

# 基于 Android 视频监控系统的数据处理及实现

周鹏飞,潘地林

(安徽理工大学 计算机科学与工程学院,安徽 淮南 232001)

**摘要:**针对平安城市、智能家居安全监控的实际发展需求,结合嵌入式、Android 多媒体框架、3G 等技术,提出了在 Android 手机端显示多路视频监控图像的设计方案,满足了视频监控的随时性、移动性的要求。该方案在基于 ARM9 的 Mini2440 上搭建了 MJPG-streamer 本地的 IP 摄像头,同时采用了目前国外公开的 IP 网络摄像机,是手机端 IP 网络监控系统良好的雏形。文中详细分析了视频采集、图像格式转换及 Android 端截取视频图像进行显示的数据处理过程,有助于进一步添加和完善视频监控系统的功能。

**关键词:** MJPG-streamer; Android; YUV; JPEG; IP 摄像头

中图分类号: TP391

文献标识码: A

文章编号: 1673-629X(2013)05-0150-04

doi: 10.3969/j.issn.1673-629X.2013.05.039

## Data Processing and Implementation of Video Surveillance System Based on Android

ZHOU Peng-fei, PAN Di-lin

(College of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232001, China)

**Abstract:** By analyzing the actual requirement of safety monitoring at safe city and intelligent home system, an idea of a multi-channel mobile video surveillance system based on Android phone is proposed, which combined with technology of 3G, embedded, multimedia framework of Android and so on. This scheme uses the local IP camera by transplanting the open source video server MJPG-streamer on Mini2440 based on ARM9 and the abroad open IP cameras, is a nice structure of IP network surveillance system on mobile phone. Give the detailed analysis on the video acquisition, the conversion of image format and the video image capture on Android phone client, which will help further add and improve the function of video surveillance systems.

**Key words:** MJPG-streamer; Android; YUV; JPEG; IP camera

## 0 引言

移植引领 IP 网络摄像机发展的 Axis Communications 率先将 IP 网络摄像机引入中国市场,但由于价格和认知度等问题,造成起初的其产品销售成绩并不理想<sup>[1]</sup>。但随着安防行业向数字化、网络化、智能化方向发展步伐的加快,以 IP 网络摄像机为主的视频监控系统逐渐向大众消费领域延伸,渗透到教育、政府、城市治安等多个领域。文献[2,3]对 IP 网络摄像机的基本概念和发展前景进行了介绍并对其应用实例进行了分析。近年来 3G 移动网络的快速发展,与 WLAN 系统的融合,加上手机的处理能力逐渐增强,基于手机的无

缝式无线监控又得以广泛的应用<sup>[4]</sup>。这样,无论何时何地,只要是在 3G 网络的覆盖之下,就可以通过 IP 的方式用手机来获取所需要的摄像头录像。

本系统采用了一些国外开放的 IP 网络摄像机地址,同时在本地搭建了基于 MJPG-streamer 视频服务器的摄像头,通过 Android 客户端软件获取监控录像,提升了监控用户在访问网络摄像机场景的随时性、移动性的需求。

## 1 系统总体设计

视频监控系统分为服务器监控终端和客户端监控中心两部分。本地是基于 S3C2440 微处理器,采用开源的 Linux 操作系统,移植了 MJPG-streamer 的视频流服务器,在监控现场通过 USB 摄像头采集图像,通过将采集的图像经过格式转换和视频压缩传输到 Android 客户端进行显示。系统总体设计如图 1 所示。

收稿日期: 2012-08-15; 修回日期: 2012-11-17

基金项目: 安徽省高校省级自然科学基金项目(KJ2011A083)

