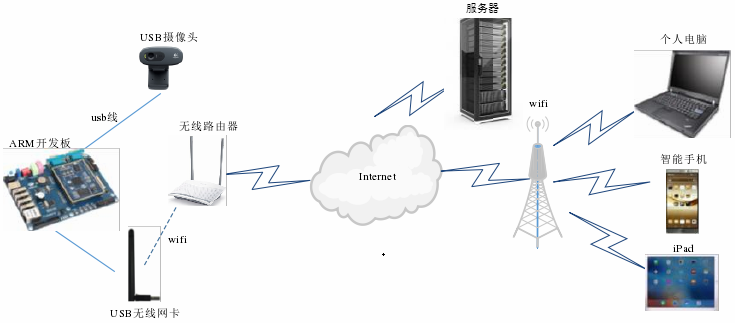
1. 具有文件保存和文件自动清除功能

用户可以设置监控画面的保存时间，当超过保存期限，系统会自动清除之前的监控信息，以此防止存储装置的空间溢出，为保存新文件预留存储空间。

2.2系统总体架构设计

为增加系统的灵活性和可扩展性，采用家庭局域网组网方式，通过路由器连入因特网，在arm板上移植web服务器，远程pc或电脑通过浏览器便可很方便的观看到实时的监控画面。

同时，使用pc平台的服务器对采集到的图像进行复杂的人工智能算法处理，比如动态物体检测识别，发现有异常入侵进行报警。



2.3硬件方案设计

根据系统的需求分析，家用智能监控系统的硬件模块分为以下部分：视频采集模块、视频处理模块、视频传输模块、入侵检测模块、报警模块、web显示模块。本课题基于ARM-LINUX平台开发，通过视频采集模块USB摄像头采集视频数据，将采集到的数据经过视频处理模块进行压缩**\*\*\*\*\*，**然后通过视频传输模块传输给pc机上的入侵检测模块进行入侵检测和存储。

视频采集模块：在arm-linux平台下，通过V4L2技术获取usb摄像头采集的视频图像。

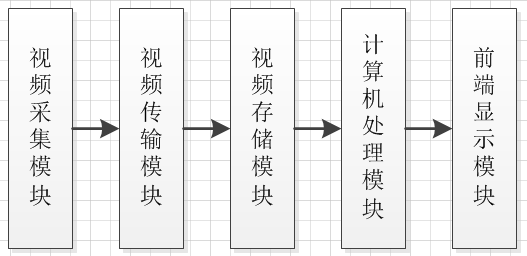
视频传输模块：将压缩后的图像通过TCP\UDP协议和无线传输协议实现传输。

视频存储模块：将获取的视频图像存储在服务器磁盘中，并且能够自动清理超过存储期限的数据。

计算机处理模块：在计算机服务器上对图像数据进行智能算法处理，检测和识别移动目标。

前端显示模块：在pc机、移动手机、pad上的浏览器进行显示。

Web显示模块：在arm上移植嵌入式web服务器，使用http超文本传输协议，在html页面中嵌入applet插件，实现web显示，用户只需在浏览器上输入对应的URL地址，便可访问制作好的web页面。



