



# Embedded Linux Implementation on a Commercial Digital TV System

*IEEE Transactions on Consumer Electronics, Vol. 49, No. 4, NOVEMBER 2003*

*Sang-Pil Moon, Joo-Won Kim, Kuk-Ho Bae, Jae-Cheon Lee and Dae-Wha Seo,  
Member, IEEE*

Student : Sheng-Song Huang

Adviser : Dr. Lian-Jou Tsai

# Outline



- Abstract
- Introduction
- Digital TV operation system
- Commercial digital TV system
- Linux porting
- Implementation
- Experimental result
- Conclusion



# Abstract



- A Digital TV system is necessary for not only video and audio, but also data processing.
- Because of many functionalities and multitasking jobs, it needs an Operating System.
- In this paper, we modified the embedded Linux kernel and the cross development environment for a big-endian system.
- Redesigned device drivers for kernel execution, and configured system memory map in order to load the Linux kernel.



# Introduction(1/2)



- In order to increase the competitiveness of ground broadcasting services, digital broadcasting services are applied widely.
- Therefore, a digital TV system needs a high-performance processor and a large amount of memory.
- An embedded operating system that can efficiently manage hardware resources is required.
- There are commercial embedded operating systems such as pSOS in digital TV systems.
- If an embedded Linux operating system with open source is used, cost of production can be reduced.



# Introduction(2/2)



- Open source device drivers and reusable applications, convenient development environment configuration using module function.
- Shell utility and file systems, and easy resolution of a problem from open source communities.
- In this paper, we will describe an embedded Linux porting in digital LCD projection TV.
- We modified the kernel library and the cross development environment into a big-endian memory access.
- Redesigned the essential device driver, and configured the system memory map for loading the kernel into main memory.
- We designed a bus interface device driver to control the entire system.



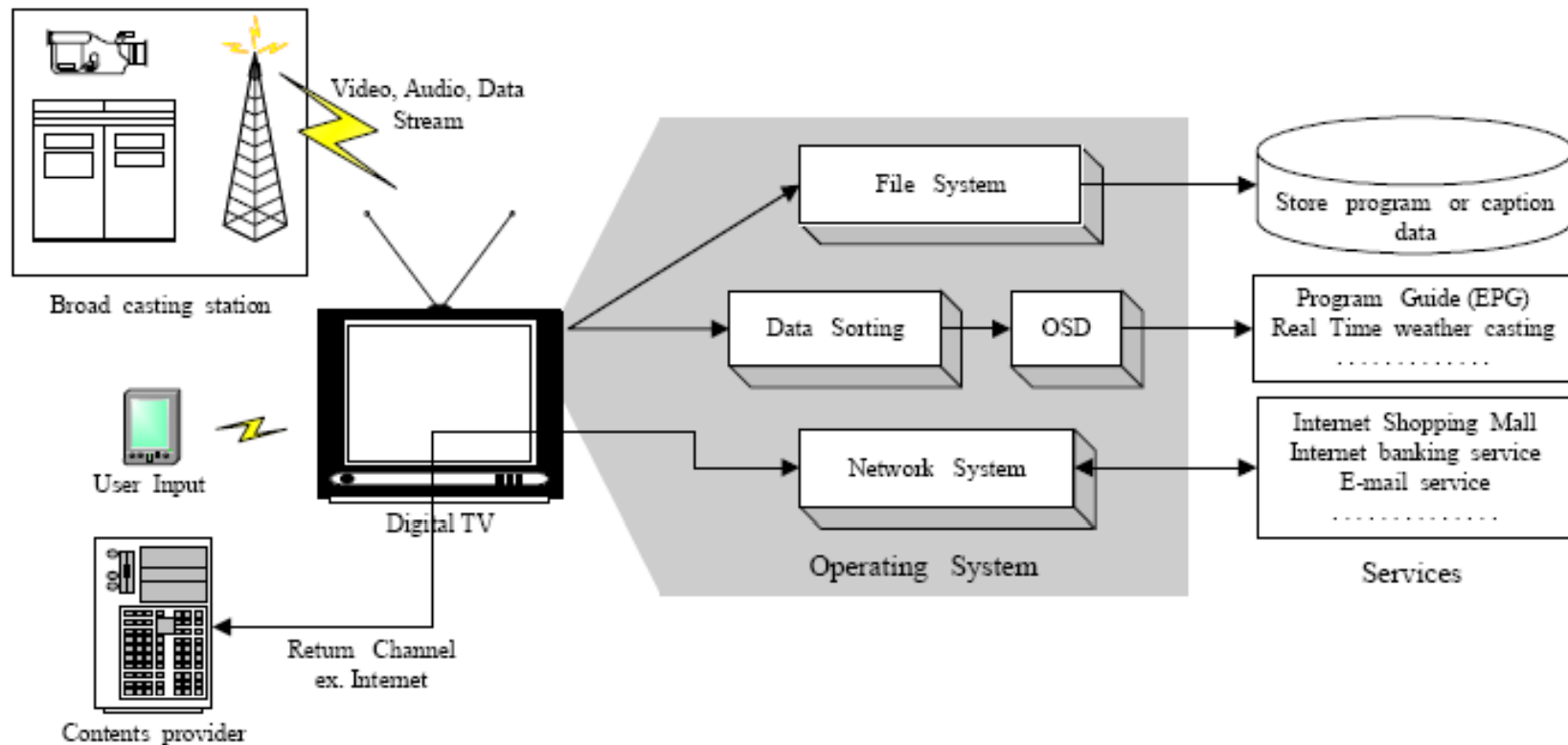
# Digital TV operation system(1/3)