作者简介: 周鹏飞(1987-),男,硕士研究生,研究方向为图形图像处理与嵌入式系统;潘地林,教授,研究方向为流体机械及图形图像处理研究。

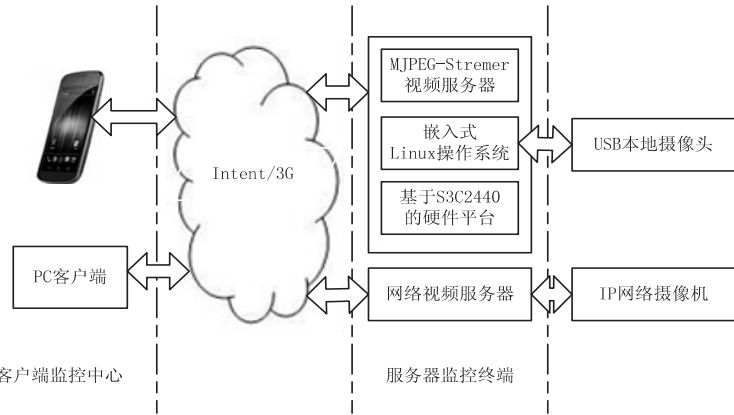


图 1 系统总体结构

2 本地视频服务器的搭建

运行环境为 Ubuntu11.10。内核采用 Linux2.6.32 版本,运行 Make menuconfig 配置万能 USB 摄像头驱动。文件系统采用的是 FriendlyArm 公司提供的 rootfs\_qtopia\_qt4-20110304,在此基础上进行裁剪,去除 Qt 界面,因为作为实时视频采集服务器,不需要 Qt 图形界面。同时移入 MJPG-streamer 插件。

MJPEG-streamer 是一款免费的轻量级的视频服务器软件,应用在基于 IP 协议的网络中,从网络摄像机中获取并传输 JPEG 格式的图像,在浏览器或用户客户端通过 IP 和端口号来显示图像。该软件采用模块化的设计方法,以功能块,即 plug-in(组件)为单位进行描述。其中最主要的组件是 input\_udev 输入组件和 output\_http 网站服务器输出组件。MJPEG-streamer 的工作就是将其中的一个输入组件和多个输出组件绑定在一起,所有的功能都是通过它的各个组件完成的。用户可以选择需要的模块,并进行必要修改,完成自己想要的功能。

具体移植过程如下:

- (1)下载 MJPEG-streamer 源码,解压进入 mjpg-streamer 目录,执行 make CC = arm-linux-gcc,生成 mjpg-streamer 可执行文件和 \*.so 文件;
- (2)将生成的 .so 文件拷至/usr/lib/下;
- (3)将生成的 mjpg-streamer 拷至/usr/bin/下;
- (4)在文件系统主目录下新建文件夹:mkdir www;
- (5)在/etc/rc.d/init.d 下设置 mjpg\_streamer 的开机启动文件 mjpg\_start,输入内容:

```
#!/bin/sh
base=mjpg_streamer
#See how we were called
case "$1" in
start)
    /usr/bin/$base -i "/usr/lib/input_udev.so"
```

```
-o "/usr/lib/output_http.so -w /www" &
;;
stop)
    pid=`bin/pidof $base`
    if [ -n "$pid" ];then
        kill -9 $pid
    fi
;;
esac
exit 0
(6)制作文件系统,执行命令:
mkyaffs2image rootfs_qtopia_qt4 fs.
```

img

- (7)订制好内核和文件系统,移植到 Mini2440 上运行。

此时就可以在浏览器上输入 http://192.168.1.230:8080/?action=streamer 查看到动态的实时监控图像,其中的 IP 是 mini2440 开发板的 IP 地址。

3 视频采集与格式转换

3.1 V4L2 简介

V4L2 (Vedio For Linux Two)是在 Linux 下开发视频采集设备驱动程序的一套规范,是 Linux 内核提供给驱动程序的开发人员的清晰的模型和统一的编程接口,应用于许多视频设备之中。而对于用户编程来说,需要了解的就是如何使用系统提供的 API。结合 MJPG-streamer 相关组件,视频采集流程如图 2 所示。

