K. V. Prashanth, K. Jail Singh, Prof. K.V.MuraliMohan / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 6, November- December 2012, pp.1075-1081

Design & Development Of ARM9 Based Live Video Monitoring System

[1]K. V. Prashanth, [2] K. Jail Singh, [3] Prof. K.V.MuraliMohan

[1]M. Tech Student, Holy Mary Institute of Technology & Science, Bogaram (V), Keesara (M), R. R Dt.- 501301.
[2]Assistant Professor, ECE, Holy Mary Institute of Technology & Science, Bogaram (V), Keesara (M), R. R Dt.- 501301
[3]Professor, HOD Dept of ECE, Holy Mary Institute of Technology & Science, Bogaram (V), Keesara (M), R. R Dt.- 501301

Abstract

Currently on the market most of the image acquisition system is based on DSP. This image acquisition system's high cost, great power consumption, and volume restriction is not suitable for some simple applications. So in this project we will be using S3C2440 along with embedded Linux. S3C2440 is being manufactured by Samsung and is a RISC microprocessor based on ARM920T.The maximum frequency can reach 533MHz.

In this project we will be using an OV9650 Color CMOS SXGA (1.3 Mega Pixel) From Omni-Vision camera connected to ARM9. One Linux device driver will be responsible for the camera. For writing to the LCD, Linux has Frame Buffer. Frame Buffer is the device for user process to write directly to screen in Embedded Linux. In Linux Frame Buffer is an interface for the display device. It describes some display device as a buffer and allows applications to access the graphics device through its defined interface without care about the specific hardware details.

We will also be using Video4Linux. Video4Linux or V4L is a video capture application programming interface for Linux. V4L is divided into two layers. The bottom is the audio and video equipment driver in the kernel, and the upper provides systern with some API.

Image data can be reconstructed through JPEG decoding. JPEG image files are divided into two parts: marking segment and compressing data. Marking segment include length, width, color information's, quantization table, Huffman table and other important information's of the image. Different informations store at different marking segment. JPEG image decoding process needs to extract the various of needed information in the marking segment, so as to facilitate decoding of compressed data. Finally the collected images transferred from JPEG format to RGB format is displayed on the LCD finally.

Keywords - ARM, Linux operating system, Image Acpuisition, image decoding

I. INTRODUCTION

Comparing with traditional CCD image sensor, CMOS image sensors using CMOS technology can integrate the pixel array and peripheral circuits (such as the image sensor core, single-clock, all the sequential logic, programmable functions and A / D converter) in a chip, with a small size, light weight, low power consumption, programming convenience, easy to control and so on.

Embedded Linux is a better embedded operation system, which has portable, strong network function and excellent GNU compile tools as well as free open source characters. S3C2410/S3C2440 is a 32 bits embedded processor that based on an ARM920T core with integrated MMU and abundant internal resources.

II. SYSTEM ARCHITECTURE

In this system, image informations were received through the OV9650 Color CMOS SXGA camera (1.3 Mega Pixel) From Omni-Vision camera connected to ARM9 , then transferred to S3C24440 chip to process, and sent to the LCD to display finally. The system's hardware architecture is as shown in Figure l.

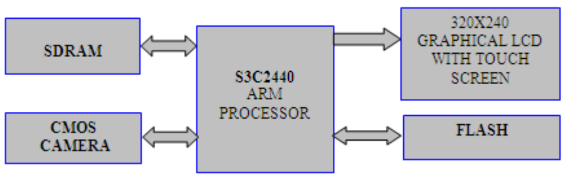

Figure 1. Hardware architecture

SAMSUNG's S3C2440 processor is the core of the system in this research. The S3C2440 is developed with RM920T core, O.13um CMOS standard cells and a memory complier. Its low-power, simple, elegant and fully static design is particularly suitable for cost and power-sensitive applications.

III. SOFTWARE & HARDWARE PLATFORM

The system uses SAMSUNG S3C2440 as a microprocessor, the camera equipment is OV9650. SAMSUNG S3C2440 uses 16/32 bit ARM920T RISC technology for the core. Its main Frequency is 533MHz. It provides a camera interface (camif) to support camera. There are two models for camif to transmit data with DMA controller: one is called Preview mode, which transform the image data sampling from the camera interface into the RGB format, and transfer it to the SDRAM under control of the DMA; the other is called code mode, which transmits the image data to the SDRAM in YCbCr4:2:0 or 4:2:2 format. Software platform is embedded Linux OS.

