Design of Real-time Image Acquisition and Display System Based on Embedded Linux

Xiaohan Guan, Wangyi Shi

College of Electronic Information Engineering, North China University of Technology, Beijing, 100041, China

Keywords: embedded system, linux, image acquisition, video4linux2, framebuffer, usb camera.

Abstract. For the need of real-time fast image processing problems in the field of pattern recognition, this paper proposes an image acquisition and display method based on embedded technology. In order to solve the problem efficiently, we present a programming technology of Video4Linux2 (V4L2), and the technology of V4L2 video capture application program interface is used to write the image acquisition program, as a result, the image acquisition of USB camera based on Ipassion IP2977 DSP control chip is realized. In addition, the Libjpeg library is used to unzip the images collected, the real-time image is displayed by framebuffer. Because the system uses new video programming interface and standard library functions of Linux, the modular structure to realize the required function has the very strong practical significance to the subsequent image application development based on Linux kernel.

1. Introduction

Traditional visual image system is based on the camera, image acquisition card and PC as a whole, this vision system can not meet the demand of real-time image processing in embedded environment. At present, the market has the general image system taking DSP as the data processing unit, but the cost is higher. With the development of domestic and international vision system, the visual image system based on embedded Linux is increasingly popular. Compared with the traditional system the embedded system not only has the advantages of small size, low cost, high stability, real time and so on, but also has practical value, so it has been widely used in intelligent transportation, computer vision, communication and other fields[1]. In terms of hardware, the USB interface camera which has high sampling rate, good versatility is favored in the embedded image acquisition applications. In terms of software, the embedded Linux is widely used in middle and low level embedded devices because of open source code, rich resources, powerful kernel function, stable performance. This paper uses the arm9-based S3C2440 processor hardware platform to achieve a better acquisition and display of digital image data in the Linux operating system.

2. Structure of the hardware and operating system

The system hardware structure is as shown in Fig.1, the core of hardware platform adopts embedded microprocessor S3C2440, outside is connected with a USB camera and LCD display device. S3C2440, the Samsung Corp production of low power consumption and high degree of integration is a 16/32 bit RISC embedded microprocessor. USB Host controller is directly connected with a video camera to get the image acquisition data which ultimately will be displayed on the LCD. In the S3C2440 and Linux operating system, the method of collecting and decoding a video image is different from traditional DSP system. In the past such as image compression, image synthesis operations involve a lot of calculation, so can only be handled by DSP, now since the ARM series processor has high performance to calculate and process, the coding and decoding of video image can be realized much better[2]. At the same time, ARM has a strong control function, the embedded Linux operating system also can be easily transplanted to ARM, so the embedded system that is composed of ARM and Linux has a great advantage.

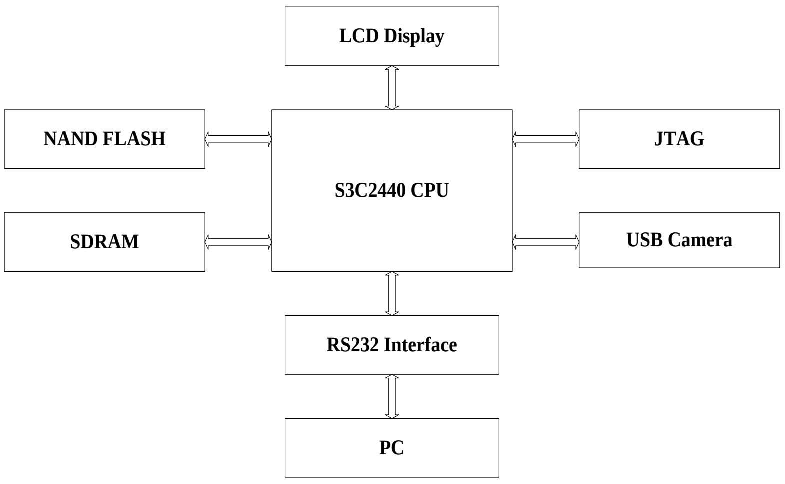

Fig.1: Structural diagram of the system hardware

3. Image acquisition based on the v4l2

V4L2 is Video For Linux Two, which provides a set of APIs for imaging equipment in the Linux kernel, it can realize image acquisition, AM / FM radio, image CODEC and channel switching function with the appropriate video acquisition device and the corresponding driver. At present, V4L2 is mainly applied to the video streaming system and embedded image system, which is widely used in teleconferencing, video telephone and video monitoring system.

The application of image acquisition of USB camera under the framework of V4L2 is mainly divided into two steps that include the successful driver of USB camera and image collection.

