

# 基于 Framebuffer 的图片处理系统<sup>①</sup>

袁 琼<sup>1</sup>, 陈 兵<sup>2</sup>

<sup>1</sup>(武汉东湖学院 计算机科学学院, 武汉 430212)

<sup>2</sup>(武昌理工学院 信息工程学院, 武汉 430223)

**摘要:** 帧缓冲 FrameBuffer 图像处理技术, 可以方便灵活地操纵液晶屏, 实现液晶屏局部刷新无闪烁, 同时为上层应用的开发带来很大的方便。鉴于此, 提出了一种基于 FrameBuffer 的图片处理系统的开发流程, 并在 Mini2440 开发板上实现了流程中的具体模块功能。

**关键词:** 帧缓冲; FrameBuffer; 图片处理系统

## Image Processing System Based on the FrameBuffer

YUAN Qiong<sup>1</sup>, CHEN Bing<sup>2</sup>

<sup>1</sup>(Wuhan Donghu University, Computer Science Institute, Wuhan, 430212, China)

<sup>2</sup>(WuChang University of Technology, Information and Engineering school, Wuhan, 430223, China)

**Abstract:** FrameBuffer image processing technology can manipulate the LCD screen easily and flexibility, and it achieve the partial refresh flicker-free LCD screen. While it has brought great convenience on the upper application development. In view of this, this paper presents a FrameBuffer-based image processing system development process, and achieve a specific process module function on Mini2440 development board.

**Key words:** FrameBuffer; image processing system

### 1 Framebuffer简介

Framebuffer<sup>[1]</sup>是出现在 2.2.xx 内核当中的一种驱动程序接口。

Linux 是工作在保护模式下, 所以用户态进程是无法象 DOS 那样使用显卡 BIOS 里提供的中断调用来实现直接写屏, Linux 抽象出 FrameBuffer 这个设备来供用户态进程实现直接写屏。Framebuffer 机制模仿显卡的功能, 将显卡硬件结构抽象掉, 可以通过 Framebuffer 的读写直接对显存进行操作。用户可以将 Framebuffer 看成是显示内存的一个映像, 将其映射到进程地址空间之后, 就可以直接进行读写操作, 而写操作可以立即反应在屏幕上。这种操作是抽象的, 统一的。用户不必关心物理显存的位置、换页机制等等具体细节, 这些都是由 Framebuffer 设备驱动来完成的。

### 2 基于Framebuffer的图片处理系统的开发流程

本文是基于 Linux 平台, 利用 Framebuffer 显示图像原理制作的图片处理系统, 整个系统的开发大概分为如下几步: Framebuffer 初始化; 获取所有图片名字; 判定图片显示格式; 显示图片预处理; 确定显示图片效果; 释放资源。其开发流程如图 1 所示。

### 3 系统中各个功能模块具体设计与实现

#### 3.1 Framebuffer 初始化

首先, 打开 Framebuffer 设备; 然后, 获取设备的参数, 比如, 屏幕宽度、高度、位数等信息; 最后, 在内存中映射一块空间, 空间大小为屏幕尺寸大小所占的空间。

在 linux 下, 每一个设备都是一个文件, 在/dev 下

① 收稿时间:2011-07-22;收到修改稿时间:2011-09-11

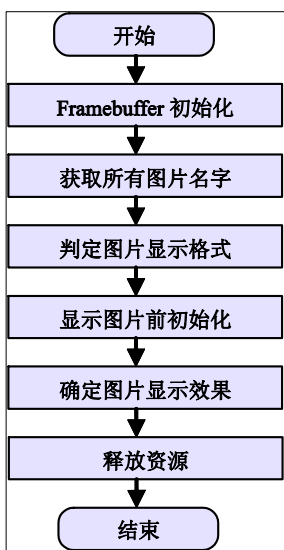


图 1 基于 Framebuffer 图片处理系统开发流程图

都有对应的设备文件，其中 fb0 就是本系统所需要的设备文件，本文所有的操作都是基于该设备 fb0<sup>[2,3]</sup>。

获取设备参数方法如下：

```

ioctl(fb_dev,FBIOGET_VSCREENINFO,&fb_var);
screen_w = fb_var.xres;
screen_h = fb_var.yres;
screen_bits = fb_var.bits_per_pixel;
    
```

Framebuffer 形象地比喻成一个水池，但是，该水池中存储的不是水，而是一些颜色信息。因此，需要在内存中分配这么一个水池，水池大小就是整个屏幕所有像素点占的空间。