2.3.1硬件设备选型

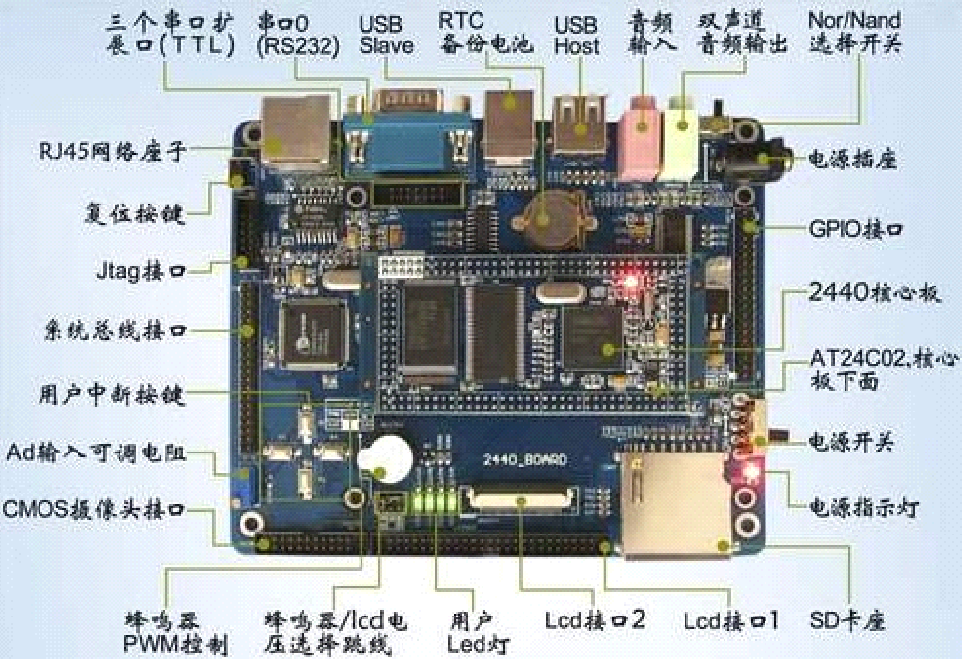
系统中硬件的选取需要考虑其功能需求、应用场景、开发成本、开发周期等因素，可以说是本课题系统的根基，对最终产品的性能有着决定性的影响，因此，对硬件的选取要十分慎重。在满足系统功能需求的前提下，在控制一定成本范围下，选取性价比较高的合适的硬件来提高系统的稳定行。

(1)嵌入式开发板的选择

C 本系统是基于ARM的嵌入式平台进行设计开发的，硬件平台不仅需要加载嵌入式操作系统，还要实现视频数据的采集、处理和传输，所以选择一款合适的硬件平台非常重要。目前ARM系列中有ARM7 , ARMS , ARM9E和ARM 11等多个典型产品「13]，考虑系统成本及多媒体处理能力，本文采用性价比相对较高的ARM9处理器，在保证视频画面质量和传输实时性的前提下，可以快速地进行图像处理。为了节约设计开发时间和提高硬件集成的稳定性，将更多精力集中在软件设计和系统功能实现方面，所以直接选用ARM9开发板进行设计开发。

本课题采用天嵌公司的tq2440开发板，其核心处理器为ARM9处理器架构，主频可达400MHZ，SDRAM达到64M，支持多种市面上的flash，拥有丰富的外部接口，方便扩展，是三星公司推出的一款功能强大，功耗极低的ARM9嵌入式CPU，支持Linux系统，具有高性价比、高可靠性和稳定性的特点。

硬件接口图如下：



表TQ2440开发板主要硬件资源

|  |  |
| --- | --- |
| TQ2440开发板主要硬件资源 | |
| 硬件 | 描述 |
| CPU处理器 | Samsung S3C2440AL，主频400MHz（最高可达533MHz） |
| SDRAM内存 | 板载64MB SDRAM(标准配置) |
| Nand Flash | 板载256MB Nand Flash（标准配置） |
| Nor Flash | 板载2MB Nor Flash |
| CorePower | 1.25V核心电压供电 |

