嵌入式系統導論

前言

- 近年來，隨著以計算機技術，通訊技術為主的資訊技術的快速發展和網際網路的廣泛應用,傳統的單晶片控制的方式正在發生改變，出現了許多新的處理器變革讓整個嵌入式的應用其及開發注入了很多的新的契機。

一、嵌入式系統如何定義：

- 一種獨立設備被用於控制(control)、監視(monitor)或協助設施、機器、工廠之操作運轉。
- 大到整個硬體設備的一個零組件，或一部大型機器。
- 嵌入式系統皆是有一個單一或多個處理器晶片為核心技術，與其他控制晶片套裝在一組件上或特殊用途的IC晶片上。
- 簡單的嵌入式系統可以執行單一或一組預先設計的功能用途(function)，比較複雜的嵌入式系統大都經由應用程式控制執行一個特殊功能，或多個複雜程序的功能表現。

- 在最簡單的傳統的嵌入式系統中，往往僅僅只有執行單一功能的控制能力，在單獨唯一的ROM 中僅能執行單一功能的控制程式，沒有具備微小型作業系統。
- 有兩種軟體型態在儲存體內。
  - 軟體型態（應用程式）。
  - 韌體(Firmware)。

嵌入式的產品 (應用系統)


Connecting
Screen Phone
Mobile Phone
Set Top Box
Handheld Game
Consumer
Game Console
Net TV
Auto PC
DSC
Smart Mobile Phone
Handheld PC
PDA
Thin Client
MP3
Palm PC
Computing

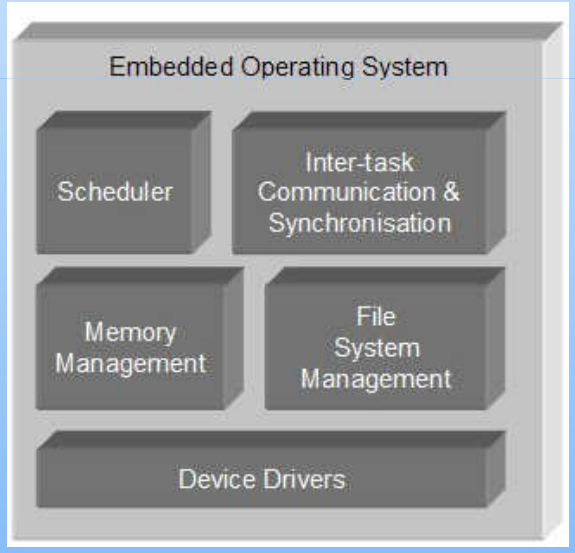
1.嵌入式系統的硬體與軟體規格

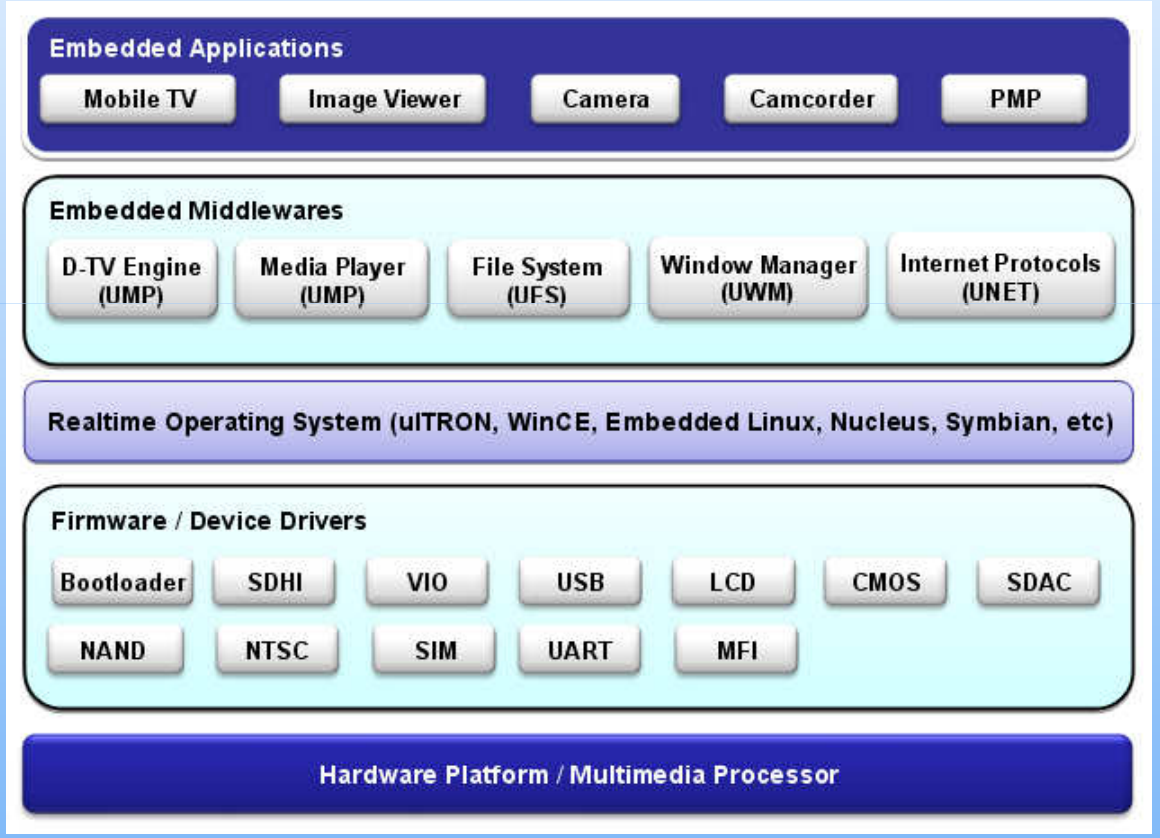
- 嵌入式系統的硬體必須根據具體的應用任務，以功率消耗、成本、體積、可靠性及處理能力等為指標性的來選擇其有對應主要要求的適合下去做最佳化選擇性規劃設計。

- 嵌入式系統的核心是系統軟體和應用軟體，由於儲存的空間是有限制的，因而要求寫軟體程式碼更為有效率及可靠，大多對即時性有嚴格要求。

Why do we need OS in the Embedded system?

