

基于 CMOS 图像传感器的嵌入式视频采集系统设计^①

Design of Embedded Video Accessing System Based on CMOS Image Sensor

廖梦云 赵利 莫金旺 (桂林电子科技大学 广西 桂林 541004)

摘要: 设计了一个基于 CMOS 图像传感器的嵌入式视频采集系统, 系统采用 OMNIVISION 公司的 MI0360 作为图像采集芯片, 使用 SONIX 公司的 SN9C120 芯片作为视频处理器, 完成了硬件电路的设计, 利用 V4L2 接口规范实现了视频图像的实时采集。该设计已应用于视频监控系统中。

关键词: SN9C120 MI0360 CMOS 图像传感器 V4L2 嵌入式

1 引言

目前图像系统的数字摄像技术中, 主要采用 CCD (charged coup led device) 和 CMOS (互补金属氧化物半导体) 图像传感器。CCD 图像传感器具有读取噪声低、动态范围大、响应灵敏度高优点, 广泛应用于各种摄录机和照相机的图像采集系统中, 但 CCD 技术难以与主流的 CMOS 技术集成于同一芯片之中, 且 CCD 图像传感器具有体积大、功耗高等缺点。CMOS 图像传感器是近些年发展较快的新型图像传感器, 采用 CMOS 技术, 可以将像素阵列与外围支持电路集成在同一块芯片上。与 CCD 传感器相比, CMOS 图像传感器具有体积小、集成度高、功耗低、抗干扰能力强和易于控制等优点, 所以广泛应用于摄像头、扫描仪、可视电话、视频会议和视频监控等领域中^[1,2]。

图像采集系统通常使用专门的 DSP 芯片作为图像数据处理器, 外加 SDRAM、FLASH 存储器和可编程逻辑器件 CPLD 来协调 DSP 芯片的时序。在芯片功能高度集成化的今天, 这种图像采集系统结构显得复杂^[3]。本设计采用 CMOS 图像传感器 MI0360 和集成了 USB 接口、大容量内部寄存器等功能的单晶片视频处理器 SN9C120, 电路简单, 功耗低, 成像清晰稳定。

2 系统构成

该视频采集系统主要由 CMOS 图像传感器 MIO

360 和单晶片视频处理器 SN9C120 构成, 系统框图如图 1 所示。

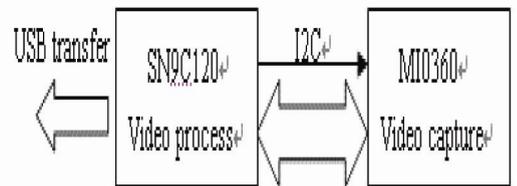


图 1 系统框图

其中, CMOS 图像传感器 MI0360 完成原始视频数据的采集, 单晶片视频处理器 SN9C120 对 MI0360 采集的视频数据进行数字化等方面的处理, 数字化后的视频流经 SN9C120 的 USB 接口传输出去。

3 系统采用的主要芯片

3.1 视频处理器 SN9C120

SN9C120 是 SONIX 科技针对 PC Camera 这类相关应用所开发出来的一种高效能、低成本及低功耗单晶片视频处理器, 可搭配 VGA (视频图像矩阵) 图像传感器的 PC Camera 控制器, 提供了一个更低功耗, 更省成本, 更高图像质量的解决方案^[4]。它将传感器接口, 图像信号处理, 图像压缩和 USB 传输功能集成在一块芯片上。其主要功能如下:

- 支持 9 位 RGB Bayer 格式数据输入的多种 CM

^① 基金项目: 广西教育厅科研项目(200808XL128); 广西硕士研究生科研创新项目(2008105950810M417)

收稿时间: 2008-10-13

OS 图像传感器;

- 同时提供如边缘增强, 自动曝光和自动白平衡控制等图像信号处理模块;

- 提供像素偏移补偿、相互独立的 R/G/B 数字色彩增益控制和用平滑滤波器进行 1/2, 1/4 幅面压缩功能;

- 内建的 JPEG 压缩引擎使得在 USB2.0 环境下 VGA 格式的视频图像帧速率能达到 30fps;

- 辅助的窗口操作和幅面压缩功能可以提供不同的图像格式, 比如 SBGGR8 和 JPEG 格式;

- 采用硬件色彩 DSP 和 JPEG 图像压缩方法, 视频数据可以通过 USB 异步模式或块模式传输;

