Ch1. Embedded System Overview
Contents

• What is Embedded System?
• Characteristics
• Application Area
• Conclusion
• Schedule
Embedded systems overview

• Computing systems are everywhere
• Most of us think of “desktop” computers
  – PC’s
  – Laptops
  – Mainframes
  – Servers
• But there’s another type of computing system
  – Far more common...
Embedded Systems Overview

- **Embedded computing systems**
  - Computing systems embedded within electronic devices
  - Hard to define. Nearly any computing system other than a desktop computer
  - Billions of units produced yearly, versus millions of desktop units

Computers are in here...

Lots more of these, though they cost a lot less each.
What is Embedded System?

• Computers inside a Product (or a System)
• Any device that includes a programmable computer but is not itself a general-purpose computer.
• “The system used for specific purpose with harmonized computer hardware and software”
  – H/W : microprocessor/controller, memory, sensor, actuator, etc
  – S/W : OS, device driver, application program, network program, etc
Example Embedded Computing Systems

Embedded System Overview

Slide courtesy of Mani Srivastava®
A “short list” of Embedded Systems

And the list goes on and on ....

<table>
<thead>
<tr>
<th>Embedded System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anti-lock brakes</td>
</tr>
<tr>
<td>Auto-focus cameras</td>
</tr>
<tr>
<td>Automatic teller machines</td>
</tr>
<tr>
<td>Automatic toll systems</td>
</tr>
<tr>
<td>Automatic transmission</td>
</tr>
<tr>
<td>Avionic systems</td>
</tr>
<tr>
<td>Battery chargers</td>
</tr>
<tr>
<td>Camcorders</td>
</tr>
<tr>
<td>Cell phones</td>
</tr>
<tr>
<td>Cell-phone base stations</td>
</tr>
<tr>
<td>Cordless phones</td>
</tr>
<tr>
<td>Cruise control</td>
</tr>
<tr>
<td>Curbside check-in systems</td>
</tr>
<tr>
<td>Digital cameras</td>
</tr>
<tr>
<td>Disk drives</td>
</tr>
<tr>
<td>Electronic card readers</td>
</tr>
<tr>
<td>Electronic instruments</td>
</tr>
<tr>
<td>Electronic toys/games</td>
</tr>
<tr>
<td>Factory control</td>
</tr>
<tr>
<td>Fax machines</td>
</tr>
<tr>
<td>Fingerprint identifiers</td>
</tr>
<tr>
<td>Home security systems</td>
</tr>
<tr>
<td>Life-support systems</td>
</tr>
<tr>
<td>Medical testing systems</td>
</tr>
<tr>
<td>Modems</td>
</tr>
<tr>
<td>MPEG decoders</td>
</tr>
<tr>
<td>Network cards</td>
</tr>
<tr>
<td>Network switches/routers</td>
</tr>
<tr>
<td>On-board navigation</td>
</tr>
<tr>
<td>Pagers</td>
</tr>
<tr>
<td>Photocopiers</td>
</tr>
<tr>
<td>Point-of-sale systems</td>
</tr>
<tr>
<td>Portable video games</td>
</tr>
<tr>
<td>Printers</td>
</tr>
<tr>
<td>Satellite phones</td>
</tr>
<tr>
<td>Scanners</td>
</tr>
<tr>
<td>Smart ovens/dishwashers</td>
</tr>
<tr>
<td>Speech recognizers</td>
</tr>
<tr>
<td>Stereo systems</td>
</tr>
<tr>
<td>Teleconferencing systems</td>
</tr>
<tr>
<td>Televisions</td>
</tr>
<tr>
<td>Temperature controllers</td>
</tr>
<tr>
<td>Theft tracking systems</td>
</tr>
<tr>
<td>TV set-top boxes</td>
</tr>
<tr>
<td>VCR’s, DVD players</td>
</tr>
<tr>
<td>Video game consoles</td>
</tr>
<tr>
<td>Video phones</td>
</tr>
<tr>
<td>Washers and dryers</td>
</tr>
</tbody>
</table>
• More than 30% of the cost of a car is now in Electronics
• 90% of all innovations will be based on electronic systems
• Average household uses ~225 embedded processors
• Today’s high-end automobile may have 100 microprocessors:
  – 4-bit microcontroller checks seat belt;
  – microcontrollers run dashboard devices;
  – 16/32-bit microprocessor controls engine
  – Millions lines of code
• Estimated 5 billion embedded process
  – 94% share of world market
  – 6%: Pentium, PowerPC, etc.