（2） USB摄像头的选择

C对整个视频监控系统而言，作为系统前端的视频采集设备，摄像头采集的视

频图像的清晰度、实时性和稳定性都是十分重要的，影响着整个系统的性能。考

虑到本监控系统的适用场景，本课题选用的摄像头是罗技C270摄像头，采用

CMOS感光元件，可达300万像素，USB2.0接口，支持YUV422和RGB24两

种数据格式，最大采集帧数为30帧/秒。选用该摄像头的一个重要原因是该摄像

头属于UVC CUSB Video Class)摄像头，拥有标准的UVC驱动程序，属于Linux

内核的一部分，无须安装额外驱动【】。

（3） USB无线网卡的选择

C在选择无线网卡时需考虑其传输速率及稳定性等一些因素，本课题USB无线网卡采用雷凌RT5572USBWiFi无线模块。它是一款高性能的USB 接口2.4G/5G 无线模块，符合IEEE802.11N标准并向下兼容IEEE802.11B/G 标准，支持IEEE 802.11i安全协议，以及IEEE 802.11e 标准服务质量，其极高的兼容性，能够快速、方便的与无线设备互相联通，支持最新的64/128 位WEP 数据加密，支持WPA-PSK/WPA2-PSK，WPA/WPA2 安全机制，无线传输速率高达300M。能够实现ARM开发板与无线路由器的互联网连接，完成视频数据的无线传输「16]。虽然会受到距离或墙体阻隔等因素的影响，但在无线网卡能够覆盖的有效范围内还是具有可行性的，其稳定性和视频质量较好，能够达到无线传输的目的。

2.4软件方案设计

软件设计与实现在整个系统中也极其重要，好的软件设计对整个系统的运行有着很重要意义。基于软件模块化和结构化的设计思想，整个监控系统的软件结构分为嵌入式操作系统软件和上层应用软件两部分。嵌入式操作系统软件部分主要实现在linux内核的基础上的USB摄像头驱动、无线网卡驱动等外围设备驱动的移植，在此基础上实现视频采集和无线信息的传输。有了操作系统软件部分的实现，上层应用软件才能得以实现。应用软件主要有web服务器、流媒体服务、入侵检测、web页面显示这几部分组成。



1. 嵌入式linux操作系统软件

Bootloader是在操作系统运行之前执行的一段程序，这段程序会初始化硬件设备、建立内存空间的映射表，建立适当的系统软硬件环境，为最终调用操作系统内核做好准备【孙弋】。而u-boot是bootloader的一种，支持ARM体系结构。本课题在裸板上移植烧写的是u-boot1.1.6。

嵌入式Linux内核是整个嵌入式系统软件的核心，就像人的心脏。此外，需要对内核进行剪裁移植，去掉不相关的服务，降低系统消耗。本课题移植的是linux-2.6.30的内核。

Yaffs2文件系统是专为嵌入式系统使用NAND 型闪存而设计的日志型文件系统，它速度更快，挂载时间很短，对内存占用较小【孙弋】。因此本课题选择yaffs2文件系统。

移植u-boot、linux内核、文件系统，构建基本的linux操作系统，在此基础上，还需要移植雷凌RT5572USB网卡驱动和C270摄像头驱动。

1. 应用层软件

移植mjpg-streamer流媒体服务器实现监控数据采集、压缩和传输。移植boa服务器和监控页面设计实现实现web服务。入侵检测模块对监控数据进行智能分析和报警。

应用层软件工作流程：首先mjpg-streamer流媒体服务接收到远程浏览器发送来的连接请求，创建一个新连接，通过linux的V4L2技术启动摄像头采集到yuv格式的图像数据，在经过格式转换、压缩传输给远程浏览器。同时，会实时的把采集到图像数据进行智能分析，检测移动目标，报警。

2.5本章小结

# 3.嵌入式系统环境的搭建

3.1系统开发硬件环境的搭建

在进行系统开发前，需要搭建开发测试的硬件环境，本课题是基于嵌入式的智能监控系统，嵌入式端采用arm9开发板，图像采集部分需要USB摄像头，图像数据及相关数据的传输需要通过USB无线网卡传输，图像存储需要sd卡。图像智能分析，需要在pc服务器上对实时的每帧图片进行处理。开发嵌入式时，嵌入式开发板需通过串口线与pc端相连，通过串口工具来开发。

3.2嵌入式交叉编译环境的搭建

3.2.1交叉编译的目的与意义

交叉编译指在一种计算机环境下运行编译程序，编译出在另外一种计算机环境下运行的程序，这个编译过程叫做交叉编译，也可认为是在一种平台下生成另外一种平台下的可执行代码。但是为什么要进行交叉编译，原因是 在进行嵌入式系统的开发时，运行程序的目标平台通常具有有限的存储空间和运算能力，比如常见的ARM 平台，其一般的静态存储空间大概是16到32MB，而CPU的主频大概在100MHz到500MHz之间。这种情况下，在ARM平台上进行本机编译就不太可能了，这是因为一般的编译工具链（compilation tool chain）需要很大的存储空间，并需要很强的CPU运算能力。为了解决这个问题，交叉编译工具就应运而生了。通过交叉编译工具，我们就可以在CPU能力很强、存储控件足够的主机平台上（比如PC上）编译出针对其他平台的可执行程序。

C一般编译程序代码需要经过四个步骤，分别是预处理(preprocessing)，编译(compilation) ,汇编((assembly)和连接((linking)，经过编译后文件为可执行文件。其中预处理过程中在识别#开头的命令中，根据编译条件来选择代码来编译，编译后将输出“.i”格式的文件，预处理过程中使用的工具是arm-linux-cpp工具。编译将文件翻译成汇编语言，编译后将输出“.s”格式的文件，编译用到的工具为ccl。汇编将输出的汇编代码翻译成符合一定格式的机器代码，编译后将输出“.o”格式的文件，Linux上为ELF文件(OBJ文件)，汇编用到的工具arm-linux-as。连接是将生成的OBJ文件和系统库的OBJ文件，库文件连接起来，最终生成在arm上运行的可执行文件，连接用到工具arm-linux-ld.本项目使用的arm架构的板子，代码在A9核心板子上运行，因此需要搭建一个能够生成可以在ARM平台上执行的编译软件，交叉编译器的移植与mcu类型、硬件定义等都有关系，一般芯片开发厂商都会提供交叉编译器。

3.2.2交叉编译环境搭建步骤

（1）搭建pc端linux操作系统，本次开发采用ubuntu12.04操作系统。

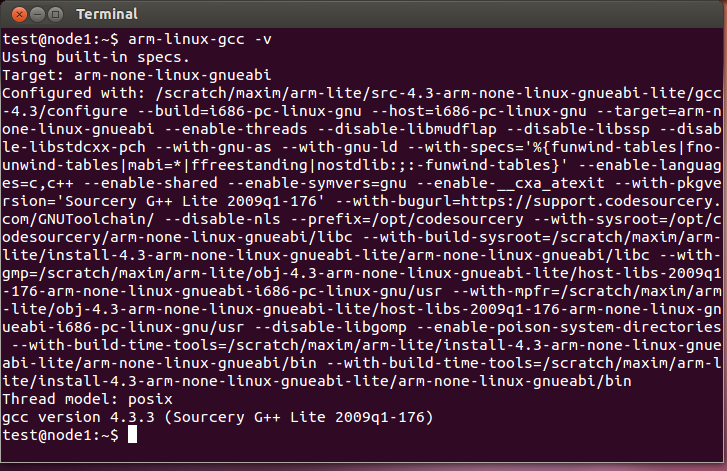
（2）用户登录pc端linux系统，本课题采用4.3.3版本的交叉编译器，将arm-linux-gcc-4.3.3.tar.bz2压缩包复制到pc端/opt/arm目录下。

（3）在pc端linux终端下输入“tar xvfj arm-linux-gcc-4.3.3.tar.bz2 –C /”将交叉编译器解压到/opt/arm/4.3.3目录下。

（4）配置环境变量，vi /etc/bashrc

打开文件，在文件中添加交叉编译工具的路径的语句这样交叉环境变量就加到了 PATH 环境变量中，我们每次使用 arm-linux-gcc 等交叉编译命令时，就在使用了 4.3.3 版本中的 arm-linux 交叉编译命令了。