- 数码变焦功能可以满足任何特殊场合的使用;

- 不需要外部存储结构, 其内部寄存器足够使用。

3.2 CMOS 图像传感器 MI0360

MI0360 是 OMNIVISION 公司生产的一种单芯片数字彩色图像器件, 采用 $0.3\mu\text{m}$ 的 CMOS 工艺, 内置了一个 640×480 的传感器阵列、数模转换器和相关双采样电路, 有效像素约 30 万, 功耗低于 90mW, 具有曝光控制、增益控制等功能, VGA 格式的最大帧率可达 30fps。它的控制部分使用 I2C 总线, 只需要一个 I2C 总线的从设备地址就可以对其内部的控制和状态寄存器进行设置和读取, 不需要外接控制器就可以直接输出数字图像, 同时输出标准的 VSYNC、HSYNC 和 VCLK, 用作外部电路读取图像数据的参考信号。通过对 MI0360 寄存器的编程, 可以设定图像的曝光时间、分辨率、帧率、RGB 增益、镜像等。MI0360 可以输出 10 位 Bayer 格式的图像数据, 其中的 RGB 颜色分量可以被数字增益所调整, 从而反馈给处理器进行色彩处理或压缩^[5]。

4 系统硬件电路设计

系统硬件电路设计主要考虑图像传感器的电源电路以及控制电路。MI0360 使用 3V 的电源驱动、24MHz 的时钟频率, 可输出 9 位图像数据, 控制部分使用 I2C 总线。

4.1 电源电路的设计

由于 CMOS 图像传感器是电源敏感元件, 如果电源不稳定会给成像效果造成很大的影响。USB 总线供电大都在 5V 左右, 而 MI0360 的供电电压为 3V, 最小 2.6V, 最大 3.1V, 并且分开了数字电源和模拟电

源, 因此必须采取相应的机制保证电源稳定的同时还需保证数字电源和模拟电源的隔离, 为了兼顾这两者的要求, 本电源设计采用专用的稳压芯片 SE33CR133A 和电容、电阻、电感等其它器件来构成。如图 2 所示, USB 总线所供的电压经电感 L1、电容 C1、C2、E1 消振后进入稳压芯片 SE33CR133A, 到 C5 端可获得稳定的 3.3V 电压供 SN9C120 使用, 经 D1 降压后可作为 MI0360 的使能信号 ENB, 使 MI0360 上电后就工作在使能状态, E5 端为稳定的 3V 电压提供给 MI0360。本电源设计很好的满足了不同芯片对不同电压的要求, 且不会对视频信号经 SN9C120 的 USB 口传输进行干扰。

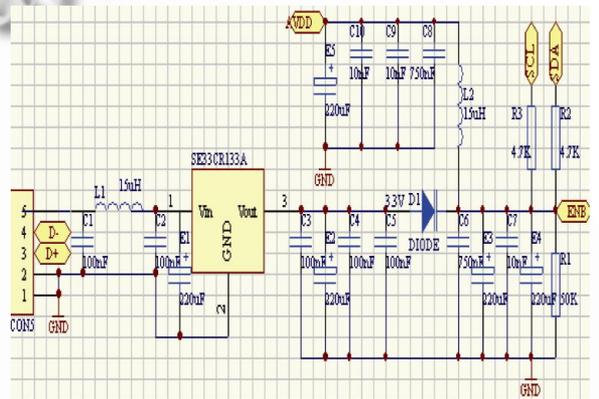


图 2 电源驱动原理图

4.2 控制电路的设计

从控制电路原理图(图 3)可以看出, SN9C120 使用外部晶振 12MHz 经内部倍频后从 SEN_CLK 引脚输出向 MI0360 提供 24MHz 的主时钟 MCLK, 从而达到时钟信号的同步; MI0360 的视频时钟 VCLK、帧同步信号 VSYNC 和水平同步信号 HSYNC 分别跟 SN9C120 的 S_PCK、S_VSYNC 和 S_HSYNC 连接; MI0360 的复位信号 RESET 可由 SN9C120 的可编程引脚 GPIO1 来提供。由于 SN9C120 只支持 9 位的 RGB Bayer 格式数据, 所以本设计中只采用 MI0360 的 9 位 D1...D9 进行图像采集。SN9C120 对 MI0360 的数据读写通过 I2C 总线进行。