典型嵌入式 OS 架構


Embedded Operating System
Scheduler
Inter-task Communication & Synchronisation
Memory Management
File System Management
Device Drivers


Embedded Applications
Mobile TV
Image Viewer
Camera
Camcorder
PMP
Embedded Middlewares
D-TV Engine (UMP)
Media Player (UMP)
File System (UFS)
Window Manager (UWM)
Internet Protocols (UNET)
Realtime Operating System (uITRON, WinCE, Embedded Linux, Nucleus, Symbian, etc)
Firmware / Device Drivers
Bootloader
SDHI
VIO
USB
LCD
CMOS
SDAC
NAND
NTSC
SIM
UART
MFI
Hardware Platform / Multimedia Processor

2.嵌入式作業系統種類

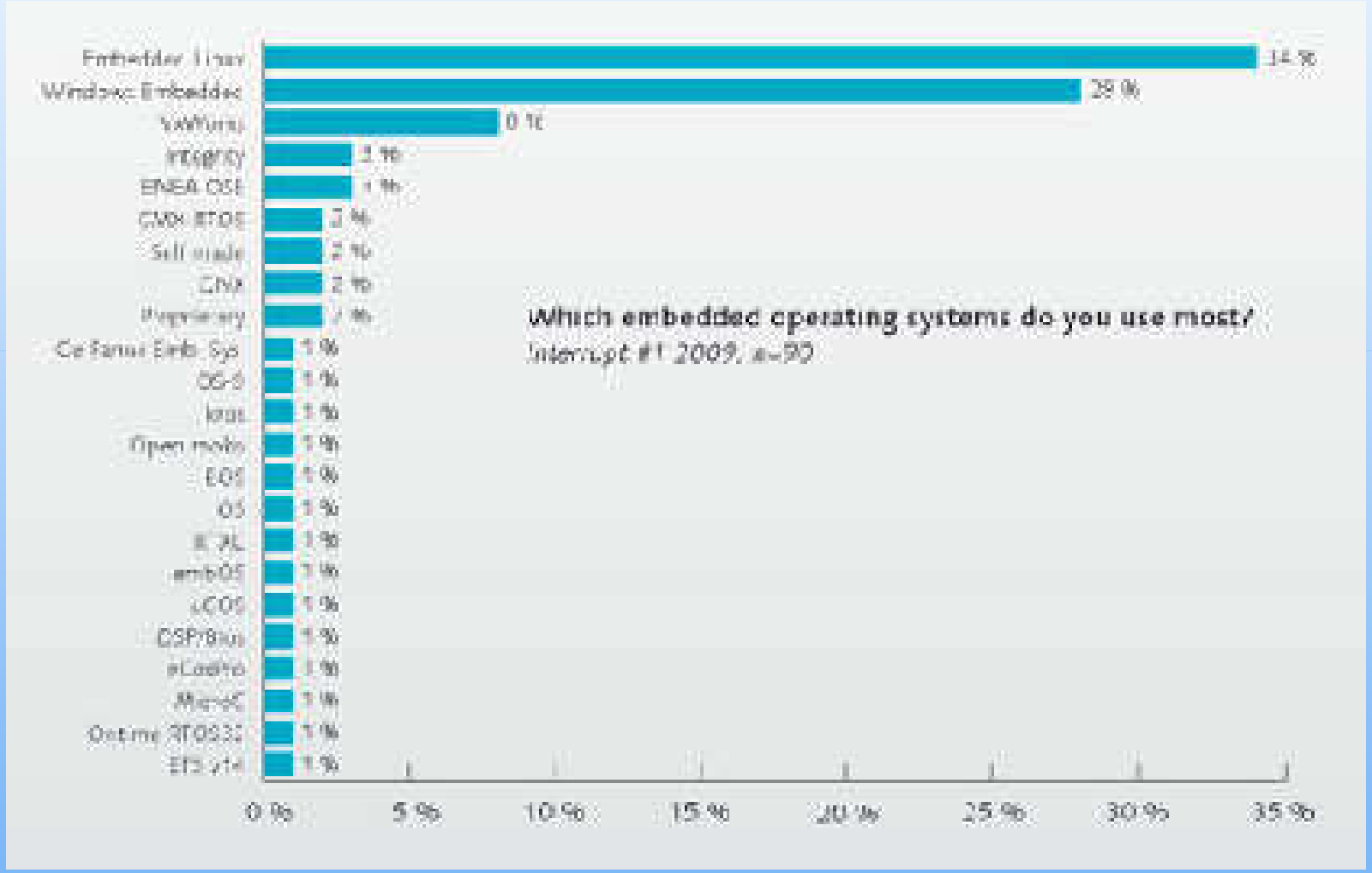
- 1.從目前在個人電腦上的作業系統及攜帶式電腦系統去移植到嵌入式系統中，形成的嵌入式作業系統，如:
  - 微軟公司的Windows CE 及其更新版本， SUN 公司的Java 作業系統，朗訊科技公司的Inferno，嵌入式Linux 等。
- 這類系統經過個人電腦或高性能計算機等產品的長期營運考驗，技術日趨成熟，其相關的標準和軟體開發模式已被用戶普遍接受，同時累積了豐富的開發工具和應用軟體資源。

2.嵌入式作業系統種類 (cont.)

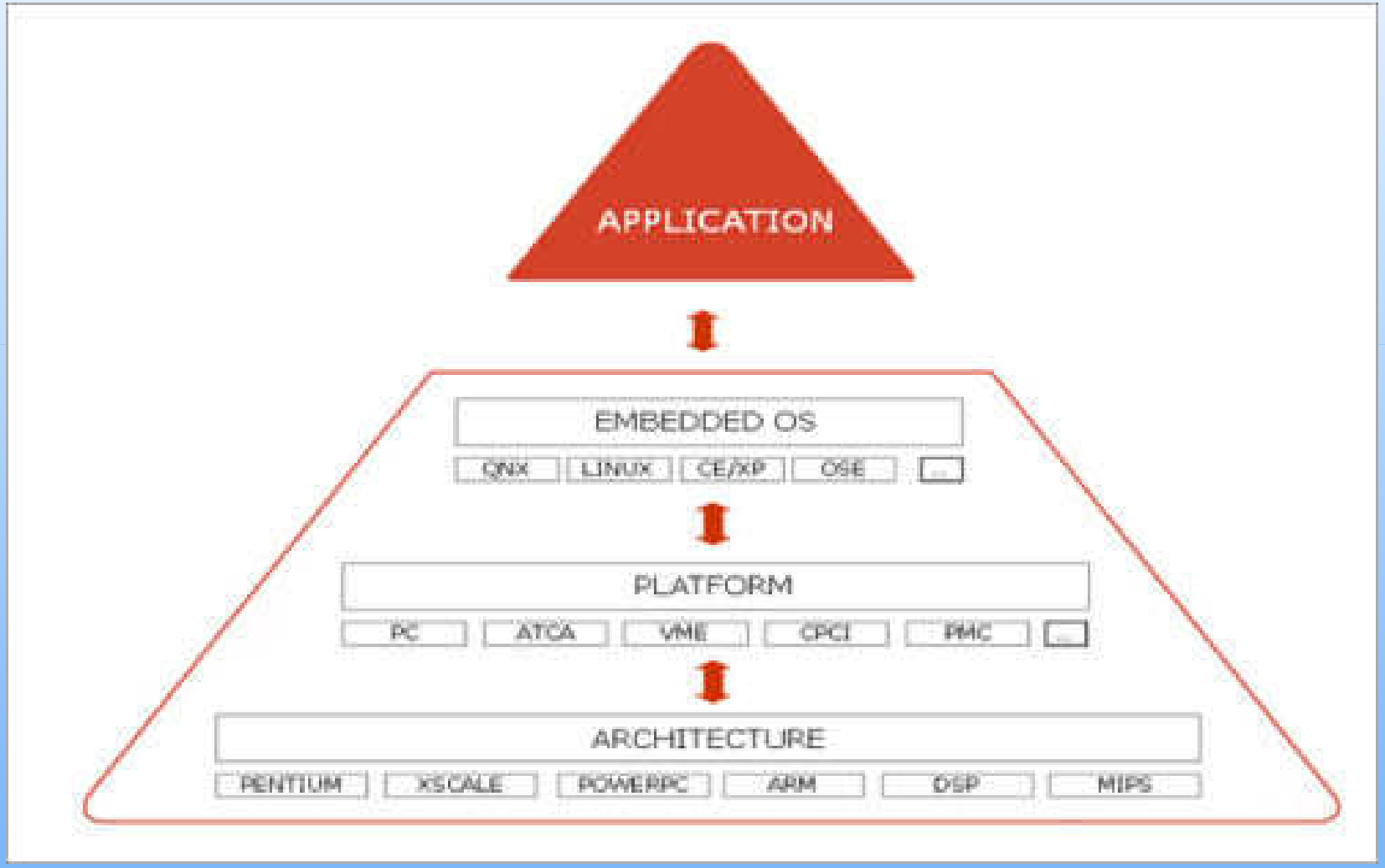
- **即時作業系統**，如ATI 的Nucleus，QNX 系統軟體公司的QNX，ISI 的pSOS，WindRiver 公司的VxWorks等。
- 這類產品在作業系統的架構和實現上都針對所面向的應用領域，對即時性的高可靠性等進行了精巧的作業系統設計，而且提供了獨立而完備的系統開發和測試工具，較多地應用在軍用產品和工業控制等領域中。

