Application Development for Video Monitoring System & Motion Detection System using ARM9 Processor

Shubhangi S. Kose¹, Prof. Mrs. Jyoti M.Varvadekar²
¹M.E, Department of Electronics & Telecommunication, K. J. Somaiya College of Engineering Mumbai University
²Professor, Department of Electronics & Telecommunication, K. J. Somaiya College of Engineering Mumbai University

ABSTRACT
Recent years monitoring facilities are necessary and useful for our daily life to make us secure. Starting from small houses to big industries, now video surveillance is necessary and plays very important role to fulfill our security aspects in many ways. In this paper video monitoring system is designed bases on real time operating system i.e. Embedded Linux.

This system presents the structure of video capture based on S3C2440 processor and it introduces the embedded system, video capture and motion detection. This embedded web monitoring system takes the powerful ARM9 chip as MPU. In the monitoring site, the system captures the video through the embedded multitask operating system. The digital video has been compressed by the MJEG algorithm. By the web browser the users can view the monitor’s video directly, by the common Gateway interface, the users who are authorized can also control the camera and observe the motion detection. Therefore, we have developed a methodology to detect the motion in a video stream environment.

Keywords – Embedded Linux, Video Capture, Video compression, video streaming, web server, SDL, FFMPEG, motion detection

I. INTRODUCTION
The video data collection is designed with the capability of both wired and wireless internet accesses. The new version of our monitoring system will be powerful and useful for several kinds of clients. So one can say this system is client security oriented system.

Video monitoring systems play very important role in many fields of our society such as in banking, personal security, finance etc. In this surveillance system, a USB camera, ARM9 board, Wi-Fi modem and a PC is used for the real time video monitoring. In this surveillance system, mjpg streamer algorithm is used as real time operation [1]. By means of mjpg streamer algorithm, live streaming from camera is converted into different frames and then each one is transmitted through the DM-9000 Ethernet controller built in ARM9 board to the web server. Here, BOA web server is used to display real time video from USB camera to the client side. First, video data is captured from camera and then using mjpg streamer algorithm and Ethernet controller in ARM9, it is transmitted to the web server. Here video surveillance from the remote place to the web server is done only within the time delay of few microseconds. Entire video monitoring of the system is under the control of S3C2440. Hence, a system is used to detect any motion in a live streaming video.

II. SYSTEM HARDWARE DESIGN
The entire system consists of a USB camera, arm 9 processor boards and Ethernet. Using the web server application written in CGI script and html, we had interfaced the server IP and developed a GUI to monitor the video streaming on the PC [2]. The preceding and existing video surveillance system which entails high end cameras, video servers, network switch and monitoring PC all these resources leads to complexity, expensive, high power consumption and also requires more area to establish. The camera detects motion of any intruder in front of the camera dimensions. The software module communicates to the intended user either via Internet network [4].

In order to overcome the hitch in the preceding and existing system, we present a proficient where it uses few hardware resources for the implementation of the video monitoring system. S3C2440 [1] is a very good ARM9 family processor providing a camera interface which is very conducive to the application and development. Embedded Linux is chosen as operating system which provides open-source, multi-task, multi-process, highly modular, multi-platform support, performance and stability to the system. The
design system achieves maximum frame rate of 22fps with a resolution of 640x480.

Fig. 1- The structure of Hardware

III. SYSTEM SOFTWARE DESIGN
The software structure of the system, it mainly includes transplanting the operation system of ARM-Linux, compiling camera driving program, calling display, storage and compression programs.

| Application Layer: Image Acquisition, display and storage procedures |
| Middle Layer: ARM-Linux                                         |
|                                                               |
| File System                                                   |
|                                                               |
| Camera Driver                                                 |
|                                                               |
| Boot Loader(Vx86)                                             |

A minimum Embedded Linux system is composed of three essential components:
1. Bootloader: It gets the operating system loaded and running on the board.
2. Kernel: It is the software that manages the hardware and the processes.
3. Root File System: It is a directory tree containing needed libraries, scripts, utilities and applications. It follows a standard convention of file system hierarchy, for example, /usr/bin/.. etc etc [8].
4. Application: The program that runs on the board. The application can be a single file or a collection of hundreds of executables.