- Digital TV provides higher video and audio qualities than analog TV.
  - Applied AC3 or MPEG2
- Data broadcasting services are program dependent, independent services or interactive services using return channel.
- That is, the data to be captured from the digital input stream need to be sorted, stored.
- Modified for screen output, and on-air TV program would be stored in an embedded hard disk.
- A digital broadcasting signal received by a TV antenna is a stream.
  - Video, audio, and data
  - It is a demodulated by a RF module
  - ATSC or DVB



# Digital TV operation system(2/3)



**Fig. 1 role of digital TV operating system**

# Digital TV operation system(3/3)



- Applying an embedded Linux into a digital TV system.
  - Reusability of open source device drivers and application programs.
  - Convenient development environment configuration using shell, file system, and the module function.
  - Fast problem resolution for problems through open source community.

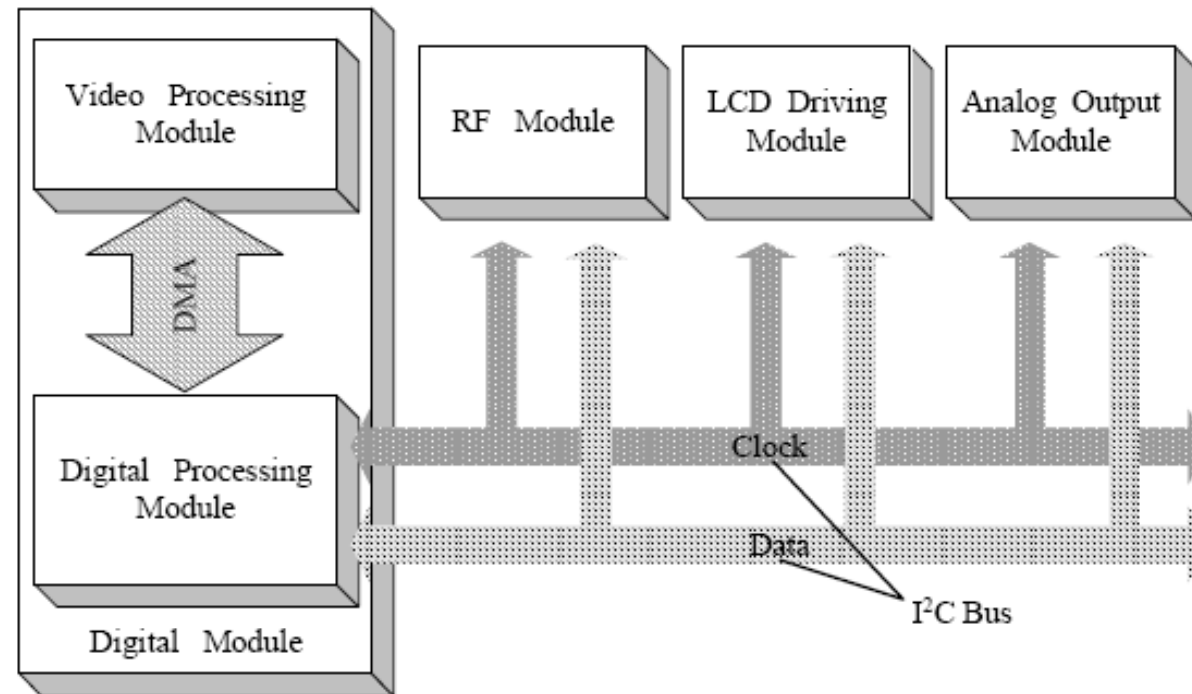




# Commercial digital TV system(1/7)



- *A. Structure of digital TV*



**Fig. 2 Digital TV system module structure**

# Commercial digital TV system(2/7)



- ***B. Hardware components***
  - CPU structure and function
  - Endianess
  - Memory map
  - Interrupt controller
  - Timer
  - Serial bus interface
  - External bus interface

# Commercial digital TV system(3/7)

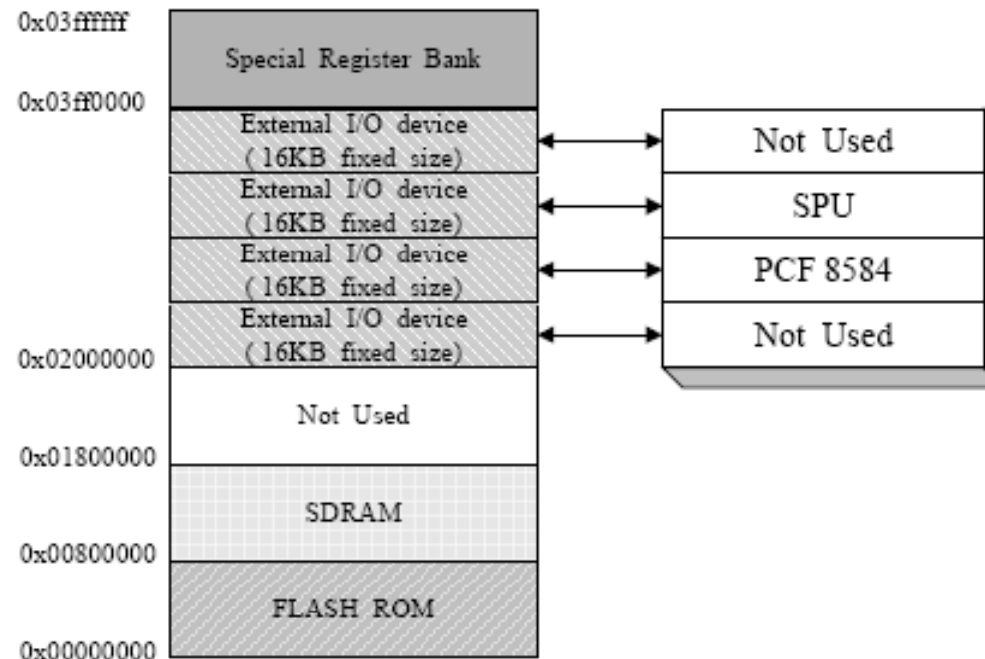


- (1) CPU structure and function
  - ARM7TDMI
  - 16/32bit RISC
  - 31 system registers and 6 status registers
  - 6 operating modes
  - 5 exception events
    - FIQ
    - IRQ
    - Abort
    - Software interrupt
    - Undefined instruction
- (2) Endianess
  - The MCU (Main Control Unit) which is used in digital TV systems supports only the big-endian memory access.