[Source: World Semiconductor Trade Statistics Bluebook]
Some Common Characteristics

• Single-functioned
  – Executes a single program, repeatedly

• Tightly-constrained
  – Low cost, low power, small, fast, etc.

• Reactive and real-time
  – Continually reacts to changes in the system’s environment
  – Must compute certain results in real-time without delay
E.g. : A Digital Camera

- Single–functioned -- always a digital camera
- Tightly–constrained -- Low cost, low power, small, fast
- Reactive and real–time -- only to a small extent
E.g. : Mobile Phone

**Hardware (H/W):**
- Micro Processor
- Memory
- Input device
- Network device

**Software (S/W):**
- OS - Kernel
- System software
- Application software

**Embedded System Overview**
E.g.: Mobile Phone Exp. Kit
Cf. Intel PXA 270 Embedded Processor

- High-performance processor:
  - Intel XScale® microarchitecture with Intel® Wireless MMX™ Technology
  - 7 Stage pipeline
  - 32 KB instruction cache
  - 32 KB data cache
  - 2 KB “mini” data cache
  - Extensive data buffering
- 256 Kbytes of internal SRAM for high speed code or data storage preserved during low-power states
- High-speed baseband processor interface (Mobile Scalable Link)
- Rich serial peripheral set:
  - AC'97 audio port
  - I²S audio port
  - USB Client controller
  - USB Host controller
  - USB On-The-Go controller
  - Three high-speed UARTs (two with hardware flow control)
  - FIR and SIR infrared communications port
- Hardware debug features — IEEE 1149.1 TAP interface with boundary scan
- Hardware performance-monitoring features with on-chip trace buffer
- Real-time clock
- Operating-system timers
- LCD Controller
- Universal Subscriber Identity Module interface
- Low power:
  - Wireless Intel Speedstep® Technology
  - Less than 500 mW typical internal dissipation
  - Supply voltage may be reduced to 0.85 V
  - Four low-power modes
  - Dynamic voltage and frequency management
- High-performance memory controller:
  - Four banks of SDRAM: up to 104 MHz @ 2.5V, 3.0V, and 3.3V I/O interface
  - Six static chip selects
  - Support for PCMCIA and Compact Flash
  - Companion chip interface
- Flexible clocking:
  - CPU clock from 104 to 624 MHz
  - Flexible memory clock ratios
  - Frequency changes
  - Functional clock gating
- Additional peripherals for system connectivity:
  - SD Card / MMC Controller (with SPI mode support)
  - Memory Stick card controller
  - Three SSP controllers
  - Two I²C controllers
  - Four pulse-width modulators (PWMs)
  - Keypad interface with both direct and matrix keys support
  - Most peripheral pins double as GPIOs
# PXA270 Board Specification

