基于 Android 平台的无线视频监控①

凡 威, 周渊平

(四川大学 电子信息学院, 成都 610065)

摘 要: 随着科技生活的发展, 人们对视频监控的要求越来越高. 提出了一种基于 Android 平台的移动音视频监控系统. 该系统利用 Mediastream2 实现流媒体的处理, ORTP 实现流媒体的安全可靠传输, 以及 FFMpeg 实现音视频的编解码. 重点讲解了音视频库 FFMpeg 中对视频和音频的编解码处理过程以及是如何将音视频同步输出的. 最后在嵌入式平台 TQ210 上测试可实现实时的视频监控.

关键词: 音视频; Mediastream2; 同步输出; TQ210

Wireless Video Monitoring Based on Android Platform

FAN Wei, ZHOU Yuan-Ping

(School of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China)

Abstract: With the development of science and technology, people have higher requirement to video monitoring. This paper proposes a mobile audio and video monitoring system based on Android platform. In this system, we adopt Mediastream2 to process the stream media, implement the safe and reliable transmission of stream media with ORTP and codec of audio and vedio with FFMpeg. This paper mainly introduces how to encode and decode the audio and video with the FFMpeg, a kind of audio and video library and output audio and video synchronously. Finally, the real-time video monitoring testing can be realized on the embedded platform TQ210.

Key words: audio and video; Mediastream2; sync output; TQ210

近年来, 随着无线网络技术, 视频压缩技术和流媒体技术的不断发展, 人们对视频监控的要求也越来越高, 而基于无线网络(wifi)将必定是今后视频监控的必然选择, 同时移动智能终端作为监控端也是一个最佳的应用平台. 随着 Android 的飞速发展, 人们也越来越多的选择 Android 平台作为客户端来代替传统的 PC 来观察监控端传来的视频. 本文设计了一种在 Android 平台上的视频监控系统, 可以达到随时随地的监控效果. 与以往的无线视频监控相比, 本系统可以在任何两个 Android 设备上进行视频监控, 同时可在两个 Android 设备上进行实时的语音通话, 从而使得此监控设备更加的智能化和实用化.

1 Android操作系统的系统架构

Android 分为以下几层, 从高层到底层分别是应用程序层(Applications)、应用程序框架层(Application Framework) 、系统运行库层(Libraries 和 Android Runtime)和 Linux 内核层(Linux Kernel), 如下图 1 所示.

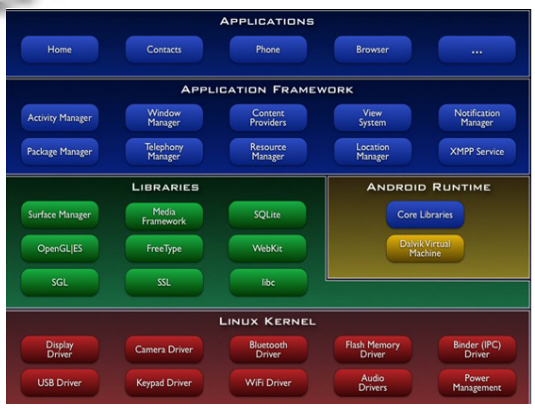

图 1 Android 系统架构图

① 收稿时间:2014-04-21;收到修改稿时间:2014-05-19

应用程序层所有的程序都是用 JAVA 语言所编写, 通过调用应用程序框架层所提供的 API 来完成的, 当然也可以使用 JAVA 通过 JNI 方式调用. 在应用程序框架层里, 为开发人员提供了访问核心应用程序所使用的 API 框架. 系统运行库层包括程序库层和 Android 运行库层两部分, 其中程序库中包含一些 C/C++库, 这些库能被 Android 系统中不同组件使用, 通过应用程序框架层为开发者提供服务. 而 Android 运行时库主要由两个方面组成, 一是核心库, 另一个是 Dalvik 虚拟机. 在核心库里, 它提供了 JAVA 语言核心库的大多数功能. 在 Linux 内核层中, 为 Android 提供了一些核心系统服务. 由于 Android 是基于 Linux2.6 内核的, 但与之又有一些区别, 在 Linux 内核的基础上加上了 Android 所独有的一些驱动程序, 如 IPC Binder, Android Debug Bridge 等[1].

