

# Face Detection Based on Embedded Linux

**Mr. S. M. Gramopadhye , Mrs. V. V. Bhiungade**

*ATS SBGI Miraj*

*Email :- gramopadhyesm@sbgimiraj.org*

*Email :bhiungadevv@sbgimiraj.org*

## **ABSTRACT**

*Continuous improvements in Linux embedded software make it secure, stable and reliable. Embedded Linux plays an important role in embedded field. Linux is widely used in embedded system due to its small kernel size, stable performance, versatility and low reward. This paper describes the reconfiguration of the Linux operating system and the launch of the CMOS device driver based on the mini2440 development board. Embedded Linux re-integration includes integrated environment development, bootloader integration, Linux kernel deployment and root file system construction. SCCB bus, camera interface and V4L2 architecture are included in the CMOS camera device driver development of the 2440 sub-board.*

**Keywords** - S3C2440 processor; bootloader; Linux 2.6.32; CMOS camera driver; V4L2; SCCB

## **I. INTRODUCTION**

Embedded Linux is built in accordance with the requirements of the embedded operating system. This is a small program with very little kernel, usually only a few hundred KB, and the required storage space is also very small even to add other required modules and applications. It is therefore very suitable for installation in another embedded system and at the same time has a multi-tasking system feature and multiple processes [2].

The camera mainly uses the CCD (Charge-Coupled Device) image sensors and the CMOS (Complementary Metal Oxide Semiconductor) image. The signal captured by the CCD requires additional rotation to convert analog light data into readable digital signal. For the CMOS sensor, each pixel has neighboring transistors that perform analog-to-digital conversion of space. The advantages of a CMOS sensor over CCD are a lower complexity in the sensor that leads to faster image capture and reduced power consumption. The CMOS sensor is used by multi-megapixel cameras because of its fast learning.

The system uses a CMOS camera to capture video under an embedded Linux system platform based on the S3C2440 micro control chip. Send data to the development board and display the captured video using the LCD display.

## **II. CERTIFICATION OF CERTIFICATES**

System context for high performance 16/32 RISC (Reduced Computer Setup Instructions) S3C2440 embedded microprocessor based on ARM920T kernel. The camera used in this program is the Omni Vision OV9650 CMOS camera. A diagram of the image capture system block is shown in Fig.1.

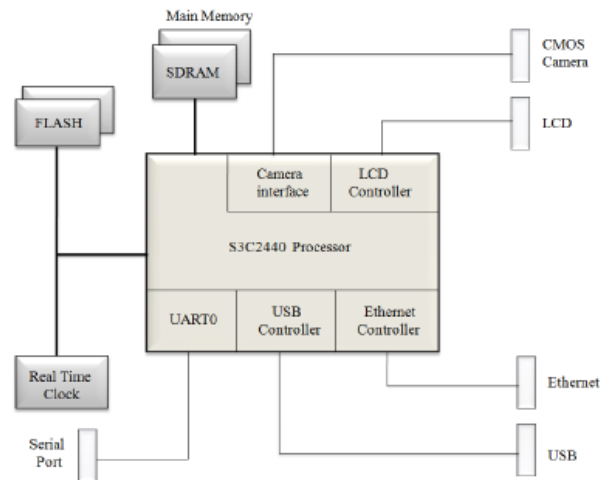


Fig. 1 :- Image Acquisition system

The S3C2440 provides platform design flexibility with support of 300,400 and 533MHz total speeds. 64MB SDRAM (Randomized Switch-in-Switched Accessal Memory) is used to execute system and user system. RS-232 is used for upgrading Linux hosting machine and Ethernet is used for system network transmission. The S3C2440 is mainly targeted at hand-held machines as well as the use of high-cost operation and low power losses. Its low power, simple design, beautiful and perfectly upright are very suitable for sensitive and cost-effective applications. Adopt a new bus design known as Advanced Micro-controller Bus Architecture (AMBA) [6]. It is currently commonly used in microcontroller multimedia terminals. This processor has USB, SDRAM and LCD controller, a virtual camera connector. Advantages of S3C2440 processor high performance and low power consumption. The OV9650 is a low-voltage CMOS camera sensor with a power of 1.3 billion for photography. Provides full-size, low-resolution or 8-bit / 10-bit images in a variety of formats, controlled by Serial Camera Control Bus (SCCB) [3]. The OV9650 is connected to the S3C2440 via a visual camera connector. The S3C2440 processor has no support for SCCB. The SIO\_C and SIO\_D PINs of OV9650 are connected to the IIC\_SCL and the IIC\_SDA PINs of the S3C2440 processor clock and data signal. The connection between the processor and the image sensor is shown in Fig.2.