<table>
<thead>
<tr>
<th>Category</th>
<th>Item</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Processor</td>
<td>Intel PXA270 520MHz</td>
</tr>
<tr>
<td>Memory</td>
<td>SDRAM</td>
<td>64Mbytes Mobile-SDRAM &lt;br&gt;Samsung K4S513233F</td>
</tr>
<tr>
<td></td>
<td>NOR type NAND</td>
<td>128Mbytes M-System DiskOnChip &lt;br&gt;M-Systems MD4331-DIG-V3Q18-X</td>
</tr>
<tr>
<td>Display</td>
<td>LCD</td>
<td>2.2” 65K Colors TFT LCD, 320x240 QVGA, Touch Screen &lt;br&gt;Controlled by DMB module &lt;br&gt;Sharp LS024Q8DD92</td>
</tr>
<tr>
<td>TV</td>
<td>TV Output</td>
<td>NTSC/PAL Encoder &lt;br&gt;Input: PXA270 LCD interface (16bit RGB565) &lt;br&gt;Chrontel CH7013B</td>
</tr>
<tr>
<td>DMB</td>
<td>T-DMB</td>
<td>DMB Module (T-DMB Baseband + Codec + RF) &lt;br&gt;Controll LCD module, I2S AUDIO interface &lt;br&gt;Nexilion NX3300</td>
</tr>
<tr>
<td>Camera</td>
<td>Mobile Camera</td>
<td>2M pixels CMOS Sensor &lt;br&gt;Output : RGB 565 or YUV422 interface &lt;br&gt;PixelPlus PO1200NC-MOR-PAF02-050705</td>
</tr>
<tr>
<td>Audio</td>
<td>CODEC</td>
<td>AC’97 Stereo Codec, ADPCM interface for Bluetooth &lt;br&gt;Wolfson WM9714</td>
</tr>
<tr>
<td></td>
<td>Speaker/Mic</td>
<td>Two Stereo Speakers, One Microphone, Hand-free interface</td>
</tr>
<tr>
<td>Wireless Network</td>
<td>CDMA</td>
<td>800MHz CDMA 1x Module (MSM6025) with UART interface &lt;br&gt;Fidelix FD810</td>
</tr>
<tr>
<td></td>
<td>Bluetooth</td>
<td>Spec. 1.1, ADPCM Interface, AT Command Control &lt;br&gt;CSR Bluecore3 GATEWAY CSP &lt;br&gt;Insung Electronics BM400G</td>
</tr>
<tr>
<td></td>
<td>WLAN</td>
<td>IEEE 802.11b (CSMA/CA), 11Mbps, PCMCIA Card Interface &lt;br&gt;Samsung SWL2450</td>
</tr>
</tbody>
</table>
e. g. X-Hyper320TKU
### HW specification

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>processor (Marvell PXA320-P (Core: 806MHz))</td>
</tr>
<tr>
<td>SDRAM</td>
<td>266MHz DDR SDRAM 128MByte</td>
</tr>
<tr>
<td>Nand Flash</td>
<td>NAND Flash 128MByte</td>
</tr>
<tr>
<td>PMIC</td>
<td>MAX8660</td>
</tr>
<tr>
<td>CPLD</td>
<td>Xilinx CoolRunner-II</td>
</tr>
<tr>
<td>Display</td>
<td>800 X 480 (7 inch Wide TFT Color LCD)</td>
</tr>
<tr>
<td>Touch Screen</td>
<td>Touch Screen Controller</td>
</tr>
<tr>
<td>DMB</td>
<td>DMB 모듈 (그림 CIF I/F)</td>
</tr>
<tr>
<td>Camera</td>
<td>CMOS Camera 1.3M Pixel</td>
</tr>
<tr>
<td>GPS</td>
<td>GPS Module</td>
</tr>
<tr>
<td>Ethernet</td>
<td>10/100 Mbps Ethernet - 2 Port</td>
</tr>
<tr>
<td>Audio</td>
<td>Audio-AC’97(WM9712)</td>
</tr>
<tr>
<td>USB</td>
<td>USB HOST/Client 1.1 1Port (Internal)</td>
</tr>
<tr>
<td></td>
<td>USB HOST 2.0 2Port (External)/ Client 2.0 1port (External)</td>
</tr>
<tr>
<td>UART</td>
<td>Debug(RS232), GPS (TTL), FFFUART (RS232) 1port (3EA)</td>
</tr>
<tr>
<td></td>
<td>Bluetooth(RS232), IrDA(TTL), RS232, TTL 1port (4EA)</td>
</tr>
<tr>
<td>IDE</td>
<td>mini IDE Interface (1.8&quot;)</td>
</tr>
<tr>
<td>RTC</td>
<td>RTC4513</td>
</tr>
<tr>
<td>VGA</td>
<td>VGA Interface (일반 모니터 연결 가능)</td>
</tr>
<tr>
<td>KEY/LED</td>
<td>GPIO 4EA</td>
</tr>
<tr>
<td>MMC/SD</td>
<td>MMC/SD 1 Slot</td>
</tr>
<tr>
<td>Speaker</td>
<td>내장 Speaker L/R</td>
</tr>
<tr>
<td>Mic, Speaker</td>
<td>Mic, Speaker 1 Jack</td>
</tr>
<tr>
<td>JTAG</td>
<td>JTAG - emulator용 20 pin</td>
</tr>
</tbody>
</table>

