

Design and Implementation of Digital TV Widget for Android on Multi-core Platform

Yu-Sheng Lu^{1,2}, Chin-Ho Lee¹, Hung-Yen Weng², Yueh-Min Huang²

1. Business Customer Solutions Lab, Chunghwa Telecom Laboratories, No. 12, Lane 551, Min-Tsu Rd. Sec. 5 Yang-Mei, Taoyuan 326, Taiwan

2. Department of Engineering Science, National Cheng Kung University, No. 1, University Rd., Tainan 701, Taiwan
yusheng@cht.com.tw

Abstract—With the rapid advancements in hand-held devices, traditional phone have also been developed into smart multi-functional mobile phones, which feature numerous functions and flexible operation. Such rapid advancement results from transformations of the embedded system architecture. In the mode of a multi-core cooperative work, the work requiring greater mathematical operations, such as video decoding, is assigned to a cooperative processor, which is specially designed for mathematical operations. In this way, the work load of a main processor can be reduced, and high operation loads can be realized. Based on the open source software model, Android may reduce time intervals in developing application programs, thus, hand-held manufacturers are paying increased attention to the system. Android has good network support capabilities, as well as a totally open software development environment, thus, there are many completed application programs available for download and use. However, until now, there was no digital TV player in Android. When a digital TV broadcast is combined with other application programs, other services will become available while TV programs are broadcast. This paper proposes how to add the mechanism for watching digital TV programs onto the multi-core platform, Texas Instrument TMS320DM6446, which is realized through cooperative works in order to use the processors of the multi-core system with Android.

Keywords—Android; DVB-T; multi-core platform; TMS320DM6446

I. INTRODUCTION

Driven by rapid improvements in video and audio quality, and the fast development of the 3G network, hand-held devices have gradually transformed into personal mobile multi-media devices. Hand-held devices tend to have a small size, high efficiency, and consume less electricity. Although conventional personal computers have many functions and high efficiency, they consume large amounts of electricity; and therefore, they are not suitable for use in hand-held devices, and instead, an embedded system designed for special purposes is utilized. The former embedded system could only deal with one piece of work at one time; moreover, its functions were limited, and thus, it could not satisfy the operational requirements of a multi-media system. The current embedded system has many functions and works with a multi-task operating system, thus, it is able to perform simultaneous tasks. For hand-held devices, a battery is the main power source. Therefore, a CPU (Central Processing Unit) with powerful functions is generally not used; however, a RISC processor, such as ARM, is utilized to deal with control-dominated works, and one DSP (Digital Signal Processor) with powerful computation functions works with the CPU, which is responsible for general events, signal distribution, and interruption handling. When a high computation load

occurs, the CPU will start the DSP, which is especially designed to process large amounts of multiply-add instructions, making it suitable for work such as video and audio compression and decoding. A mode with several processors engaged in cooperative works is called a multi-core platform.

Since Google launched a development platform for hand-held devices in 2007, Android has aroused great attention and intense discussion in many free software associations and forums around the world [1]. Android is obviously characterized by its completeness and publicity, and as such, it has become a development kit and program framework for open software. If mobile phone manufacturers adopt Android for mobile phone development, the costs of software development could be largely reduced. In the Android operating system, application programs work in the form of a widget, thus, programs can be attached to other programs for synchronous execution. For example, users can watch TV while reading mobile phone messages and browsing a network photo album. Although Android has complete Internet functions, including the wireless network functions of 3G or 802.11, it is not provided with the program or the kit for receiving digital TV programs. Based on the concept that Android was developed for use in hand-held devices, this paper chooses a multi-core embedded system as a hardware platform, and develops a digital TV watching platform using the Android operating system in a multi-core framework. The framework involves a mechanism for receiving network serial streams, dealing with multi-core cooperative works, and decoding and broadcasting video and audio. The operating interfaces are from software development kit designs of Android. The contributions of this paper are to transplant Android into a multi-core platform, and then, add the mechanism in the Android platform for watching digital TV programs and achieve cooperative works with the processors of a multi-core system

The remainder of this paper is organized as follows: Section 2 introduces the hardware platform architecture; Section 3 describes the software architecture; Section 4 describes the system architecture and implementation; Section 5 discusses experiments and tests; Section 6 gives conclusions and suggestion to future work.

II. HARDWARE PLATFORM ARCHITECTURE

This study uses TMS320DM6446 development board (Davinci EVM, DM6446), provided by TI, as the main development platform. DM6446 is employed to develop a user interface that has features of high efficiency and extended battery life to integrate the multi-media video and audio, and

network multi-media encoding/decoding system. It can be used to develop applications such as digital multi-media, multi-media encoding/decoding based on network transmission, and image processing [2]. Figure 1 is a block diagram of DM6446 functions [3]. DM6446 is divided into three major sub-systems, which are ARM sub-system, DSP sub-system, and video signal processing sub-system. The DM6446 processor is a System on Chip (SoC) consisting of ARM926EJ-S and TMS320C64x+™, which is a double-core framework

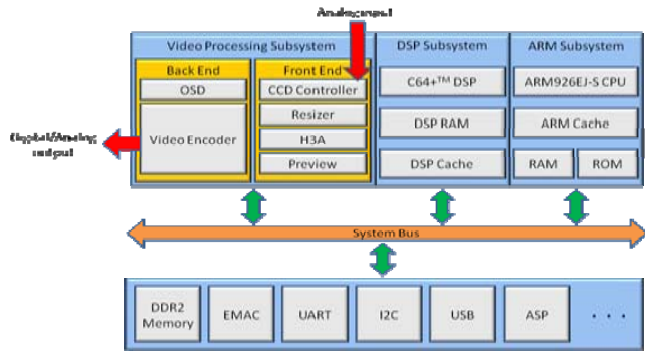


Figure 1. Block diagram of DM6446 functions

A. ARM Sub-System

ARM is 32-bit reduced instruction set computer (RISC), which mainly processes 32-bit, 16-bit, and 8-bit data, and a supporting pipeline framework, thus enabling the processor and memory unit to continuously process data.

B. DSP Sub-System

TMS320C64x+™ is a highly efficient digital signal processor, which is able to rapidly process a large number of multiply-add instructions. It is mainly used to encode/decode video/audio signals, and is suitable for integrating multi-media video & audio systems. TMS320C64x+™ includes 2 arithmetic logical units (ALU), 2 multipliers, 64 32-bit buffers, etc. It can carry out arithmetic operations of 1 32-bit group, 2 16-bit groups, or 4 8-bit groups, and the multiply-add instructions of 4 multiplying 16-bit and 16-bit groups, or 8 multiplying 8-bit and 8-bit groups, within a unit time.

C. video signal processing sub-system

This sub-system includes front and rear units for video signal processing. The front unit is used to acquire video signals from image acquisition equipment, while the rear unit is employed to display messages on the output equipment. The former includes a charge-coupled device (CCD) controller, auto focus (AF), auto white balance (AWB), auto exposure (AE), and preview engine (PE). Of these, the CCD controller receives images acquired by an external CCD camera, and sends the original images to other devices; H3A adjusts the receiving parameters in the CCD controller; PE previews current images received by the CCD controller, and enlarges or lessens images. The rear unit includes on-screen display (OSD) and a video encoder (VE). OSD separates output images into those with superposing effects at several levels. For example, strips or numbers of volume adjustments or

image quality adjustments appearing on the main display menu of a TV screen utilizes the superposing effects of video images. VE provides 4 sets of codec for converting digital signals to analog signals, and encoding video signals into NTSC, PAL, or S terminals, as well as output Component terminals composed of RGB or YCbCr.

III. SOFTWARE PLATFORM ARCHITECTURE

A. Google Android

Android is a software stack [4] created by Google and Open Handset Alliance for mobile devices in 2007. It includes an operating system, agency software, and an application. The operating system core is based on Linux, and is composed of several driver modules independently developed by Google. Agency software abstracts the interface between the operating system core and the application program. Developed application programs should match with the application programming interface (API) provided by agency software, thus, system resources may be properly accessed. If any change takes place in the bottom hardware, with suitable adjustments in the agency software, application programs can be executed without any changes. Application programs are developed with Java programming language, thus, it is very easy to install the same application programs in different hardware devices.

B. Android System Architecture

Figure 2 shows the framework of the Android software stack. To explain the hierarchical mode of Android, a brief introduction is given for each layer.

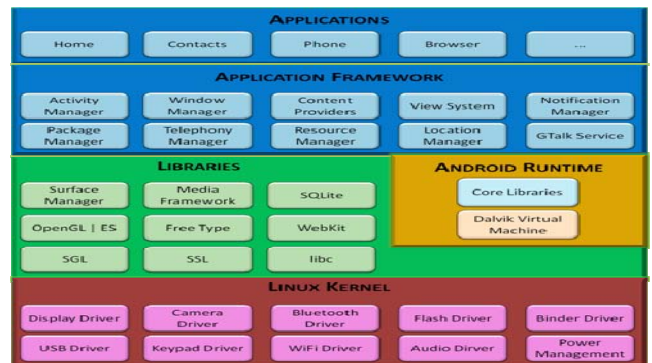


Figure 2. Android software stack

- **Application**
Android provides the common application programs of mobile devices, such as E-mail receiving and sending, brief messages, calendar, Google maps, browser, address book, etc. These programs are developed with the Java programming language.
- **Application Framework**
The Application Framework provides many programming interfaces. By calling the functions of these interfaces, application programs can be easily designed, thus simplifying processes and re-utilizing

resources. In addition, through the functions provided by the Application Framework, original core application programs of Android can be re-written, such as adding a photo display function into the address book.

- The Android Application Framework provides the following functional interfaces:
 1. Content Providers: to allow application programs to access the data of other application programs.
 2. Resource Manager: to manage non-program parts, such as multi-lingual encoding, photos, and sound effects.
 3. Notification Manager: to allow application programs to notify the interface of user messages.
 4. Activity Manager: to manage the life cycle of each application program.

- Libraries

Android Libraries include C and C++ link libraries, connecting upwards with the Application Framework, and downwards with the operating system core. The main libraries include a System C library, a multimedia library, a network library, a database library, a 3D library, and a font library. Most of these libraries are improved programs of the embedded system, with higher execution efficiency, smaller program codes, and reduced functions.

- Android Runtime

This layer includes a core library and a Dalvik virtual machine, which is used to execute Java programs. The Dalvik virtual machine is especially improved for the embedded system, and its programs occupy smaller memory space; at the same time, while it is able to simultaneously execute several application programs.

- Linux core

The Linux 2.6 version-based Android operating system provides a framework for modules, such as safety, memory management, executive program management, a network stack, and drivers. The Linux core is an abstract layer between the application program or the library and the hardware, which manages all hardware resources.

C. Codec Server and Codec Engine

The frequent multi-core co-processing platform has two common working modes among the cores. One is a master and slave type framework, where one main processor is responsible for the operations of all the hardware platform. When co-processing work is required, algorithms or programs are loaded in the co-processing core. Such modes as interruption, memory sharing, and an information channel are utilized to realize information exchange between the two processors. In the other mode, the two cores work together, and they may execute different operating systems. When data

or a program exchange is conducted between these two cores, a complete data structure is required. Different from the master and slave framework, the co-processor mode can load several algorithms or programs. These algorithms or programs are managed by the co-processor's operating system, and the communication mode between these two cores is memory sharing.

The multi-core cooperative working mode uses DM6446, where two processors work together. The main processor ARM applies the Linux operating system, and the co-processor DSP uses DSP BIOS. Information exchange is realized between the two processors through two programs, namely, a codec server and a codec engine. The codec server is based on DSP, which serves as the server end and provides various registered algorithms; the codec engine is based on ARM, which serves as a communication bridge between the codec server and the application programs. The communication mode between ARM and DSP is shown in Figure 3. Sharing memory mainly transfers parameters, while EDMA and ASP transfer data.

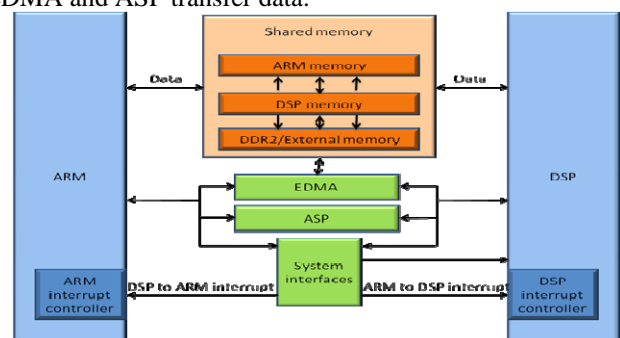


Figure 3. Communication mode of ARM and DSP of DM6446

When DSP is used for encoding and decoding algorithms in DM6446, the programming interfaces specified by the codec engine should be conformed to, as the algorithms are encapsulated in a special format. Through the DSP Link of the Linux driver module, DSP BIOS is registered into the codec server to provide services. These programming interfaces and data structures should also be conformed to when application programs call these algorithms, and a codec engine demands services from a codec server. The flow of algorithm registration is as follows: application programs call the VISA (Video, Image, Speech, Audio) interface, and make registration demands to the codec engine with a xDM data structure, then codec engine loads the programs or algorithms into DSP through the DSP Link of Linux, on the bottom. After the DSP BIOS receives a registration demand, it notifies the resource manager to generate new space, and registers the algorithms in the algorithm list of the codec server.

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION

A. Processing flow of digital TV programs:

Processing flows of digital TV programs from input to output are shown in Figure 4.[6][7]

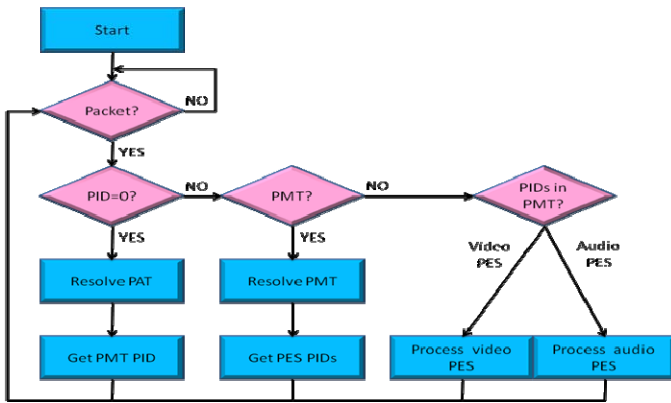


Figure 4. shows the flow chart of encapsulation for analyzed transmission serial streams

- Adjust the parameters of the modulator, such as the Quadrature Amplitude Modulation (QAM), bandwidth, the transmission rate, and the guard interval used in modulation, in order to receive transmission serial streams.
- Analyze transmission serial streams:
 - Reject incomplete packages.
 - Read program index table with PID equal to 0, remove the program information therein, and locate the PID of the mapping table of each program. Serial streams under certain frequencies and modulation parameters may include many programs, which are all included in the program index table.
 - Read the package with the program mapping table equal to one PID, from which the PID of video, audio, subtitle, and multi-lingual elementary streams can be analyzed.
 - Analytic mode of video and audio reads corresponding PID packages, and re-composes the load of the same PID into one serial stream in order to form the video/audio elementary streams.
 - Processing mode of video serial streams: as elementary streams are in an MPEG2 compressed format, they should be decoded before being broadcast. For this purpose, the codec engine and codec server are used to call the MPEG2 decoding algorithm of DSP for decoding.
- Processing mode of audio serial streams: the codec engine and codec server are used to call the MPEG1 Layer2 decoding algorithm of DSP for decoding.
- The decoded video data frames are in the YUV color format, with a resolution of 704x480; however, the resolution of the display is 720x480; therefore, they should be resized before being displayed. Resolution adjustment adopts the interpolation method, i.e. the mean values of the adjacent left and right data in the horizontal frames are used as the data for the row.

- Adjusted frames are sent to the video frame buffer, which is located in the memory block of the system and corresponds to the display memory of the solid hardware, and is mainly used to output menus.
- The video encoder at the rear of the video processing sub-system converts the digital data of the frame buffer into simulative composite signals, which it outputs to the screen, and conforms to the NTSC standard.
- Decoded audio waveform data are transmitted to the output port through a sequence audio port.

V. EXPERIMENTS AND TESTS

Digital TV broadcast is divided into three steps:

- Receiving transmission serial streams and re-building elementary streams: the processes of receiving transmission serial streams to constructing video/audio elementary streams are shown in Figure 5. First, adjust the modulator's parameters in order to receive the transmission serial streams; then filter the video/audio PID packages out of serial streams, which form the video/audio packetized elementary streams (PES); finally, remove the package information, namely, head and tail, and the acquired load forms the elementary streams. Elementary streams can then be broadcast or displayed

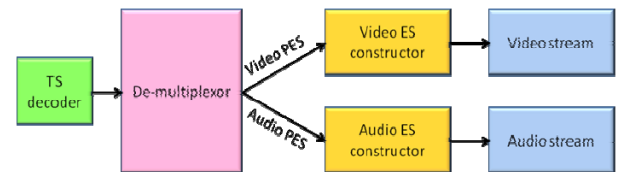


Figure 5. Process of rebuilding video/audio serial streams from transmission serial streams

- Decoding MPEG2 serial streams: call the MPEG2 video decoding algorithm of the digital signal processor to decode the rebuilt video serial streams into frames; call the MPEG1-L2 audio decoding algorithm for decoding audio serial streams that form the audio wave. As the DSP BIOS in a digital signal process serves as resource management, the said two algorithms can be simultaneously executed.
- Broadcasting decoded video frames and audio wave form: decoded video frames are sent to the video buffer device, which further resizes these frames; after they are in agreement with the resolution of the display; the video encoder encodes them into the output format. This paper uses Composite signals. The decoded audio wave form is sent to an audio serial port, and is then output to the amplifier for broadcasting. Video signals for broadcasting digital TV programs are shown in Figure 6 and Figure 7. The original resolution of the video serial streams is 704x480, while the resolution of the display is 720x480. The figure shows that the video signals have been resized to within the visible range of the display.



Figure 6. Video signals for broadcasting digital TV programs



Figure 7. DM6446 broadcasting digital TV programs

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a digital TV broadcasting platform based on Android, and describes how transmission serial streams received from the hardware modulator are converted into original data. Then, these data are sent to the digital signal processor, which decodes them into a broadcast-able format.

The Android image interface is used to operate the digital TV broadcaster. Its content includes a co-processing mode of a two-core platform, analysis of serial stream data, image interface application programming, interface connecting Android application programs with the original programs of the system, and reading and calling of shared libraries. In future research, two services, namely, an electronic program guide (EPG) and signal strength detection (SSD), can be added into the digital TV functions. Based on an image user interface, the touch screen input mode of this paper can be improved. For hand-held devices using a keyboard input mode, keyboard position and size should be taken into consideration in the hardware design. If the keyboard is designed in the software, and simulated keyboard is clicked on the screen, then the hardware design space can be increased.

VII. ACKNOWLEDGMENTS

Yueh-Min Huang's research in this paper was supported by Sustainable Growth Project (98WFA0900277) of Department of Engineering and Application Science, National Science Council (NSC), Taiwan.

REFERENCES

- [1] Gozalvez, J., "First Google's android phone launched [Mobile Radio]," *Vehicular Technology Magazine, IEEE*, vol.3, no.4, pp.3-69, Dec. 2008
- [2] Talla, D.; Gobton, J., "Using DaVinci Technology for Digital Video Devices," *Computer*, vol.40, no.10, pp.53-61, Oct. 2007
- [3] "TMS320DM6446 DaVinci Digital Media System-on-Chip", <http://focus.ti.com/docs/prod/folders/print/tms320dm6446.html>, retrieved on July 2009.
- [4] "What is Android?", <http://developer.android.com/guide/basics/what-is-android.html>, retrieved on July 2009.
- [5] "Codec Engine Server Integrator User's Guide", <http://focus.ti.com/lit/ug/sprued5b/sprued5b.pdf>, retrieved on July 2009.
- [6] Brett, M.; Gerstenberg, B.; Herberg, C.; Shavit, G.; Liondas, H., "Video processing for single-chip DVB decoder," *Consumer Electronics, 2001. ICCE. International Conference on*, vol., no., pp.82-83, 2001
- [7] "The Java Native Interface Programmer's Guide and Specification", <http://java.sun.com/docs/books/jni/html/start.html#769>, retrieved on July 2009.