# Commercial digital TV system(4/7)



- (3)Memory map



**Fig. 3 System Memory Map**

# Commercial digital TV system(5/7)



- (4)Interrupt controller
  - The interrupt controller of the MCU includes 21 interrupt resources (17 from integrated internal devices, 4 from external devices).
  - It sends the interrupt request within an interrupt pending register to the CPU.
  - It switches the operating mode to IRQ or FIQ modes.
  - Into a proper service routine.
- (5)Timer
  - The timer of the digital processing module is a 32bit interval mode timer.

# Commercial digital TV system(6/7)



- (6)Serial bus interface

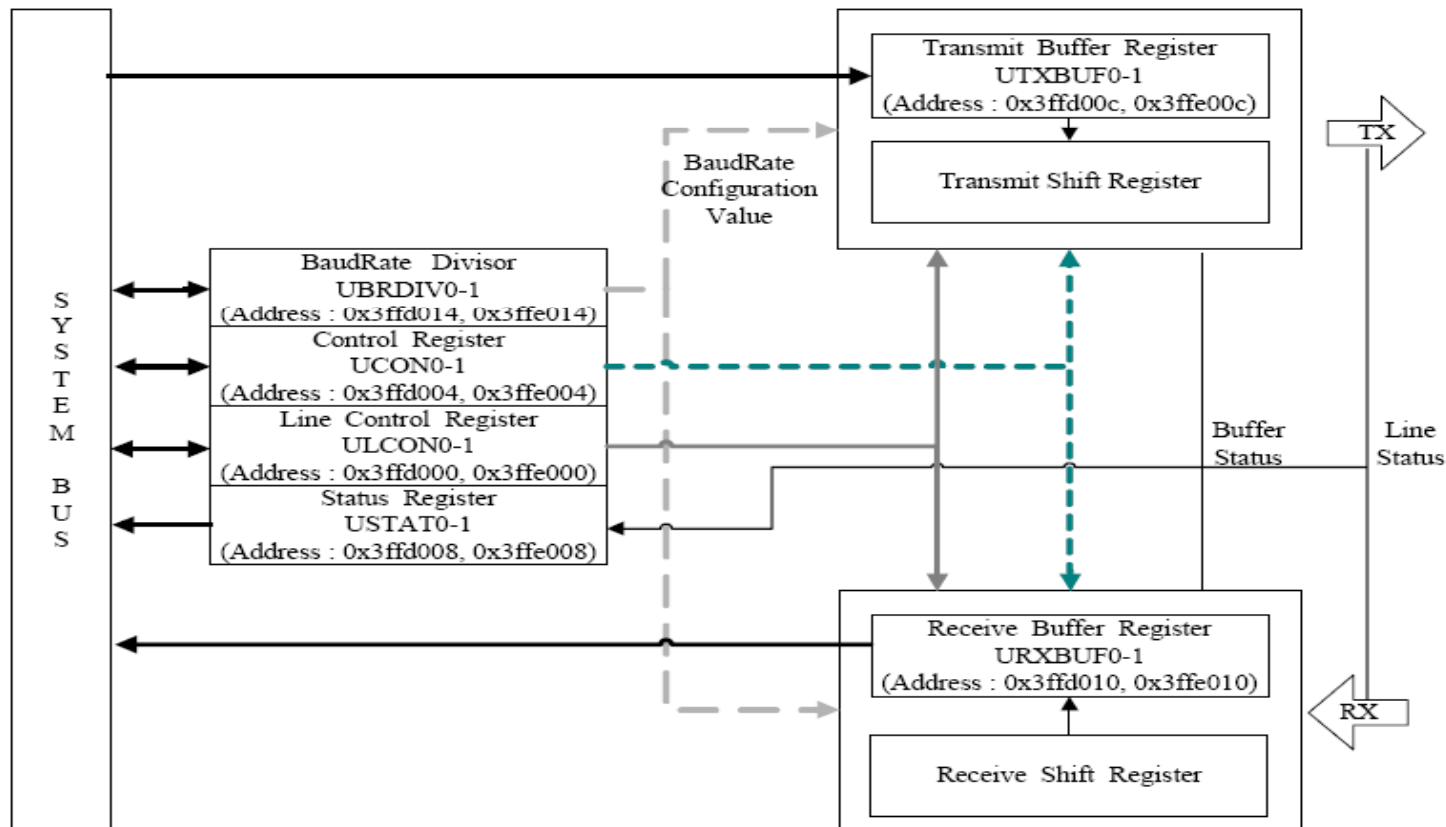


Fig. 4 UART structure and control block diagram

# Commercial digital TV system(7/7)



- (7)External bus interface

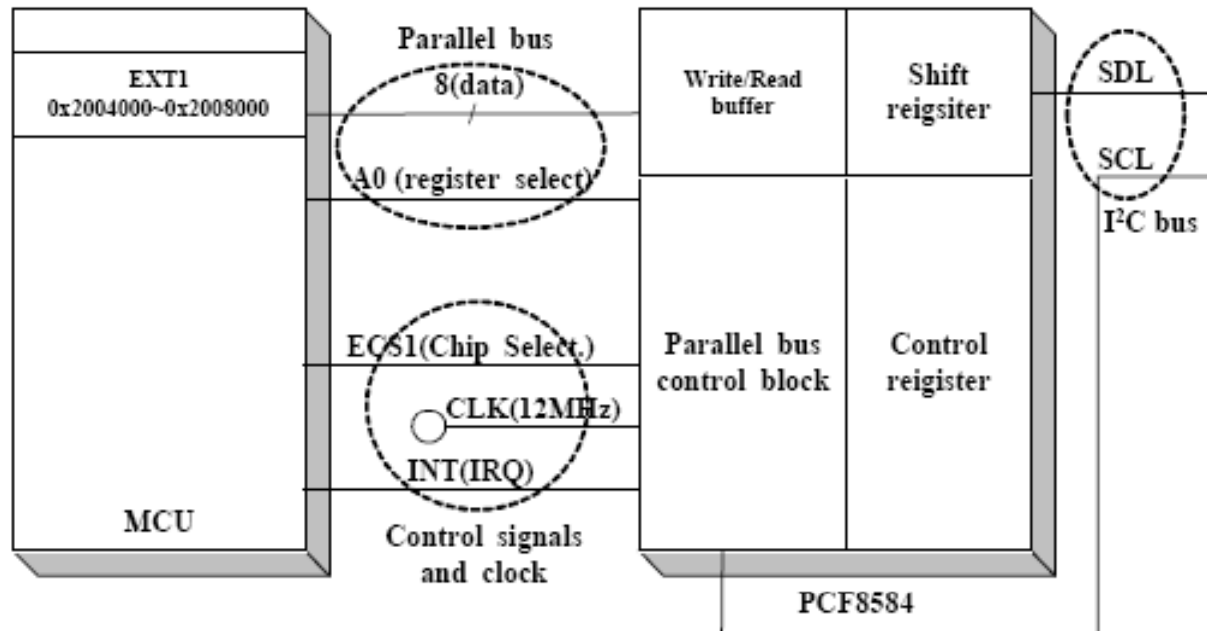


Fig. 5 Connection between MCU and I<sup>2</sup>C bus

# Linux porting(1/7)



- We will explain the sequence of porting Linux with respect to the analysis of the component.
- Modification of the kernel initialization codes and the creation of the device driver.
  - A. Rebuilding of kernel initialization code
  - B. Device driver design



# Linux porting(2/7)



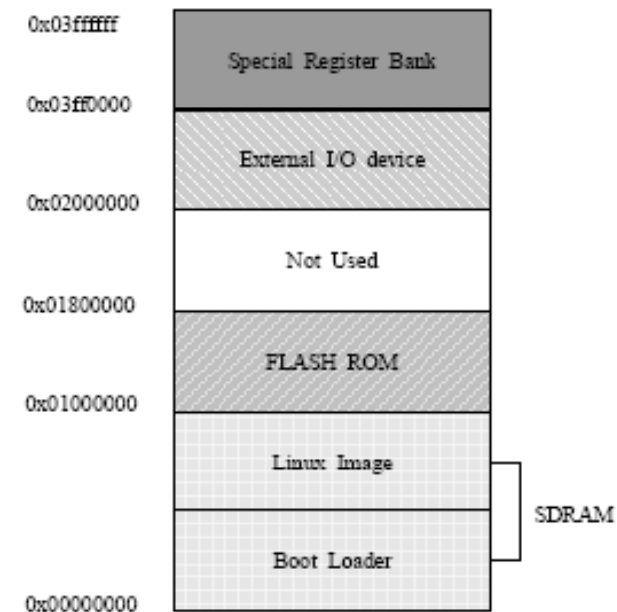
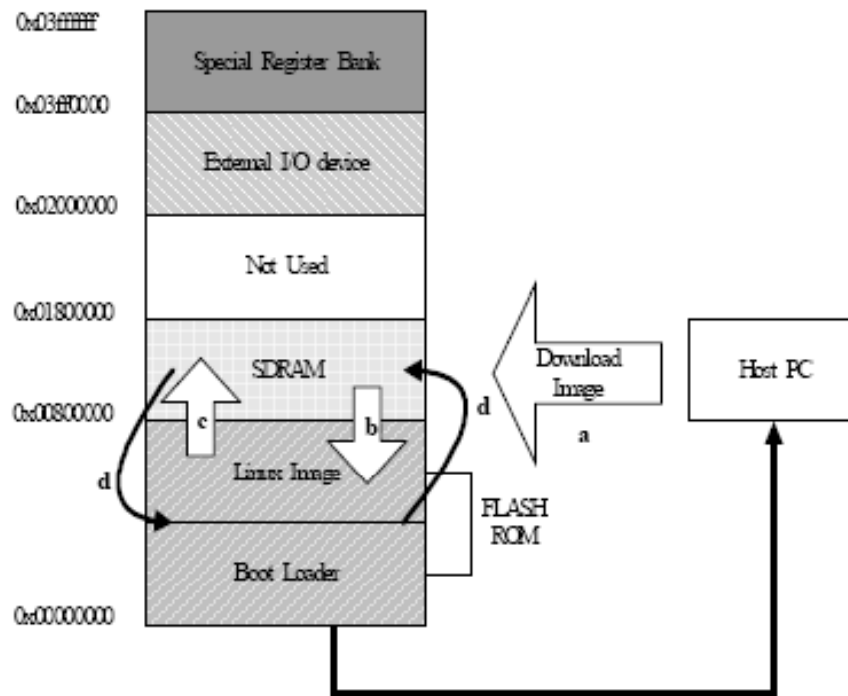
- ***A. Rebuilding of kernel initialization code***
- (1) Setting the memory map
  - The Linux image to be zipped is contained in a flash memory, moved to a SDRAM and unzipped while in execution.
  - The unzipped kernel codes are relocated into the beginning address of the SDRAM to remove the codes for the zipped Linux image.
  - When the system resets, the CPU recognizes only the flash ROM area and executes a boot loader at the bottom of flash rom.

# Linux porting(3/7)



- The boot loader copies the Linux image of a host PC into a SDRAM in the digital processing module.
- The downloaded Linux image in the SDRAM is stored in a flash rom above the boot loader.
- The system is initialized and the boot loader calls an execution code (*head.o*) included in the Linux image.
- The CPU exchanges the beginning address of the flash ROM and with that of the SDRAM.

# Linux porting(4/7)



**Fig. 6 Kernel image loading sequence**    **Fig. 7 Final memory map**

# Linux porting(5/7)

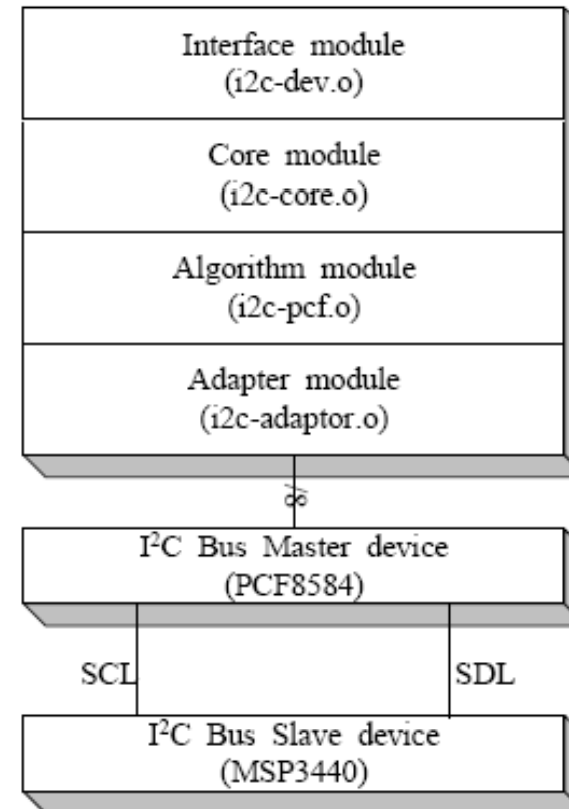


- (2)Essential devices for the kernel
  - interrupt controller
  - timer
  - console device

# Linux porting(6/7)



- ***B. Device driver design***
- There are a video processing device and an analog output device as well as essential devices for the kernel in a digital TV system.
- In this paper, we designed and implemented a device driver for the I2C bus interface (PCF8584).
  - Interface module
  - Core module
  - Algorithm module
  - Adaptor module



**Fig. 8 Structure of device driver**

# Linux porting(7/7)



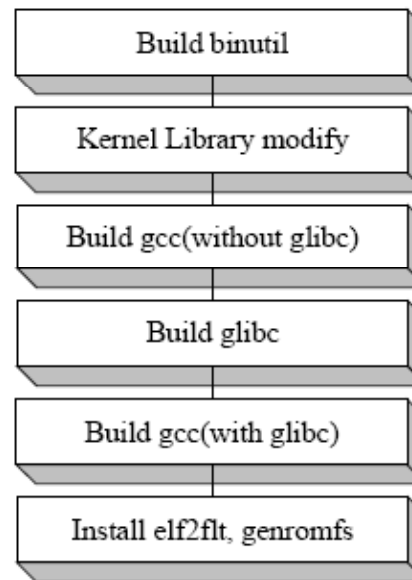
- In order to drive the PCF8584 interface device, module initializing are required.
- In general Linux systems, device drivers are included and initialized at run time.
- Control data are transferred sequentially beginning from the interface, core, algorithm, and finally to the adaptor modules.



# Implementation(1/3)



- For Linux porting, we installed a cross development environment on the host PC, set a test environment using the test board which has similar specification, and finally, applied it to the digital TV system.
- ***A. Installation of cross development environment***



**Fig. 9 Procedure for cross development environment**

# Implementation(2/3)



- *B. Development environment*

TABLE 1 DEVELOPMENT ENVIRONMENT

	Composition	Function
<b>SNDS100 Test Board</b>	Samsung s3c4510B (ARM 7TDMI), 16MByte SDRAM, 1MByte flash ROM, status LED, RJ45 Connector.	Network application development board.
<b>Digital TV Board-LG electronics</b>	Samsung s3c4500(ARM 7TDMI), 16MByte SDRAM, 8MByte flash ROM, pcf8584, SPU (LG electronics), 32MByte SDRAM.	Digital processing module.
<b>JEENI™</b>	EPI, Two hardware breakpoints, Ethernet host interface, Internal ARM 710A cached processor.	Remote debugging environment.
<b>Software</b>	gcc-2.95.3, glibc-2.1.3, binutils-2.12, genromfs-0.5.1, elf2flt, gdb-5.0, ADS(ARM™ Developer Suite) evaluation ver. 1.1.	Tools for cross development and debugging.



# Implementation(3/3)



**Fig. 10 Development environment**

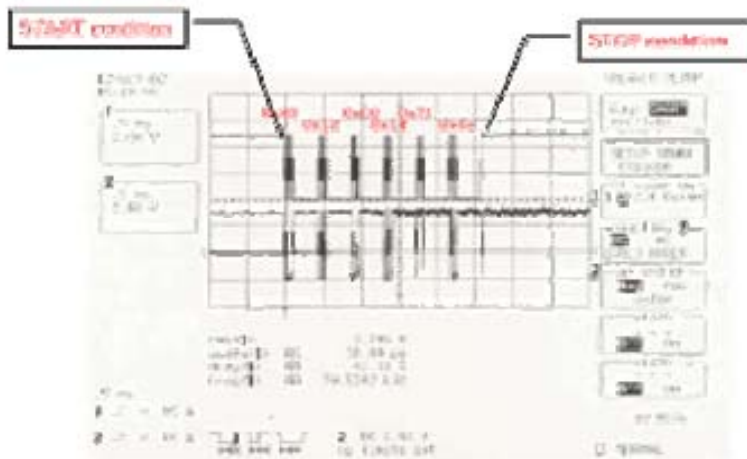


**Fig. 11 Development board**

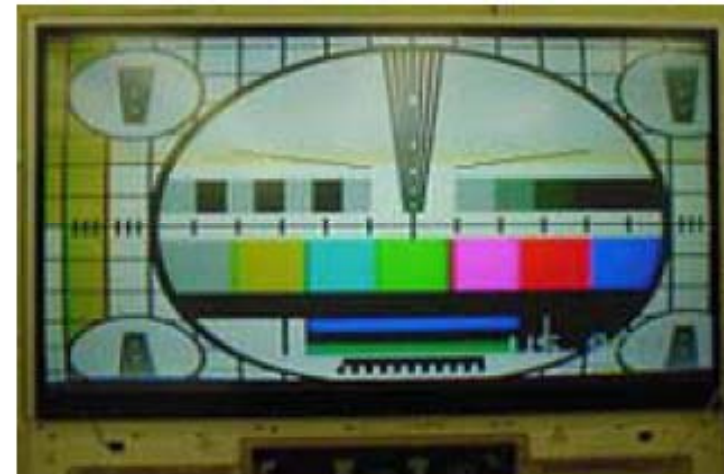
# Experimental result



- As a result, we controlled an audio output IC through the PCF8584.
- That is, which converts digital data to audio output signals.
- We could hear a beep sound and checks I2C bus signal using a digital oscilloscope.



**Fig. 12 Entire data output**



**Fig. 13 Digital TV output**

# Conclusion



- General digital TV systems have used expensive commercial operating systems, which have advantages of code optimization and reliability.
- If we apply an embedded Linux instead of a commercial operating system, we are able to reduce the cost of the products since Linux is an open-source program, and to provide a more convenient development environment.
- In this paper, a digital TV system which is currently in the market and is able to receive Korean type digital broadcasting programs is used.

# References



- [1] Korea Broadcasting System 2001th annual report-data broadcasting  
<http://tri.kbs.co.kr/pdf/publish/01/6.pdf>.
- [2] *Digital Video Broadcasting (DVB): Specification for Service Information (SI) in DVB systems, ETSI EN 300 468 V1.5.1.*
- [3] *ATSC Digital Television Standard, Rev.B, ATSC Standard A/53B with Amendment 1.*
- [4] Steve Furber, Stephen B., *ARM system-on-chip architecture*, Addison-Wesley, 2000.
- [5] *s3c4510B user manual* ,[http://www.samsung.com/Products/Semiconductor/SystemLSI/Networks/Personal/NTASSP/CommunicationProcessor/S3C4510B/um\\_s3c4510b\\_rev1.pdf](http://www.samsung.com/Products/Semiconductor/SystemLSI/Networks/Personal/NTASSP/CommunicationProcessor/S3C4510B/um_s3c4510b_rev1.pdf)
- [6] *I2C bus specification*, <http://www.semiconductors.philips.com/acrobat/literature/9398/39340011.pdf>.
- [7] *PCF8584 specification*, [http://www.semiconductors.philips.com/acrobat/datasheets/PCF8584\\_4.pdf](http://www.semiconductors.philips.com/acrobat/datasheets/PCF8584_4.pdf).