5 视频采集的实现

本系统的视频采集在 linux 平台下进行, 主要分两步来实现: 一是 SN9C120 驱动在 linux 内核中的实现; 二是利用 Video4Linux2(V4L2)接口函数编写

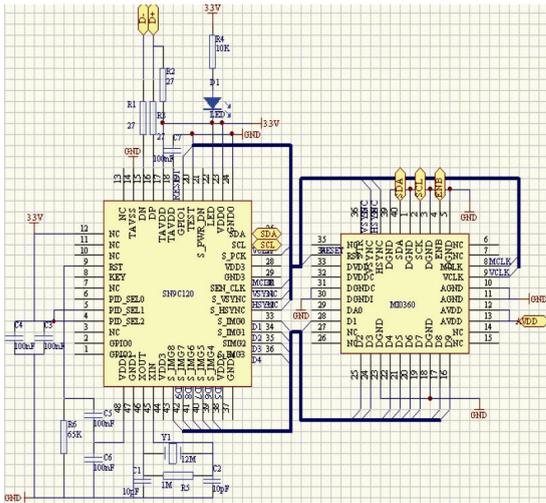


图 3 控制电路原理图

应用程序实现视频采集。本文着重讨论第二步的实现方法。

5.1 设备驱动的设计

在 Linux 下，设备驱动程序可以看成是 Linux 内核与外部设备之间的接口。由于设备驱动程序向应用程序屏蔽了硬件实现上的细节，所以应用程序可以像操作普通文件一样来操作外部设备。驱动程序的主要任务就是调用系统的标准接口来完成对硬件设备的打开、关闭、读写和 I/O 控制操作。

V4L2 是为 inux 下视频采集设备驱动程序的编写提供统一接口而重新设计的一套规范(API)，相比与 Video4Linux，V4L2 接口功能更加强大。V4L2 上层是 videodev 模块，当 videodev 初始化后，把自身注册一个主设备号为 81 的字符设备，同时注册自己的字符驱动成员函数；下层是 V4L2 驱动程序，当其初始化后，把一个包含 V4L2 驱动程序成员函数，次设备号及其它相关信息的结构传递给 videodev，从而把要处理的设备注册到 videodev。当应用程序触发了一个驱动程序调用时，控制权首先传递给 videodev 中的函数，videodev 负责将应用程序传递的文件或节点结构指针转化为相应的 V4L2 结构指针，并调用 V4L2 驱动中的处理函数^[6]。

利用 V4L2 接口规范来编写视频采集设备的驱动程序所涉及的流程主要有如下三个：

(1)设备的注册和打开。注册函数 `int video_register_device(struct video_device *vfd,int type,int nr)`,vfd 是设备结构体，type 是总线类型，nr 是附设备号，成功返回值为 0.打开设备函数 `open()`。

(2)基本的 I/O 控制。V4L2 非常依赖于 `ioctl()` 接口，

具体代码在内核/drivers/medie/Video/videodev.c 文件中。这些代码主要用来处理数据在用户和内核空间的移动，且为驱动分配单独的 `ioctl()` 调用。

(3)基本的帧输入输出。用户空间和驱动之间的视频数据帧格式传输方式在 V4L2 中实现了两种方法，一种是 `read()`和 `write()`系统调用，速度很慢；另一种是帧数据通过流向缓冲区从而到达应用程序，大部分驱动使用此种方法。

5.2 V4L2 下的图像采集

V4L2 支持音视频文字广播等内容，其中定义了很多数据结构，模块的源代码主要在 linux 源码中的 /linux/videodev.c 和/linux/videodev.h 两个文件中定义。本设计中用到的相关数据结构有：

①struct video_capability grab_cap 包含摄像头的基本信息，如设备名称 name(32),支持的最大最小分辨率 maxwidth、maxheight、minwidth、minheight,类型 type 等；

②struct video_picture grab_pic 包含设备采集图像的各种属性，如亮度 brightness、色调 hue、对比度 contrast、色度 whiteness 和深度 depth 等；

③truct video_mbuf grab_vm 利用 mmap 进行映射帧的信息，包括帧的大小 size、最多支持的帧数 frames、每帧相对基址的偏移量 offsets。