After the boot loader loads the kernel and the kernel mounts the root file system, it looks for a program to execute by default, or the user can supply the program with given environment variables. This program runs as the first process and it continues to run. When this process stops, the kernel, and thus the entire system, stops running. On the desktop Linux system, this program is likely init, which is known as the first user process. The init process performs low level initialization and system modes. In embedded devices various startup scripts are written to initialize the user program as the environment variables are not specified in the init script [9]. All these components are interrelated and thus depend on each other to create a running system. Working on an embedded Linux system requires interaction with all of these, even if our focus is only on the application.

The system selected Linux operating system [3][4] as software platform, the build environment using ARM-Gcc cross compiler debug mode, use embedded Linux 2.6 kernel. and there are mainly three function modules, that is, Video capture module, Video Compression module, and Video Streaming module. The flow chart system software shown in Figure 2.

Fig. 2- Flow chart of software system

IV. VIDEO CAPTURING MODULE
Video capture system is mainly based on the S3C2440 microprocessor, which is connected with the SDRAM, NAND Flash, camera, and GPIO port to construct a hardware platform that can support the
embedded Linux operating system and network. For the Linux version with 2.6.16 kernel or above, the driver of the cameras is called GSPCA [3]. And this system will select Linux with the 2.6.35 kernel. So, the driver for the camera can be implemented by burning the zImage file which is generated by compiling the kernel with the program of GSPCA into the NAND Flash.

V. VIDEO COMPRESSION
The increasing demand to incorporate video data into telecommunications services, the corporate environment, the entertainment industry, and even at home has made digital video technology a necessity. A problem, however, is that still image and digital video data rates are very large, typically in the range of 150Mbits/sec. For this reason, Video Compression standards have been developed to eliminate picture redundancy, allowing video information to be transmitted and stored in a compact and efficient manner.

Fig. 3-V4L Capture Flow

Discrete Cosine Transform (DCT) based compression algorithms and international standards were developed to alleviate storage and bandwidth limitations imposed by digital still image and motion video applications. Today there are three DCT-based standards that are widely used and accepted worldwide:
1. JPEG (Joint Photographic Experts Group)
2. H.261 (Video codec for audiovisual services)
3. MPEG (Motion Picture Experts Group)

Each of these standards is well suited for particular applications: JPEG for still image compression, H.261 for video conferencing, and MPEG for high-quality, multimedia systems.

VI. VIDEO STREAMMING
Once web camera is connected through master USB interface to arm board make minicom- s settings in the terminal window, during the settings we run the application related shell script in terminal which will execute application in board resulting video streaming on web browser using http protocol, entering a static IP address by user. Here the web browser is based on MJPG streamer for streaming captured video from camera placed in remote location.

MJPG-streamer", is a command line application that copied JPG-frame from a single input plug-in to multiple outputs plug-in. It can be used to stream JPEG files over an IP-based network from the webcam to a viewer like Firefox, Video lan client or even to a Windows Mobile device running the TCPMP-Player. It was written for embedded devices with very limited resources in terms of RAM and CPU. Its origin, the "uvc streamer" was written, because Linux-UVC compatible cameras directly produce JPEG-data, allowing fast and reformat M-JPEG streams even from an embedded device running Open WRT. The input module "input_uvc.so" captures such JPG frames from a connected webcam.

In multimedia, Motion JPEG (M-JPEG or MJPEG) is a video format in which each video frame or interlaced field of a digital video sequence is compressed separately as a JPEG image. Originally developed for multimedia PC applications, M-JPEG is now used by video-capture devices such as digital cameras, IP cameras, and webcams; and by non-linear video editing systems [5]. It continues to enjoy native support by the QuickTime Player, the PlayStation console, and browsers such as Google Chrome, and Mozilla Firefox.