3.1 Driver of USB camera

The key to drive USB camera which is based on IP2977 chip is to ensure the successful loading of USB bus driver, V4L2 standard and IP2977 chip driver in the kernel[3]. Therefore, it is necessary to reconfigure and compile the kernel. Specific operation method is: run command “cp config_ok ./.config、make menuconfig” one after the other in the Linux-3.4.2 source file directory, enter the configuration options of system kernel, and then separately add the USB bus driver, V4L2 standard and the IP2972 driver.

i) The addition of USB bus driver

After running the command “make menuconfig”, select the following in the system configuration to support the USB bus driver.

```
Device Drivers  --->
  [*] USB support  --->
     {*}   Support for Host-side USB
     [*]    USB device filesystem (DEPRECATED)
     [*]    USB device class-devices (DEPRECATED)
     <*>     OHCI HCD support
     <*>   USB Mass Storage support
  [*] HID Devices  --->
     {*}   Generic HID support
     [*]    /dev/hidraw raw HID device support
  SCSI device support  --->
     <*> SCSI device support
     [*] legacy /proc/scsi/ support
     <*> SCSI disk support
     <*> SCSI tape support
```

ii) The addition of V4L2 and IP2977 device driver

Loading the driver of IP2977 control chip has two ways which are static loading and dynamic loading[4].Static loading refers the drivers are compiled into the kernel directly, which can be called directly after the boot without requiring any loading or unloading command. Dynamic loading refers to the use of Linux module features, through the command “insmod or modprobe” to mount the .ko kernel object file and load the module after the boot, and through the command “rmmod” to unload the module when not needed. As the module itself has not been compiled into the kernel, once the module is loaded by the kernel, it is used flexibly the same as the kernel of other parts, but the shortage is that each call is required for loading and unloading module through the command, the operation is trouble.

In view of hardware resources are adequate, for the convenience of using, this chip driver is loaded by static way, which is the IP2977 chip camera driver is compiled into the kernel, the specific configuration options are as follows:

```
-> Device Drivers
<*> Multimedia support  --->
   <*> Video For Linux
   [*] Video capture adapters (NEW)  --->
        [*] V4L USB devices (NEW)  --->
            <*> USB Video Class (UVC)
```

After completion of the above addition, exit the kernel configuration, save the configuration menu, successively run the command “make, make uImage” in the terminal, compile the kernel image file uImage, burn the generated uImage to development board, and reboot development board with the new kernel. At this point, you can see the camera driver has been successfully compiled into the kernel from the startup information, the information is as follows:

```
ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
s3c2410-ohci s3c2410-ohci: new USB bus registered, assigned bus number 1
usbcore: registered new interface driver uvcvideo
USB Video Class driver (1.1.1)
```

When the USB camera is inserted, the system can correctly identify the USB camera, and the output information in serial port is as follows:

```
usb 1-1: new full-speed USB device number 2 using s3c2410-ohci
uvcvideo: Found UVC 1.00 device <unnamed> (1b3b:2977)
input: UVC Camera (1b3b:2977) as /devices/platform/s3c2410-ohci/usb1/1-1/1-1:1.0/input/input0
```

3.2 The specific implementation of image acquisition

After development board recognizes camera successfully, images can be captured in the embedded Linux. For the image acquisition, there are two ways that are read()-based direct read mode and mmap()-based memory mapped mode[5]. The read()-based method is to read data from the kernel buffer to the user space memory, while in the method of mmap() process achieves shared memory each other by mapping the same file, which is that the image data buffer of camera and the image data area accessed by users share a memory region, thereby bypassing the kernel buffer district. The file is mapped into the user process address space, files can be accessed by the process the same as ordinary memory, without calling the operations of read()and write(). Because the process can directly read and write memory without copying any data, the use of shared memory approach accelerates I/O access, improves efficiency. In view of the above advantages, this paper adopts the way of mmap() to get image. Video capture process based on V4L2 is as shown in Fig.2.

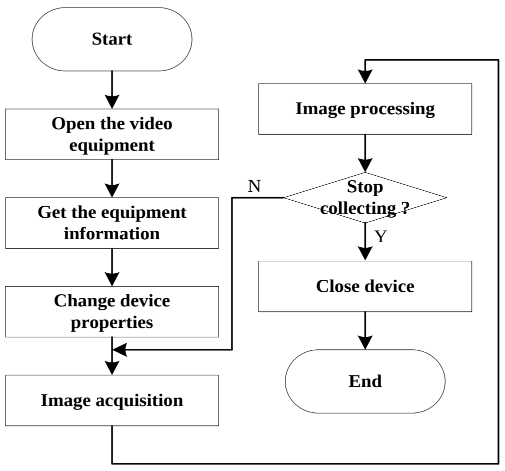

Fig.2: the video capture process

The concrete realization process of the image acquisition is as follows[6]:

i) Open the camera equipment

As the system equipment file is required to access the camera device in the Linux, we must create and set up the camera device file, before the file descriptor can be obtained by the following program code.

```
iFd = open(strDevName, O_RDWR);
```

ii) Get the camera equipment information

First of all, we can find the header file videodev2.h in the kernel source code, this header file defines all the data structure and function of the application that we will write. Of course, firstly we need to obtain the camera information, understand the camera performance through the v4l2_capability structure in the header file, and read out the unit according to the following program code.

```
struct v4l2_capability tV4l2Cap;
iError = ioctl(iFd, VIDIOC_QUERYCAP, &tV4l2Cap);
```

iii) Set the image collection format

After modifying the last step camera equipment information obtained, the following function can be called to set the property of camera.

```
iError = ioctl(iFd, VIDIOC_S_FMT, &tV4l2Fmt);
```

