

基于 Android 的手持终端 CoreProcess 系统的设计

韩冰¹, 闫红岩²

(1. 北京理工大学 软件学院, 北京 100081;

2. 郑州轻工业学院 计算机与通信工程学院, 河南 郑州 450001)

摘要:为了在手持终端系统上实现音/视频的转发、播放、录制等功能,设计了手持终端 CoreProcess 系统.该系统分为嵌入式客户端和 Windows 服务器端 2 部分,采用三星公司以 S5PC110 为核心芯片的 Android 系统,使用了 V4L2 视频驱动、ALSA 音频驱动等技术.交付前期的测试验证表明,系统能够稳定工作,在网络传输过程中出现的丢包、码流控制方案不稳等问题也都得到了解决.

关键词:Android;手持终端;CoreProcess

中图分类号:TP393.09 **文献标志码:**A **DOI:**10.3969/j.issn.2095-476X.2012.06.014

Design of handheld terminal CoreProcess system based on Android

HAN Bing¹, YAN Hong-yan²

(1. School of Software, Beijing Institute of Technology, Beijing 100081, China;

2. College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China)

Abstract:To realize the functions of sound/video transmission, playing, recording etc. on the handheld terminal system, the handheld CoreProcess terminal system was designed, which is divided into two parts: embedded client and Windows server. Besides V4L2 video driver, ALSA audio driver, it adopts Android system with the Samsung S5PC110 as the core chip. The test results showed that it can work stably, the problems of losing package, the instability of controlling code stream project and so on are solved in the process of net transmission.

Key words:Android; handheld terminal; CoreProcess

0 引言

随着物联网技术的推广应用,智能视频监控系統越来越多人引起人们的关注.而智能视频分析技术作为智能视频监控的核心技术,引起了国际上许多著名科研机构以及研究人员的兴趣,尤其在美国、英国等国已经开展了大量相关研究,包括运动检测、基于三维模型的车辆与行人的定位识别和跟

踪、基于移动摄像机的视觉监控等.智能视频技术的研究已经取得初步的成果,但是还处于初级阶段.

音/视频编解码适用于专业的音/视频图像分析和处理,基于音/视频编解码技术的高清高精度采集卡可以应用于工业检测、工业测量、智能交通、显微成像等领域多种图像采集处理分析.然而目前市场上大部分音/视频编解码板都没有手持终端,且只有一些专用的功能.鉴于此,本文拟设计基于

收稿日期:2012-09-14

基金项目:国家“十一五”科技支撑计划项目(2006BAK01A38);郑州轻工业学院2009年校科研基金项目(No.24)

作者简介:韩冰(1990—),男,河南省新野县人,北京理工大学本科生,主要研究方向为软件工程;闫红岩(1975—),男,河南省三门峡市人,郑州轻工业学院讲师,硕士,主要研究方向为数据库安全、嵌入式系统.

Android 的手持终端 CoreProcess 系统,以实现音/视频的转发、播放、录制等功能,并解决网络传输过程中的丢包、码流控制方案不稳等问题。

1 CoreProcess 系统主要技术

在 CoreProcess 系统中,主要用到了 Android 和 S5PC110 处理器、串口驱动、V4L2 视频驱动、ALSA 音频驱动等技术。

1.1 Android

Android 是以 Linux 为核心,由软件堆迭的架构延伸发展而来的一套软件平台与操作系统,主要用于便携设备。相比其他手机操作系统,Android 更具开放性,有丰富的硬件选择、开发商不受任何限制、无缝结合的 Google 应用等优势^[1]。

1.2 S5PC110 处理器

三星公司的 S5PC110 处理器采用主频为 1 GHz 的 45 nm 制程的数据处理芯片,是目前功能最强的移动单核处理器。S5PC110 采用的图形显示芯片是 Power VR SGX540,可以达到 9 000 万个/s 多边形输出和 10 亿/s 的像素填充率,S5PC110 处理器上集成了多个 Power VR SGX540 芯片。

S5PC110 内部集成了 4 Gb MuxOneNAND, 2 Gb OneDRAM 和 1 Gb mDDR,不但减小了硬件电路设计的复杂度,而且低功率技术可确保电池有更长的使用寿命。S5PC110 配备功能超强的内建 3D 图形引擎,能支持复杂的 3D UI 及高显示能力的游戏,整合了 1 080 p 高画质编码引擎,能支持 30 f/s 的高画质的影片播放与录像。S5PC110 支持以太网口, VGA 接口,液晶触摸屏, TV OUT, TV IN, I2S 和 AC97 接口, 4 个 USB HOST, 1 个 USB OTG, 4 个串口和 3 路 SD/SDIO/MMC 接口,硬件配备完全能够满足音/视频编解码的需要^[2]。

1.3 串口驱动

在 Linux 系统中,终端是一种字符设备,通常使用 Tty(Teletype 的缩写)来简称各种类型终端设备。串行端口终端是其中的一类,它是使用计算机串行端口连接的终端设备。串行端口所对应的设备名称是 /dev/ttyS0 (或 /dev/tts/0), /dev/ttyS1 (或 /dev/tts/1) 等。

1.4 V4L2 视频驱动

V4L(Video4Linux 或 Video for Linux) 是 Linux 内核中标准的视频驱动程序,目前其版本是 Video4Linux2,简称 V4L2。在 Linux 系统中,摄像头

视频一般规范到了使用 V4L2 驱动程序。V4L2 可以支持多种设备,可以有多种接口。V4L2 驱动的 Video 设备可以支持捕获及视频输出方式,通常使用它作为摄像头的驱动程序。V4L2 驱动的 Video 设备在用户空间通过各种 ioctl 调用进行控制,并且可以使用 mmap 进行内存映射^[3]。

1.5 ALSA 音频驱动

高级 Linux 声音体系 ALSA (advanced linux sound architecture) 是为音频系统提供驱动的 Linux 内核组件,以替代原先的开放声音系统 OSS (open sound system)。

ALSA 是一个完全开放源代码的音频驱动程序集,除了像 OSS 那样提供一组内核驱动程序模块之处,还专门为简化应用程序的编写提供了相应的函数库,与 OSS 提供的基于 ioctl 等原始编程接口相比,使用 ALSA 函数库要更加方便一些。利用该函数库,开发人员可以方便、快捷地开发出自己的应用程序,细节则留给函数库进行内部处理。

ASOC (ALSA system on chip) 是 ALSA 在 SOC 方面的发展,对 CPU 和 Codec 的相关代码进行了分离。本文对于嵌入式系统的声卡驱动开发,采用 ASOC 框架,主要由 Codec 驱动、平台驱动、板驱动 3 部分组成。前 2 部分是通用的驱动,只有板驱动是不通用的,它由特定电路板上具体的 CPU 和 Codec 确定^[4]。

2 系统架构设计

CoreProcess 系统分为嵌入式客户端和 Windows 服务器端 2 部分。客户端是 S5PC110 为核心芯片的 Android 系统,可将模拟视频转化为数字信号,经过 H. 264 main profile 编码,再使用 RTP/RTCP 协议传输到网络中;客户端还可以接收别的客户端或者 Windows 端传送过来的 RTP/RTCP 码率,对音/视频码流进行解码显示。在服务器端,系统提供了登录一个 Android 终端和登录多个 Android 终端、音/视频转发、音/视频的本地播放、录制文件、播放文件等大部分 API 的 demo 实例,用户可以参考这些 demo 实例,研发出符合自己需求的 Windows 端产品。CoreProcess 系统硬件使用 S5PC110 CPU,具有硬编和硬解的功能,系统硬件架构设计如图 1 所示。

CoreProcess 系统的软件设计架构如图 2 所示,CoreProcess 底层是一个 Linux 系统,应用程序通过 JNI 对硬件进行控制。

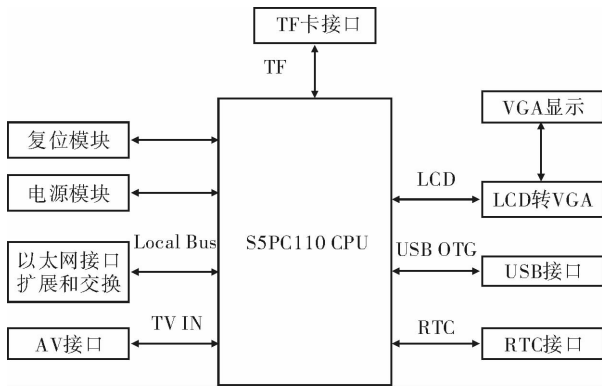


图1 CoreProcess 系统硬件架构图

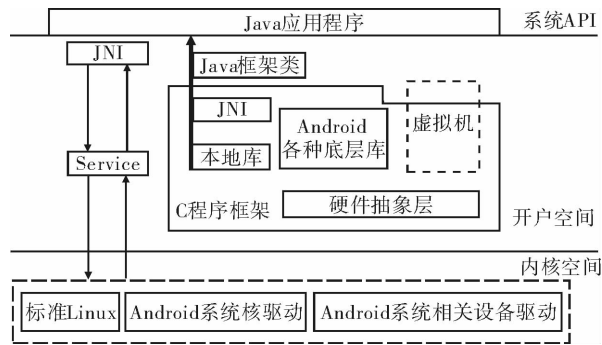


图2 CoreProcess 系统软件架构图

3 系统的移植程序设计

CoreProcess 系统的移植程序设计主要包括串口移植、VGA 驱动、板移植和平台移植。

3.1 串口移植

在 .config 文件中添加 CONFIG_CMDLINE = "console = ttySAC0,115200 init = /linuxrc"^[5]。

3.2 VGA 驱动

在文件 drivers/video/samsung/s3cfb_lte480wv.c 中修改 s3cfb_lcd_lte480wv 结构^[6]。

```
static struct s3cfb_lcd_lte480wv =
{.width = 800, .height = 600, .bpp = 24, .freq
= 60,
.timing = {.h_fp = 40, .h_bp = 470, .h_sw =
110, .v_fp = 2, .v_fpe = 1, .v_bp = 18, .v_bpe
= 1,
.v_sw = 3 },
.polarity = {.rise_vclk = 1, .inv_hsync = 0, .
inv_vsync = 0, .inv_vden = 0}
};
```

3.3 板移植

针对板卡 saa7113 的移植,修改 saa7113_init

函数:

```
static int saa7113_init(struct v4l2_subdev *sd, u32 val)
{ struct i2c_client *client = v4l2_get_subdevdata(sd);
  unsigned char data8;
  unsigned char ID = 9;
  int i = 1, k;
  u32 pup;
  reg_con = ioremap(T_GPD1_CON, 24);
  t_gpio_init(reg_con);
  v4l_info(client, "%s: camera initialization start \n", __func__);
  for(k = 0; k < SAA7113_INIT_REGS; k++)
  { I2C_Start();
    I2C_reg_Write(0x4a, saa7113_init_reg[k][0],
saa7113_init_reg[k][1]);
    i = I2C_reg_Read(0x4a, saa7113_init_reg[k][0],
&ID);
  }
  i2c_deinit();
  return 0;
}
```

3.4 平台移植

对于音/频驱动要有 Codec 和 CPU 的平台驱动,本系统的平台板卡选用 uda1341,对平台驱动做如下修改。

1) 在文件 sound/soc/s3c24xx_wm8580slv.c 中修改函数 smdk64xx_hw_params。

```
static int smdk64xx_hw_params(struct snd_pcm_substream
* substream, struct snd_pcm_hw_params * params)
{ ...
  eppll_out_rate = rclk * psr; ret = set_epll_rate(eppll_out
_rate);
  if (ret < 0) return ret;
  ret = snd_soc_dai_set_sysclk(cpu_dai, S3C64XX_CLK-
SRC_CDCLK, 0, SND_SOC_CLOCK_OUT);
  if (ret < 0) return ret;
  ret = snd_soc_dai_set_sysclk(cpu_dai, S3C64XX_CLK-
SRC_MUX, 0, SND_SOC_CLOCK_IN);
  if (ret < 0) return ret;
  ret = snd_soc_dai_set_fmt(&s3c64xx_i2s_dai[I2S_
NUM], SND_SOC_DAIFMT_I2S
| SND_SOC_DAIFMT_NB_NF | SND_SOC_DAIFMT_CBS
_CFS);
  if (ret < 0) return ret;
  ret = snd_soc_dai_set_clkdiv(cpu_dai, S3C_I2SV2_DIV_
PRESCALER, psr - 1);
```

```

if (ret < 0) return ret;
ret = snd_soc_dai_set_clkdiv(cpu_dai, S3C_I2SV2_DIV_
BCLK, bfs);
if (ret < 0) return ret;
ret = snd_soc_dai_set_clkdiv(cpu_dai, S3C_I2SV2_DIV_
RCLK, rfs);
if (ret < 0) return ret;
ret = snd_soc_dai_set_sysclk(codec_dai, 0, rclk, SND_
SOC_CLOCK_OUT);
if (ret < 0) return ret;
return 0;
}