### SW specification

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>O/S</td>
<td>Embedded Linux Kernel 2.6.x / WINCE 5.0</td>
</tr>
<tr>
<td>Compiler</td>
<td>PlatromBuilder / GNU Tool for ARM</td>
</tr>
<tr>
<td>JTAG</td>
<td>XDB OPTION</td>
</tr>
<tr>
<td>Bootloader</td>
<td>EBOOT / hybus-boot320</td>
</tr>
<tr>
<td>FileSystem</td>
<td>JFFS2, YAFFS, EXT2/3, FAT</td>
</tr>
<tr>
<td>GUI</td>
<td>WINCE 5.0 / TinyX for X-Hyper320 &amp; QT/Embedded</td>
</tr>
<tr>
<td>Ethernet Driver</td>
<td>Ethernet Device Driver (100 BaseT/Wireless)</td>
</tr>
<tr>
<td>USB Client</td>
<td>USB 1.1 Host/Client Device Driver</td>
</tr>
<tr>
<td>USB Device</td>
<td>USB 2.0 Host/Client Device Driver</td>
</tr>
<tr>
<td>Codec Driver</td>
<td>AC’97 codec Device Driver</td>
</tr>
<tr>
<td>Touch Screen</td>
<td>Touch LCD Device Driver</td>
</tr>
<tr>
<td>Device Driver</td>
<td>RTC4513 Device Driver</td>
</tr>
<tr>
<td>PMIC Device</td>
<td>PMIC Device Driver</td>
</tr>
<tr>
<td>MMC/SD Device</td>
<td>MMC/SD Device Driver</td>
</tr>
<tr>
<td>LCD driver</td>
<td>Char LCD / Dot Matrix Device Driver</td>
</tr>
<tr>
<td>Motor Driver</td>
<td>7 Segment / Stepping Motor Device Driver</td>
</tr>
<tr>
<td>Key Button</td>
<td>Key Button/ADC_DAC Device Driver</td>
</tr>
<tr>
<td>GPIO/LED</td>
<td>GPIO Test Button (입력), LED (출력)</td>
</tr>
<tr>
<td>Camera</td>
<td>CMOS Camera Device Driver</td>
</tr>
<tr>
<td>DMB Device</td>
<td>DMB Device Driver</td>
</tr>
<tr>
<td>VGA Device</td>
<td>VGA Device Driver</td>
</tr>
</tbody>
</table>
Characteristics of Embedded System

- Dedicated to a specific task or tasks
- Rich variety of microprocessors (over 300 types)
- Designs are cost-sensitive
- Have real-time performance constraints
- Used with Real-Time Operating Systems (RTOS)
- **Software is tightly coupled with hardware**
- May have constraints on power consumption
- Operate over a wide-range of environmental conditions
- Fewer system resources than a desktop system
- All code might be stored in ROM
- **Require specialized design tools**
  - E.g: WINCE Platform Builder
- May have on-chip debugging resources
Design Challenge

• Obvious design goal:
  – Construct an implementation with desired functionality

• Key design challenge:
  – Simultaneously optimize numerous design metrics

• Design metric
  – A measurable feature of a system’s implementation
  – *Optimizing design metrics is a key challenge*
Design Challenge

• Common metrics
  – Unit cost: the monetary cost of manufacturing each copy of the system, excluding NRE cost
  – NRE cost (Non-Recurring Engineering cost): The one-time monetary cost of designing the system
  – Size: the physical space required by the system
  – Performance: the execution time or throughput of the system
  – Power: the amount of power consumed by the system
  – Flexibility: the ability to change the functionality of the system without incurring heavy NRE cost
Design Challenge

• Common metrics (continued)
  – Time-to-prototype: the time needed to build a working version of the system
  – Time-to-market: the time required to develop a system to the point that it can be released and sold to customers
  – Maintainability: the ability to modify the system after its initial release
  – Correctness, safety, many more
Design metric competition -- improving one may worsen others

- Expertise with both software and hardware is needed to optimize design metrics
  - Not just a hardware or software expert, as is common
  - A designer must be comfortable with various technologies in order to choose the best for a given application and constraints
Applications Area

• Electronic Home Appliances: internet washer, internet refrigerator, HDTV etc.
• Control: factory automation, home automation, robotics, process control, etc.
• Mobile Hand-held Devices: Mobile phone, PDA, Smart phone, etc.
• Network Devices: Switch Board, Router, IP sharer, home gateway, etc.
• Game: PS2, XBox, intelligent toys, etc.
• Space/Military: plane, space-craft, rocket, GPS & GIS devices
• Finance: ATM terminal, RFID, POS terminal, etc.
• Car/Traffic: car, ITS (Intelligent Transport System), etc.
• Office, medical instruments: phone, printer, heart pacer, operation robot
Home Appliance(정보가전)
- 기술의 발전에 따라 일반 가전 제품에 다양한 기능이 요구됨
- 다양한 기능의 처리를 위한 가전제품 전용 임베디드 시스템 적용
- 가정 내 네트워크 구성에 따른 원격 제어, 정보 수집 등이 가능해짐
- 홈 오토메이션, 홈 네트워킹과 함께 가정 자동화의 핵심 부분
- 인터넷 냉장고, HDTV, 인터넷 전자레인지, DVR, 세탁기 등
Electronic Home Appliances (1)

- **HDTV/Internet TV**
  - Internet surfing, function setup

- **Internet Washer**
  - Function control & working

- **Internet Refrigerator**
  - Control, web surfing & downloading

- **Microwave Oven**
  - Searching the cooking information & downloading

- **Heater**
  - Control by internet

- **Camcorder**
  - Network connection
Electronic Home Appliances (2)

- Digital TV
- Internet Refrigerator
- Internet Washer
- Microwave Oven
Home Automation

- Every device is connected to the home network
- For this, every device contains embedded system
Factory Automation

- 공장자동화 : FA (Factory Automation)
  - 특정 기계나 장비를 통해 생산 과정을 자동적으로 관리하는 시스템
  - 센서와 제어 시스템, 로봇 등으로 구성하여 무인시스템을 구축
  - 공장 자동화 및 로봇은 실시간 시스템과 임베디드 시스템 발전의 원동력
  - 생산성증대: 인건비감소, 오류감소, 품질의 균일화, 생산기간단축
  - 로봇, conveyor belt
Mobile Hand-held Device

• 정보단말기기
  - 단순한 통화 중심의 이동 전화기에서 각종 정보검색, 오락, 메시징 등의 복합 기능이 수행되는 디지털 정보단말기기로 발전
  - 단말기기 각각의 기능에 맞는 마이크로프로세서, 메모리, 운영체제, 응용 프로그램 등으로 구성
  - 앞으로는 다양한 단말기가 하나의 기기로 통합될 것으로 예상됨
  - 핸드폰, PDA, 스마트폰, MP3 플레이어, 게임기기 등
Space/Aviation

- Aviation
  - Equipped with a few hundred or more processors
- Space-craft
  - Pathfinder – equipped with RTOS VxWorks
  - Representative real time system

NASA Pathfinder (mission to MAR 1997)
Traffic/Intelligent Toys

Embedded System Overview
Game…

- High performance microprocessor
- E.g. Xbox, PS2, PSP, etc
  - 32-Bit ARM processor
  - 2.9 inch TFT
Finance/Office…

- Finance
  - POS terminal, ATM terminal
- Office
  - printer, scanner, fax, copier, multifunction printers
Network...