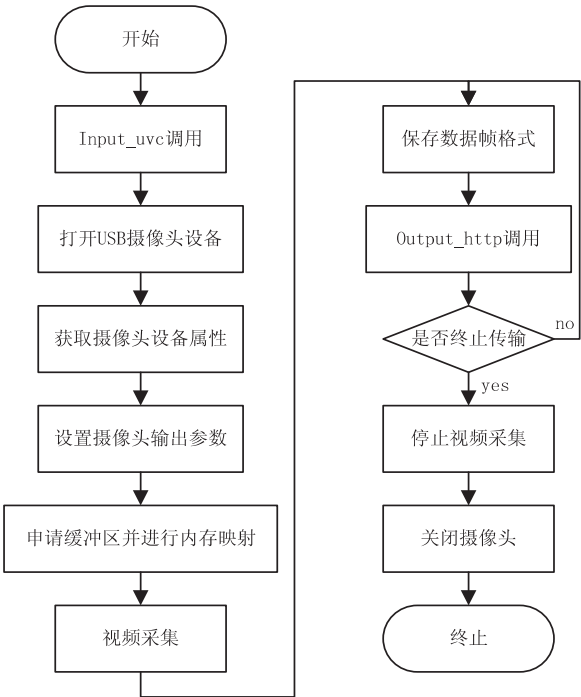


图 2 视频采集流程图

3.2 RGB 与 YUV 格式转换

RGB 色彩模式是一种颜色标准,是通过对红(R)、

绿(G)、蓝(B)三个颜色通道的变化以及它们相互之间的叠加来得到各式各样颜色的。

YUV 是一种颜色编码方法,亮度信号 Y 和两个色差度信号 Cb 和 Cr。RGB 诉求于人眼对色彩的感应,YUV 则着重于视觉对于亮度的敏感程度。采用 YUV 色彩空间的重要性是它的亮度信号 Y 和色度信号 U、V 是分离的。YUV 主要的采样格式有 YCbCr 4:2:0、YCbCr 4:2:2、YCbCr 4:1:1 和 YCbCr 4:4:4,所代表的含义是 Y、Cb、Cr 三个成分的数据取样比例。摄像头常见的输入格式为 YUV 4:2:2。

RGB 与 YCbCr 转换方程如下:

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0.299 & 0.578 & 0.114 \\ 0.500 & -0.419 & -0.081 \\ -0.169 & -0.331 & 0.500 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

经过数学运算得到转换公式:

$$\begin{cases} R=Y+1.4075*(Cb-128) \\ G=Y-0.3455*(Cr-128)-0.7169*(Cb-128) \\ B=Y+1.779*(Cb-128) \end{cases}$$

在计算机中,用移位的方法实现整型运算替代浮点运算,具体实现代码如下,转换同时防溢出处理:

```
u = yuyv[1] - 128;
v = yuyv[3] - 128;
r = (y + (359 * v)) >> 8;
g = (y - (88 * u) - (183 * v)) >> 8;
b = (y + (454 * u)) >> 8;
*(ptr++)=(r>255)? 255:((r<0)? 0:r);
*(ptr++)=(g>255)? 255:((g<0)? 0:g);
*(ptr++)=(b>255)? 255:((b<0)? 0:b)。
```

3.3 格式转换

市场上常见的两种摄像头视频输出格式有 YUV 和 MJPEG。这样,摄像头采集到的视频数据并不能直接发送给浏览器显示,需要进行图像格式、文件格式等系列转换,转换成最终传送给 WEB 浏览器的 JPEG 格式。为了减少码流通常还需要图像压缩。

JPEG 是 CCITT 和 ISO 联合制定的静态图像的压缩编码标准,是所有图像压缩技术的基础,它在变换编码的基础上,综合应用了离散余弦变换(DCT)和哈夫曼编码两种方法,达到了很好的图像压缩效果,DCT 编码是 JPEG 算法的核心<sup>[5]</sup>。MJPEG 是在 JPEG 基础上发展起来的动态图像压缩技术,它只单独地对某一帧进行压缩,而基本不考虑视频流中不同帧之间的变化。MJPEG 的优点是画质比较好,能产生高质量、全屏、全运动的视频,每帧可任意存取,可以动态调整帧率、分辨率,而且编码相对来说比较容易。

JPEG 文件大体上可以分成以下两个部分:标记码(Tag)加压缩数据。标记码部分给出了 JPEG 图像的

所有信息,如图像的宽、高、Huffman 表、量化表等<sup>[6]</sup>。这里对压缩编码时和后面章节中 Android 端从视频流中读取图像内容时用到的标记码做下说明(如表 1):

表 1 JPEG 文件格式标记码

符号	标记代码	单位
SOI	0xFFD8	图像开始(Start Of Image)
APPn	0xFFE0-0xFFEF	JFIF 应用数据块(n,1~15)
DQT	0xFFDB	量化表
SOFO	0xFFC0	帧开始
DHT	0xFFC4	霍夫曼表
SOS	0xFFDA	扫描线开始
EOI	0xFFD9	图像结束(End Of Image)

