The design of Image transmission system Based on STM32F429ZI

Deshun Fan, Jichun Zhao

(The Research Center of Beijing Engineering Technology for Rural Remote Information Services, China, 100097)

(Email: deshunfan@163.com)

Keywords: Image; Transmission; STM32F429ZI; uClinux

Abstract: The text gives a method of image transition based on STM32F429ZI. First the STM32F429 platform and develop software environment is introduced based on the Embedded uClinux. And then the realization method of achieving the image collection, compression, control, and transmission is introduced. The system can be used where people can't be easily gone, and could be widely used in environment monitoring.

1. Introduction

With the fast development of multimedia and Internet, it is possible that we realize embedded wireless image monitoring system based on wireless network. In order to realize the application, real time image capture is an important process. The article realizes image collection and compression based on embedded uClinux platform. We use the USB interface camera which is produced by Omni Vision Company and supported by uClinux operation system. The image can be saved for file form after compression, and then it is transmitted by image.

Wireless network has the advantages of always online, logging on promptly, charging by quantity, switching freely and so on. Therefore, it ensures the stability and greatly reduces the investment and running expenditure.

image (Multimedia Messaging Service) depends on WAP (Wireless Application protocol), and can transmit image paragraph, photo, sound and characters, realized phone terminal to terminal transmission, phone terminal to networking or networking to phone terminal. WAP Forum and 3GPP formulate image industry standard. Image runs on WAP, and not confined in the transmission pattern, not only supporting CSD (Circuit Switched Data) but also wireless network (General Packet Radio Service) pattern.

2. Hardware and software platform

The system uses ST’s STM32F429 32-bit RISC microprocessor which is designed to provide hand-held devices and general applications with cost-effective, low-power, and high-performance micro-controller solution in small die size. To reduce total system cost, the STM32F429includes the following components separate2MB FLASH and 256KB internal RAM, LCD Controller (TFT), NAND Flash Boot Loader, System Manager (chip select logic and SDRAM Controller), 4-ch UART, 2-ch DMA, 14-ch Timers, 114 I/O Ports, RTC, 24-ch 12-bit ADC and Touch Screen Interface, 3 IIC-BUS Interface, 2 IIS-BUS Interface, USB Host, USB Device, SD Host & Multi-Media Card Interface, 6-ch SPI and PLL for clock generation. The hardware development figure is shown as Fig.1.

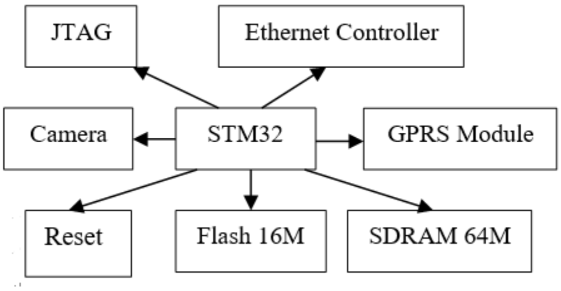

Fig.1 System hardware diagram

The STM32F429ZI offers outstanding features with its CPU core, a ARM 32-bit Cortex-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator) allowing 0-wait state execution from Flash memory, frequency up to 180MHz, MPU, 225 DMIPS/1.25DMIPS/MHz (Dhrystone 2.1), and DSP instructions.

The camera is connected with STM32F429ZI by USB interface which the data of image can be put in MCU buffer. The software platform system is uClinux operation system which the kernel is the heart. The kernel sits at the lowest level and manages the hardware. If a running program wants to interact with the hardware, for example to print a document, it doesn't communicate directly with the printer, but with the kernel, which then manages the printer. In addition to managing peripheral devices like the printer, keyboard, and mouse, the kernel controls the hard drive, memory usage, concurrent program execution, networking, and system security.

3. Making the development environment

3.1 customizing uClinux kernel

The first, we make a kernel of uClinux for using USB to collect image, which supports USB bus and ov511 camera.Ov511 is a special numeral IC chip connected with USB interface which has numeral camera interface, DRAM interface, real time image compression engine, USB interface,I2C interface, FIFO function and so on. The development hardware deal with image data through USB interface directly.OV511 chip’s driver must be correct that can support host USB controller UHCI or OHCI, also Video4Linux. When customize kernel, check

[*] Video for Linux,
[*] Support for USB,
[*] UHCI,
[*] OHCI,
[*] USB OV511 Camera Support.
Then compile uClinux kernel, all is prepared for image capture.

3.2 Cross development environment

There is no enough resource to run development and compile tools because of embedded system localization, so embedded system development uses cross development form which is built guest machine. The target board is embedded STM32F429ZI. We use arm-linux-gcc to compile the code which can run in STM32F429ZI processor and the bin code can be debugged and planted.

4. The realization of image capture

4.1 Camera driver and introduction of V4L

The device driver program can be regard as a interface under the uClinux, the task of driver program is to realize the transfer of system function. The camera chip OV511 is supported by uClinux operation system. When the operation runs, it will recognize USB device and take the device module into uClinux kernel automatable. We can get the information and data by system transfer directly, which is convenient for development. Driver program need to provide the basic

I/O Operation with the function open, read, write and close.Video4Linux is the driver about image device kernel which provides a series of application interface. We design the program based on Video4Linux API function.

4.2 Image capture

The image capture flow is shown as Fig.2. The first, CPU should open the image device which is a camera. After the processes get the image information, it will process the image data. The program will judge whether it will shop capture image. If the result has a symbol which is going on capture image, then it will continue get image information, otherwise it will close the image device.

The image capture date program needs all kinds of image data parameter. An important data structure image device of Video4Linux provides image device information. The structure is as Fig.2.