- Digital Switch Board, PABX (private automatic branch exchange), etc
- Router, Gateway, IP sharer, etc
- Set-top box
Key Recent Trends

- Increasing computation demands
  - e.g. multimedia processing in set-top boxes, HDTV
- Increasingly networked
  - to eliminate host, and remotely monitor/debug
  - embedded Web servers
    - e.g. Mercedes car with web server
  - embedded Java virtual machines
    - e.g. Java ring, smart cards, printers
  - cameras, disks etc. that sit directly on networks
- Increasing need for flexibility
  - time-to-market under ever changing standards!

*Need careful co-design of h/w & s/w!*
An Analogy: Life on Earth

Insects/Embedded Systems

Humans/Desktop

Source: Scientific American, 7/01

Embedded System Overview
패러다임의 변화

- Mainframe (one computer, many people)
- PC (one person, one computer)
- Ubiquitous Computing (one person, many computers)
임베디드 시스템의 향후 전망 - H/W

- 프로세서
  - 대부분의 프로세서가 임베디드 시스템용으로 사용
  - 데이터 처리용량 증가: 32비트 코어 => 64비트 코어로 발전.
  - 고성능의 파이프라인, DSP, 자바 처리 전용 H/W 추가
  - 많은 종류의 마이크로프로세서/컨트롤러들 중에서 응용에 최적인 제품을 찾아내는 것이 설계에서 매우 어렵고 중요한 작업

Where Has CS Focused? Where Are the Processors?

Source: DARPA/Intel (Tennenhouse)
새로운 시대의 가능성 - 임베디드

Embedded System Overview

~ 1980 ~ 2000 ~ 2010 ~ 2020

1대 컴퓨터 : 다수 (Server-to-Client)

1대 컴퓨터 : 1인

다수 컴퓨터 : 1인

사람+컴퓨터+사물 (Things to Things)
Conclusion

• Embedded **software is tightly coupled with hardware**

• To design an embedded system, we should know well about computer hardware, system software and device driver.

• In this course, we need the following backgrounds
  – Linux OS : kernel architecture, file system, command, …
    • **We will use an embedded linux operating system(2.6 version)**
  – Linux Utility : shell, make, vi, tar, mount/umount …
  – Computer Architecture : internal microprocessor architecture and external interface, bus, interrupt handling, I/O interfacing, …

• In this course, we will experience the embedded system with our experimental boards
Schedule

Final Labs: build the hardware control device driver program based on Embedded Linux which has the following functions

• Dynamic device driver program which can be uploaded into the kernel after booting
  – How to develop the boot-loader, kernel and the application software for the target embedded system on the host computer?
  – How to download the developed software to the permanent ROM/Flash memory of the target embedded system?
  – How to debug the target embedded system?
  – What is the module program? What is the device driver?
  – How to relate the device driver program with the embedded Linux kernel?
Schedule

• Trigger the interrupt from the key pad connected to the GPIO pin of PXA270 or PXA320 processor on our experimental board
  – Internal architecture/components of embedded processor(PXA270 or PXA320)?
  – What is ARM core? Why ARM core?
  – What is the GPIO pins? How to control the GPIO pins?
  – How to trigger and process the interrupt from the external device?

• Handle the interrupt using the blocking I/O
  – How/when to register the interrupt handler?
  – What is the blocking/non-blocking interrupt handling?
  – Dynamic memory allocation in the kernel?
  – Kernel program vs application program?