In program testing this device cannot be initialized to the RGB format that can be shown directly by LCD, so in the subsequent image processing, converting JPEG format to RGB format is required.

iv) Apply for cache space

Compared with the original V4L application programming interface, the biggest change is that equipment driving cache numbers can be customized in the new interface, so the performance of the program is improved significantly in the practical application. With the following function to apply for a few pieces of cache space at the start of image acquisition, the image data collected will be firstly stored in the cache space which has been applied for.

```
iError = ioctl(iFd, VIDIOC_REQBUFS, &tV4l2ReqBuffs);
```

v) Start the camera equipment

First, use the order parameter VIDIOC_QUERYBUF IOCTL in ioctl function to define each cache offset and size, then use mmap () function to map the device cache into user space, and the starting address of application in the memory mapping will be returned. After the memory mapping is completed, it also cannot start the image acquisition, need to put the memory determined into the queue by the command parameter VIDIOC_QBUF in ioctl function, and finally the following function is used to start the camera to capture the image data.

```
iError = ioctl(ptVideoDevice->iFd, VIDIOC_STREAMON, &iType);
```

vi) Close the camera equipment

After the acquisition is completed, to turn off the device, recover the system resources is required, so as to avoid memory leaks. The details are as follows:

```
iError = ioctl(ptVideoDevice->iFd, VIDIOC_STREAMOFF, &iType);
```

4. Image decompression based on libjpeg

Because the camera data collected based on the IP2977 control chip is JPEG coding format, the data cannot be shown directly in the Framebuffer equipment, extracting the RGB encoding format of the data is needed, before the data can be written to the Framebuffer device to display.

Libjpeg is a widely used JPEG compression/decompression library[7], it can read/write the image file in accordance with the standard of JPEG compression, through the Libjpeg library, every time the application can read one or a plurality of scanning lines from the image of JPEG compression, such as color space conversion, down sampling/up sampling, color quantization and so on are done by Libjpeg, the decompression process of Libjpeg is as shown in Fig.3.

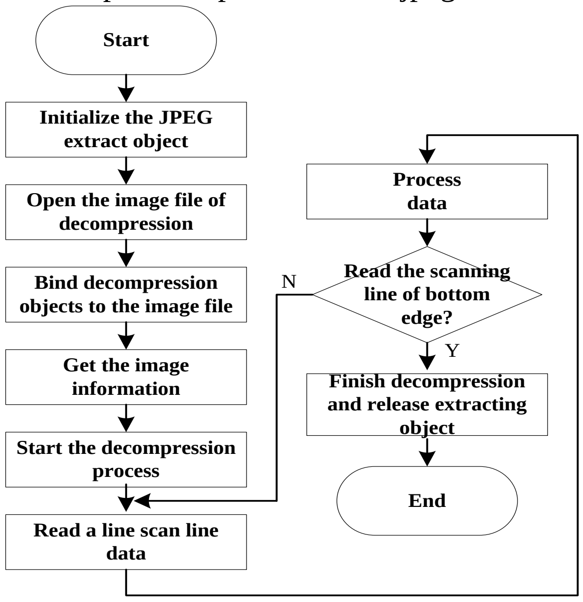
Start
Initialize the JPEG extract object
Open the image file of decompression
Bind decompression objects to the image file
Get the image information
Start the decompression process
Read a line scan line data
Process data
Read the scanning line of bottom edge?
N
Y
Finish decompression and release extracting object
End

Fig.3: Process of JPEG decompression

The image data extracted by libjpeg is stored in the form of a scan line that is a row of pixels in the image. For color images, each component is composed of three bytes, three bytes constitute a pixel on scanning line according to the sequence of R, G, B. For the scanning line the reading is in accordance with the top-down order, that is to say, the scanning line at the top of the image is read into storage space firstly by the function jpeg_read_scanline () ,followed by the second scanning line, and finally the scan line of the bottom edge of the image is read into memory space.

5. Image display based on the framebuffer

FrameBuffer, it serves as the basis for graphical facilities, is the basic function library of other senior graphics or graphical applications. This interface summarizes the display device to a frame buffer. The user can think of it as an image of display memory, but does not need to care about the physical memory location, form feed mechanism and other details, these are accomplished automatically by FrameBuffer device driver[8]. As long as it is mapped into the process address space, you can conduct read/write operation directly, and a write operation can be immediately reflected on the screen.