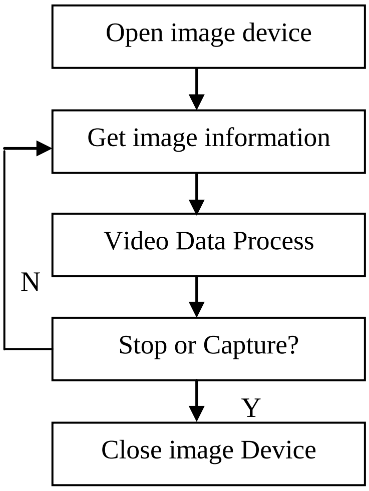

Fig.2 The image capture flow

```
typedef struct{
int Ima;
int use_Nad;
int wid, heig;
int frame_rate;
int frame_size;
struct image_size size;
struct image_buf    buf;
struct image_window wind;
struct image_pic pic;
struct image_ Nad Nad;
struct image_mbu mbu;
unsigned char *frame_buf;
} image_device;
```

The image_capability, image_picture, image_buffer date structure contain the information of device name, the maximum and minimum resolution, signal source, brightness, color, contrast, depth, frame size and so on. The function image_mmap and vdieo_mbuf are used for memory map.

The function ioctl (grab_fd, VIDIOCGMBUF, &memoryBuffer)   is used for initialization.

The mapping buffer and filing code is as follows:

```
grab_data = mmap (0, memoryBuffer.size, PROT_READ|PROT_WRITE, MAP_SHARED,
grab_fd, 0);
    mmaps.frame=0;
    mmaps.format=VIDEO_PALETTE_RGB24;
    mmaps.width=width;
    mmaps.height=height;
```

The function ioctl (grab_fd, VIDIOCMCAPTURE, &mmaps) is used for capture image data.

5. Image data compress

If the image data is not compressed, then it is too large to transmit in network. We compress image data in form of JPEG which is a standard form based on Disperse Cosine Transform. The main code is as follows. When the compression is finished, we turn off the camera. The main code is as follows.

```
jpeg_create_compress (&cinfo);
 jpeg_stdio_dest (&cinfo, fp);
 cinfo.image_width = width;
 cinfo.image_height = height;
 cinfo.input_components = depth;
 cinfo.in_color_space = JCS_RGB;
jpeg_set_defaults (&cinfo);
jpeg_set_quality (&cinfo, FIXEDQUALITY, TRUE);
  jpeg_start_compress (&cinfo, TRUE);
line=grab_data;
for (i = 1; i <= height; i++) {
      jpeg_write_scanlines (&cinfo, &line, 1);
      line=grab_data + (linesize * i);
      }
jpeg_finish_compress (& (cinfo));
jpeg_destroy_compress (& (cinfo));
fclose (fp);
```

6. Image transmission based on wireless network

6.1 Image package pattern

Image data will be compressed in order to reduce transmission data quantity, overcome the characteristic such as narrow bandwidth, long timeout, and low stability, when image is sending. The basic compression coding mechanism is defined by WAP-209-image Encapsulation. The transmission content is image PDU, when transmission and receive image communication. Image PDU is regard as message body to transmit, which is made of message head and message body. Image head describes the specifically message of PDU, which described by SMIL language. Image data packet code is as follows:

```
X_image_ Transaction_ID: 10887
X_image_ image_Version: 1.2
Subject: test
X_Image_ Message_ Class: Personal
X Image Delivery Report: Yes
Content_Type: application/vnd.wap.multipart.related
Type= application/smil
```

6.2 Transmission process

SEND.image is a packed image file, which can be packed with image development tools including image header and image body. Image.lib integrates WAP, PPP, and TCP/IP. The main transmission code is as follows which we can transmit several pattern image.

```
#define image_TEXT    0xAB
#define image_JPG     0x3E
#define image_GIF     0x27
#define image_PNG     0xDD
open ("image.png", O_RDWR, 0777);
```

```
    len=read (fd, buff, 23000);
    i=send_image ((u_char *)"13512344321", (u_char *)"test", len, buff);
```

Other function code is as follow.

```
    new_image ();
    i=image_add_content (image_PNG, (u_char*) "image.png", Len, buff);
    image_add_content (image_TEXT, (u_char *)"siasun.txt", 9, "agnh12341");
i=image_send ("test subject");
```

7. Conclusion

Embedded system is widely used in different field because of its high real-time and reliability. Embedded wireless image monitoring system can be used in monitoring field. The image effect after compression is stable. It’s a fact that embedded Linux system has many advantages in image process. The system can be widely used in monitoring field.

References

[1] Wireless Application Protocol Wireless Session Protocol Specification. WAP-203-WSP [EB/OL].WAP (Wireless Application Protocol) Forum Ltd, 2000, 5. 1999,1347-1384.

[2] Aljadhai, A.; Znati, T.F.; “Predictive mobility support for QoS provisioning in mobile wireless environments”, Selected Areas in Communications, IEEE Journal on, Volume: 19 Issue: 10, Oct. 2001, pp. 1915-1930.

[3] Wireless Application Protocol Wireless Session Protocol Specification. WAP-203-WSP[EB/OL].WAP(Wireless Application Protocol) Forum Ltd, 2000,5

[4] Joachims T., Freitag D., Mitchell T. Web Watcher: A Tour Guide for the World Wide Web[C]. Proceedings of the 15th International Joint Conference on Artificial Intelligence. Nagoya, Japan, 1997: 770-775.

[5] Martin-Bautista M J., Martinez-Folgoso S., Vila M A., A New Approach for Representing and Querying Textual Attributes in Databases [J]. INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, 2015, 30(9): 1021-1045.

[6] Joe deBlaquiere. Building Devices with Embedded Linux. Red Hat [M], 2000