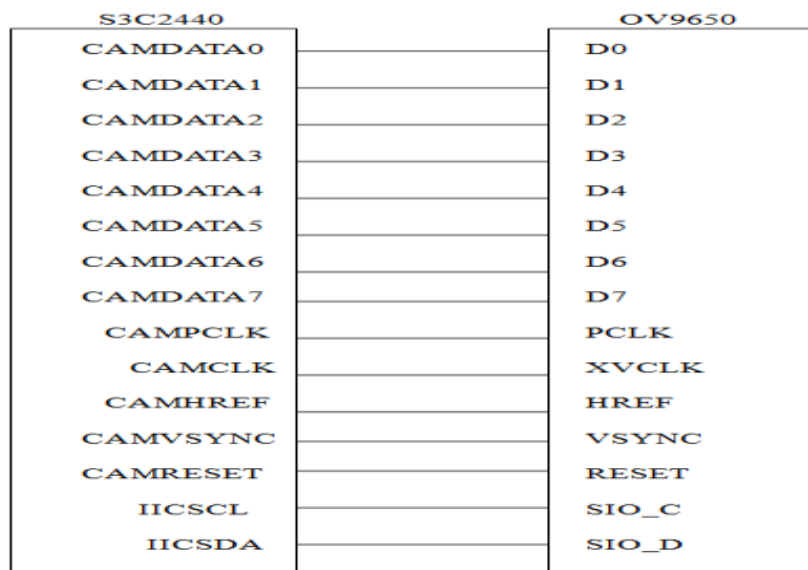


Fig.2. The interconnection of S3C2440 and OV9650

### III. LINUX PROMOTION

A detailed description of the various steps involved in Linux re-investment is presented in the following sections.

#### A. Bootloader Transmission: -

Bootloader is a small program that works before the kernel of the operating system. Its main role is to launch hardware, create a memory space map and bring the system hardware and software environment to a perfect level. The bootloader used in this paper is Universal bootloader whose name is U boot. The U-boot is customized to provide both a rich feature set and a small existing binary printer.

U-boot 1.3.2 was downloaded to the RAM of the target board via a network port connection, and the bootloader was labeled FLASH on the hard drive storage device for the target machine. U-boot is widely used to load and activate kernel image, but also allows you to change the kernel image and root file system stored in FLASH. B. Linux kernel deployment: -

Linux kernel 2.6.32.2 is used in this paper. The kernel integration has the following steps,

- Download the Linux source package 2.6.32.2 tar.bz2 from the official Linux website;
- Uninstall the source code for the Linux kernel;



- Create cross-compiler space in a Linux host: The opposite combination is: the use of certain types of compact machines to integrate the source system and generate object code that works on another machine. Download and install arm-linux-gcc compiler, toolchain;
- Customize the Kernel Makefile: Select architecture as ARM and provide an arm-linux-gcc path from the system;
- Configure the kernel: create a menuconfig, with an easy-to-navigate menu, user interface, allowing the user to select Linux kernel features to be integrated [2].
- Kernel integration: After configuration with the kernel creation command, Merge the kernel using the following command,

#make Image - Create a Uboot kernel image. After merging, three kernel image files "Image", "Image" and "Image" will be generated in arch / arm / boot directory. Image is the normal size of an image file, but zImage is a compressed kernel image file. The Uboot image is made up of a mage and a U-bootloader title file. Image file for Linux kernel zImage, which must be reinstalled on the target board. The generated zImage is shown in Fig.3.

```
swap1@ubuntu: ~/project/linux-2.6.32.2
File Edit View Search Terminal Help
CC lib/prio heap.o
CC lib/prio tree.o
CC lib/proportions.o
CC lib/radix-tree.o
CC lib/ratelimit.o
CC lib/rbtree.o
CC lib/reciprocal_div.o
CC lib/rwsem-spinlock.o
CC lib/sha1.o
CC lib/show_mem.o
CC lib/string.o
CC lib/vsprintf.o
AR lib/lib.a
LD vmlinux.o
MODPOST vmlinux.o
GEN .version
CHK include/linux/compile.h
UPD include/linux/compile.h
CC init/version.o
LD init/built-in.o
LD .tmp_vmlinux1
KSYM .tmp_kallsyms1.5
AS .tmp_kallsyms1.o
LD .tmp_vmlinux2
KSYM .tmp_kallsyms2.5
AS .tmp_kallsyms2.o
LD vmlinux
SYSMAP System.map
SYSMAP .tmp_System.map
OBJCOPY arch/arm/boot/Image
Kernel: arch/arm/boot/Image is ready
AS arch/arm/boot/compressed/head.o
GZIP arch/arm/boot/compressed/piggy.gz
AS arch/arm/boot/compressed/piggy.o
CC arch/arm/boot/compressed/misc.o
LD arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
swap1@ubuntu:~/project/linux-2.6.32.2$
```

Fig.3. The generated z-Image after compilation

The generated zImage can be transferred to the board via the TFTP (Trivial File Transfer Protocol) protocol. The tftboot command enables U-boot to download the kernel image from a TFTP server. The kernel image is downloaded from RAM and embedded in a respected memory address of the target system. Finally, the U-boot command "bootm" command is used to start operating system images. The first argument "bootm" is the memory address (RAM, ROM or flash memory) in which the image is stored, followed by an optional application based on Application. The natural model of cross-compiler is shown in Fig.4.

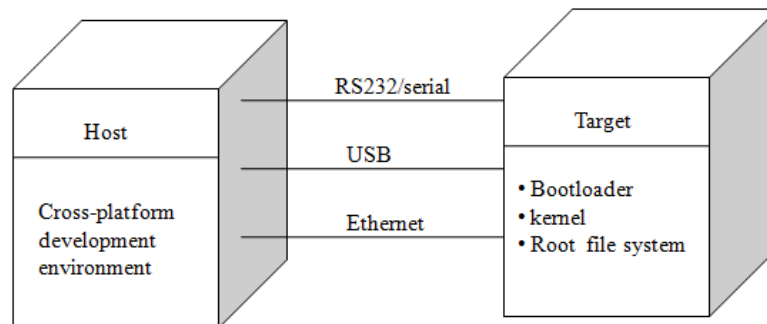


Fig.4. The cross-compiler environment model

#### IV. SOFTWARE DESIGN SYSTEM

The S3C2440 processor communicates with the OV9650 camera via a camera interface. The camera interface has two scalars. Another preview image that produces a small image like PIP (Picture in Picture) and a codec scale is provided to produce a useful codec image as a type of aircraft YCbCr 4: 2: 0 or 4: 2: 2 followed by two DMA modes. Preview mode is used to store RGB image data in PIP memory. The codec method stores image data of YCbCr 4: 2: 0 or 4: 2: 2 in codec memory as MPEG-4 [6].

The design of the CMOS camera driver is similar to the character driver. The camera driver basically has two parts. The first is the implementation of the OV9650 registers and the second is the main image capture system using the S3C2440 processor camera interface.

##### A. SCCB Driver Design: -

The main device is the S3C2440 processor and the slave device is the OV9650 CMOS camera. If the bus does nothing, the manager will call the SIO\_D signal up. Transfers will start when SIO\_C is high and SIO\_D is low. The supervisor will begin the literacy work, only after the initial situation arises. Completion of the writing task occurs only when the king asserts a state of suspension. Similarly the expert asserts the condition of suspension in order to complete the learning task. The transmission will stop only if the main one will hold the SIO\_D up and keep the SIO\_C signal up.

