

# 自动售货机智能多媒体控制单元的研究与设计

## Research and Design of Intelligent Multimedia Controller in Vending Machine

何益波 金 瓯 贺建飏 (中南大学 信息科学与工程学院 湖南 长沙 410083)

**摘要:** 针对自动售货机中的显示终端显示信息不够生动、人机界面简单等问题,提出了一种基于嵌入式软硬件平台的智能多媒体控制单元,阐述了基于 Intel XScale PXA255 的硬件平台及嵌入式 Linux 操作系统环境下实现智能多媒体控制单元的开发过程。采用组态软件思想实现智能显示软件和基于 ffmpeg 构建多媒体播放软件,并详细介绍了多媒体信息处理的模块化设计及主要模块的构架。该方案实现自动售货机具备多媒体信息处理能力和拥有良好的人机界面。

**关键词:** 自动售货机 嵌入式 Linux 组态软件 智能显示 多媒体播放

近年来,以自动售货机为代表的自助服务在我国的大众城市发展势头非常的迅猛,给人们的生活带来了极大的方便。但是自动售货机在多媒体控制方面表现的不尽人意,表现在利用小屏幕的单色 LCD 显示屏,人机界面单调,只能显示单色的简单字符,显示单一,不够生动,同时不带有音频视频播放功能。

随着信息技术和互联网技术的应用,自动售货机将朝着娱乐化、智能化、信息化方向发展。比如具有良好的人机界面,操作提示界面生动活泼,带有操作提示音频,具有定时自动播送广播新闻、歌曲或视频的节目功能,且能在消费者购物之后自动播放广告,满足行业推广应用的娱乐型售货机。本文提出了一种基于嵌入式软、硬件平台的网络化多媒体智能控制单元,在自动售货机上实现良好人机界面,为自动售货机在多媒体方面提供整体的解决方案。

### 1 系统结构概述

传统的自动售货机显示控制单元是以单片机为处理核心,外接 LCD 显示屏,一般为  $128 \times 64$  的单色 LCD 显示屏,只能显示简单的字符信息。本系统选用 Intel 公司提供的基于 ARM 体系架构的 Xscale PXA255 处理芯片为核心,主板上提供丰富的外部通信接口,包括标准的 RS232 串口、10/100M 自适应

以太网接口和 USB 口等。在嵌入式硬件平台上搭载经过裁剪的嵌入式 Linux 内核和嵌入式文件系统,提高了多任务处理和文件管理等性能。用户层上的多媒体软件采用组态思想和模式设计相结合的方式,实现不同显示情况下的快速重组的功能,使信息的输出和交换更加直观。同时拓展了对网络的支持,可实现网络的远程视频和广播音频的播放。系统扩展图如图 1 所示。

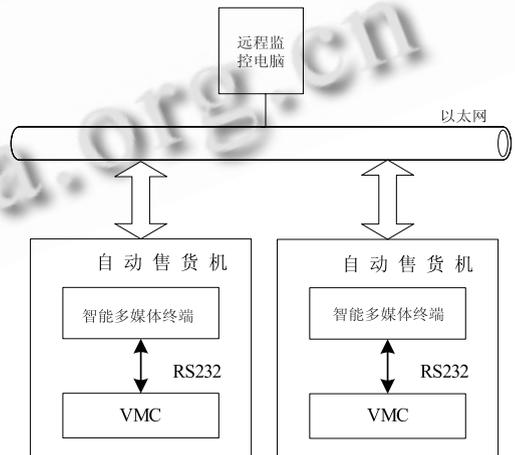


图 1 系统扩展图

系统端接入以太网,接受远程控制 PC 的视频和音频信息,而下端则通过串口和自动售货机的主控制

① 基金项目:国家自然科学基金(60503007)

收稿时间:2008-11-29

器(vending machine controller, VMC)连接, 实时接受 VMC 传过来的显示或者播放命令及相应的显示数据, 然后进行处理。软件环境是已经裁剪过的内核为 2.6.18 的嵌入式 Linux 操作系统<sup>[1]</sup>, 利用 arm-linux-gcc3.3.2 工具链为交叉编译环境。

## 2 软件总体结构

软件系统分为操作系统和应用软件两个部分。操作系统选用了功能和可靠性都很成熟的嵌入式 Linux 操作系统。利用其核心小巧而可靠, 对网络支持比较完善, 同时利用嵌入式 Linux 内核对多文件系统的支持, 搭载了 cramfs、jaffs2 共存的混合文件系统。此方案不仅提供了可操作的用户空间, 并且保证了系统核心文件的不可更改行, 使系统更加强壮, 避免了因使用者的误操作或者频繁的非正常关机等原因导致的系统崩溃。