Image Data Acquisition & Display

In this project we will be using an OV9650 Color CMOS SXGA (1.3 Mega Pixel) From Omni-Vision camera and the output device was LCD.

A.CMOS Camera Driver

The OV9650 Camera chip is a low voltage CMOS image sensors that provides the full functionality of a single chip SXGA (1024x1024) camera and image processorin a small foot print package. The OV9650 provides full frame , sub-sampled or windowed 8-bit/10-bit images in a wide range of formats,controlled through the Serial Camera Control Bus(SCCB) Interface.

In Linux operating system, device driver provides access interfaces for underlying data structures, hardware device drivers and advanced data structures .Device driver shields the hardware details for the application. From Linux 2.6 kernel version, Linux operating system using kobject kernel object facility, which treat hardware device as a file.System calls and drivers are linked by file_operations. Every member of this structure corresponds to a system call. The S3C2440 does not provide sccb interface, so the driver has to include a sccb driver.

SCCB is short of serial camera control bus, which is customized by OminiVision Company . It is used for writing and reading the registers of the camera in order to control the output of the camera. SCCB data transmission is controlled by master device which send the data transmission start signal, the clock signals and stop signal. Usually the master device is microprocessor, and the sub device addressed access to it. Each device connected to the SCCB has a unique address (ID). At present, SCCB communication protocol supports only 100Kb/s or 400Kb/s transmission speed, and supports two kinds of address forms. Here, S3C2440 is the master device, OV9650 is the sub device. The important function: in SCCB driver are sccb_write(u8 IdAddr, u8 SubAddr, u8 data) which is used to write data into internal register of OV9650 and u8 sccb_read(u8 IdAddr, u8 SubAddr) which is used to read data from internal register of OV9650. \

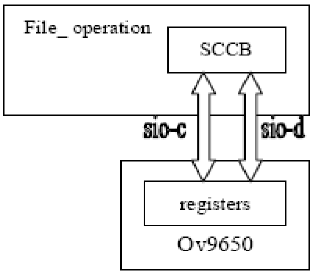

Figure 2 SCCB Protocol

Part of the code for sccb_read(u8 IdAddr, u8 SubAddr) is as follows

```
u8 sccb_read(u8 IdAddr, u8 SubAddr)
{
u8 data;
down(&bus_lock);
sccb_start();
sccb_write_byte(IdAddr);
sccb_write_byte(SubAddr);
sccb_stop();
sccb_start();
sccb_write_byte(IdAddr|0x01);
data = sccb_read_byte();
sccb_stop();
up(&bus_lock);
return data;}
```

Function sccb_read_byte () and sccb_write_byte () is respectively to complete 8 bit read and write according to SCCB timing logic.

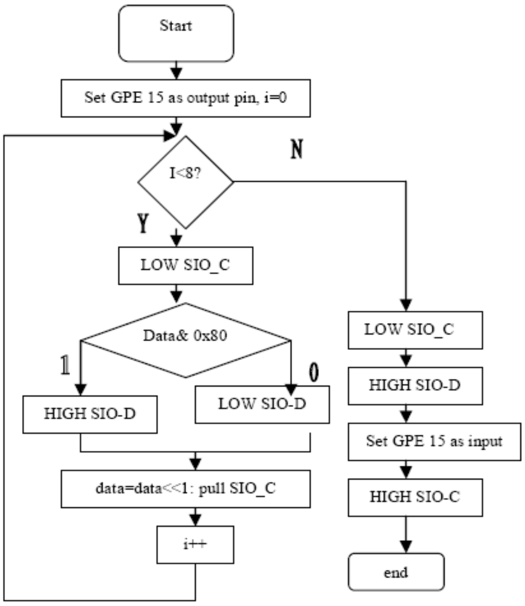

Figure 3 Algorithm flow chart to reading 8 bit

Registration and Initialization

Here register the camera as a miscdevice. All the miscdevice shara one major device number: 10. Using minor = MISC_DYNAMIC_MINOR to get a sub device number.All the miscdevices form a list. Calling the function misc_register() to add the device to the list, and registration is completed.

Initialization includes:
1) Using s3c2410_ gpio_cfgpin to set GPIO to camera mode. Part code is as follows:
s3c2410 _gpio_cfgpin(S3C2440_GPJ8,S3C2440_GPJ8 _CAMPCLK);
2) Using request_mem_ region() to ask for memory resources for camera. Using ioremapnocache () to map the memory resources to kernel space.
3) Initialize SCCB. Set SIC_C and SIC_D to high, which is the waiting status.
4) Initialize camera. Write the parameters (output format, AGC, sampling clock) into an array, and then using a cycle function to write parameters into the internal register of OV9650 through SCCB one by one.

Some important functions

When a application calls the camera, the system using sub device number to find the miscdevice, and then call its file_operation struct, and finally operate the API. There are some important function to provide: open(), release(),read().

1) Open():

a) Ask for continuous blank memory for P mode and C mode. To speed up data transfer, using a PingPong structure data buffer for data transfer, asking for 4 memory buffer for P mode and C mode respectively, and recycling this 4 buffer. When a frame data transmission is completed, that is, one buffer of the PingPong structure is full, interrupt handling program can capture the output data while the next frame data is transmission.
b) open the interrupt for P mode and C mode. Part of the code is as follows:
request_irq(IRQ_S3C2440_CAM_C,on_camif_irq _c, IRQF_DISABLED, "CAM_C", pdev);
request_irq(IRQ_S3C2440_CAM_P,on_camif_irq_ p, IRQF_DISABLED, "CAM_P", pdev);

2) Read():

*a)*Close interrupt. Part of the code is:
disable_irq(IRQ_S3C2440_CAM_C);disable_irq (IRQ_S3C2440_CAM_P);
b) Call start_capture() to start capture under corresponding mode.
c) Using copy_to_user() to transmit the data in driver buffer into user space.
d) Open interrupt when capturing is end.

3) Release():

a) Close camera interface.
b) Release interrupt.
c) Release buffer memory.
d) Call misc_deregister() to deregister the device.

B. LCD Driver[3]

FrameBuffer is the device for user process to write directly to screen in Embedded Linux. In Linux FrameBuffer is an interface for the display device. It describes some display device as a buffer and allows applications to access the graphics device through its defined interface without care about the specific hardware details.

In Linux FrameBuffer equipment can be seen as a complete subsystern, generally consisted of fbmern.c file and xxxfb.c file. On the one hand it provides application with some API(Application Programming Interface)which performed by fbmem.c file, such as read, write, ioctl. On the other hand it provides the hardware operation with some interfaces which should be setten to meet LCD controller hardware needs. In the S3C2440, LCD controller is integrated into the chip as a independent unit relatively, so it's a platform device for Linux.

In the kernel , /arch/arm/plat-s3c24xxJdevs.c, the file defined a platform for LCD-related equipment and resources. In /arch/arm/mach-s3c2410/include/mach/fb.h, the file defined s3c2410fb_mach_info structure to record parameters information of LCD, such as LCD screen size, Variable parameters of the screen, etc. In this paper, the model of LCD is the NEC256K color, 24Ox320/3.5 inch, TFf true color LCD screen.

In the configuration menu, select configure is as follows.

```
Device Drivers--->
    Graphics support--->
        Support for frame buffer devices
        S3C2410 /2440 LCD framebuffer support
        Framebuffer Console support
```

C. Vide04Linux programming[4]

Camera belongs to video equipment, followed the Vide04Linux standard(V4L). V4L is intended to provide a common programming interface for the many TV and capture cards now on the market, as well as parallel port and USB video cameras. It is programming interface that linux kernel provides for the user space. V4L is divided into two layers. The bottom is the audio and video equipment driver in the kernel, and the upper provides system with some API. Video41inux(short for V4L) is the API interface in Linux that is used to acquire video and audio data. Adding appropriate video acquirement device and corresponding driver to it, we can realize the video/image acquirement, AF/FM broadcast and switch channel function. This can be widely use in remote meeting, video phone and video surveillance system.

Every extemal device can be regard as special file in Linux, these files are called device file. We can read or write them as access ordinary files. Generally speaking, the camera device file that adopt V4L structure is "/dev/v4l/video0". V4L supports two methods to capture image: mmap and read. V4L defined some important data structure in"/usr /include/linux/videodev.h". In course of image acquirement, we can operate these data structure to get the image data.Video capture process flow chart shown in Figure 2.