There are three types of transmission cycles depending on the literacy phase.

(1). Three-phase writing cycle- A three-phase writing cycle is a complete writing cycle. In this case the master will write one byte of data to a particular slave. The master will identify a slave by his address and the address of a particular registrar by his minimum address. The 8 bit writing data used to write over the content of a particular address is key. The ninth of the three categories does not matter.

(2). Two-phase writing cycle- To read data from a certain slave the master must know his minor address. The use of a two-phase writing cycle to identify the sub-address of a particular slave. The master reads the data from a certain slave to the transfer cycle of the two-phase reading.



(3). Two-phase reading cycle - The sub-address of a particular slave cannot determine the transmission cycle of two-phase reading. There is therefore a need for a three- or two-phase writing cycle to identify a small address. Then the master can read the data for the two-phase transmission cycle reading. The two-phase reading transfer cycle consists of 8-bit and 9-bit reading data, NA.

To record 8 bit data on OV9650 camera's internal registers, initially the SIO\_C signal must be low. MSB 8 bit data test checked high or low status. The last bit bit effect can be inserted into the SIO\_D signal. The data is then slowly moved away, circling 8 times until the process is complete.

The 8 bit data can be read by the S3C2440 processor from the internal registers of the OV9650 CMOS camera. To read data, set PIN SIO\_D as an input. Drag SIO\_C down and set the data content as zero. Set the SIO\_C signal up and left move the data by 1 bit. Then ORed data with SIO\_D signal status. Reset SIO\_C signal down around 8 times until the process is complete. The 8-bit data reading flow is shown in Fig.5.

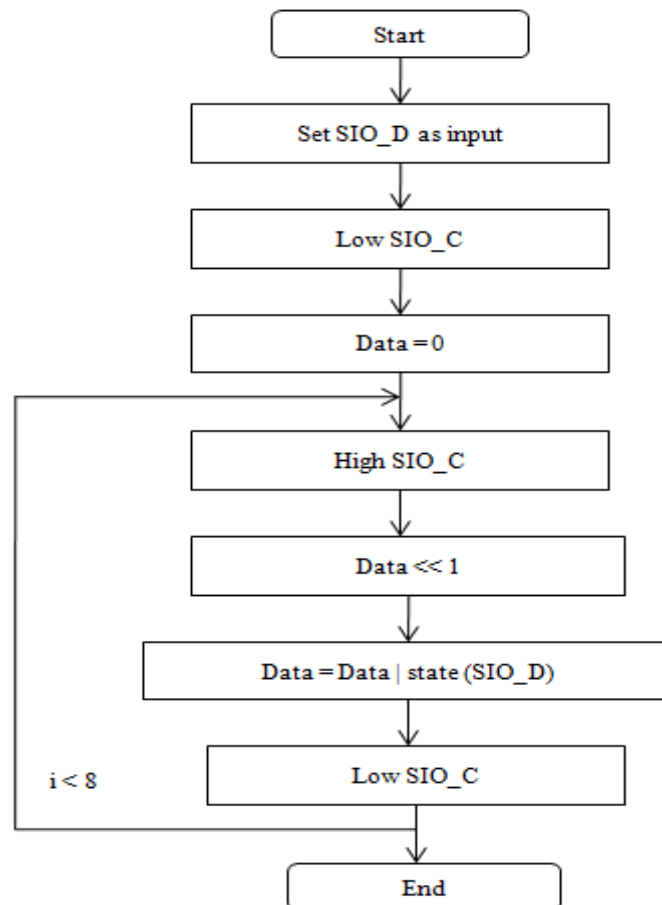


Fig.5. The flow of reading 8 bit data

#### B. V4L2 Driver Use: -

The camera driver is made under Linux in the form of V4L2 (Video4Linux2). The image can be taken with the V4L2 drive interface. Video4Linux is a Linux video recording program. Using the V4L2 format,



the video device is packaged as a file that can be read and transcribed directly. The camera function of the camera device is as follows,