获取视频图像后,在进行转换压缩的过程中,pglobal->stop 是一个全局变量,控制着线程的终止;vd->formatIn 表示图像格式,对 YUYV 和 MJPEG 进行不同的处理。此外,还应注意同步机制的使用:互斥锁和条件变量。具体转换如图 3 所示。

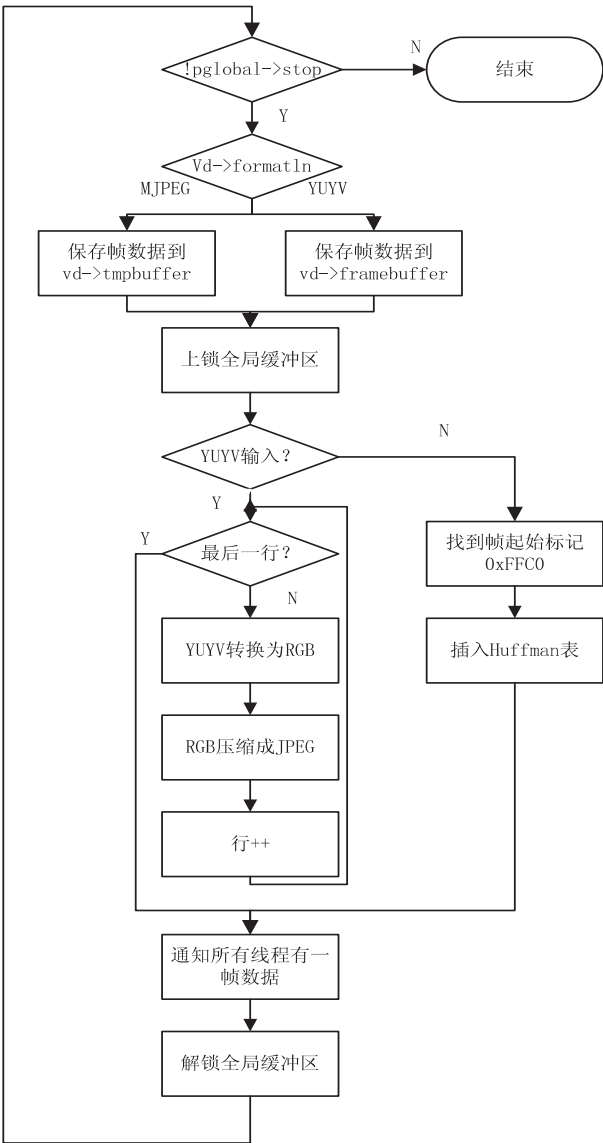


图 3 格式转换流程图

## 4 Android 客户端获取数据

### 4.1 获取视频流数据

Android 是 Google 公司推出的针对移动设备的操作系统,底层采用 Linux 内核,应用程序采用了 Java 语言编写,并运行于类 JVM 虚拟机的 Dalvik 虚拟机之上。在 Android 开发中,Android SDK 附带了 Apache 的 HttpClient,它是一个完善的客户端,提供了对 HTTP 协议的全面支持,可以使用 HttpClient 的对象来执行 HTTP GET 和 HTTP POST 调用。HttpClient 的一般使用步骤:

- (1)使用 DefaultHttpClient 类实例化 HttpClient 对象;
- (2)创建 HttpGet 或 HttpPost 对象,将要请求的 URL 通过构造方法传入 HttpGet 或 HttpPost 对象;
- (3)调用 execute 方法发送 HTTP GET 或 HTTP POST 请求,并返回 HttpResponse 对象;
- (4)通过 HttpResponse 接口的 getEntity 方法返回响应信息,并进行相应的处理。

对已获取的数据流按照 JPEG 标准文件头进行分段处理。通过标记码 SOI 和 EOI 计算出数据的长度,读取数据,其流程如图 4 所示。



图 4 视频帧数据获取流程

Android 开发中位图是最常用的资源,使用 BitmapDrawable 和 BitmapFactory 来获取资源中的位图。BitmapFactory 这个类,API 中这样解释:

Creates Bitmap objects from various sources, including files, streams, and byte-arrays.