1) The defined data structure of V4L program

In the image acquisition application, we can call some structure that the V4L defined. The following is the data structure defined in this article.

```
struct _ v41_ struct{
int fd;
struct video_capability capability;
struct video_buffer buffer;
struct video_window window;
struct video ""picture picture;
struct video _ mmap mmap;
struct video _ mbuf mbuf;
unsigned char *map;
int frame_current;
int frame _ using[VIDEO _ MAXFRAME];} ;
```

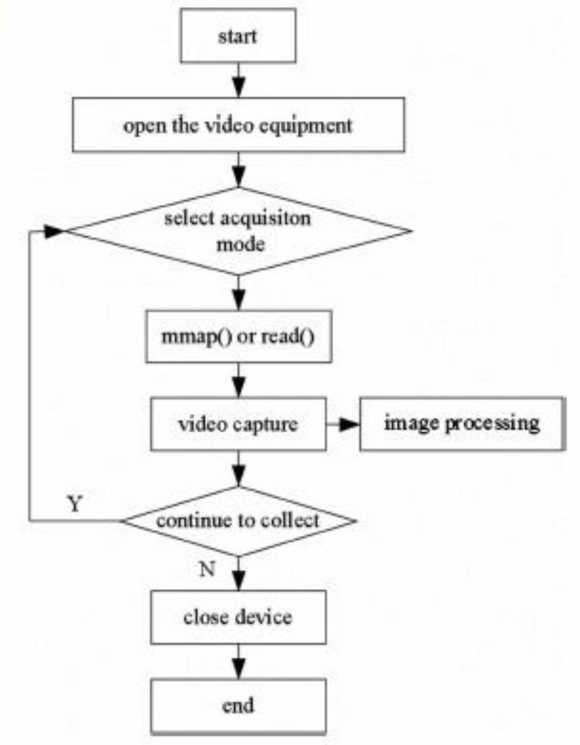

Figure 4. Video Capture flowchart

2) Program design of image acquisition

a) Open the video equipmen

Video equipment is used as device file in the Linux. The device name of USB camera in Linux is /dev/vidooO. The main program code is as follows:

```
if{(vd->fd=open( dev,O _ RDWR» <O)
{
perror("v41_ open:");
return -1; }
```

*vd is a structure pointer of the defined struct _ v41_ struct.

Through the open function to read the device file, it returns device descriptor when read successfully, else returns -1.

b) Read the video information

Reading video information is mainly to read the basic information and images property of equipment, which could be performed through the ioctlO function's control commands. Take reading the image properties for example, here's part of the program codes:

```
if (ioctl(vd->fd, VIDIOCGPICT, &(vd->picture» <
0) {
perror("v41□et ""picture:");
return -1; }
```

Of course, in user space program the informations could be changed according to the actual needs. The method is to assign a value to a parameter , then call the control command VIDIOCSPICT.

c) Video Capture

In this paper we got Video through mmapO function. In order to get the information of mapped buffer, video_mbuf must be initialized firstly. After got the map memory size, calling mmapO function, then the vd.map pointer points to the memory space that shall be collected image data.

```
vd->map = mmap(O, vd->mbuf.size,
PROT _ READIPROT _ WRITE,MAP
_SHARED,vd->fd,O)
```

In this way the real program code to obtain the image is as follows:

```
if (ioct1(vd->fd, VIDIOCMCAPTURE, &(vd-
>mmap» < O){
perror("v41_get _capability: ");
return -1; }
```

Control command VIDIOCSYNC was used to determine whether the interception of the frame completed. The image data could be saved as a file after the image acquisition finished. In order to improve image acquisition speed, it used double buffering, that is, a frame was dealing with collection the other.

vd->frame_using[frame] = TRUE;
vd->frame _current = frame;

d) Close device
The video equipment must be closed after Video Capture. Close(vd->fd);

IV. IMPLEMENTATION OF IMAGE DECODING

Through the above image acquisition, the following will introduce the image decoding.