### 3.2 获取所有图片名字

从图片文件夹中，搜索出所有图片的文件名，并存入名字文件链表中，以便于管理和搜索图片。搜索图片模块流程如图 2 所示。

把所有图片名字存入到一个 txt 文本中，然后对该文本操作。为了方便后面的操作，需要把 images.txt 中的每个图片名字提取出来，存入一个链表中，链表中每个节点包含图片 id（为了以后升级预留接口）、图片名字，具体节点定义如下：

```

{
int images_id;
char images_name[IMAGES_NAME_SIZE];
struct images_list *next;
}IMAGES_LIST;
    
```

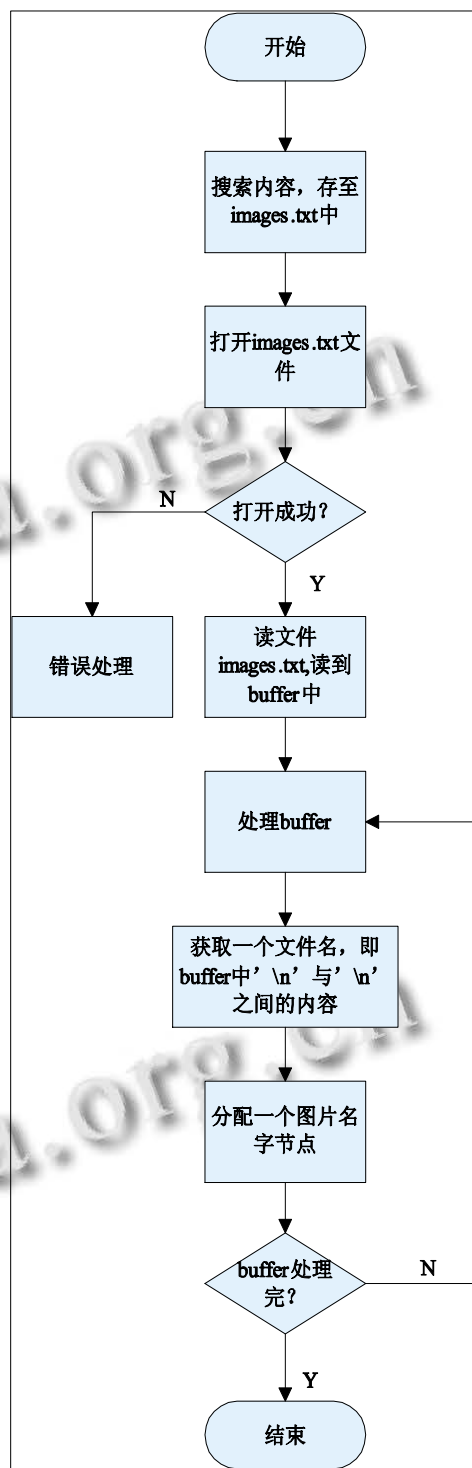


图 2 图片搜索流程图

```

typedef struct images_list
    
```

由于 images.txt 文件中每一个图片名字是一行一行存储的，所以可以根据 '\n' 来截取图片名字，具体方法如下：

首先，一次读取 1024 字节个文件

```
memset(buffer, 0, sizeof(buffer));
```

```
fread(buffer, sizeof(buffer), 1, images_fd);
```

其次，循环的获取每个图片名字，找出合法的图片名字，创建一个图片节点，把 id 赋值，图片名字赋值，把该节点插入到链表中去，如果遇到不是合法的图片名字，比如 images.txt，就把该内容过滤，进行下一次循环，直至整个内容搜索完毕。

### 3.3 判定图片显示格式

从图片文件名链表中取出一个节点，得到图片文件名，打开该图片，读取前 8 个字节，根据这 8 个字节判断图片类型，常见图片类型：jpeg/jpg 是 0xff，png 是 0x89，bmp 是"BM"。判定图片格式模块流程如图 3 所示。

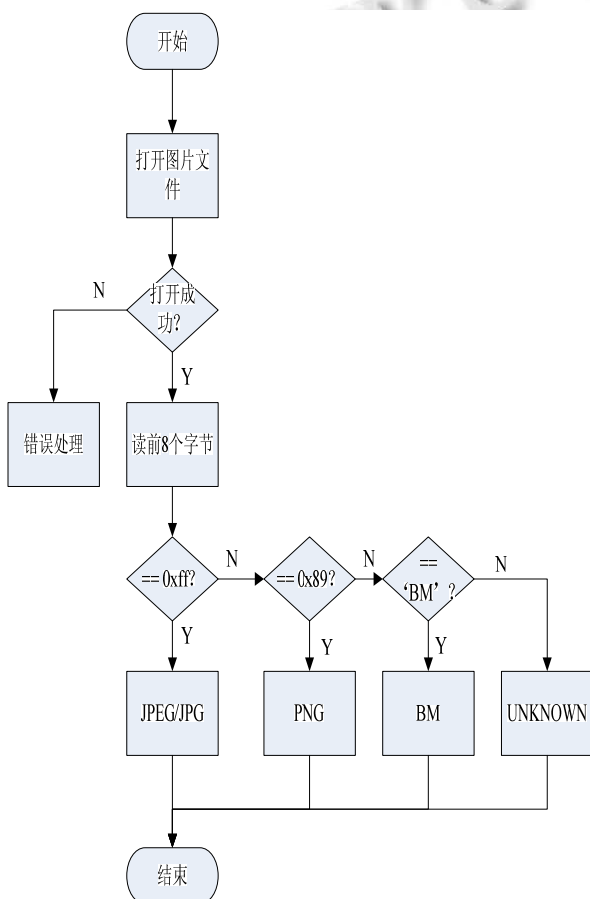


图 3 判定图片格式流程图

### 3.4 显示图片预处理

本文主要是显示 jpeg/jpg 格式图片，对于该类型图片，在 linux c 开发下有支持的 libjpeg 库，使用该库

首先确定解压缩对象，其次，解压，获取图片信息，最后要释放解压缩对象<sup>[3]</sup>。

#### 3.4.1 创建一个解压缩对象

```
jpeg_create_decompress(&cinfo);
```

#### 3.4.2 指定要解压缩的图像文件

```
jpeg_stdio_src(&cinfo, infile);
```

#### 3.4.3 获取图像信息

```
jpeg_read_header(&cinfo, TRUE);
```

#### 3.4.4 开始解压

```
jpeg_start_decompress(&cinfo);
```

#### 3.4.5 获取图片参数

在解压完成后，可以获取一些图片常见的参数，比如图片宽度、高度、颜色位数等基本信息。

```
picture_w = cinfo.output_width;
```

```
picture_h = cinfo.output_height;
```

```
picture_bits = cinfo.output_components;
```

#### 3.4.6 判定处理

考虑本系统的健壮性，有时会出现图片大小大于屏幕大小的情况，如果不对该情况做预处理，可能会导致本系统崩溃，所以本文在该部分预加了判定处理，具体如下：

```

if((picture_w > screen_w) || (picture_h > screen_h))
{
    printf("%s\n", ERR_BIG);
    exit(1);
}
  
```

### 3.5 确定显示图片效果

本文实现了 7 种效果，如图 4 所示，分别是：普通效果、自上向下覆盖效果、自左向右覆盖效果、上下百叶窗效果、左右百叶窗效果、原形效果、风车效果，在选择图片如何显示时，本文采用随机数确定显示图片效果，用时间作为随机种子。