- Turn on the video device;
  - Read device information
  - Provide device frame storage;
  - Download image using V4L2 interface.
  - Turn off the video device
- When the program is running the camera, the app uses a small device number to locate a specific device. Camera launch function includes I2C device registration, device driver registration, V4L device registration, camera visual memory launch, camera clock launcher and sccb\_init. A member of the camera interaction file functions in the camif device framework to use basic functions such as camif\_open (), camif\_read (), camif\_ioctl2 (), camif\_poll (), and camif\_release () [5]. Some of the functions are described as follows,

#### 1) camif\_open ()

a. To change the appropriate registers of the S3C2440 internal camera module. Provide file handle memory and check if the device is an open master or an open slave.

b. Turn on P-mode and M

c. Two interfering management systems C-mode and P-mode use a pdev value -> cmdcode to achieve a consistent response, such as switching between P-mode and C-mode, installing windows, zooming in and updating the format, by changing the appropriate registers of the internal camera module of S3C2440.

#### 2) camif\_read ()

a. Turn off P mode interruption and interrupt C mode by driving consecutively to disable interruption tasks.

b. Call start\_capture fuction to start taking pictures. c. Use the copy\_to\_user function to transfer data to user space from the driver's bath.

d. Call the stop\_capture function to unlock the bug and return from the disruptive capture system.

#### 3) camif\_release ()

a. Turn off camera interaction



- b. Stop shooting with a camif clock
- c. Exit mode P and C mode are disruptive
- d. free private bath memory.

The `camif_cleanup ()` function is used for removing the camera interface module from the kernel. This task includes the transfer of the hard disk memory, the task of cleaning the `sccb` and removing the device registration.

With the V4L2 framework the system can easily call the system interface interface and control the camera, while the system calls the CMOS video device.

C. Driver Launch: -

There are two ways to load the camera driver. One direct route where the driver is connected directly to the kernel and another alternative module where the driver is loaded with the kernel as a module. Driver combines three main programs: `sccb.c`, `s3c2440camif.c`, and `s3c2440_ov9650.c`. `Sccb.c` is used for data transfer. `S3c2440_ov9650.c` is used for machine activation and communication between user and kernel. Camera activation can be done with `module_init ()` function. Includes testing and selection of image capture format. The `module_exit` function is used to relieve distractions and memory. The driver is removed at the end of the application.

Programs are compiled using Makefile. Makefile contains a list of programs needed for integration. And Makefile includes a kernel method that is essential for driver integration. The command "make" is used to create a kernel object file. The built-in file is transferred to the development board. The object file is then uploaded to the development board which completes the CMOS camera driver installation process.

## V. CONCLUSION

The image capture system is enhanced using the mini2440 enhancement board. The reboot process includes steps such as bootloader integration followed by a reduction and kernel integration and the root file system construction has begun.

## REFERENCES

- [1] Geng Qingtian, Sun Zhanchen, Zhao Hongwei, Gu Jianhao. "The U-boot Transplantation Based on S3C2440". *International Conference on Mechatronic Science*, 2011, pp. 2168-2171.
- [2] Sun Yanpeng, Peng Peng, Zhang Yuan. "Linux Transplantation Based on The Processor S3C2440". *Electronic Measurement And Instruments*, 2009, 2: 306-309.
- [3] OmniVision, *OV9650 datasheet version 1.3*, September 24, 2004.





[4] Jia Liu, Wusheng Chou. "Design and Implementation of Embedded Image Capture Device for Mobile Robots". *The Second International Conference on Image and Signal processing*,2009,pp. 1-5.

[5] Kuang Shunming, He Xiaojian. "Design and Application of CMOS Device Driver Based on S3c2440". *The Tenth International Conference on Electronic Measurement & Instruments*,2011,pp.110-114.

[6] Samsung. *S3C2440 32bit CMOS Microcontroller User's Manual[Z]*. Samsung Electronics Corp,2003.

[7] Christopher Hallinan. *Embedded Linux Primer*. "Embedded Linux Primer".Prentice Hall Publications,Second Edition,2009.