在文件中加入语句“export PATH=/opt/arm/4.3.3/bin/:$PATH”，保存并关闭文件。在终端中输入source /etc/bashrc 使交叉编译器生效。在终端上输入 arm-linux-gcc -v 来查看交叉工具的版本。



3.3嵌入式linux系统的搭建

Linux系统是一个应用很成熟的操作系统，系统安全可靠，支持多平台和多处理器。一方面它的源代码是开放的，方便对其裁剪以适合设计需求，另一方面具有良好的可移植性，这些使其成为嵌入式开发的首选。从软件角度来看，移植嵌入式Linux操作系统一般可以分为三个层次【】:引导程序(Bootloader) ， Linux内核(Kernel)以及根文件系统(Filesystem )。本文选用了Linux3.1.8内核，系统为嵌入式操作系统，下面将阐述移植嵌入式Linux系统的过程和方法。

3.3.1 u-boot的移植

Bootloader是嵌入式Linux系统中引导内核启动的一段程序，首先初始化硬件，建立一个合适的环境引导系统在硬件平台上运行。由于Bootloader的运行实现依赖于嵌入式系统的硬件，所以不同的嵌入式硬件系统需要不同的Bootloader。目前常用的Bootloader有RedBoot， ARMboot， U-boot等。其中U-boot是一种开源的、功能强大的Bootloader，它不仅包含各种类型的处理器，还提供了丰富的外设驱动【】。最大的优势就是加快了项目开发的进度，节省开发时间。

因此，本系统选用的引导程序是U-boot1.1.6，移植操作如下:

（1）制作U-boot镜像文件

将源码包uboot\_1.1.6.tar.gz复制到/opt/uboot工作目录，并进入目录进行

解压，然后通过make 该开发板cpu所对应的配置文件，进行开发板匹配配置、编译。

（2）烧录U-boot镜像

成功编译后，将会在uboot 1.3.4目录下生成u-boot.bin的二进制文件，将其

烧写到ARM开发板的Nandflash中完成移植。

3.3.2 linux内核的移植

Linux内核是一种系统软件，可以进行多任务操作，Linux内核移植过程就

是针对具体的目标平台，然后对Linux内核进行裁剪和修改，最后编译生成的内

核镜像文件安装到目标平台中运行。本系统将移植的是Linux2.6.30版本的内核，

移植操作如下:

（1）源代码的获取

内核源码的获取可以在Linux内核官网上下载，网址如下

<http://www.kernel.org/>

本课题使用的内核源码版本是linux2.6.30。

（2）添加对arm的支持

首先，解压内核源码到pc中，使用解压命令“tar xvfj linux-2.6.30.tar.bz2 -C /opt/arm/”,然后解压到pc的“/opt/arm/”目录下。进到内核源码，修改Makefile文件，将“ARCH ?=(SUBARCH)”修改为“ARCH=arm”，将“CROSS\_COMPILE=arm-linux-”，然后保存。

（3）修改平台输入时钟

找到内核源码arch/arm/mach-s3c2440/mach-smdk2440.c文件，在函数 static void \_\_init smdk2440\_map\_io(void)中，修改成s3c24xx\_init\_clocks(12000000)。

（4）添加nandflash支持

修改文件arch/arm/plat-s3c24xx/common-smdk.c文件中的smdk\_default\_nand\_part、smdk\_nand\_info结构体，同时在内核配置中添加对nandflash的支持。

（5）修改机器号

修改S3C2440的机器号

由于Bootloader传递给Linux内核的机器号为168，为与Bootloader传递参数一致，修改 arch/arm/tools/math-types文件。

s3c2440 ARCH\_S3C2440 S3C2440 362

修改为：

s3c2440 ARCH\_S3C2440 S3C2440 168

（6）配置linux内核

进入Linux-2.6.30内核主目录，通过命令“make s3c2410\_defconfig”将2410的默认配置文件写到当前目录下的.config。S3C2410的配置和S3C2440差不多，在这基础上进行修改。