主要函数的功能如下：

```
int init_videoln(struct vdiIn *vd,char *device,int width,int height,int format,int grabmethod);
//初始化设备;
static int GetVideoPict (struct vdiIn *vd);
//获取图像属性信息;
static int SetVideoPict (struct vdiIn *vd);
//设置图像属性;
int convertframe(unsigned char *dst,unsigned char *src, int width,int height, int formatIn, int size);/
/把共享缓冲区中的数据放到一个变量中，通知系统已获得一帧;
put_image_jpeg(FILE *out,char *image,int width,in theight,int quality)//保存视频图像;
int close_v4l2 (struct vdiIn *vd);//关闭设备;
经传感器采集的视频帧的通道排序为 BGR,SN9C120 驱动把数据流转换成 RGB 排列，然后自己交叉编译 jpeg 库添加到项目中，把 RGB 视频数据格式转换为 jpeg 格式，实现图像压缩保存。视频采集基本流程如图 4 所示：
```



图 4 图像采集流程图

首先打开视频设备,摄像头在系统中的对应的设备文件为 `/dev/v4l/video0`,采用系统调用函数 `dev=open("/dev/v4l/video0",O_RDWR)`,`dev` 是设备打开后返回的文件描述符(打开错误返回-1),以后系统调用函数便可使用它来对设备文件进行操作。然后调用 `ioctl` 函数读 `video_capability` 中的信息。成功后可读取 `vd->capability` 的各分量,接着读 `video_picture` 中的信息,成功后可读取视频设备中图像的的属性,随后调用函数保存图像(jpeg 格式),最后关闭设备。

图像的获取有两种方式,一种是 `read()`直接读取,另一种是 `mmap()`内存映射。`read()`通过内核缓冲区来读取数据;而 `mmap()`通过把设备文件映射到内存中,绕过内核缓冲区,加速了 I/O 访问,此外,`mmap()`系统调用使进程之间通过映射同一文件实现内存共享。基于 `mmap()`以上的优点,本设计采用 `mmap()`方式进行视频的截取,具体操作如下:

(1)使用 `ioctl(grab_fd,VIDIOCGMBUF, &grab_vm)`函数获得摄像头存储缓冲区的帧信息,之后修改 `video_mmap` 中的设置,如重新设置图像帧的垂直及水平分辨率、彩色显示格式。可利用如下语句

```
grab_buf.width=640;
grab_buf.height=480;
grab_buf.pixelformat=V4L2_PIX_FMT_JPEG;
```

(2)接着把摄像头对应的设备文件映射到内存区,具体使用 `grab_data=(unsigned char*)mmap(0, grab_vm,size,PROT_READ|PROT_WRITE,MAP_SHARED,grab_fd,0)`操作。该函数成功时返回映像区内存的指针,失败时返回值为-1。

6 结束语

在宿主机 PC 上使用交叉编译器 `arm-linux-gcc` 编译 `SN9C120` 驱动,生成 `sn9c120.o` 模块,在超级终端用 `insmod` 命令把 `sn9c120.o` 加载到 EP9302

开发板上,将出现如下信息:

```
/mnt/share/sn9c120 # insmod sn9c120.o
Using sn9c120.o
usb.c: registered new driver sn9c120
sn9c120_core.c: USB SN9C120 camera found.
SONIX sn9c120 + MI0360
sn9c120_core.c: sn9c120 driver 00.57.06LE
registered
```

从上面的信息可以看出,摄像头 `sn9c120` 已注册成功,还可用 `lsmod` 来查看已加载的模块:

```
/mnt/share/sn9c120 # lsmod
Module              Size  Used by    Not
tainted
sn9c120              76008  0 (unused)
```

在 PC 机上利用 `SDL` 和 `spcview` 可查看采集到的视频图像,采集效果如图 5 所示。



图 5 视频采集效果

测试证明:所设计的驱动很好的实现了 Linux 下的 V4L2 视频采集。该视频采集系统接入网络时,可应用于监控等场合,具有广泛的市场和应用前景。

参考文献

- 1 李巍,邱跃红,董佳.基于 CMOS 图像传感器的视频采集系统.电子设计,2008,4(2).
- 2 邓兴波,高强,任德学.基于 TMS320DM642 的 CMOS 图像采集系统的设计.天津理工大学学报,2008:1-4.
- 3 魏永明.LINUX 设备驱动程序.中国电力出版社,2002.
- 4 <http://www.sonix.com.tw/sonix/asearch.do>.
- 5 <http://mi0360-image-sensor>.
- 6 http://lwn.net/Article/203924.The_Video4linux2_API_series.