2 Mediastream2的简介

Mediastream2 是一个功能非常强大, 框架小又十分灵活的流处理引擎, 在音视频开发方面有着非常广泛的应用. 它能够捕获和接收音视频流, 并且可以将其发送、编码, 同时完成解码以及渲染等功能[2]. 因此本文采用的也就是这个库, 来对其进行二次开发, 从而完成编译以及移植, 来达到我们所需要的效果.

Mediastream2 里可以实现 RTP 数据包的接收和发送. 同时它也可以支持多种音视频编解码的格式, 主要的一些音频格式有 Speex, G722,G711,GSM,ILBC 等. 主要的一些视频格式有 H263,theora,MPEG4,H264 和 VP8 等. 在捕获摄像头的 YUV 图片格式的同时也可以优化渲染 YUV 图片.

Mediastream2 中每一个处理实体都包含一个 MSFilter 对象, 都有一个或若干个输入和输出, 可以将每个 MSFilter 对象进行连接. 如 MSRtpRecv 接收网络上传来的数据包, 在其后进行解包, 并且将他们输出到下一个 MSFilter. MSFilter 可以被串接成一个 filter 链, 如下面的结构关系式:

MSRtpRecv-->MSSpeexDec-->MSFileRec 在 Mediastream2 中有一个独立的线程 MSTicker, 它是在媒体流处理中的调度对象, 每 10ms 被唤醒一次, 唤醒之后会处理其管理的媒体链中的数据. 几个 MSTicker 可以同时运行, 在这里采用的就是让其一个来处理音频数据, 另外一个处理视频数据, 从而使他们能够同步输出.

3 视频的编解码以及RTP协议

在视频监控系统中, 视频的编解码以及压缩传输都是十分重要的, 而目前采用 H.264 是世界上比较优秀的一种视频压缩算法. H.264 采用视频编码层(Video Coding Layer, VCL)与网络抽象层(Network Abstraction Layer, NAL)相分离的编码结构. 由于其对编码效率和图像质量进行很多的改进, 于此同时它的抗丢包性能和抗误码性能也很好, 能够适应各种各样的网络环境, 尤其是对压缩率要求比较高的视频监控中[3].

在这里需要移植开源的音视频解码库 FFMpeg 进行 H.264 编解码, 其步骤简要的概括如下: 在 jni 目录下建立一个 Android.mk 文件, 编辑内容如图 2 所示需要注意和仔细的一点就是, Android.mk 文件的每一行的最后不能够出现空格, 否则导致编译无法通过.

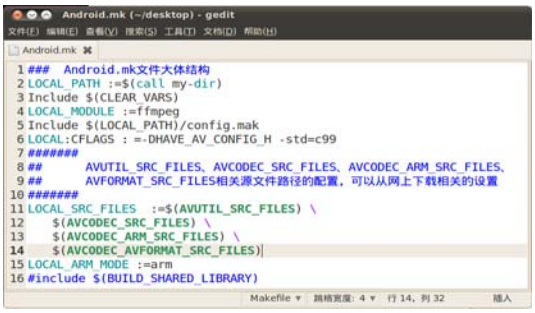

图 2　Android.mk 大体结构

Android.mk 文件编辑完成后, 为其添加可执行权限: # chmod a+x Android.mk.在 jni 目录下执行 ndk-build 命令, 经过大概十分钟左右的时间后, 将出现如图 3 所示的结果, 说明 FFmpeg 编译成功, 会在 jni 同级目录下生成 libs/armeabi/libffmpeg.so 文件, 这就是在 Android 中需要加载的库文件.