The MJPG streamer is cross-compiled and loaded in to the S3C2440 board to act as a web streaming server. The server periodically obtain
videos from camera through the private network, such videos are transmitted from camera to the server. They are accumulated as MPEG video temporally into the internal buffer of the server. The script file .sh contains the following command:

```
cvlc vlc -vvv --color v4l2://dev/video0:size=640x480:fps=22:adev=pl ghw:1,0 --sout "#transcode {acodec=mpga,ab=128,samplerate=48000,channels =1,vcodec=mp2v,vb=1000000}:standard{access=ht tp,mux=ts,dst=192.168.1.230:8080}" on executing the script file the streaming process works. Streaming server should support some default plug-in tools for streaming video like ffmpeg, vlc for displaying video consisting still of images. VLC is the video LAN projects media player [5]. VLC can also be used as a streaming server that duplicates the stream it reads and multicasts them through the network to other clients or serves them through http. Gstreamer ffmpeg video plug-in supports a large number of audio and video compression formats through the use of FFMPEG library. Gstreamer is a streaming media framework which operates on media data. TCP/IP [5] application protocols included in the TCP/IP suite of protocols is an extensive list of applications designed to make use of the suites services. It is through these entities that resources can be made available data can be moved between hosts and remote user can communicate.

1. The board must be configured with network address over the NFS service.
2. Open the command prompt.
```
cmd ipconfig ping 192.168.1.230
```

And then the network connection is established between ARM9 Board and PC via Ethernet cable. Open webpage, the processed stream data packets are uploaded into server using TCP/IP protocol and transmitted through wireless or wired device so that user input the corresponding IP-address http://192.168.1.230:8080/project.html and the output video is streaming as shown in below Figure 4.

VII. MOTION DETECTION

Motion detection is the process of detecting a change in position of an object relative to its surroundings or the change in the surroundings relative to an object. In motion detection previous image compared with present image i.e here comparison of pixels of two images in this comparison if there is no difference indicates nothing is present, and then we see green color, if any difference is present indicates something is present then we see the green color on system.

If there exists a difference in the previous and present images RGB values, the motion is said to occur which is detected. Here motion detection algorithm is based on frame difference calculation in terms of RGB values and brightness threshold values stored in byte arrays. The algorithm compares two consecutive frames previous and present, pixel by pixel to generate a difference value. If they differ set color to max and If the difference value is greater than a fixed value (randomly taken), then motion is detected. Else, If they difference is below threshold set difference to 0 and there is no difference between previous and present frame’s and then present image is set to previous and present is next new image [6].

Here, the difference i.e. canvas() object to draw with and threshold specifies with how much the color value of a pixel must differ before they are regarded to be different [7].

```
function compare(image1,image2,canvas,threshold)
```

With every new picture, a compare() is performed. Then new picture is img1 and the previous image is now img2. The canvas() color is silver
In this system, previous image always compared with the present image. Now type command prompt

1. ifconfig
2. ping 192.168.1.230
3. sudo ifconfig eth0 down
4. sudo ifconfig eth0 192.168.1.10 up

Open the web browser & go to link 

VIII. RESULTS

<table>
<thead>
<tr>
<th>Sr.no</th>
<th>Parameters</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Video length</td>
<td>60 sec</td>
</tr>
<tr>
<td>2.</td>
<td>Frame rate</td>
<td>22 fps</td>
</tr>
<tr>
<td>3.</td>
<td>Resolution</td>
<td>640x480=307200</td>
</tr>
<tr>
<td>4.</td>
<td>No. of frames</td>
<td>60x22=1320</td>
</tr>
<tr>
<td>5.</td>
<td>Bit rate</td>
<td>60x512kbps=30720kbps=30.720Mbps</td>
</tr>
</tbody>
</table>

IX. CONCLUSION

In this paper we have designed and implemented video monitoring and motion detection system using Arm9 board. USB Camera module to capture the video and video streaming server on the embedded board for surveillance, using this we save the transmission bandwidth and memory. We have adopted Embedded Linux as the operating system on the embedded board because Linux supports well on the network, and many free required modules can be selected. Video 4 Linux is
used to get the camera video data, which is transferred to the Web Server, and the data is displayed on the client browser or on client. The system can be applied in intelligent anti-theft, intelligent transportation, intelligent home, medical treatment, as well as all kinds of video surveillance systems. Compared with video capture system based on digital signal processor (DSP), this system has the advantages of fewer modules, lower cost, higher intelligence, higher system stability, and higher security.

ACKNOWLEDGEMENT

I take immense pleasure to express my sincere and deep sense of gratitude to my guide and mentor Prof. Mrs. Jyoti M. Varvadekar Ma’am, K.J.Somaiya College of Engineering, for her sustained enthusiasm creative suggestions, motivation and exemplary guidance throughout the course of my project.

REFERENCES

Journal Papers:
[2] Research and Implementation of CMOS Camera Device Driver Based on S3C2440, ©2010 IEEE.

Books: