

Design and Implementation of Embedded Media Player Based On ARM9

M.Sushanth¹, K.Somasekhara Rao², M.Ravi Prakash³

M.Tech (Embedded systems), Geethanjali collage of Engineering and Technology AP

M.Tech Professor Geethanjali collage of Engineering and Technology AP

M.Tech, PhD Electronic Communication, JNTU Hyderabad AP

ABSTRACT:

In this paper of Embedded Media Player we have selected Samsung S3C2440A as microprocessor from the family of Samsung ARM9. And we have been using 64MByte Nand Flash kwon to be K9F1208 an external memory and a Nor Flash of 2MByte SST39VF1601 to play the Video and audio files and to display the picture frames effectively and smoothly. The memory external interface is been updated with micro SDcard and USB which is meant to be an expandable memory. We have installed the optimized Linux operating system in the processor and transplanted the SDL_FFMPEG library into S3C2440 after the cross compilation. Of this whole system come together in playing the audio/video and picture formats files smoothly and effectively.

KEYWORDS: ARM9, Linux, Nand Flash, Nor flash, SDcard, SDL, SDL_FFMPEG, SST39VF1601

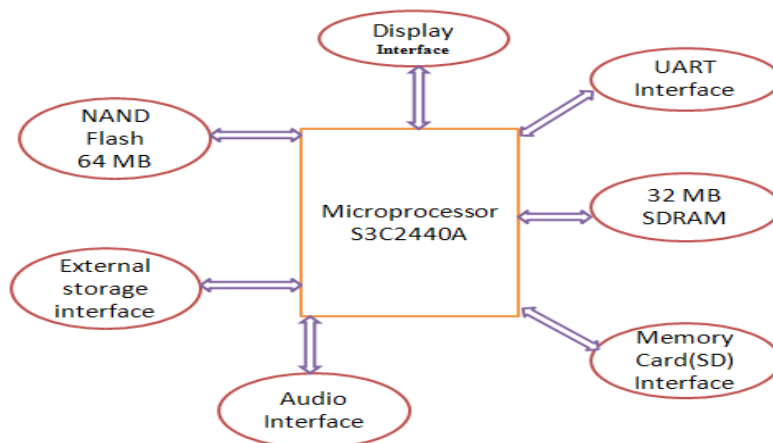
I. INTRODUCTION

Embedded media player is one of the popularly used electronic products for the entertainment purpose. People especially need them when they are in travel, outings and tourism. Because of the popularity the manufacturers make their market targets and produce a large number of products year by year which increases the market rate. In the market competition the manufacturer with best quality and effective cost will gain profits in the market. Because of this we have been using S3C2440 as microprocessor with an extended memory interfacing of USB and SDcard. The external interfacing are done with LCD, UART, EATHERNET, audio ports. In the software optimized Linux operating system is imaged on to the S3C2440. The SDL library which is been included in the OS uses SDL_FFMPEG library to encode and decode the audio/ video or picture frames.

II. SYSTEM HARDWARE DESIGN

We have been using 32-bit RISC processor, 64Mbytes Nand flash, 32Mbytes SDRAM externally S3C2440 as core processor. The main reason in using the S3C2440 core processor is famous in its low power consumptions and working in high frequencies. Where the frequency can be maximum up to 400MHZ. OF all those price comparatively low, which means it is cost effective. The LCD which is of 3.5 inches ensures the better picture quality with multiple touch and TFT (Thin Film Transistor) technologies respectively for the better touch and picture qualities. Since the video files would be big in size the extended USB and SDcard are interfaced to use the externally memory effectively.

The System hardware structure is shown below in the Figure 1. From the above figure 1 we notice that operating system uses NandFlash for the data storage for Boot Loader and other software applications placed in it. SDcard and USB acts as expanded external memory to store the files which are needed. When the system works SDRAM is used as dynamic memory for the whole system. Audio information or data is played through the audio interface. To display and to choose the menu options we use Touch screen LCD. One of the provisions observed is the UART port acts as the bridge to the developers and the system for the designing and developing purpose after which it can be dropped or removed as per the convenience.



