Introducing A Unique Strategy For Simplified Media Processing Engine

B.SNEHA REDDY
M. Tech Student, Embedded Systems
Department of Electronics and Communication Engineering
Vijaya Krishna Institute of Technology and Sciences

L.CHANDRASHEKAR
Guide & Hod , Associate Professor
Department of Electronics and Communication Engineering
Vijaya Krishna Institute of Technology and Sciences

Abstract: **A media player engine with scalable software connects is developed. Eventually, on Real 6410 platform, an embedded multi-media is made, which utilizes LINUX Frame buffer device driver interface to show video for testing the machine. Presently the extensive use of embedded multimedia processing requires dependence on hardware acceleration and multi-function expansion; it firmly requires a scalable and efficient style of system architecture for embedded multimedia processing terminal. Then, embedded multimedia processing software programs are implemented in line with the system architecture design. A scheduling program which schedules modules is implemented. The particular operation result proves that system architecture design simplifies the multimedia processing and supports multiple media formats in addition to hardware acceleration. Initially, system architecture with distinct structures and reasonable modules was created; a simplified multimedia framework is made for embedded application, with a scheduler along with a multimedia processing engine according to FFMPEG. The style of engine connects concentrates on compatibility and scalability to handle a number of applied needs. The scalability and efficiency allow it to be appropriate for embedded application.**

Keywords: **Scalability; System Architecture; Embedded Multimedia Framework**

I. INTRODUCTION

The extensive application urges for any simplified framework for embedded application, which not just meets the overall needs of video and audio process, but additionally has good scalability by way of adding or getting rid of preferred modules to apply custom functionality. At the moment, embedded multimedia processing continues to be broadly applied. For example, multimedia gamers, wise phones, digital set-top boxes, digital media advertising and security monitoring are associated with it. GStreamer word press plugin is a kind of dynamic link library including multiple elements, elements in pipeline gets in the plug-in. Core loads and registers plug-in after which schedules the execution of elements [1]. Therefore, developers could accomplish a plug-along with personalization features. GStreamer is effective but complex, it must be decline in functionality and enhanced software performance for particular embedded application. Based on our project demand, we design a scheduler along with a media processing engine to develop the multimedia framework:(1) scheduler provides scheduling functions: mode scheduling, message loop, module registration (2) a simplified media processing engine according to FFMPEG is made to integrate parser, demux, codec, event loop along with other functions together, connects are suitable for a number of I/O implementation and codec techniques. The look concentrates on functionality growing easily and following a principle of "strong cohesion, loose coupling” to create modules. In comparison, DSP can't provide memory protection without MMU it brings a burglar risk to system and boosts the burden on program development. Therefore, we decide S3C6410. S3C6410 adopts ARM1176JZF-S core and internally integrates with multiple multimedia co-processors including MFC (Multi Format CODEC) which could encode or decode MPEG4 Simple Profile, H.264/AVC Baseline Profile, H.263 P3 and VC-1 Primary Profile in additional than 30fps. Dual-core design may is more expensive power and needs greater programming workload. A Leg processor which integrates with multimedia acceleration abilities has good cost performance.

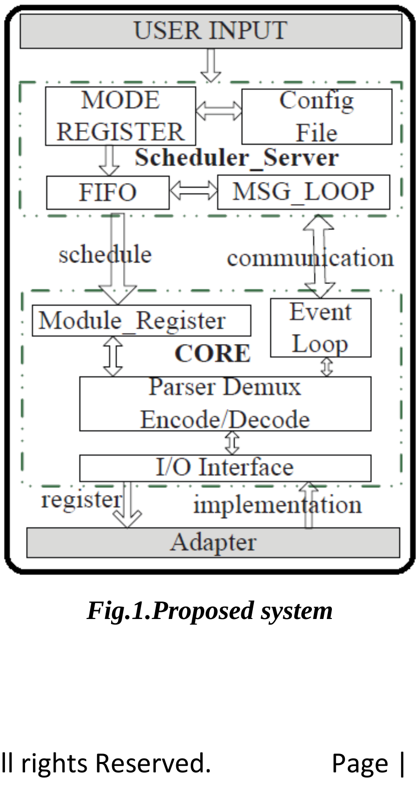

Fig.1.Proposed system

II. PROPOSED MODEL

A scalable multimedia processing architecture mainly reflects on three aspects: (1) a definite overall hierarchy (2) an acceptable division of functional modules (3) a simplified multimedia framework with compatible software connects makes functionality implemented easily. System layer includes system software, including UBOOT, Linux Kernel, device motorists, busy box, UBIFS file system and system libraries [2]. They compose a fundamental embedded LINUX atmosphere to supply support for programs in upper layer. Engine layer includes multimedia processing programs and libraries. Engine mainly comprises a scheduling support known as Scheduler Server, a multimedia processing core and a few dependent libraries including codec libraries, font libraries, etc. Display layer includes some graphical interface implementation programs and libraries which cooperate with multimedia processing core to process video display and audio playback. Cellular overall modules division, unified interface design is needed. Modules ought to be mutually independent in functionality, without disturbing one another whenever possible and self-contained. Cellular the detailed implementation of the module, a module ought to be split into sub-functions. Each function accomplishes a tiniest execution and keeps source code in fewer than 50 lines. Its modification should modify the module less than possible. In line with the principle above, software product is split into five primary modules: mode scheduling module, file input module, video output module, audio output module and multimedia processing engine module. The multimedia framework consists of two important modules: a Scheduler along with a multimedia processing engine, they cooperate to attain multimedia processing: Within this paper, a type of multimedia processing application is known as a mode. Scheduler Server may be the mode scheduler. Connecting in 2-way circular linked list, MCB precursor and subsequent search time complexity is O (1) .FIFO scheduling formula is used to mode switching. After mode registration, MCB is going to be placed in to the list rear .Current MCB is going to be selected from list mind to begin mode execution. First, in compliance with mode ID, Scheduler reads mode configuration from the XML file known as Scheduler_Config. Then enable mode registration function to see configuration of modules which fit in with this mode [4]. Call module registration function to complete module loading and write module information into MCB. Multimedia Processing Engine Module according to FFMPEG may be the core from the software this module could be split into six smaller sized modules: File Input, Demux_Parser, Synchronization, Playback Control, Event Loop, Audio Output Interface and Video Output Interface. Like a control signal, Compatibility mainly reflects in the style of I/O interface. Some function prototype is abstracted to satisfy compatibility. Connects cover overall process by removing key functions. There are many possible techniques for I/O, these techniques require different implementation. Gui libraries and LINUX FRAMEBUFFER device driver both does apply to output video. Engine requires abstract input and output interface to process file input and audio/video output .Adapter is really a specific I/O module registered by Scheduler. Interface utilizes function pointer to feed an interface realization with a specific adapter to engine. Real6410 platform has provided a couple.6.28 LINUX kernel that could be customized by modifying .confer. FFmpeg requires C dynamic link library libc.so version more than 2.7 to ensure that we develop our UBIFS. Media Processing Engine uses audio PTS as audio clock and primary clock. Based on audio clock, video clock changes its PTS through event loop. The whole process of multimedia processing software on Real6410 platform works stably .Mode scheduler and media processing engine cooperate to apply mode switching and modules applied. File input modules could read media file from internet disk and native UBIFS. The engine drives video display through FRAMEBUFFER and audio playback through ALSA effectively. The end result represents that: Multimedia framework design is affordable and it is connects are simple to be implemented by specific adapter .Its functionality grows easily by modules scheduling and combination [5]. The press processing engine nicely cooperates with assorted codec libraries and hardware accelerator program to aid audio and video stream processing easily in D1 resolution. The multimedia framework consists of two important modules: a Scheduler along with a multimedia processing engine, they cooperate to attain multimedia processing:

III. CONCLUSION

Engine mainly comprises a scheduling support known as Scheduler Server, a multimedia processing core and a few dependent libraries including codec libraries, font libraries, etc. Within this paper, embedded multi-media architecture with enhanced scalability was created and implemented, following a principle of "strong cohesion, low coupling”. The try Real6410 board proves the simplified multimedia framework design is efficient and fits for embedded application. software product is split into five primary modules: mode scheduling module, file input module, video output module, audio output module and multimedia processing engine module. Some function prototype is abstracted to satisfy compatibility. Connects cover overall process by removing key functions. There are many possible techniques for I/O, these techniques require different implementation. Compatible connects the perception of media processing engine could facilitate media processing procedure and satisfy the custom needs of embedded multimedia application. Compatibility mainly reflects in the style of I/O interface.

IV. REFERENCES

[1] Hong Zhuang, Zhaoqing wang, “IP-based real time video monitoring system with controllable platform, ”Proceedings of the IEEE/ASME International Coference on Mechatronic and Embedded Systems and Applications, Aug. 2006, pp. 1-4.

[2] T. Ho et al., “Byzantine Modification Detection in Multicast Networks Using Randomized Network Coding,” Proc. IEEE Int’l Symp. Information Theory (ISIT), 2004.

[3] Michele Sliger,Stacia Broderick, “The Software Project Manager's Bridge to Agility,” Addison Wesley,2008.

[4] Samsung Electronics Co. Ltd, “S3C6410X USER'S MANUAL,” February 2008, pp.50-58.

[5] Haibin Zhang,Hui Li,Dan Wu,Husheng Yuan,Tao Sun,Peng Yi, Hongchao Hu, Bingqiang Wang,“The design and implementation of an embedded high definition player,” Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference, 26-28 Feb 2010,vol.4,pp.301 – 304.