配置内核模块的功能，有几种方式可以进行界面选择：

make menuconfig（文本选单的配置方式，在有字符终端下才能使用）

make xconfig（图形窗口模式的配置方式，图形窗口的配置比较直观，必须支持Xwindow下才能使用）

make oldconfig（文本配置方式，在原内核配置的基础修改时使用）

本课题使用make menuconfig命令，根据需求配置linux内核。

（7）编译烧写

使用命令“make zImage” 成功编译后，将内核源码目录arch/arm/boot下生成的zImage内核映像文件烧写到ARM开发板中，完成Linux2.6.30内核移植。

3.3.3 yaffs文件系统的移植

文件系统是一种用户和操作系统进行交互的机制，如果Linux中缺少文件系

统，用户与操作系统之间的交互操作就不存在了，就无法使用shell命令来运行

可执行文件。嵌入式Linux操作系统支持Yaffs2, Cramfs, Jffs2等多个文件系统，

其中Yaffs2是专门为Nand闪存而设计的，运行速度更快，占用的内存空间也更

少，而且对于大容量的Nandflash芯片能够提供很好地支持。由于开发板所用存

储设备为Nandflash，所以本课题选择使用Yaff2嵌入式文件系统。

移植操作如下：

（1）获取yaffs源码git://www.aleph1.co.uk/yaffs2。在内核中添加对yaffs的支持,需要在linux内核中打上yaffs2的补丁，并且在配置单中添加对yaffs的支持，方法如下：

#cd yaffs2/

#./patch-ker.sh c /opt/arm/linux-2.6.30.4/

（2）获取busybox源码<http://www.busybox.net/downloads/>，配置，编译并安装busybox，在busybox目录下生成一个名为“\_install”的目录。

（3）开始构建文件系统

①在工作目录下新建一个“root\_2.6.30.4”的目录，创建根文件系统的目录，复制上一步编译出来的busybox的“bin”、“sbin”、“usr”、“linuxrc”目录到“root\_2.6.30.4”目录下：

#mkdir –p /opt/arm/root\_2.6.30.4

#cd /opt/arm/root\_2.6.30.4

#cp –rf /opt/arm/busybox-1.13.0/\_install/\* .

#mkdir –p dev etc home/sky lib mnt opt proc root sddisk sys tmp udisk usr/lib usr/share var

②创建设备文件，dev目录下的内容在内核挂载完文件系统后，系统会使用mdev自动建立。在使用mdev构建“dev”目录前，init进程需要用到“/dev/console”和“/dev/null”这两个设备文件，所以需要手动创建。

#cd /opt/arm/root\_2.6.30.4/dev

#mknod console c 5 1

#mknod null c 1 3

③安装建立etc目录下的配置文件，比如/etc/fstab、/etc/group、/etc/inittab、/etc/passwd、/etc/profile等文件

④使用命令 “mkyaffsimage\_2 root\_2.6.30.4 root\_2.6.30.4.bin”编译好文件系统镜像，烧写到arm板上。

3.4外设驱动的移植和编写

对ARM开发板硬件的操作是通过对Linux内核中的设备驱动的调用来实现的，驱动的作用就是把这些调用映射到设备操作。要使外围设备在ARM开发板上能正常工作，Linux系统需要加载该设备的驱动程序，通常有两种实现方式:一种是把外设相应的驱动模块编译进内核;另一种是将其编译为功能模块，即生成.O文件，在需要时动态加载此文件[[23]。本文采用的是第一种方式，直接将设备驱动编译进Linux内核，使其即随内核启动而自动加载。 在Linux系统中，驱动程序[31 32]作为内核的一部分提供对设备操作的支持，主要包括设备的打开和释放、设备的读写、设备的控制管理等。应用程序通过系统调用与内核通信，Linux将所有设备看作设备文件，对设备文件的操作如同普通文件类似，提供open(), closeQ, read(), writeQ, ioctlQ等几个主要调用接口。在应用程序中使用系统调用如readQ时，最终会转换成内核中对具体设备的read()操作，而我们编写驱动程序的目的就是提供这些打开、关闭读、写、控制接口。

3.4.1摄像头驱动移植

先移植摄像头驱动至内核，重新配置内核，将对应的摄像头支持模块勾选上，再重新编译内核，使摄像头的驱动程序加入内核，实现开机驱动自动加载。罗技C270摄像头是标准的UVC摄像头，在linux-2.6版本以后的Linux内核里己经集成了UVC的驱动，由于本系统的Linux2.6.30内核中己集成了C270的摄像头驱动，所以只需配置Linux内核即可。摄像头驱动模块的内核配置过程如下:

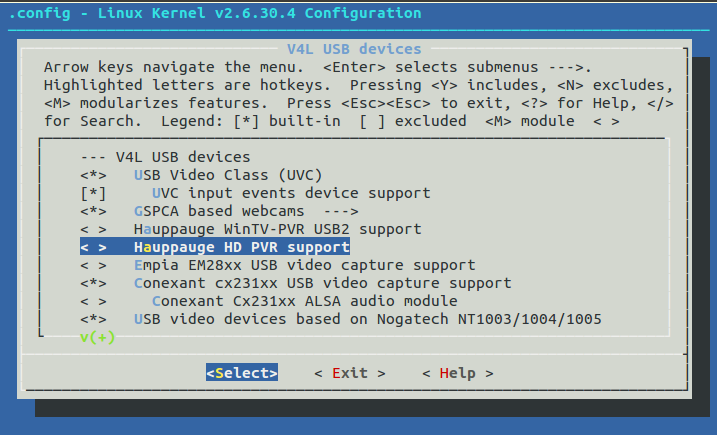
（1）配置内核

①进入内核配置选项

#make menuconfig

②在Device Drivers菜单里面选择在Device Drivers菜单里面选择Multimedia devices;

③选择Video capture adapters进入V4L USB devices，选择支持uvc，具体配置见下图。



（2）编译内核，内核配置完成后，使用命令make zImage 编译内核，将编译好的内核烧写到开发板上。开发板重启后，会自动加载摄像头驱动，这时插上摄像头会出现以下提示。

3.4.2无线网卡移植

本系统选用的是雷凌RT5572无线网卡，通过USB接口将其连接到开发板，由于Linux2.6.30内核没有带RT557x的驱动程序，所以这里需要rt5572无线网卡的驱动源码。无线网卡驱动模块的内核配置过程如下:

（1）从官网或其他途径获取RT5572的linux驱动源码。

（2）解压后，修改源码中的makefile、config.mk等文件，使之与本地交叉编译环境匹配。

（3）编译源码生成.ko文件，去掉调试信息。

（4）重新配置编译内核，勾选 Device Drivers --->Network device support --->Wireless LAN(non-hamradio) ---> Wireless LAN driver(non-hamradio)& Wireless Extensions。

（5）拷贝驱动到目标系统

把./RT2870STA.dat拷贝到开发板（/etc/Wireless/RT2870STA）目录下

把./os/linux/rt3070sta.ko拷贝到开发板

把./common/rt2870.bin拷贝到开发板（lib/firmware）目录下

（6）加载驱动模块

$ insmod /xxx/rt3070sta.ko

无线网卡驱动移植完成后，插入USB无线网卡会出现以下提示信息。

3.5本章小结

# 4.移动目标检测的设计与实现

# 5.应用层软件的设计与实现

# 6.系统测试与结果分析

# 8.总结与展望