“Figure 1. System Hardware Design”

III. SYSTEM SOFTWARE DESIGN

Linux is derived from the UNIX-based operating system which was originally developed from INTEL-compatible PC's OS

3.1. Customization of Operating System

These days Linux is available for wide range of PAD's (Personal Digital Assistance) ranging from wrists watches to mainframes. Linux is known as “modern operating system” for its silent features such as Multitasking, Multiuser, Multi platform, Multithreading, crash proof, virtual memory, memory protection etc...For which it is been developed by a large number of international community of users and developers as an open-source software. These developers believe it is an alternative for the mostly used Windows and Solaris operating systems and few applications such as MS Office, Internet Explorer, Outlook and other application software. Before the optimization the Linux operation system is huge it use in such a portable electronic devices. Since that we need don't need the full functioning of the huge operating system. For which it have been optimized and scaled to a portable devices friendly. In the optimization of Linux the few of the functions removed such as power management, network, wireless communication, hard disk etc...The command ARCH for arm and CROSS_COMPILE for arm-linux has to be entered in the root directory to remove and configures the menu configurations of Linux kernel.

3.2. Software Interface Design of Media Player

Simple Direct Media Layer (SDL) is the basic cross-platform used for the multimedia library which provides a unified programming interface and also used for the software development which intern provides convenient for the code transplantation. SDL is written basic C language which provides functions to the user through library and also provides programming language interface in large number. One of the most important features of SDL is it is an open source, through which the user can get the source code from the SDL's homepage it self which can be used freely by using the licensed second version of GNU LGPL. SDL is very simple to use, it contains eight subsystems such as event handler, timer, video, sound etc...To initialize the system the user need to call the function `SDL_Init()`, he can also initialize the system by setting the parameters of `SDL_Init()` by which he can choose the audio interfacing by calling the function `SDL_Init(SDL_INIT_AUDIO)` in this paper we need multiple software subsystems which can call functions to initialize with the `SDL_Init(SDL_INIT_AUDIO|SDL_INIT_VIDEO|SDL_INIT_TIMER)` together. For the particular and specific equipment SDL provides that particular function to accesses information through that equipment such as `SDL_VideoInfo` to access the video information through display equipment and we can call the function `SDL_SetVideoMode` which Set up a video mode with the specified width, height and bits-per-pixel. Similarly we have functions such as `SDL_ListModes` which returns a pointer to an array of available screen dimensions for the given format and video flags. For audio files functions like `SDL_OpenAudio()` to access to the sound information through sound equipment similarly functions like `SDL_CloseAudio` -- Shuts down audio processing and closes the audio device, `SDL_BuildAudioCVT` -- Initializes a `SDL_AudioCVT` structure for conversion, `SDL_PauseAudio` -- Pauses and play the audio callback processing for necessary corrections and configuring the files provided with the hardware equipment. In designing this media player its takes lot of time for the research in the video subsystem. As per video is concern we concentrate on the unity

video buffer to be shown, any thing operated on the display screen it corresponds to the video buffer. The video buffer can be selected as physical memory or a physical video graphic device memory at the time of video initialization. Here in this paper we selected the video graphic device memory which is known as physical memory and used as a buffer. So that we call functions like `SDL_SetVideoMode` -- Set up a video mode with the specified width, height and bits-per-pixel, and we select `SDL_SWSURFACE` for the parameter flag and we use `SDL_ListModes` which Returns a pointer to an array of available screen dimensions for the given format and video flags. The software interface I broken down into small pieces implemented on the paper and the drawing software with decorative things such as window types, buttons, display pattern and other display parameters on the screen. SDL acts as a stick pad and allows sticking all the display elements on it to me pasted in the display for the better view of the display screen. We cannot use the source code provided in the SDL website directly into the hardware system. Before using the source code we need to compile the using the cross compiler `arm-linux-gcc`. Then source code can be copied to the PC's Linux operating system to the following directory `/root/work/SDL/` and get on to the newly created directory to compile with the following command line up.

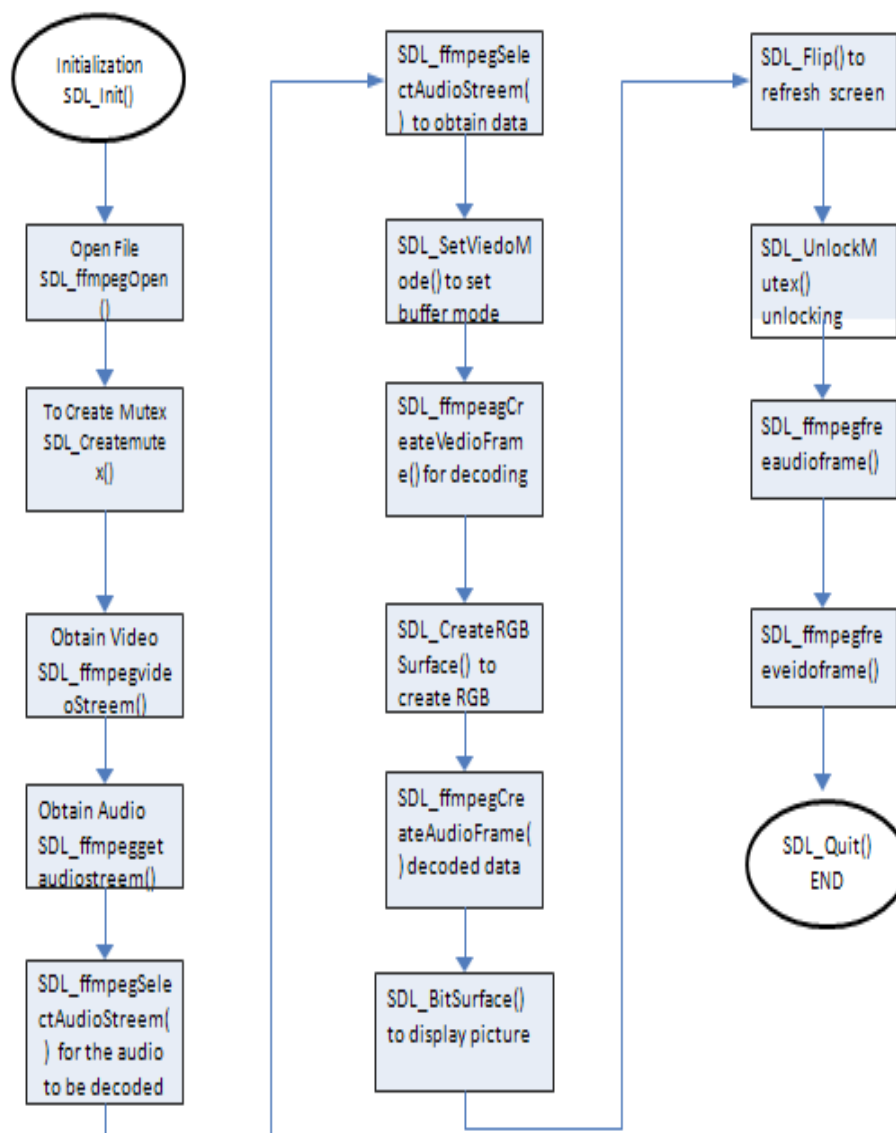
3.3.Commands

```
./configure
--disable-video-gtopia --disable-video-dummy
--disable-video-fbcon --disable-video-dga --disable-esd
--disable-alsa --disable-cdrom --disable-video-x11
--disable-nasm --target=arm-linux --host=arm-linux
--enable-video-fbcon
```

are the functions used to compile the source code. After the compilation is done it can be copied to the hardware system.

3.4.Core Design of Media Player

Encoding and decoding of the video or audio files and to synchronies the audio stream and video stream between them is the core of the media player. The SDL is contained with the powerful library kwon as FFmpeg which provides perfect and complete solution for the audio and video files. Since the FFmpeg library is difficult to understand a simple modified version developed for the beginner's kwon as `SDL_FFmpeg`, this version library is called `libavcodec`. `SDL_FFmpeg` is packed with some multiple functions for the interfacing purpose. Such that users need not know the typical FFmpeg commands rather uses `SDL_FFmpeg` functions to operate. `SDL_ffmpegFile*` `SDL_ffmpegOpen(const char* filename)` is used to open the video files, which is also call with the command `SDL_ffmpegAddVideoFrame()` to add an video frame for the `SDL_ffmpegFrame` directory. At the time of video streaming we have to call the functions `SDL_ffmpegVideoDuration()` and `SDL_ffmpegAudioDuration()` to synchronize the two streams. To set the speed of the media player we need to cal the function `SDL_ffmpegGetFrameRate()` if we want to record the current playing video on the screen `SDL_ffmpegGetPosition` is to be called and to seek the position of the video which is playing `SDL_ffmpegSeek()` is to be realized.



“Figure 3. Media Player Flow Chart”

3.5.Functions and their Effects

```

--SDL_ffmpegInit () */Initializes the SDL_ffmpeg library/*
-- SDL_ffmpegFree (SDL_ffmpegFile *file) */Use this to free an SDL_ffmpegFile/*
--SDL_ffmpegFreeAudioFrame(SDL_ffmpegAudioFrame*frame)*/Use this to free an SDL_ffmpegAudioFrame/*
--SDL_ffmpegFreeVideoFrame (SDL_ffmpegVideoFrame *frame) */Use this to free an SDL_ffmpegVideoFrame/*
--SDL_ffmpegOpen (const char *filename) */Use this to open the multimedia file of your choice/*
--SDL_ffmpegCreate (const char *filename) */Use this to create the multimedia file of your choice/*
--SDL_ffmpegAddVideoFrame (SDL_ffmpegFile *file, SDL_ffmpegVideoFrame *frame) */Use this to add a SDL_ffmpegVideoFrame to file/*
--SDL_ffmpegAddAudioFrame (SDL_ffmpegFile *file, SDL_ffmpegAudioFrame *frame) */Use this to add a SDL_ffmpegAudioFrame to file/*
--SDL_ffmpegCreateAudioFrame (SDL_ffmpegFile *file, uint32_t bytes) */Use this to create a SDL_ffmpegAudioFrame/*
--SDL_ffmpegCreateVideoFrame (const SDL_ffmpegFile *file, const uint32_t format, SDL_Surface *screen) */Use this to create a SDL_ffmpegVideoFrame/*

```

```
--SDL_ffmpegGetVideoFrame (SDL_ffmpegFile *file, SDL_ffmpegVideoFrame *frame) */Use this to get new
video data from file/* SDL_ffmpegStream *
--SDL_ffmpegGetAudioStream (SDL_ffmpegFile *file, uint32_t audioID) */Get the desired audio stream from
file/* SDL_ffmpegStream *
--SDL_ffmpegGetVideoStream (SDL_ffmpegFile *file, uint32_t videoID) */Get the desired video stream from
file/*
```

The above are the few functions used to access the video/audio file that need to be accessed



“Figure 4. A Designed Media Player with Hardware”

IV. CONCLUSION

After the software and the audio/video and picture files are observed with their effective synchronization between audio and audio. The practical implemented media player is shown in the figure 4. The practical implementation emphasizes the advantages for being low power consumption, less memory footprint, cost reliable, simple structure, compressed size for the comfortable portability with simple structure, stable with the audio video synchronization for the effective and quality output of the media files when executed. By all these seasons make the product with the effective development in the fierce in the market and to improve the sales with simple structure and implementation design.

REFERENCES

- [1] Wenhao Wang, Mingyu Gao, Hangzhou, Zhejiang “Design Of Embedded Media Player Based on S3C2440 and SDL_FFMPEG”
- [2] SAMSUNG, “S3C2440A 32-BIT CMOS MICROCONTROLLER USER'S MANUAL Revision 1”.2004 Samsung Electronics.
- [3] www.friendlyarm.com
- [4] Sam Lantinga: <http://sdl.org>
- [5] SDL's home page: <http://www.libsdl.org/index.php>.
- [6] SDL_FFMPEG's home page: http://www.arjanhouben.nl/SDL_ffmpeg
- [7] www.dranger.com
- [8] <http://en.wikipedia.org/wiki/Video>
- [9] <http://en.wikipedia.org/wiki/mp>
- [10] <http://en.wikipedia.org/wiki/codecs>