### 3.6 释放资源

主要是释放动态分配的内存，释放图片解压缩对象，关闭打开的文件及设备。

## 4 结语

帧缓冲 FrameBuffer 图片处理技术在嵌入式多媒体系统中的应用非常合适，文中给出的图片处理系统的设计具有实际参考价值。

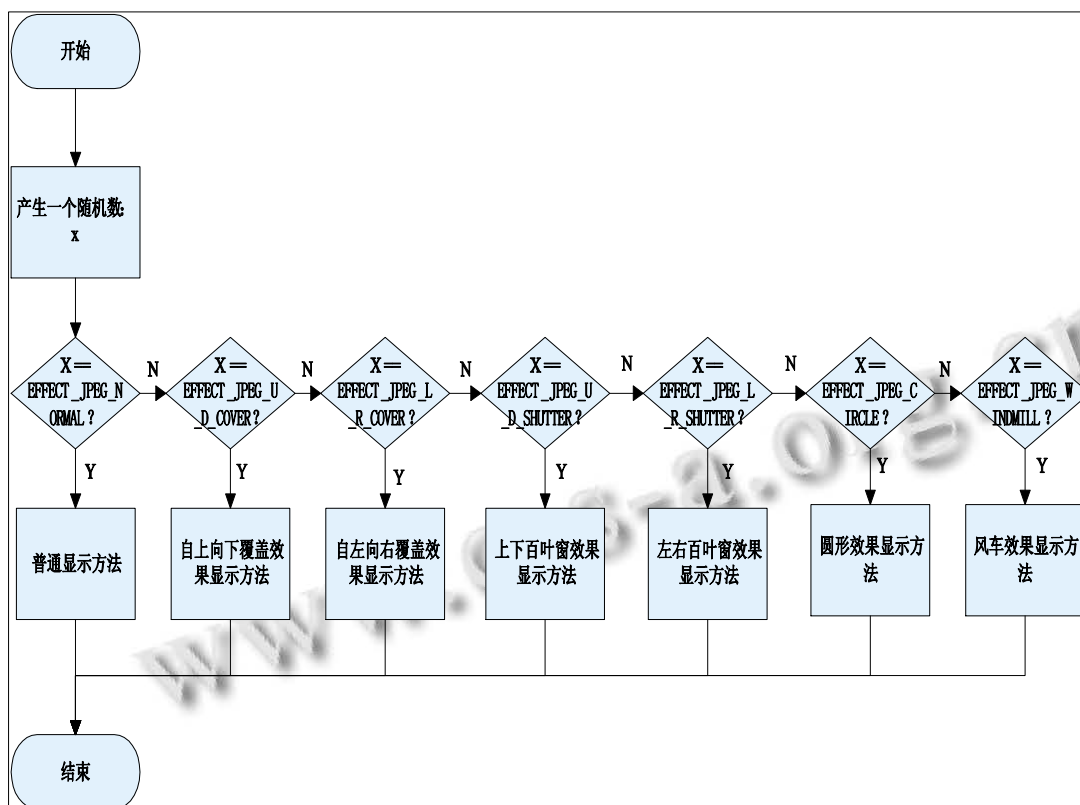


图 4 显示图片流程图

参考文献

- 1 储光,刘京南.基于嵌入式 ARM 的 LCD 图像显示系统设计. 电子工程师,2007,8.
- 2 Sally. Linux 嵌入式系统高级程序设计.北京:人民邮电出版社,2010.
- 3 刘刚,赵剑川.Linux 系统移植.北京:清华大学出版社,2010.
- 4 Donoho DL, Kerkyacharian G, Picard D, Johnstone IM, et al. Density estimation by wavelet thresholding. The Annals of Statistics, 1996,24(2):508-538.
- 5 朱呈祥,邹云.基于小波分析的分数阶系统辨识信号降噪的变尺度阈值方法.计算机应用,2011,(2):543-547.
- 6 Daubechies I. Orthonormal Bases of Compactly Supported Wavelets II. Variations on a Theme. SIAM Journal on Mathematical Analysis, 1993,24(2):520-527.
- 7 王莉,张浩.基于 Web 的现场总线控制系统的远程监控.上海电力学院学报,2008,24(1):35-38.
- 8 刘红.TC35 无线模块的数据传输实现.电信工程技术与标准化,2007,(3):61-63.
- 9 田丰,敦旭峰,孙小平,吕鸣.基于 TC35 的无线传感器网络远程监控系统.计算机工程,2008,(23):105-109.

(上接第 50 页)

- 用.无损探伤,2002,(4):1-4.
- 2 王微微,朱小倩,王平,樊尚春,任东顺.凝析天然气流量测量新方法.自动化仪表,2010,(8):36-40.
- 3 彭玉华.小波变换与工程应用.北京:科学出版社,2006.
- 4 Donoho DL, Kerkyacharian G, Picard D, Johnstone IM, et al. Density estimation by wavelet thresholding. The Annals of Statistics, 1996,24(2):508-538.
- 5 朱呈祥,邹云.基于小波分析的分数阶系统辨识信号降噪的变尺度阈值方法.计算机应用,2011,(2):543-547.

- 6 Daubechies I. Orthonormal Bases of Compactly Supported Wavelets II. Variations on a Theme. SIAM Journal on Mathematical Analysis, 1993,24(2):520-527.
- 7 王莉,张浩.基于 Web 的现场总线控制系统的远程监控.上海电力学院学报,2008,24(1):35-38.
- 8 刘红.TC35 无线模块的数据传输实现.电信工程技术与标准化,2007,(3):61-63.
- 9 田丰,敦旭峰,孙小平,吕鸣.基于 TC35 的无线传感器网络远程监控系统.计算机工程,2008,(23):105-109.