Embedded Linux

- Linux 是90 年代以來逐漸成熟的一個開放源代碼的作業系統。 PC 機上的Linux 版本在全球數以百萬計愛好者的合力開發下，得到了非常迅速的發展。
- 90 年代末uClinux， RTLinux 等相繼推出，在嵌入式領域得到了廣泛的關注，它擁有大批的程式員和現成的應用程式，是在研究開發工作上的最寶貴的資源。


Which embedded operating systems do you use most?
Interrupt #1 2009, n=90
14 %
29 %
0 %
1 %
1 %
2 %
2 %
2 %
1 %
1 %
1 %
1 %
1 %
1 %
1 %
1 %
1 %
1 %
1 %
0 %
5 %
10 %
15 %
20 %
25 %
30 %
35 %

三、嵌入式系統核心架構組成介紹



嵌入式開發的系統架構

1. 單核心或多核心嵌入式的硬體系統。
2. 單核心或多核心啟動載入核心系統。
3. 單核心或多核心的微型化小容量的作業系統
4. 嵌入式的應用程式開發系統。

四、嵌入式系統的應用介紹：

嵌入式系統的應用層面分類

- 獨立的微處理器： 目前在許多小型功能設備時實現。
  - 例如：溫度感應器、煙塵瓦斯偵測器、智慧型家庭自動化電路....等等。
- 微型處理器設備： 應用一些小型的處理器執行單一功能的程式。
  - 例如：流量控制、信號擴大器、定位感應器、和行動電話 。

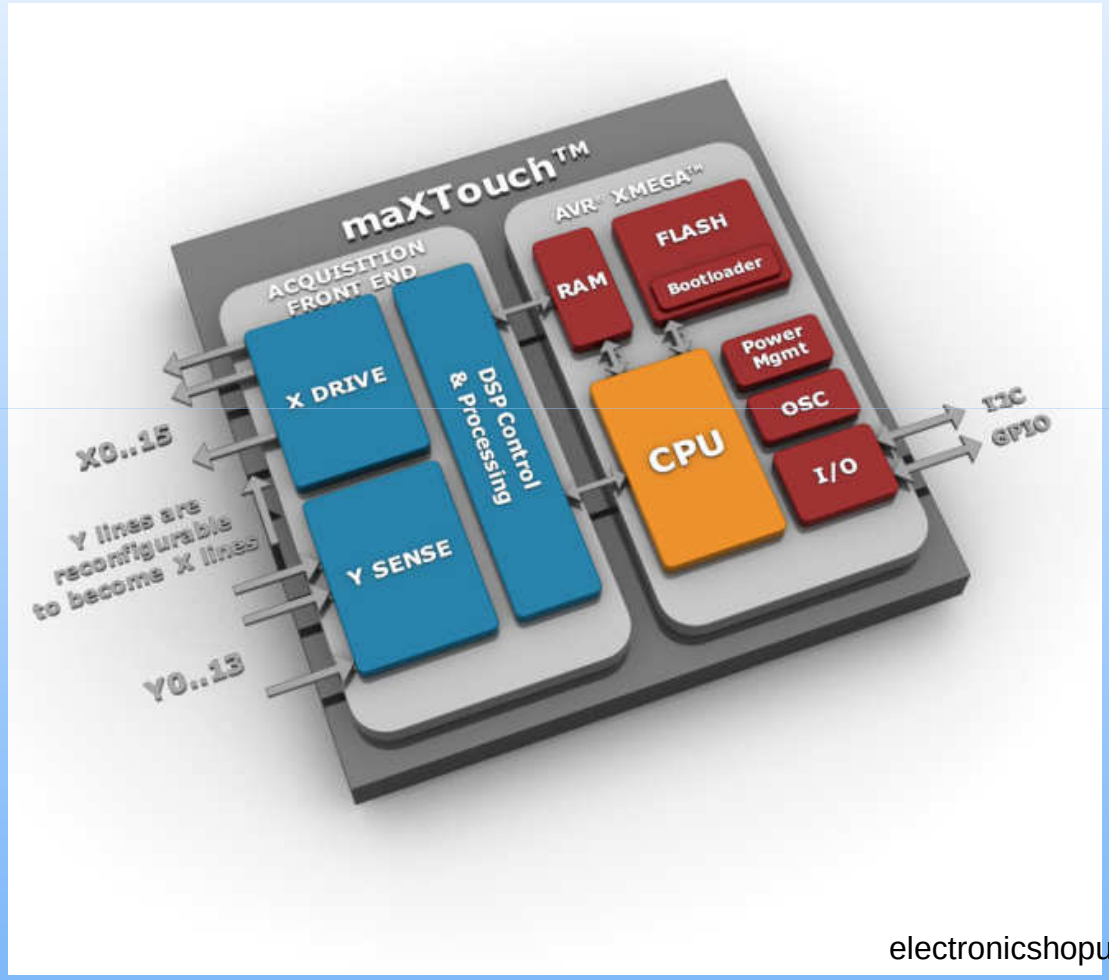
嵌入式系統的應用介紹

- — 次零組件設備 ： 通常為特定一個功能的整體特殊功能的設計，例如：齒輪開關器、交通指揮與導航系統控制器、網路電話交換機、電子化升降機、影像數位的資料攫取和視訊監視系統、視訊網路診斷和即時反應回報安全控制系統等都屬此範圍。其可能與PC網路合作，或與資料庫連接互動成為一個具有儲存資料的機器設備。
- — 以電腦嵌入式核心系統用在製造或流程控制設備 ： 連接或控制工廠、機器或設備，包括自動化貨物補給、倉庫儲存和貨物自動化運送系統，電腦嵌入式核心系統與商業上的資料庫管理資訊處理系統很緊密的連結。
- 例如：出貨系統、庫存系統、 安全庫存料件系統、運輸派遣系統等等都與自動化生產線有緊密的相關連。