多媒体控制单元软件作为自动售货机的人机界面生成器, 不仅具有显示界面生成功能, 还具有音视频播放功能, 采用了组态软件的思想构建。软件由界面组态模块、音视频播放组态模块、串口通信模块、以太网通信模块、命令解释模块和文件管理模块 6 个功能独立的模块组成, 其模块构成如图 2 所示。

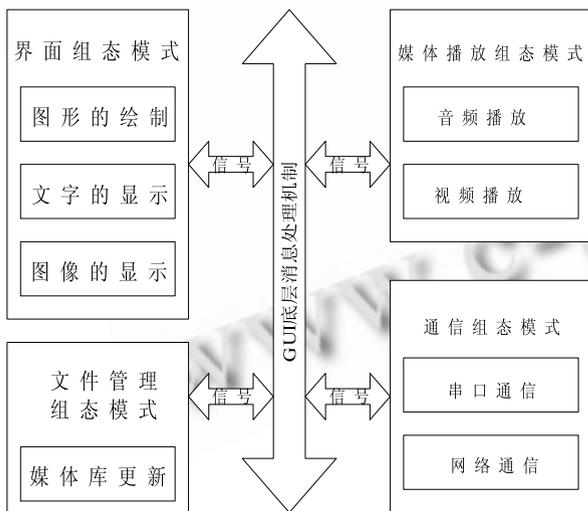


图 2 模块总体结构

从上图中可以看出, 其主要的模块间通过信号的方式相互通信, 避免了过多的相互调用导致耦合性增大等不利因素。考虑到嵌入式系统中存储设备空间不大的特点, 只有 64M 的 FLASH 芯片存储, 音视频播

放采用流媒体播放的方式。当然, 对于自动售货机中的操作提示音, 比如“请投币”、“请选择货物”、“请从掉物仓中取走您买的商品”等体积小而且经常使用的音频, 则保存在多媒体控制单元的存储设备中。界面组态模式是实现良好的人机显示界面, 其中的图像显示模块提供了 GIF 图片动画的支持, 可以用来提供用户提示操作的界面显示。

## 3 多媒体控制软件的详细设计与实现

该系统在图形界面开发上, 直接利用操作 Framebuffer 进行图形开发; 在音视频播放方面, 利用 ffmpeg 音频、视频编解码库, 实现对 mpeg4 文件的播放。

### 3.1 图形显示

FrameBuffer 设备是运行在 Linux 控制台下的一个优秀的图形接口, 作用在于把不同的图形设备抽象为一个逻辑设备, 能将显存里的内容映射到屏幕上相应的位置显示。通过 FrameBuffer 构建一个图形库, 提供一些基本的绘图功能和对图像显示的支持, 向上层提供统一的接口<sup>[2]</sup>。

#### ① FrameBuffer 初始化

FrameBuffer 的设备节点是 /dev/fb。由于 FrameBuffer 工作在终端之上, 所以首先要屏蔽终端的输入, 使终端进入图形工作模式, 这样可以通过下面的代码完成。

```
tty = open( "/dev/tty" , O_RDWR);
ioctl(tty, KDSETPRM, KD_GRAPHICS);
```

接着打开 fb 设备, 读取 fb\_fix\_screeninfo 和 fb\_var\_screeninfo。从 fb\_fix\_screeninfo 和 fb\_var\_screeninfo 中可以获得分辨率、色深和颜色比例等信息, 并根据这些计算出需要映射显存的大小, 然后通过 mmap() 函数把显存映射到用户空间。将显存映射到用户空间之后, 我们就可以直接操作用户空间来控制, 而不需要用 read、write 等底层函数去操作显存来控制显示, 这样可以加快显示速度<sup>[3]</sup>。完成 fb 设备的初始化, 接下来就可以构建图形库。

#### ② 图形库的构建

在这一个部分只要完成画点、画线、填充区域等基本功能的绘图功能以及对 GIF 和 JPG 图片显示的支持。图形设备的色深、像素映射方式等属性决定了显示模式的不同。本系统中获得 LCD 设备的参数是

640\*480, 色深为 16 位, 颜色组成为 r5g6b5 模式。则对于屏幕上的一个点, 存储需要两个字节的空  
red 占高 5 位, green 占中间的 6 位, blue 占低 5 位。以屏幕的左上角为坐标原点, 屏幕上的一个点(x, y), 在用户空间的内存中的偏移量计算如公式 1 所示。

$$\text{offset} = (y * 640 + x) * 2 \quad (1)$$

offset 的偏移量的单位为字节。绘制一个点, 则只要找到该点在映射显存的内存中的偏移量, 然后写入该点的两个字节的颜色信息即可。完成画点之后, 则画线和填充区域都好办了, 这些都有成熟的图形学算法[4], 就不再赘述了。

在建立图形库的过程中, 采用组态软件的思想[5]。图形界面组态是组态软件中所有功能输入接口及输出结果的直接表现, 是用户通过组态软件提供的界面组件将采集到的数据按事先预订的某种方式在界面上显示出来的结果。在界面组态模块的设计中, 需要对数据和其界面呈现进行充分的解耦, 以便在实际组态时, 可以动态选择并配置合适的呈现方式, 而不是将工位固定绑定于某一种界面上呈现出来[6]。考虑上述原因, 选择面向对象设计中的 Bridge 模式。Bridge 模式允许一个抽象的基类有多种实现, 通过将抽象部分与实现部分完全解耦, 使两者可以相互独立的变化, 并灵活地组合起来使用, 具体的设计如图 3 所示。

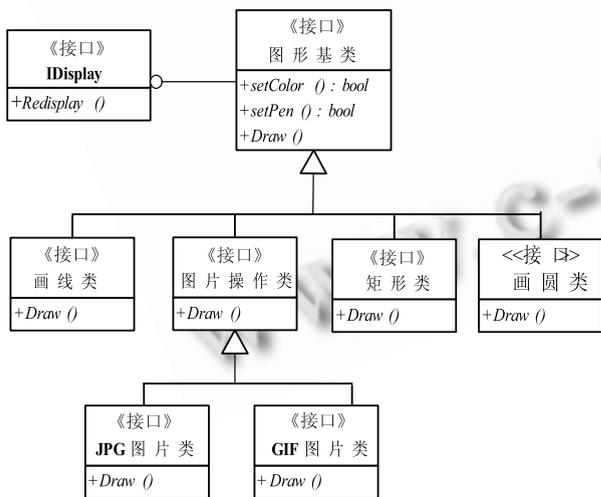


图 3 界面组态结构

在图片显示方面, 提供了 JPG 和 GIF 图片显示的支持。首先读取图片, 然后利用 Linux 中的 libjpeg、libgif 库对图片解码, 得到图片的大小、色深以及显

示数据, 然后调用画点函数实现在屏幕上的绘制。

### 3.2 多媒体播放

多媒体播放包括音频播放和视频播放。音频播放功能主要是是在用户购物的时候, 在界面提供显示 GIF 图片形式的操作提示的同时, 能够配以音频提示操作。这些提示操作的音频信息不大, 只有一、两句话长的音频, 且播放频繁, 故采用 WAV 的格式保存在多媒体控制单元的本地 Flash 存储器中。由于 WAV 格式的音乐直接写入音频设备文件/dev/dsp中即可播放, 故可以避免频繁的解码。视频主要是用在某一时间段, 能够播放视频节目。由于视频文件比较大, 而本地的 Flash 存储器容量有限, 故采用利用 mpeg4 压缩的流媒体播放的方式。

本系统采用 ffmpeg 库进行多媒体播放器开发。ffmpeg 是一个开源免费的项目, 它提供了录制、转换以及流化音视频的完整解决方案。它包含了非常先进的音频/视频编解码库 libavcodec, 能对 mpeg4 格式提供很好的解码方案, 并集合了 RTSP 协议和 HTTP 协议[7]。在利用 ffmpeg 的编程中, 主要利用其提供的两个重要的库文件: libavformat 和 libavcodec: libavformat 用来处理解析视频文件并将包含在其中的流分离出来, 而 libavcodec 则处理原始音频和视频流的解码。系统模块划分如下:

① GUI 模块。利用 FrameBuffer 接口, 将视频解码后得到的每一帧数据——PIX\_FMT\_YUV420P 帧结构, 转化为 PIX\_FMT\_RGB565 帧结构模式, 然后将这一帧的数据拷贝到映射显存的内存空间中。

② RTSP 模块。流式传输需要合适的传输协议, 由于 TCP 需要的开销比较大, 不适合做传输的实时数据。RTSP 协议虽然比 HTTP 协议复杂, 但是能提供双向性的控制会话建立和多媒体数据的播放、暂停、结束等操作, 故选用 RTSP 协议。

③ 音视频解码模块。利用开源库 ffmpeg 中的音视频解码库, 能实现 mpeg4 格式音视频解码。

④ 音频设备属性设置模块。解码音频之后, 得到音频的属性, 然后设置音频设备的播放属性和音频的属性一致, 否则不能正常播放。需要设置的属性包括采样频率、量化位数、声道数。