A. The basic steps of JPEG decoding

As a static image data compression, JPEG is used very broadly. Image data can be reconstructed through JPEG decoding. The process includes pre-processing, entropy decoding, inverse quantization and Inverse Discrete Cosine Transform(IDCT). The smallest encode unit is MCU consisted by some 8x8 pixel blocks, the specific number of the blocks determine by sample mode of Y, Cr and Cb. Decoding is to carry out Circularly decoding to every MCU individually, until detect EOI mark.

1) Marking segment decoding

JPEG image files are divided into two parts: marking segment and compressing data. Marking segment include length, width, color informations, quantization table, Huffman table and other important informations of the image. Different informations store at different marking segment. JPEG image decoding precess needs to extract the various of needed information in the marking segment, so as to facilitate decoding of compressed data.

2) Entropy decoding

Entropy decoding refers to the process that restoring compressed image from the quantitative data block which consist of D.C coefficient and A.C coefficients. In the JPEG decompression algorithm process, because of the unique of the code word in Huffman coding, it is simplely to decode by lookup table[5] After Huffman decoding finished, DC coefficient could be gotten form direct component with DCPM(Differential Pulse Code Modulation) and AC coefficient could be gotten form alternating component with RLE(Run Length Encoding).

3) Inverse quantization

JPEG decoder makes use of the quantization table in the quantitative segment information to decode the quantitative values. JPEG file usually contains two quantization tables: one is luminance component of the quantization table, and another is chroma component. Inverse quantization is that coefficient matrix from Huffman decoding multiply by the corresponding quantization matrix. It will get 8x8 luminance arrays and chrominance arrays after inverse quantization and Z-shaped transform to a MCU.

4) Inverse Discrete Cosine Transform

When restore the original image information, it is necessary to do inverse discrete cosine transform with encoded and compressed information. The 8x8 array of the IDCT transform matrix as shown in the following formula:

$$f(x,y) © \frac{1}{4} \overset{7}{\underset{u©0}{;}} \overset{7}{\underset{v©0}{;}} (C(u)C(v)F(u,v)\cos\frac{(2x J 1)u\pi}{16}\cos\frac{(2y J 1)v\pi}{16}) \quad (1)$$

f(x,y) is the original image pixel value, F(u, v) is the size of the every frequency components. x, y © 0, 1 ... 7. When both ***u*** and ***v*** is 0, C(u) , C(v) ©1/√2. Other cases: C(u) , C(v) © 1.

In the process of image decoding, IDCT has the largest part of the calculation, so it is important to adopt a fast and efficient IDCT algorithm for image decoding. In this paper, taking advantage of decomposable properties of two-dimensional DCT/IDCT transform. Following is tranformation on Formula (1):

$$f(x,y) © \frac{1}{2} \overset{7}{\underset{u©0}{;}} (C(u)g(x,v)\cos\frac{(2x J 1)u\pi}{16}) \qquad (2)$$

$$g(x,v) © \frac{1}{2} \overset{7}{\underset{v©0}{;}} (C(v)F(u,v)\cos\frac{(2y J 1)v\pi}{16}) \qquad (3)$$

B. Color space conversion

After the above series of treatment, JPGE image decoding is basically over, but there is need to do some post-processing with decoded image. One of the post-processes step is to complete the color space conversion, put JPEG images from Y crCb color space conversion to RGB color space[6], the conversion formula is as follows:

$$\begin{cases} R © Y J 1.402Cr \\ G © Y ſ 0.344Cb ſ 0.714Cr \\ B © Y J 1.772Cb \end{cases} \qquad (4)$$

Conversion by the formula derive from R, G, B values may be beyond its domain. If it greater than 255, then truncated to 255; if less than 0, then truncated to O.

An MCU decoding has been completed now, as long as composing a full image with each MCU.

V. CONCLUSIONS

This project based on ARM9 processor and embedded Linux operating system, realize a CMOS camera image data acquisition, image decoding and image display. The figure 5 is the actual view of the project. The entire system is simple, small size, low cost. It can be applied to many areas after expanded, such as video phones, cameras, surveillance systems, etc.

Nowadays, more and more image capture system uses CMOS camera. With the continuous progress of CMOS technology, CMOS image sensors are more and more high resolution, low-cost, small size and easy programming, its application will become increasingly widespread. In this paper, the method of designing the CMOS camera driver based on S3C2440 developing board and Image acquirement in embedded linux is realized .

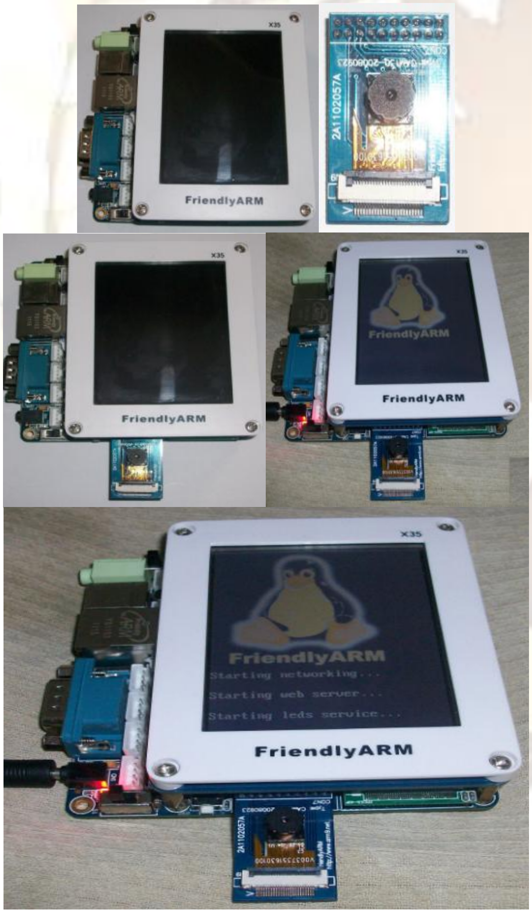

Figure 5. Picture of real products

REFERENCES

[1] Electronic Publication: weijing, Migration of gspca camera’s driven http://blog.chinaunix.netlu2174310/showart_1217552.html

[2] Gong Sheng-feng, ZHANG Xi-huang, " Implementation of image capturing and decompressing based on ARM Linux system", Computer Engineering and Design, 2009, 30 (6), pp 1397-1403.

[3] Embedded Linux C language application programming, Huaqingyuanjian Embedded Training Center, Posts & Telecom Press, Aug. 2007

[4] Dai Li, "Implementation of USB camera image capturing based on Vide04Linux", unpublished

[5] Cai Shijie, Digital Compression and Coding of Continuous-tone Still Images, Nanjing University Press, 1995

6] Alan Bovik, Hand of image and video processing, Publishing House of Electronices Industry, 2006

[7] El Gamal A.Eltoukhy H.CMOS image sensors[J].USA Circuits and Devices Magazine IEEE, 2005, 21(3) : 6- 20

[8] Yuan Weiqi, Tang Yonghua. Application of SCCB of OmniVision camera in DSP[J]. Chinese Journal of Scientific Instrument. 2006_27_6__1687-1688. (in Chinese)

[9] DU Chun-lei.ARM Architecture and Programming[M].Beijing _Tsinghua University Press,2003. (in Chinese)

[10] Zhang Da-bo. Principles, Design and Application of Embedded System[M].Beijing_Machinery Industry Press_2004. (in Chinese)

[11] Samsung. S3C2440A 32Bit CMOS Microcontroller User’s Manual[Z]. Samsung Electronics Corp, 2003.

[12] Jonathan C, Alessandro R, Greg K. Linux device drivers[M].3rd ed. Sebastopl, CA:O’ Reilly & Associates,2006:324-327.

[13] Daniel P, Cesati M. Understanding the Linux kernel [M]. 3rd ed.

This is **K.V.Prashanth** pursuing **M.tech Embedded Systems** from **HOLY-MARY INSTITUTE OF TECHNOLOGY & SCIENCE(College of Engineering).** Bogaram,Keesara,RR Dist. Affiliated to JNTUH. I did my M.Sc (Applied Electronics) with specialization Microwaves from GULBARGA UNIVERSITY and stood 1st rank in the acedemic year 2004.My interesting areas are Embedded systems, Microwaves, Antenna & Wave propagation, Real Time Operating Systems(RTOS) , Computer Networks.

Mr. K.Jail singh working as Assistant Professor in **HOLY-MARY INSTITUTE OF TECHNOLOGY & SCIENCE(College of Engineering).**
I did my **M.E in Telecommunication** from IISC , Bangalore. My interesting areas are micro processor and micro controller