Terminology

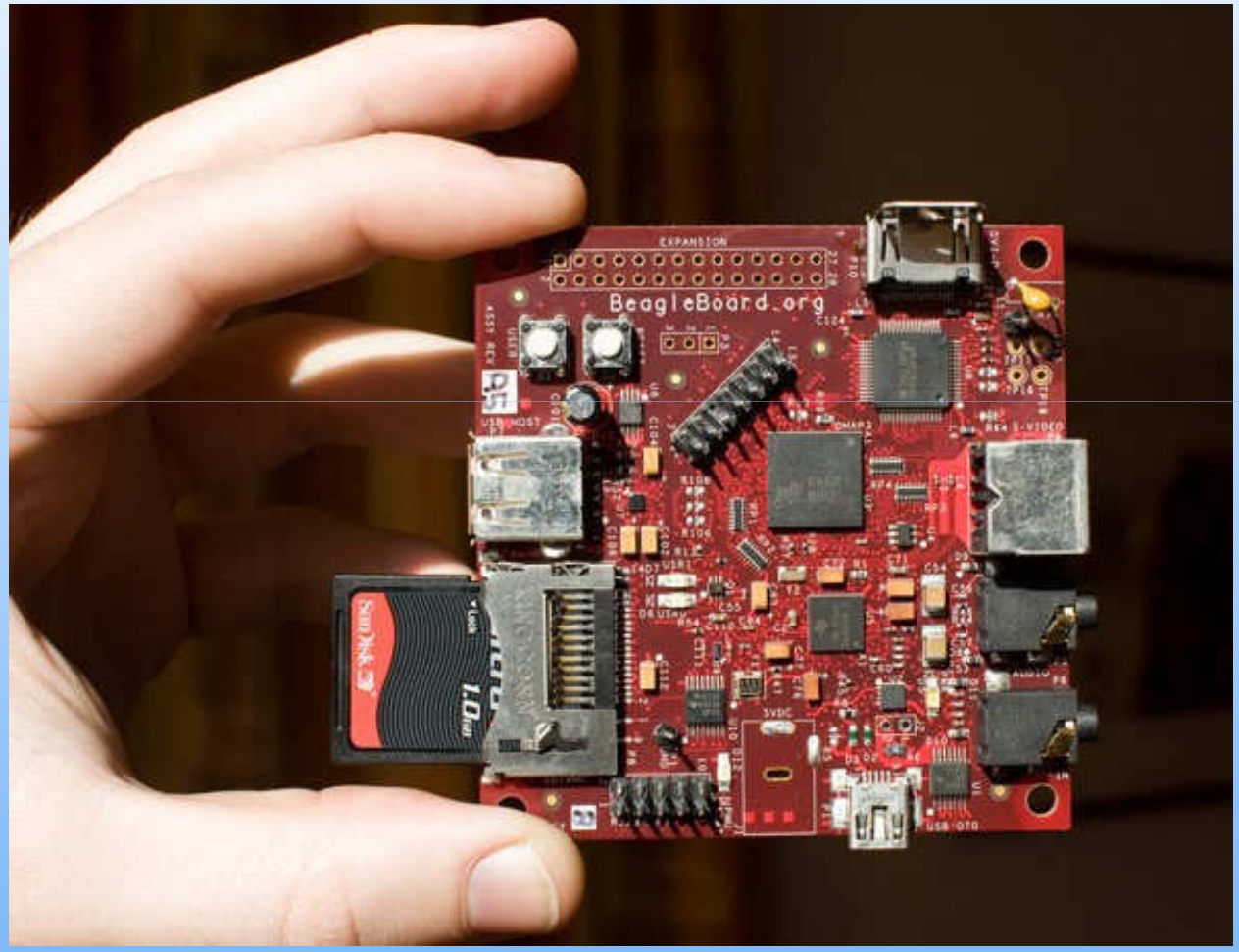
- MCU vs SoC
- DSP
- FPGA
- IC
- µP vs. µC
- CPU architectures:
  - Von Neumann vs. Harvard
  - RISC vs. CISC
  - 8/16/32 bits
  - Big vs. little endian