首先从数据源(多媒体文件)读入数据, 然后注册 ffmpeg 库中含有的所有可用的文件格式和编码器。由于音视频数据是分开传输的, 通过寻找音频流和视

频流标号,就找到音频流和视频流的入口,而不用经过分离器就可以将它们分别存放在视音频缓冲区中,然后寻找音频流和视频流的解码器。在同步方面,采用视频同步音频的方式。设置好声卡设备的属性,主要包括采样频率、量化位数、声道数,然后对音频流解码,得到脉冲编码调制(Pulse Code Modulation, PCM)数据,就可以直接向/dev/dsp 设备文件写入数据即可。对视频流解码之后,调用 GUI 模块将视频帧数据显示在 LCD 上。播放流程如图 4 所示。

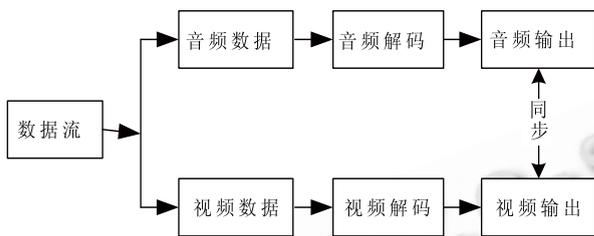


图 4 播放器数据流程

### 3.3 通信模块和文件管理模块

为了实现对现场数据的输出与控制,必须要和自动售货机的 VMC 及远程管理服务器进行通信。其中,该智能多媒体控制单元与 VMC 通信利用串口,与远程管理服务器的通信采用以太网。

VMC 发送过来显示命令,通过解析之后,调用界面图形组态,迅速构建显示界面。该系统采用基于多线程技术的串口通信的模型<sup>[8]</sup>来构建串口通信,使得前端人际交互部分、中间处理部分和后台的串口操作部分并行处理,让耗时的 I/O 操作在后台运行。在大数据量通信的情况下对改善程序的响应速度是相当有效的,并且对多个串口设备同时操作成对一个串口同时进行读写操作的处理也非常的成功,提高了程序的响应速度和资源的利用率。串口通信波特率设置为 9600bps、8 位数据位、1 位停止位、奇校验和无数据流控制。同时在该控制单元和 VMC 通信过程中,需要等待 VMC 发送控制命令。为了让程序等待 VMC 发送控制命令,串口在读控制命令的函数设计中,应使用阻塞操作。在串口设置中,将 c\_cc[MIN]设置为大于 0, c\_cc[TIME]设置为等于 0,在这种情况下,read 调用一直等待,直到有大于或等于 MIN 个字符可以读取时才返回,返回实际读取的字符数。

多媒体控制单元与远程管理服务器通信,主要是

完成流媒体文件播放和存储在本地的多媒体文件的升级。由于一些多媒体文件比较小,故保存在本地的 Flash 中。比如:操作提示音、操作提示的 GIF 图片动画、GIF 图片动画形式的商品广告等。这些文件有时候需要升级更新,采用以太网的方式,从远程管理服务器发送这些多媒体文件,然后将其保存到本地的 Flash 中,同时也可以通过名字相同而覆盖掉原有的多媒体文件,从而实现多媒体文件升级。

## 4 结语

本文在研究了自动售货机的显示单元现状及智能多媒体控制单元的发展方向之后,提出了一种基于 PXA255 嵌入式处理器的硬件平台、内嵌嵌入式 Linux 操作系统、图形 GUI 的智能化显示及多媒体音视频播放系统。该系统可以取代现有的自动售货机显示设备,同时提供了音视频播放功能,并且提供了对网络的支持,可以方便的实现多媒体文件的升级和流媒体的播放。本系统在自动售货机行业有很强的推广性,促使售货机朝着网络化、娱乐化方向发展。

## 参考文献

- 1 Matthew N, Stones R. Linux 程序设计(第三版).陈健,宋健健译.北京:人民邮电出版社,2007.
- 2 吴峰,王自强.基于 FrameBuffer 的嵌入式 GUI 系统设计.计算机应用与软件,2005,22(3):128-130.
- 3 赵霞.基于 Framebuffer 的嵌入式 GUI 系统实现.微计算机信息,2007,23(12-2):100-102.
- 4 Ploskas N, Berger M, Jiang Z, Wintterle GJ. A knowledge management framework for software configuration management. USA: 2008 IEEE 32nd International Computer Software and Applications Conference (COMPSAC):593-600.
- 5 倪明田,吴良芝.计算机图形学.北京:北京大学出版社,2004:43-88.
- 6 洪中.组态软件设计中的模式应用.微计算机信息,2007,23(10):219-281.
- 7 任严,韩臻,刘丽.基于 FFMPEG 的视频格式转化与发布系统.计算机工程与设计,2007,28(20):4962-4963.
- 8 张晓华,陈曦.多串口通信系统设计与实现.计算机与数字工程,2005,33(12):147-150.