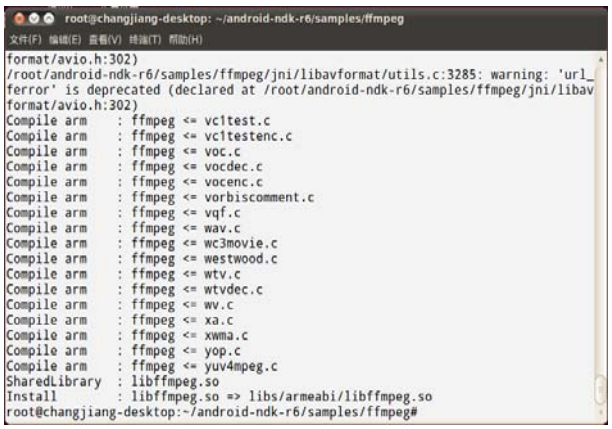

图 3　libffmpeg.so 库文件编译成功

RTP(Real-Time Transport Protocol, 实时传输协议)是一种在网络上传输多媒体数据流中所使用的传输协议, 它位于 UDP 之上, 功能上与传输层和网络层相独立, 但不能单独存在, 一般使用 UDP 协议传输多媒体数据, 而 RTP 则依靠 RTCP(Real-time Transport Control Protocol, RTCP)为数据的顺序传输提供可靠的保证, 同时也提供流量控制和拥塞服务[4].

Mediastream2 在此负责流媒体的处理而 ORTP 负责流媒体如何安全可靠的传输. ORTP 作为一个开源库, 它实现了 RTP 协议, 提供简单且容易使用的 API, 同时又可以支持多种 RTP 格式, 发送接收的实时调度, 在单线程下还可以支持多个 RTP 会话. 由于在 Mediastream2 中利用 ORTP 作为传输部分使用, 因此也可以认为是其在流媒体中的一部分功能. 将接收到的音视频数据进行打包, RTP 发送. 以发送视频包为例, 初始化流程为 ortp_init(),ortp_scheduler_init().再来创建视频会话为:

RtpSession*video_sess,video_sess=rtp_session_new (RTP_SESSION_SENDONLY); 需要调用发送函数 sendBytes=rtp_session_sen-d_with_ts(video_sess,sendBuffer,len,m_curTimeStamp).在发送接收的实时调度之中有一个比较关键的就是时间戳的计算, 它是以第一个到达 rtp 包为准, 依照其到达的时间来算出下一个包到达的时间[5].

4 音视频同步的实现

由于在 MS 中只是用一个全双工 session 来传送音频或者视频, 不管是本机还是远端主机, 运行的都是同一个程序, 因此一次只能选择一种 payload. MS 的总体逻辑关系非常简单, 它利用函数指针将一系列的 FILTER 组合起来, 通过一个线程来调度这些 FILTER 上的函数指针. 如图 4 为 Mediastream2 的媒体流框架图.

要想实现同步需在结构体 struct MediastreamDatas 中定义两个负载 payload 和 payloadv, 一个为视频一个为音频. 同时就需要创建两个 session, RtpSession *session, RtpSession *sessionv. 如实现音频通话的过程就需要建立两个 graph.其中一个 graph 创建三个 filter,从声卡捕获数据, 然后进行编码, 最后把解码后的数据发送. 两外一个 graph 也创建三个 filter, 从 RTP session 接收数据, 然后解码, 最后把数据解码后发送给播放设备. 同理, 视频传输的过程也类似, 因而所需要它们同步来传输, 就需要两个 session 来同时工作了[6].

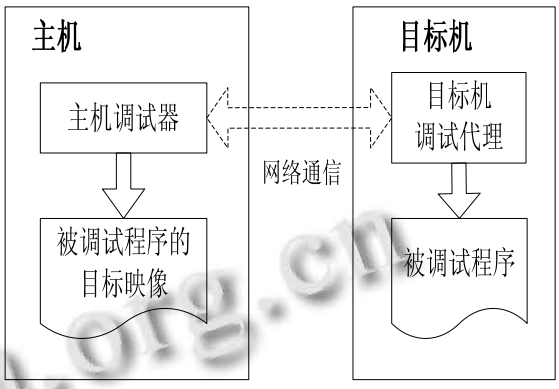

图 4 Mediastream2 的媒体流框架

为了对 mediastream2 的编译, 首先需要从 google 上下载 Android ndk, 然后安装一些自动工具 autoconf, automake, aclocal, libtoolize.然后再对 prepare_sources 进行配置 make./prepare_sources.sh . 最后再编译 ndk-buildNDK_APPLICATION_MK=jni/Application.mk 经过一段时间的编译(图 5 为编译结果图)后最后生成了 armeabi-v7a 和 x86 两个文件夹. 其中在 armeabi-v7a 文件夹中可以看到有我们所需要的.so 文件.