IC Inside



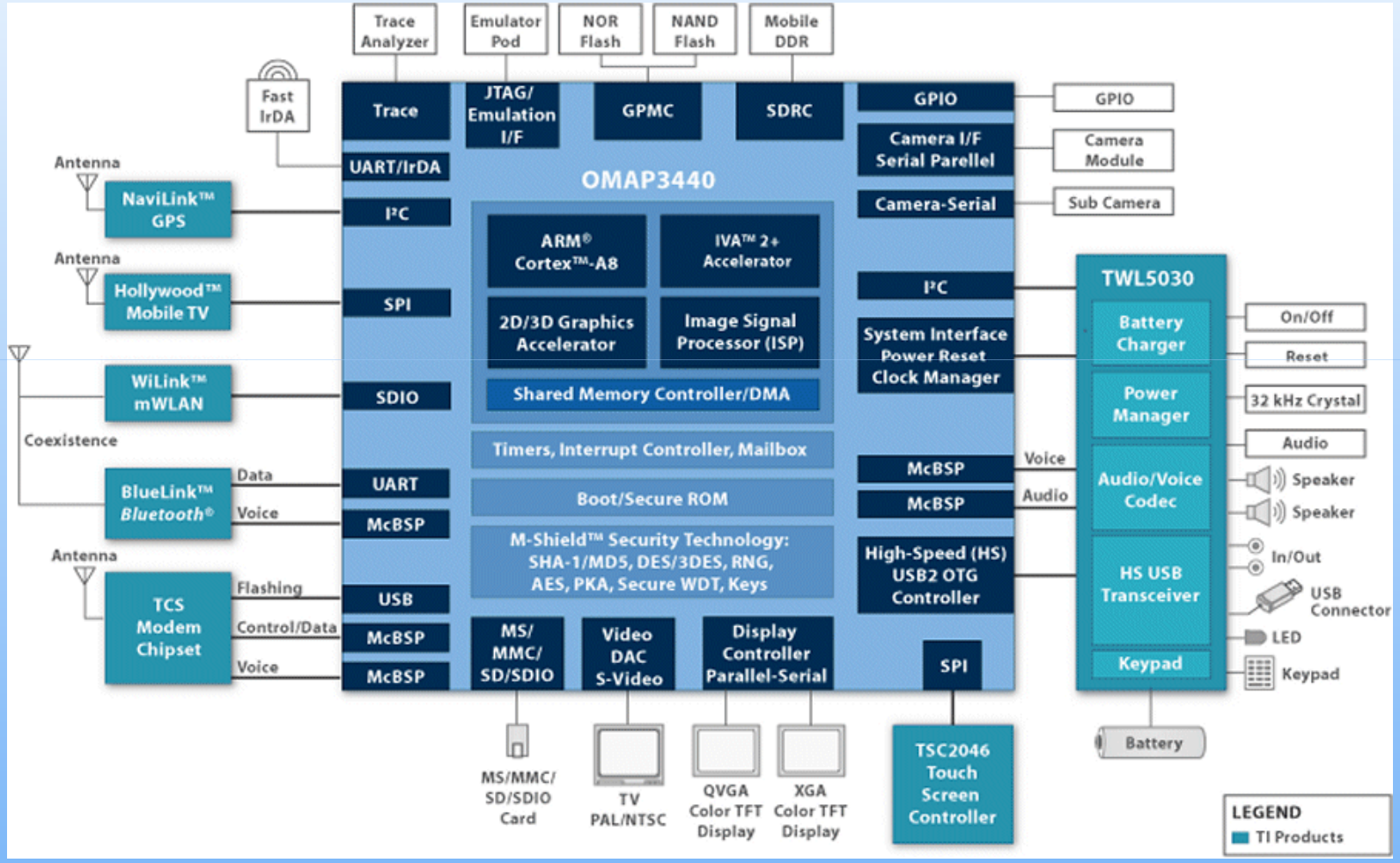
electronicshopusa.blogspot.com

OMAP 3530 EVB (Beagle Board)

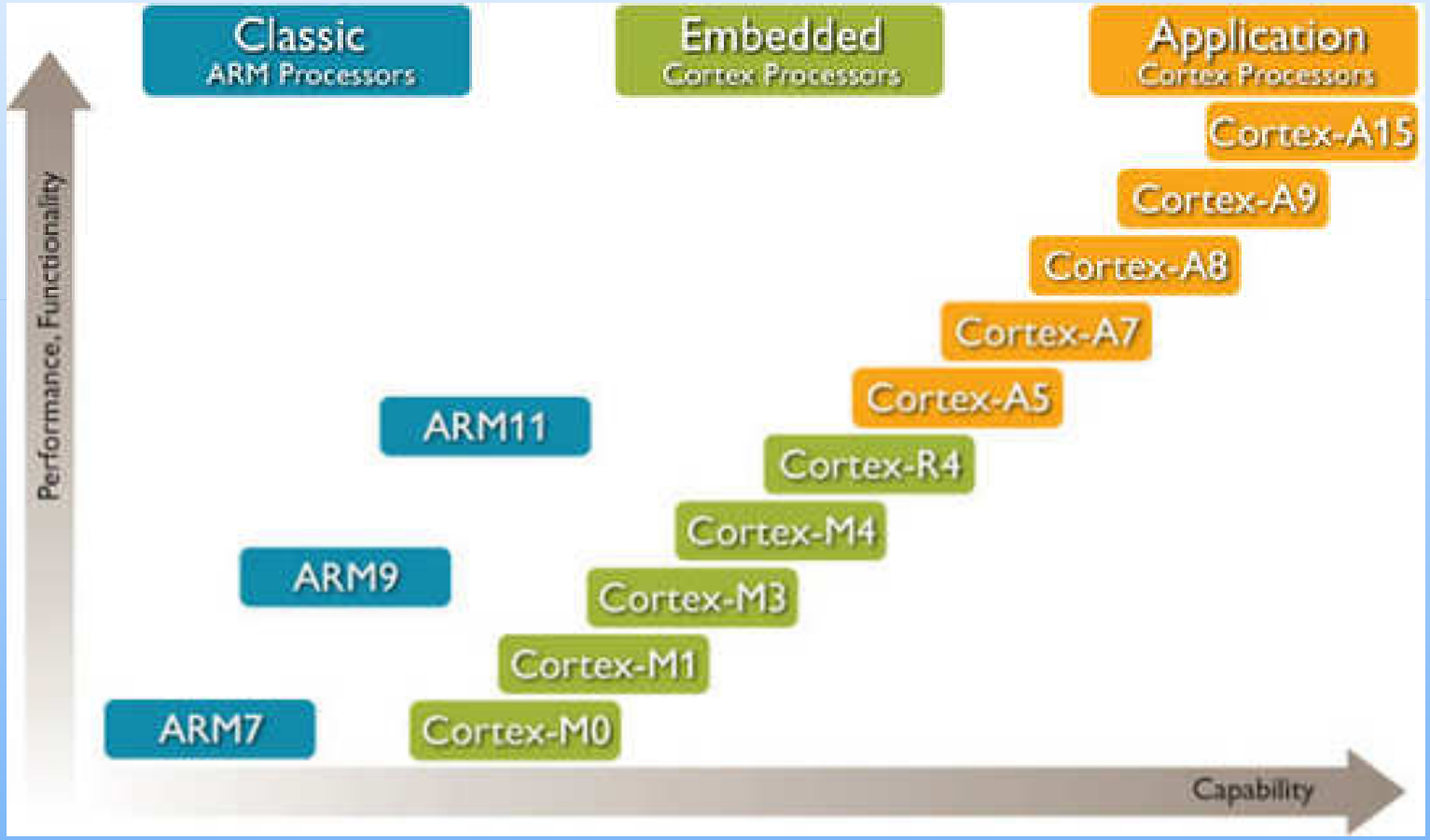


tech.nocr.at

SoC (OMAP3)


Trace Analyzer
Emulator Pod
NOR Flash
NAND Flash
Mobile DDR
Fast IrDA
Trace
JTAG/ Emulation I/F
GPMC
SDRC
GPIO
Camera I/F Serial Parellel
Camera-Serial
GPIO
Camera Module
Sub Camera
UART/IrDA
OMAP3440
Antenna
NaviLink™ GPS
I²C
ARM® Cortex™-A8
IVA™ 2+ Accelerator
Antenna
Hollywood™ Mobile TV
SPI
2D/3D Graphics Accelerator
Image Signal Processor (ISP)
I²C
System Interface Power Reset Clock Manager
TWL5030
Battery Charger
On/Off
Reset
Power Manager
32 kHz Crystal
WiLink™ mWLAN
SDIO
Shared Memory Controller/DMA
Audio
Coexistence
Timers, Interrupt Controller, Mailbox
McBSP
Voice
Audio/Voice Codec
Speaker
Speaker
BlueLink™ Bluetooth®
Data
Voice
UART
McBSP
Boot/Secure ROM
McBSP
Audio
In/Out
Antenna
M-Shield™ Security Technology: SHA-1/MD5, DES/3DES, RNG, AES, PKA, Secure WDT, Keys
High-Speed (HS) USB2 OTG Controller
HS USB Transceiver
USB Connector
TCS Modem Chipset
Flashing
Control/Data
Voice
USB
McBSP
McBSP
MS/ MMC/ SD/SDIO
Video DAC S-Video
Display Controller Parallel-Serial
SPI
LED
Keypad
Keypad
MS/MMC/ SD/SDIO Card
TV PAL/NTSC
QVGA Color TFT Display
XGA Color TFT Display
TSC2046 Touch Screen Controller
Battery
LEGEND
TI Products

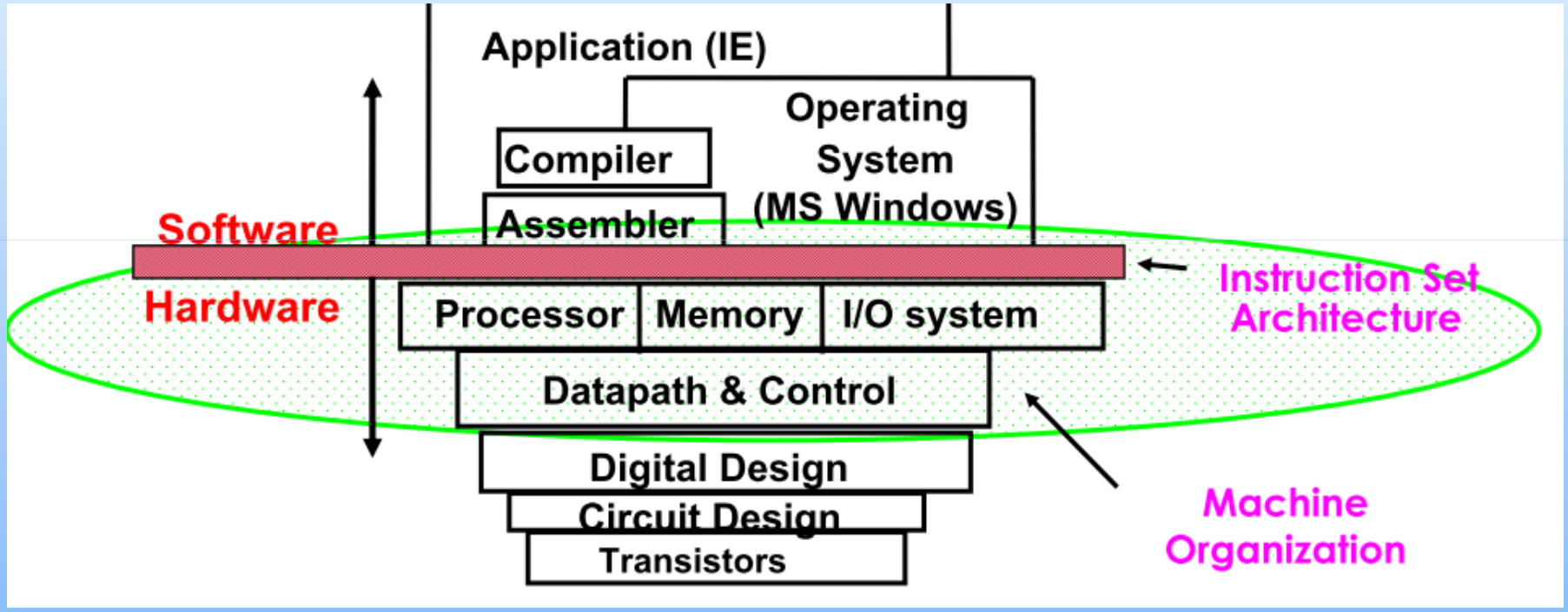
ARM uProcessor



ARM 架構與版本

架構	處理器家族
ARMv1	ARM1
ARMv2	ARM2, ARM3
ARMv3	ARM6, ARM7
ARMv4	StrongARM, ARM7TDMI, ARM9TDMI
ARMv5	ARM7EJ, ARM9E, ARM10E, XScale
ARMv6	ARM11, ARM Cortex-M
ARMv7	ARM Cortex-A, ARM Cortex-M, ARM Cortex-R

ARM的經營模式在於販售其半導體知識產權核心（IP core），授權廠家依照設計製作出建構於此核的微控制器和中央處理器


Application (IE)
Operating System (MS Windows)
Compiler
Assembler
Software
Hardware
Processor
Memory
I/O system
Instruction Set Architecture
Datapath & Control
Digital Design
Circuit Design
Transistors
Machine Organization

BSP (Board support packages)

- Device software to aid you with board bring-up and design.
- BSP provides a development platforms for your specific architecture supporting the latest processors.
- Typically, BSP is provided by silicon vendors and hardware manufacturers
- *processor-* or *platform*-dependent code
    - *Drivers for most on chip peripherals*
    - *USB Host mass storage class application*
    - *USB device mass storage class application*
    - *Ethernet driver*
    - *TCP/IP stack and file system*

BOM Cost

Pricing in U.S. Dollars	Part
$22.80	8Gbyte NAND Flash Memory
$20.00	Improved Touch screen
$20.00	Display
$13.50	Application Processor
$5.00	1Gbit SDRAM - Mobile DDR
$15.00	HSDPA Digital Baseband
$7.00	2 Megapixel Camera Module
$4.00	WLAN chipset
$4.00	Battery
$3.60	GPS
$2.00	Motion Sensor / Accelerometer
$4.25	RF Transceiver
$1.89	64Mbit NOR Flash Memory, 32Mbit PSRAM
$1.80	Charger (USB)
$2.00	Bluetooth 2.0 + Enhanced Data Rate (EDR)
$126.84	Total of Above Items
$37.16	Other Costs
$164.00	Total Bill of Materials (BOM) Costs (Direct Materials Only)
$173.00	**Total BOM & Manufacturing Costs**
$199	Initial retail price, 8Gbyte Version

* Research includes other costs, including software development, shipping and distribution, packaging, and miscellaneous accessories included with each phone.

ECR and ECN

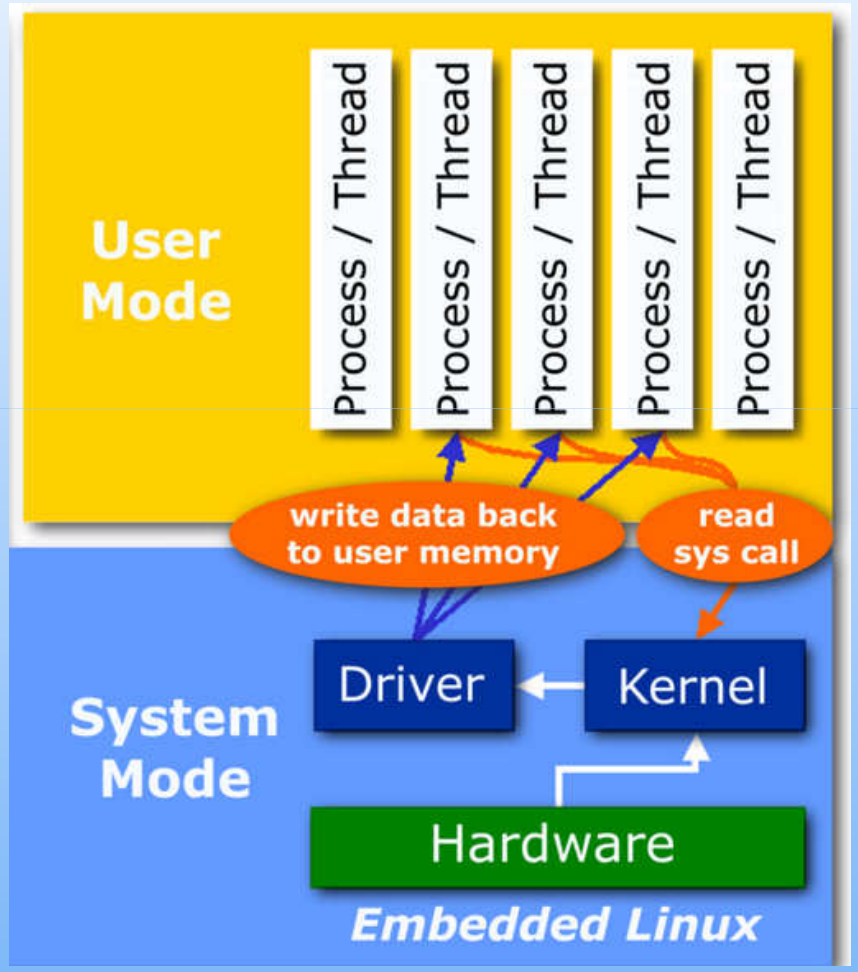
- ECR Engineering Change Request 是用在提出變更需求時，提出的人可以視業務因為客戶的需求改變，RD因為設計的改善，工程因為產線的良率考量等等。所以只是提出需求讓各個相關單位，大家一起來考慮引響的程度。

