

Embedded Systems

Ch 3B

Linux Development Environment



Byung Kook Kim

Dept of EECS

Korea Advanced Institute of Science and Technology

Overview

- 1. Embedded Linux
- 2. Cross-Development System
- 3. Setting Linux Development Environment
- 4. **개발 시스템 환경 구축**
- 5. Linux Basics
- 6. Cross Compiler **구축**
- 7. Cross Compiling
- 8. Host Network Environment **환경 설정**
- 9. PXA255 Product Brief

6. Cross Compiler 구축

- Cross compiler
 - Compiler for non-native target
 - Target: EZ-X5 Board with Xscale and Embedded Linux
 - Host: PC with Linux
- Where to get?
 - EZ-X5 CD
 - <http://www.falinux.com>
- Files (in RPM format)
 - armv5l-linux-binutils-2.13.90.0.16-ez1.i386.rpm
 - Assembler, loader, etc.
 - armv5l-linux-gcc-3.2.1-ez1.i386.rpm
 - Compiler
 - armv5l-linux-glibc-2.3.1-ez1.i386.rpm
 - Library

Cross Compiler (II)

■ RPM

- Redhat Package Manager
- Program의 설치를 간략하게 해주는 기술

■ 장점

- 누구나 사용할 수 있는 개방된 package system
- 배포, 설치, 갱신, 삭제할 때 편리
- 여러 개의 file과 program을 묶어 하나의 package로 만듦으로 관리가 편하다
- 각 package에는 어떤 file이 어떤 directory에 속하는지에 대한 정보가 기록된다.
- 각 package를 이용하기 위해서 어떤 file이 필요한지에 대한 정보도 포함된다.

Cross Compiler (III)

■ RPM Usage

- # rpm -qa | more ; **설치된 RPM package 목록 보기**
 - bzip2-libs-1.0.2-6
 - MAKEDEV-3.3.2-5
 - ...
- # rpm -qi ftp ; ftp package에 대한 구체적 정보 표시
- # rpm -ql ftp ; Package에 포함된 file list
- # rpm -qf ls ; 실행 binary file이 어떤 package에 속해있는지 조사
- # rpm -Uvh ---.rpm ; Package 설치
- # rpm -V ftp ; Package가 제대로 설치된 것인지 알려줌
- # rpm -Va ; 시스템에 설치된 모든 package를 검사
- # rpm -e ftp ; Package를 삭제
- # rpm -help ; Display help message
- # rpm -showrc ; RPM 설정환경 표시

Cross Compiler (IV)

■ Cross Compiler 설치

■ 1. Software cdrom을 disk로 복사

- # mount /dev/cdrom /mnt/cdrom
- # cd /mnt/cdrom
- # mkdir some_directory/ezx5
- # cp -R /mnt/cdrom/* some_directory/ezx5
; e.g., /home/cdrom/ezx5

■ 2. Compiler directory 설정 및 rpm file 복사

- # mkdir /home/ezx5/cross_compiler
- # cd /home/ezx5/cross_compiler
- # cp /home/cdrom/ezx5/cross_compiler/rmp-redhat8.0/* .
- # ls
- armv5l-linux-binutils-2.13.90.0.16-ez1.i386.rpm
- armv5l-linux-gcc-3.2.1-ez1.i386.rpm
- armv5l-linux-glibc-2.3.1-ez1.i386.rpm

Cross Compiler (V)

■ Cross Compiler 설치 (II)

■ 3. RPM 설치

- # rpm -Uvh --nodeps --force armv5l-linux-binutils-2.13.90.0.16-ez1.i386.rpm
- # rpm -Uvh --nodeps --force armv5l-linux-gcc-3.2.1-ez1.i386.rpm
- # rpm -Uvh --nodeps --force armv5l-linux-glibc-2.3.1-ez1.i386.rpm
- /usr/armv5l-linux directory **밑에** cross-compiler가 설치된다.

■ 4. 환경설정 (in /root/.bash_profile)

- # cd /root
- # vi .bash_profile
- Include: PATH=...:\$PATH:/usr/armv5l-linux/bin:...
- # logout and login

OR

- # PATH=\$PATH:/usr/armv5l-linux/bin
- # export

7. Cross Compiling

- Ex) Compile and run the Problem 1.1.
 - *Use the PC cross-development system*
 - Edit the program "hello.c" using vi editor
 - # mkdir embeddlab
 - # cd embeddedlab
 - # mkdir lab1
 - # cd lab1
 - # vi hello.c
 - Cross-compile and link the program to produce hello.
 - # /usr/armv5l-linux/bin/gcc -o hello.x hello.c

Cross Compiling (II)

- Ex) cont'd
 - *Use the minterm on the PC , and Boot the EZ-X5 to run Linux*
 - Download using zmodem protocol
 - \$ rz
 - Press Control+a s
 - Select zmodem
 - Select file to download as "hello.x"
 - Check that the file is already downloaded.
 - \$ ls -la
 - hello.x
 - Run hello, which should print the following on the console:
 - \$./hello.x
 - Hello, Embedded board!

8. Host Network Environment

8.1 TFTP Environment

■ TFTP 필요성

- 다운 받아야 할 내용이 매우 큰 램 디스크 이미지를 시리얼로 다운 받게 되면 개발속도가 현저히 떨어진다.
- LAN을 이용하여 파일을 EZ-X5에 download하는 방법으로 이지부트와 TFTP를 이용하면 매우 빠른 속도로 다운 받기 때문에 개발속도가 향상될 수 있다.

■ TFTP (Trivial File Transfer Protocol) 란

- 이더넷을 이용하여 파일을 다운 받는 프로토콜로서 UDP 방식을 사용한다.
- TFTP은 ftp와 같은 파일 전송 프로토콜이지만, 매우 간단한 프로토콜로 구성되므로 부트로더와 같은 작은 크기의 프로그램에서 수행할 수가 있다.

TFTP (II)

■ TFTP의 동작

- 클라이언트가 서버에 접속하여 파일을 요청
- 서버는 해당 파일을 전송하고 이를 클라이언트는 응답하는 방식을 취한다.
- 매번 파일의 블록을 전송 시 블록의 길이를 전송하는데 이 블록의 크기는 고정되어 있어 이 크기보다 작은 블록이 오면 이를 마지막 블록으로 인식한다.

■ 단점

- 하나의 블록 전송이라도 실패하면 전송을 중지하므로 패킷 손실이 많은 환경에서는 사용이 곤란하다.

TFTP (III)

- Setting the environment of tftp

- # vi /etc/xinetd.d/tftp

- Service tftp

- {

- disable = **no**

- socket_type = dgram

- protocol = udp

- wait = yes

- user = root

- server = /usr/sbin/in.tftpd

- server_args = -s /tftpboot

- }

- Restarting xinetd

- # ./xinetd stop

- # ./xinetd start

- # netstat -al | grep tftp

; Check if tftp daemon is started

TFTP (IV)

- Test tftp
 - Make directory in the host
 - # make /tftpboot
 - Copy files into tftpboot directory
 - # cp -a /mnt/cdrom/sw/image/ezboot /tftpboot
 - # cp -a /mnt/cdrom/sw/image/zImage.x5 /tftpboot
 - # cp -a /mnt/cdrom/sw/image/ramdisk.x5.gz /tftpboot
 - In the target, use EZBOOT
 - EZBOOT> tfb ; Tftp file transfer of EZBOOT
 - EZBOOT> tfk ; Tftp file transfer of zImage.x5
 - EZBOOT> tfr ; Tftp file transfer of ramdisk.x5.gz

8.2 NFS Environment

■ NFS 필요성

- 개발 호스트에서 작업한 것을 NFS 시스템을 이용하여 Ez-X5의 리눅스 상에서 마운트 시켜 사용하면 download 할 필요가 없다.
- 개발 호스트상의 파일이 EZ-X5 Embedded Linux File System 위에서 접근이 가능하고 실행이 가능하다,
- 램 디스크상에서 올리기에 너무 큰 파일도 NFS 상에서는 호스트의 기억용량에 의존하기 때문에 쉽게 처리할 수 있다.

■ NFS (Network File System)란

- RPC를 이용하여 remote host 상의 file을 user가 마치 local file에 access 하듯이 할 수 있게 한다.
- Remote host file system을 사용하는 client 측의 kernel 기능과 server 측의 NFS server 의 혼합으로 가능한 것이다.
- NFS의 file access는 다양한 server와 host architecture에서 동작한다.

NFS (II)

■ 이점

- 개발 호스트에서 개발된 내용을 EZ-X5 에서 바로 사용이 가능
- 여러 개발자가 하나의 PC에 개발된 내용을 동시에 공유가 가능
- 개발된 어플리케이션의 크기에 관계 없이 수행이 가능

■ 주의 점

- 특수 파일은 NFS에 연결된 directory에 만들 수 없다 (예: 장치 파일)
- 읽고 쓰는 속도가 빠른 파일로는 사용이 곤란하다 (예: 멀티미디어 파일)

■ More info

- Rick Sladkey: NFS 커널 소스와 NFS 서버의 많은 부분을 썼는데, 후자는 원래 Mark Shand가 쓴 unfsd user-space NFS 서버와 Donald Becker가 쓴 hnfs Harris NFS서버에서 유래한 것이다.
- <http://kldp.org/Translations/html/NAG/11.html>

NFS (III)

■ 환경 file 편집

- # mkdir nfs
- # chmod 777 nfs
- # chmod nobody nfs
- # chgrp nobody nfs

- # vi /etc/exports
 - /nfs localhost(rw,insecure)
 - /nfs 192.168.10.0/24(rw,insecure)
; 192.168.10.x 모두 허용

■ Running NFS daemon

- # /etc/rc.d/init.d/nfs stop
- # /etc/rc.d/init.d/nfs start

NFS (IV)

- Mounting NFS

- # mkdir /tmp/nfs
- # mount -n localhost:/nfs /tmp/nfs

- Connecting NFS server in EZ-X5

- \$ mkdir /mnt/nfs
- \$ mount -t nfs -o nolock 192.168.10.1:/nfs /mnt/nfs



9. PXA255 Product Brief

Intel® PXA255 Processor with Intel® XScale™ Technology

Continuing the advance in handheld multimedia functionality.

Product Highlights

High Performance:

- Low-power, high-performance 32-bit Intel® XScale™ core-based CPU (200, 300 and 400 MHz)
- ARM* Architecture v.5TE compliant
- Intel® Superpipelined RISC Technology utilizing advanced Intel® 0.18μ process for high core speeds at low power
- Intel® Media Processing Technology including 40-bit accumulator and 16-bit SIMD to enhance audio/video decode performance
- High-performance glue-less burst and page mode interfaces with Synchronous Intel StrataFlash® Memory



PXA255 Product Brief (II)

Product Highlights (cont'd)

Low Power:

- Low-Power and Turbo modes enables enhanced battery life
- 32 KB data and 32 KB instruction caches
- 2 KB Mini data cache for streaming data
- Support for 2.5 and 3.3V memories.

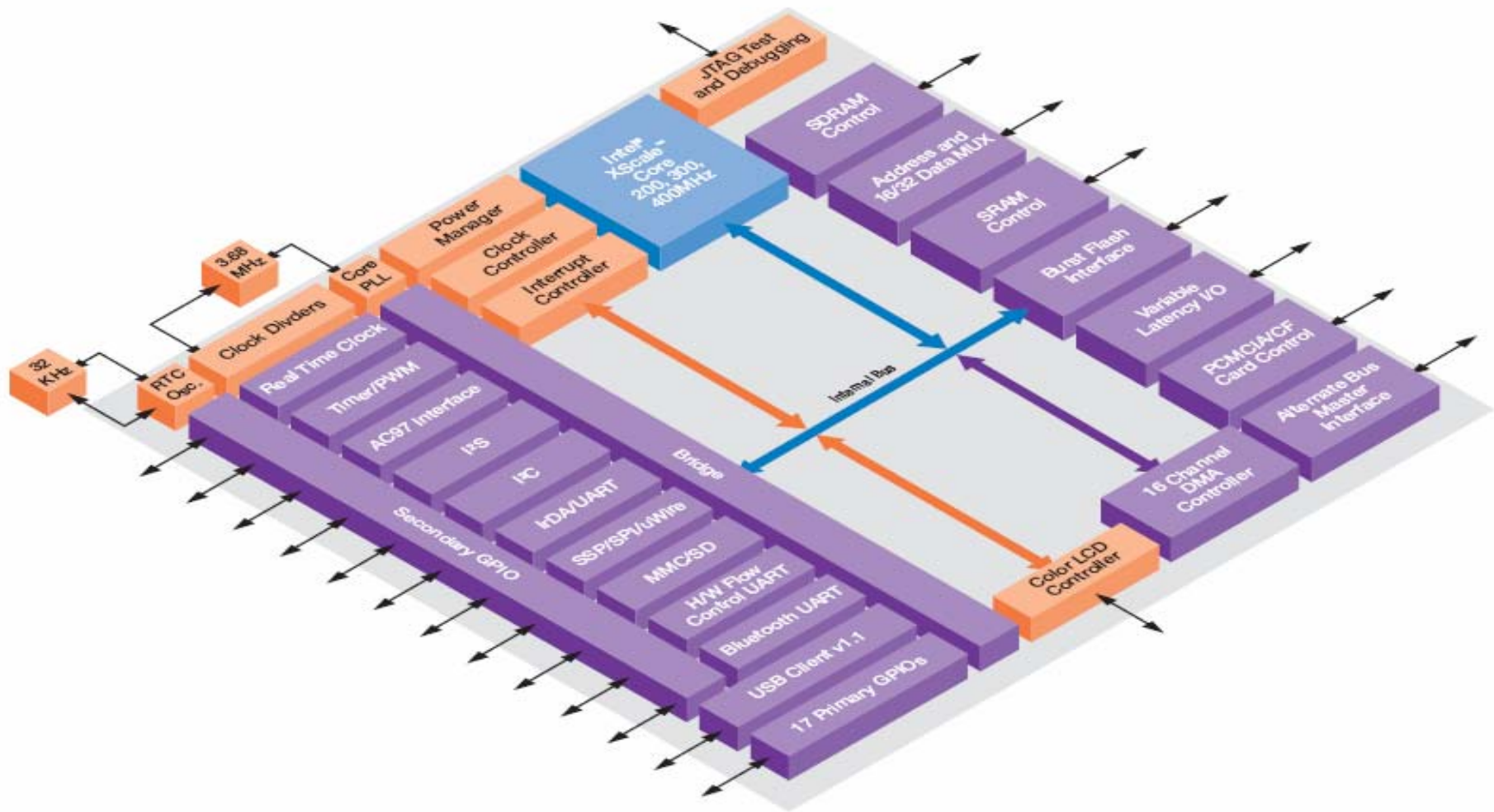
I/O Expansion:

- Integrated Memory and PCMCIA/Compact Flash Controller with 100 MHz Memory Bus, 16-bit or 32-bit ROM/Flash/SRAM (six banks), 16-bit or 32-bit SDRAM, SMROM (four banks), as well as PCMCIA and Compact Flash for added functionality and expendability
- System Control Module includes 17 dedicated general-purpose interruptible I/O ports, real-time clock, watchdog and interval timers, power management controller, interrupt controller, reset controller, and two on-chip oscillators.

Wireless:

- Peripheral Control Module offers 16 channel configurable DMA controller, integrated LCD controller with unique DMA for fast color screen support, Bluetooth** I/F, serial ports including IrDA, I2C*, I2S*, AC97, three UARTs (1 Full H/W flow control), SPI and enhanced SSP, USB end point interface, and MMC/SD Card Support for expandable memory and I/O functionality
- 17x17 mm 256-pin PBGA
- Extended temperature support.

PXA255 Product Brief (III)



Intel® PXA255 Block Diagram

PXA255 Product Brief (IV)

The Intel® PXA255 Processor Advantage

Feature	Benefit
High-performance, low-power Intel® XScale™ core at 200, 300 and 400 MHz. New power management for low power	Ideal for enabling enhanced battery life and performance for high-performance personal digital assistants and wireless communicators.
Faster internal system bus (At 400 MHz core, internal bus runs at 200 MHz vs. 100 MHz on the Intel® PXA250 processor)	Improved application performance
Intel Media Processing Technology	Optimized audio and video multimedia functionality
Enhanced Memory Controller	Supports low power 2.5V and 3.3V 32-bit and 16-bit memories including enhanced support for low-power SDRAM as well as glue-less burst and page mode interfaces with Synchronous Intel StrataFlash® Memory
MMC/SD and PCMCIA/CF Card support	Expandable storage and I/O device support
USB Client	Fast host synchronization
1.84 MHz cellular baseband interface	Efficient communications integration
920 Kbps Bluetooth** interface	Broad inter-device communication
Variable latency I/O	Add-on functionality capabilities

PXA255 Product Brief (V)

Support Collateral

Item	Description
Developer's Manual	Intel® PXA255 Processor
Design Guide	Intel® PXA255 Processor
User's Manual	Intel® XScale™ Microarchitecture for the Intel® PXA255 Processor
Specifications	Intel® PXA255 Processor Electrical, Mechanical, and Thermal Intel® PXA255 Processor (Update)

Intel Access

Developer Web Site	developer.intel.com
Intel® PCA Processors Home Page	developer.intel.com/design/pca/applicationsprocessors