The device file is /dev/fb* for Framebuffer, where the value is 0~31. The /dev/fb0 is often used in Framebufer equipment. Because the embedded system has only a frame buffer device in general, the method of operation is that the Framebuffer equipment is mapped to memory through memory mapping mechanism, which can improve the efficiency. If the function mmap() is called successfully, the memory space mapped can be used to read/write in the program, and all read / write operations will be converted to I/O operation by the operating system kernel, the simple use of the program is as follows:

```
static struct fb_var_screeninfo g_tFBVar;
static struct fb_fix_screeninfo g_tFBFix;
g_fd = open(FB_DEVICE_NAME, O_RDWR);
ret = ioctl(g_fd, FBIOGET_VSCREENINFO, &g_tFBVar);
ret = ioctl(g_fd, FBIOGET_FSCREENINFO, &g_tFBFix);
```

It is necessary to obtain initial physical address of memory, resolution, color depth and other information from the variable g_tFBVar and g_tFBFix, before mapping memory size can be calculated according to these, the codes are as follows:

```
g_dwScreenSize = g_tFBVar.xres * g_tFBVar.yres * g_tFBVar.bits_per_pixel / 8;
g_pucFBMem = (unsigned char*)mmap(NULL, g_dwScreenSize, PROT_READ | PROT_WRITE, MAP_SHARED, g_fd, 0);
```

So we can operate the memory space of the size of g_dwScreenSize, the starting address of g_pucFBMem. When the image data collected based on V4L2 which is decompressed by Libjpeg database is mapped to this area of memory, it can directly display in the LCD.

6. System test

After sending the executable file of application program to a ARM board through the super terminal tool, entering the ARM board through the super terminal tool, executing the command "./video2lcd /dev/video0", continuous image data will be got, the test result is shown in Fig.4.

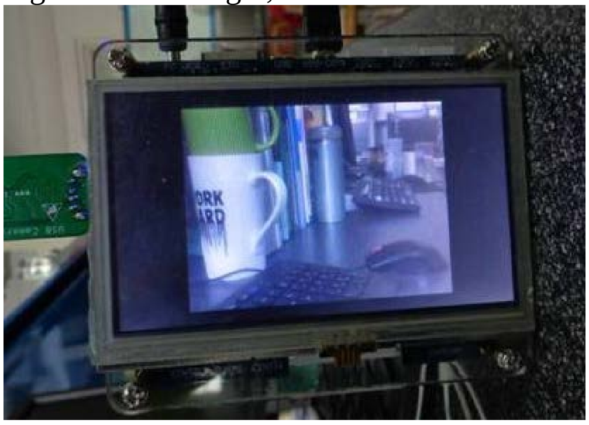
Fig.4: System test chart

7. Conclusion

On the S3C2440 embedded Linux platform this paper uses V4L2 programming interface to realize image acquisition of USB camera which is based on IP2972 DSP chip. Experiment shows that the image acquisition scheme used in this paper has good real time, in the practical application, the user can change the application according to the actual demand, so that real-time images can be displayed by LCD through the FrameBuffer equipment. In addition, the system adopts modular design in hardware and software, so it has good portability and function expansion, in practical application, users can also transfer the image data collected to the image processing algorithm for further processing with the help of OpenCV computer vision library.

8. References

[1] Weihua Ma. Embedded Systems Principles and Applications[M].Beijing University of Posts and Telecommunications Press, 2006.

[2] Weihu Zhou, Chenliang Shi, and Jiayang He, Embedded system design and development guide, 3th ed., vol. 2. Beijing: China Electric Power Press, 2009,pp.18-19.

[3] Lihua Song, Ke Gao, Implementation of USB camera driver based on embedded Linux, Computer Engineering, vol. 36, May. 2010,pp.282-284.

[4] Tianze Sun, Wenju Yuan and so on. Embedded Design and Linux Driver Develop Guidence[M]. Beijing Electronic Industry Press, 2005.

[5] Dirks B. Video for Linux Two API Specification Draft 0.24.http://v4l2spec.bytesex.org/spec/book1.htm, 2008.

[6] Yongqing Wang, Bo He, Image acquisition of USB camera based on ARM920T and Linux, Microcomputer Information, vol. 23, Jan. 2007, pp.17 6-177.

[7] Shengfeng Gong, Xihuang Zang, Implementation of image capturing and decompressing based on ARMLinux system, Computer Engineering and Design, 30 (6),pp.1397-1396, 2009.

[8] Linag liu, Wanchang Lai, and Ming Li, Design and implementation of image transmission system based on ARM9, Computer Engineering and Design,vol. 31, Apr.2010,pp.1477-1480.