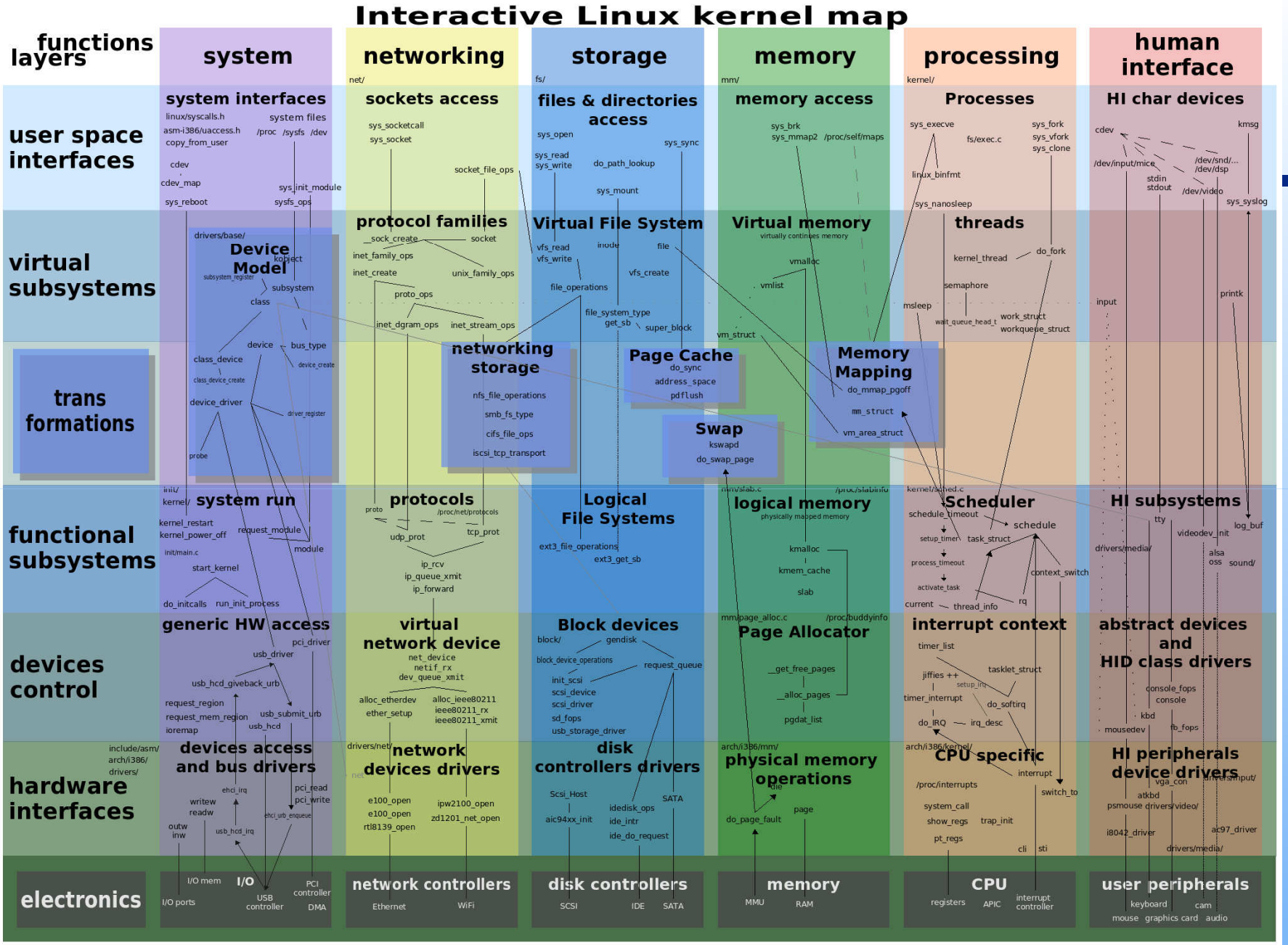
- ECN Engineering Change Notice 是用在變更通知時，這時候設計變更已經確知，引響的程度已先評估過，更改項目也必須要正式執行時提出。提出的人通常是RD或是工程單位，提出的內容是要各個單位去執行的。

Embedded Linux

- Embedded Linux is the use of a Linux OS in embedded computer systems
- Use of Linux in embedded devices continues to grow at an exciting pace :
  - mobile phones, PDA, media players, set-top boxes, and other consumer electronics devices, networking equipment, industrial automation, navigation equipment

Embedded Linux架構





Linux kernel

- Kernel Architecture Overview
  - User Space
  - Kernel Space
- Kernel Functional Overview
  - File System
  - Process Management
  - Device Driver
  - Memory Management
  - Networking

Pros and Cons

- Advantages of embedded Linux :
  - no royalties or licensing fees, a stable kernel, a huge variety of applications and networking protocols, a huge number of developers
- Disadvantages of embedded Linux :
  - Larger memory footprint (kernel and root filesystem), complexities of user mode and kernel mode memory access and complex device drivers framework