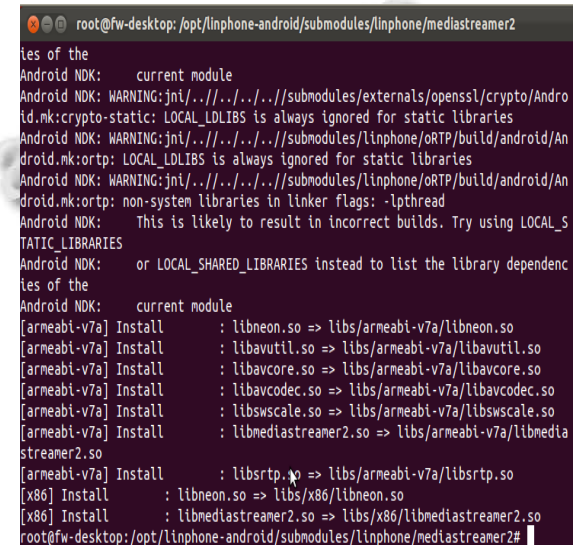

图 5 编译 mediastream2

其中一些库文件简单解释如下 libavcodec: FFMpeg 中进行多媒体编解码的所有函数 libavutil: 一些工具函数库. libswscale: 对视频做缩放处理的函数库. 再将

这两个文件夹拷贝至 Android 工程中 libs 目录下面，如图 6 所示.

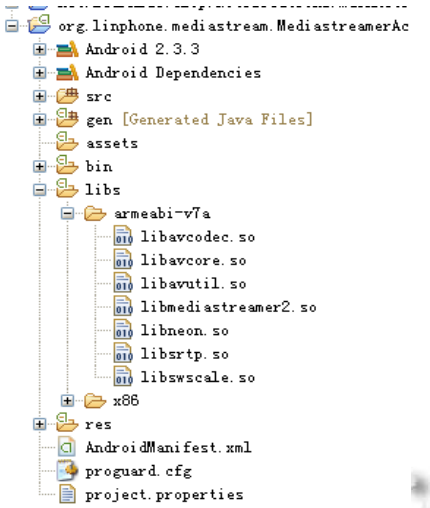

图 6　Android 工程目录

在此类中加载本地库文件，从而就可以调用我们库中的函数来实现我们所想要的函数功能，如图 7 所示.

```
static {
    // FFMPEG (audio/video)
    loadOptionalLibrary("avutil");
    loadOptionalLibrary("swscale");
    loadOptionalLibrary("avcore");
    loadOptionalLibrary("avcodec");
    loadOptionalLibrary("srtp");

    // Main library
    System.loadLibrary("mediastreamer2");
}
```

图 7　JAVA 代码中加载本地库代码

5　测试过程及结果

在本次实验中，采用的是天嵌 TQ210 开发板，测试系统是 Android4.0 系统. 在开发板上安装上经过编译生成的 apk 文件，这个里面的 IP 地址设置成 192.168.1.28(远端设备的 IP 地址). 监控端采用的是 Android 手机，其设置的 IP 地址是 192.168.1.29.

如图 8 所示，上面的为视频采集端，下面的为监控端. 上述两画面各增加了一个本地预览，用来检测是否是实时监控的画面. 由上图可以得出在局域网的环境内，可以起到实时的监控效果.

图 8　采集端与监控端的画面比较

参考文献

1 李宁.Android 开发权威指南.北京:人民邮电出版社,2011.
2 http://www.linphone.org/eng/documentation/dev/media streamer2.html
3 肖飞.Android 视频监控系统的研究与设计.山东工业技术,2013.
4 杨佳胜.基于 Android 终端的视频通话系统设计与实现[学位论文].大连:大连理工大学,2011.
5 丁鹤洋,李太君,徐瑛.3G 无线视频监控系统的设计与实现.通信技术,2012,71,(2):2.
6 吴茜.IP 可视电话中的 SIP 和 Mediastreamer2 技术研究[硕士学位论文].北京:北京邮电大学,2011.