BitmapFactory 的所有函数都是 static,这个辅助类可以通过资源 ID、路径、文件、数据流等方式来获取位图。使用 BitmapFactory 类 decodeStream (InputStream is)解码位图资源,获取位图。再通过 Canvas 的 BitmapDrawable 来将 Bitmap 绘制到 Canvas,显示到 View 中。设置一线程 MjpegViewThread 不间断获取视频数据显示在视图中,更新图像内容。

### 4.2 本地摄像头测试与应用案例分析

将应用程序的 apk 安装至手机,保持手机上网状态,最好是利用 WIFI 资源,这样程序更加流畅。图 5 为程序主界面,界面被分为 4 个区域,同时显示 4 个摄像头的监控画面,界面块 4 (图 5 中数字 4 标识的区域)是本地摄像头监控的实验室场景,其余是国外开放的 IP 摄像机的监控画面,依次为小区监控、交通监控、广场监控。点击其中任一区域进入该摄像头的单个监控画面,可以对画面进行放大、缩小及抓拍保存图片至 sdcard 中。

从引用的几个案例可以看出可以同时监控不同场景,只要拿有一部 Android 手机,知道要访问的 IP 摄像机的地址,就可以打开其监控画面。这比传统的视频监控更加体现出了便携性,随时访问性,模块独立性,“分散监控、集中管理”的理念。



图 5 多摄像头监控画面

## 5 结束语

3G 移动网络将促使手机监控成为网络监控的主流,引领着网络监控朝着移动化的方向发展。对于监控端,智能化的多功能 IP 摄像机已备受关注<sup>[7]</sup>,而其涉及到的最为关键也是作为摄像机最为基础的技术就是在监控场景提取相关的运动目标,如何做到提取目标的准确性、效率性及自动化处理能力,这也是在整个监控视频处理中研究的热点<sup>[8]</sup>。Android 手机端监控系统比传统的监控系统更注重用户体验,除了对摄像头权限的管理、远程控制、本地录像存储与回放等基础功能外,若监控场地发生意外情况,可以通过监控端服务器向手机发送报警短信等智能化功能。至此,手机端的视频监控系统的的功能前景将会更加明朗。

系统改进思路:WLAN 成为无线通信使用热点,通过移植 WIFI 模块可以实现无线网络视频采集。在 MJPG-streamer 软件中有一个 input\_control. so 的输入插件,加上该插件可以实现摄像头位置的控制,另外可以加入智能模式识别算法<sup>[9]</sup>,从摄像头读取的视频信息识别运动物体,必要时通过短信告警通知用户<sup>[10]</sup>,从而使系统功能更加强大。

### 参考文献:

- [1] 胡青松. IP 网络摄像机的研究与应用[J]. 微计算机信息,



```
I2C: ready
SPI: ready
DRAM: Using DDR SPD. . .
DDR: 2 GiB (DDR3, 64-bit, CL=6, ECC off)
FLASH: I2:512 KB enabled
NAND:1024 MiB
MMC:FSL_ESDHC: 0
SF: Detected S25FL128P_256K with page size 256, total 16
MiB
```

由上述打印信息可知,板卡各项硬件设备功能正常,运行稳定。

成功启动 uboot 之后,再烧入内核文件 uImage,设置树文件 p2020rdb. dtb,及文件系统镜像 rootfs. ext2. gz. uboot。

```
设置系统启动环境变量如下[12]:
setenv hwconfig "usb1;dr_mode=host;usb2;dr_mode=host;
esdhc"
setenv bootargs root=/dev/ram rw console = ttyS1, $ baudrate
$ othbootargs;
setenv bootcmd "sf probe 0 40000000;sf read 2000000 80000
80000;sf read 3000000 100000 400000;sf read 4000000 900000
600000;bootm 3000000 4000000 2000000";
boot;
重启系统。系统在启动 uboot 后,自动加载环境
变量,启动内核,成功进入操作系统,打印信息如下:
Welcome to the LTIB Embedded Linux Environment
P2020RDB login: PHY: 0;01 - Link is Up - 1000/Full
[root@P2020RDB /root]# uname -r
2.6.35.9
[root@P2020RDB /root]#
至此,SPI FLASH 启动方式顺利实现。
```

4 结束语

文中在 P2020 硬件平台之上,对 SPI FLASH 启动模式的原理、实现与优化进行了详尽介绍。SPI FLASH 启动方式有着独特的优势,相信在嵌入式领域

必将发挥更大的作用。采用 SPI FLASH 启动 uboot 时,还存在着诸如不能保存环境变量等一些问題,这也是今后努力与改进的方向。

参考文献:

[1] 白 潇,徐智勇,张 耀. 基于 PowerPC 架构的嵌入式 Linux 系统开发和 RAID 技术[J]. 仪器仪表用户,2012 (2):49-51.

[2] 袁堂夫. 嵌入式 PowerPC 处理器浅[C]//2003 国际有线电视技术研讨会论文集. 杭州:出版者不详,2003.

[3] Freescale Semiconductor Corp. QorIQ™ P2020 Communications Processor Product Brief[M]. America: Freescale Semiconductor,2009.

[4] 陈柏喜,曾桂根. 基于 omap5912 的 uboot 平台构建[J]. 中国新通信,2009(3):53-55.

[5] 郭建磊,杨厚俊. 基于 S3C44B0X 的 U-Boot 及 μClinux 的移植分析[J]. 计算机技术与发展,2009,19(4):13-15.

[6] 刘邦运,别红霞. PowerPC 平台的 U-Boot 启动分析和移植[J]. 微计算机信息,2010(23):172-174.

[7] Freescale Semiconductor Corp. QorIQ™ P2020 Integrated Processor Reference Manual[M]. America: Freescale Semiconductor,2009.

[8] Freescale Semiconductor Corp. PowerPC™ e500 Core Family Reference Manual[M]. America: Freescale Semiconductor, 2005.

[9] 王长清,林桂竹. 基于 PowerPC 双核处理器嵌入式系统 U-Boot 移植[J]. 河南师范大学学报(自然科学版),2010 (1):70-74.

[10] Freescale Semiconductor Corp. Booting from On-chip ROM (eSDHC or eSPI)[M]. America: Freescale Semiconductor, 2010.

[11] Spansion Inc. S25FL128P. pdf[M]. America: Spansion Inc., 2007.

[12] Freescale Semiconductor Corp. BSP Targeting P2020RDB Board User Manual[M]. America: Freescale Semiconductor, 2010.

(上接第 153 页)

2004,20(2):68-69.

[2] 刘红彬,周 强. IP 网络摄像机及其在远程视频监控系统的应⤿[J]. 设备管理与维修,2004,9(14):31-33.

[3] 刘红彬,周 强. IP 网络摄像机及其发展前景[J]. 煤矿现代化,2005(4):44-46.

[4] 高 成. 基于移动网络的智能视频监控系统的设⤿[D]. 长沙:国防科技大学,2010.

[5] 张 弘. 数字图像处理与分析[M]. 北京:机械工业出版社,2007.

[6] Dirks B,Schimek M H,Verkuil H,et al. Video for Linux Two API Specification(Revision 0.24)[S]. 2008.

[7] 张 翔. 多功能 IP 摄像机的设计实现及流媒体传输的研究[D]. 杭州:浙江大学,2008.

[8] Collins R,Lipton A J,Kanade T,et al. A system for video surveillance and monitoring; VSAM final report[R]. America: Carnegie Mellon University,2000;45-50.

[9] 侯宏录,李宁鸟,刘迪迪,等. 智能视频监控中运动目标检测的研究[J]. 计算机技术与发展,2012,22(2):49-52.

[10] Yang Xingyu, Wu Haibin, Xu Songqing. The application of GPRS wireless transmission in remote image monitored control system[J]. Microcomputer Information,2005(3):64-66.

# 基于Android视频监控系统的数据处理及实现

作者: [周鹏飞, 潘地林](#)  
作者单位: [安徽理工大学 计算机科学与工程学院, 安徽 淮南 232001](#)  
刊名: [计算机技术与发展](#)  
英文刊名: [Computer Technology and Development](#)  
年, 卷(期): 2013(5)

本文链接: [http://d.g.wanfangdata.com.cn/Periodical\\_wjfz201305041.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjfz201305041.aspx)