```

2) 在 sound/soc/codecs/l3.c 中添加.

```

static unsigned int read_l(u32 ptr)
{ return *((volatile unsigned int *)ptr); }
static void write_l(u32 value, u32 ptr)
{ *((volatile unsigned int *)ptr) = value; }
void setclk(u8 value)
{ u32 pup; pup = read_l(reg_dat); pup &= 0xfffffd; if
(value) pup |= 2; write_l(pup, reg_dat); }
static void sendbytes(struct l3_pins * adap, const u8 *
buf, int len)
{ int i;
for (i = 0; i < len; i++)
{ if (i) { udelay(adap->mode_hold); setmode(0); ude-
lay(adap->mode); }
setmode(1); udelay(adap->mode_setup); t_sendbyte
(buf[i]); }
}
int l3_write(struct l3_pins * adap, u8 addr, u8 * data,
int len)
{ setclk(1); setdat(1); setmode(1); udelay(adap->
mode); setmode(0);
udelay(adap->mode_setup); t_sendbyte(addr); udelay
(adap->mode_hold);
sendbytes(adap, data, len); setclk(1); setdat(1); set-
mode(0); return len;
}

```

4 测试验证

在测试阶段,对动态壁纸、SD卡、图片查看器、媒体播放器、音乐播放器、有线网卡/无线网卡上网来检测音/视频的连接、解码、播放、录像等功能进行了验证.通过对VC客户端进行的模块调调用例测试,测试结果如下:VC端播放音/视频5路;输入

码流为 800 000 b/s; CPU 占用率 8%; 内存使用 153 208 K; 带宽 402 kb · s⁻¹/1 Gb · s⁻¹; 实际收到的码流为 805 000 b/s; 音/视频质量良好; 延迟时间为音频 500 ms, 视频 2 s; 稳定运行时间 3 h; 循环切换视频稳定, 为 6 次/s.

在 30 min 内把环境温度提升到 70 ℃, 对 2 个 Android 终端在 2 h 内进行压力测试. 结果表明: 在测试环境中, 音频能够正常播放和使用, 视频也能够正常进行编解码和播放. 后期, 又进一步把 Recorder 做成服务、RecorderSetting(完成 Recorder 部分参数设置)、更正 Recorder bug、增加 OSD 参数设置功能等; 同时, 分别解压 app-jni.rar 到 NDK/samples 目录、解压 SerialPort.rar 到 NDK/samples 目录, 并进行了编译, 实现了 JNI 层的底层操作.

5 结论

本文设计了一个基于 Android 平台的嵌入式手持终端 CoreProcess 系统, 它集成了音/视频编解码卡, 采用 S5PC110 CPU, 进行了音/视频驱动的移植设计. 音频驱动使用了 ALSA 架构, 视频驱动使用了 V4L2 架构; 此外, 还做了 LED 灯、VGA 等驱动, 设计了一个 Android 应用程序, 对 Android 底层类库进行了修改. 该系统已经应用在了某部队的夜视镜上, 到目前为止, CoreProcess 系统能够进行长时间稳定的音/视频编解码, 同时, 在网络传输过程中出现的丢包、码流控制方案不稳等问题也都得到了解决.

参考文献:

- [1] 郭宏志. Android 应用开发详解[M]. 北京: 电子工业出版社, 2010: 40-69.
- [2] 韦东山. 嵌入式 Linux 应用开发完全手册[M]. 2 版. 北京: 人民邮电出版社, 2008: 50-111.
- [3] 宋宝华. Linux 设备驱动开发详解[M]. 北京: 人民邮电出版社, 2008: 21-92.
- [4] 张石锐, 郑文刚, 申长军, 等. 嵌入式手持无线农产品价格信息采集终端[J]. 计算机工程与设计, 2012, 33(2): 514.
- [5] 孙弘羿. 基于智能移动终端的数据监控系统的解决方案[J]. 软件, 2012, 33(2): 61.
- [6] 宋立波. 嵌入式手持智能导航终端摄像头模块设计与实现[D]. 武汉: 华中科技大学, 2011.