TEXAS INSTRUMENTS

Products | Applications | Design Support | Sample & Buy

Wireless Solutions

Overview

TI Home > Wireless Handset Solutions > OMAP™

Operating Systems: Linux

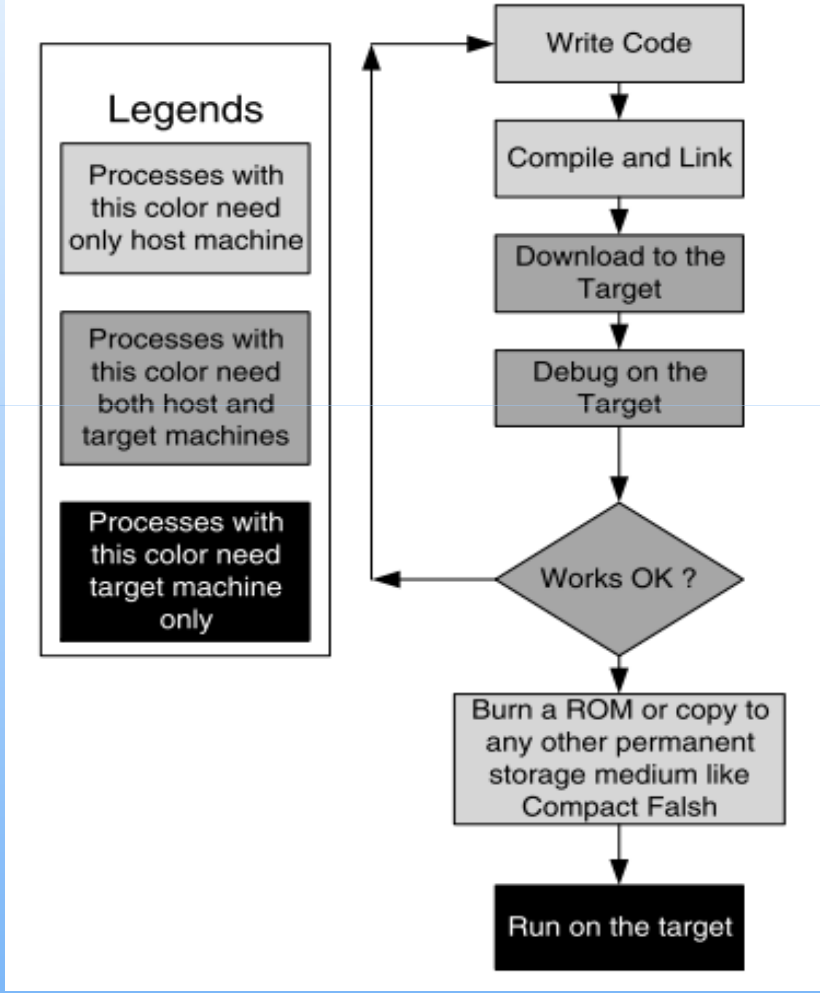
Linux development tools

Industry-standard Linux development tools provide OMAP developers a fully tested, full-function Linux environment that streamlines software development and delivers software programs to market faster.

With a full board support package (BSP) for Linux, and TI´s development tools like the award-winning Code Composer Studio integrated development environment (IDE) and the Innovator™ Development Kit for the OMAP Platform, software developers can quickly get started with the high-performance and power efficiency of the OMAP platform.

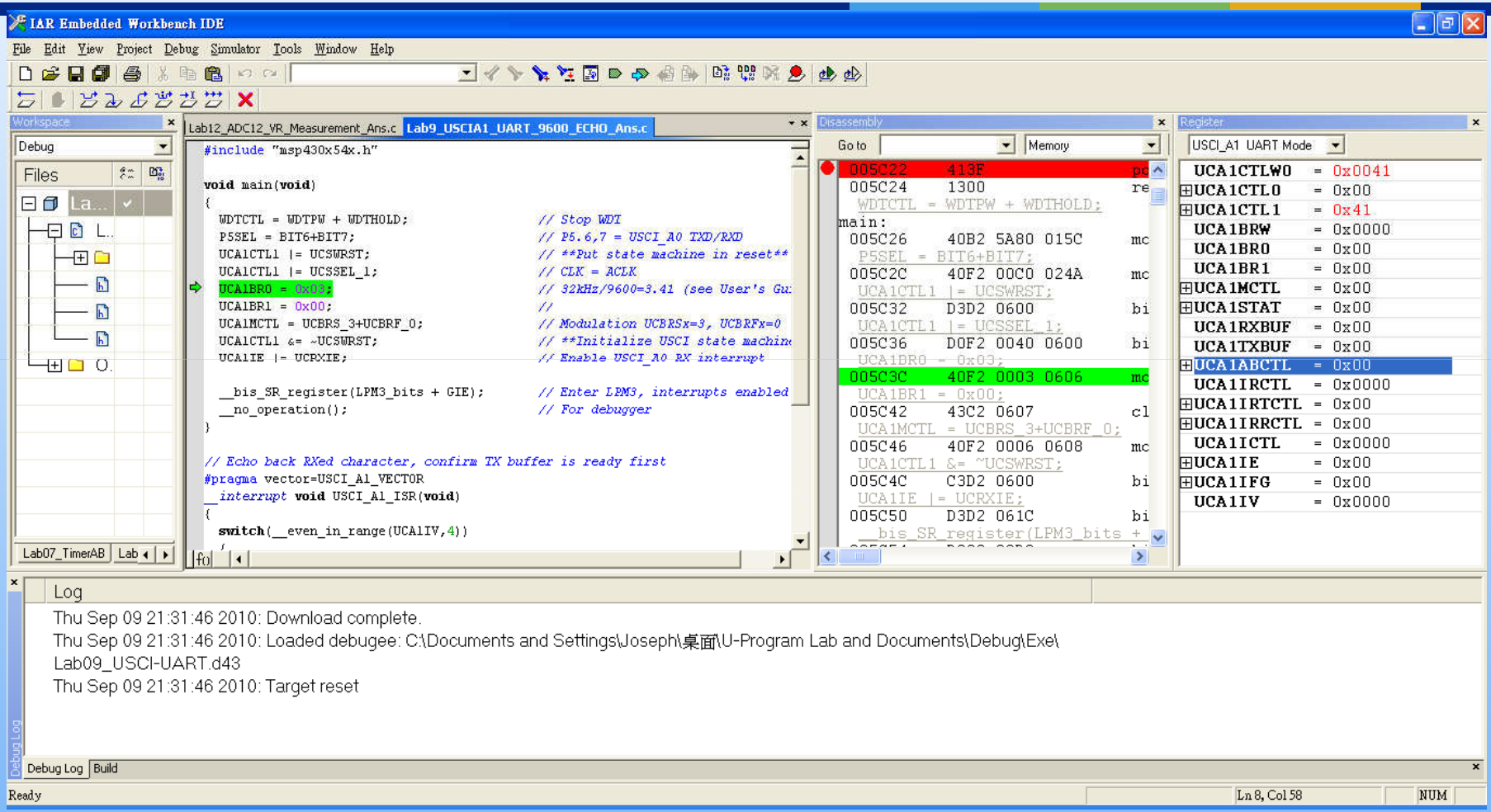
Visit linux.omap.com for mailing lists, downloads, documentation, and other resources.

Cross-platform Development.



- ISS
  - Gives us control over time – set breakpoints, look at register values, set values, step-by-step execution, ...
  - But, doesn't interact with real environment
- Download to board
  - Use device programmer
  - Runs in real environment, but not controllable
- Compromise: emulator
  - Runs in real environment, at speed or near
  - Supports some controllability from the PC

IAR Embedded Workbench IDE



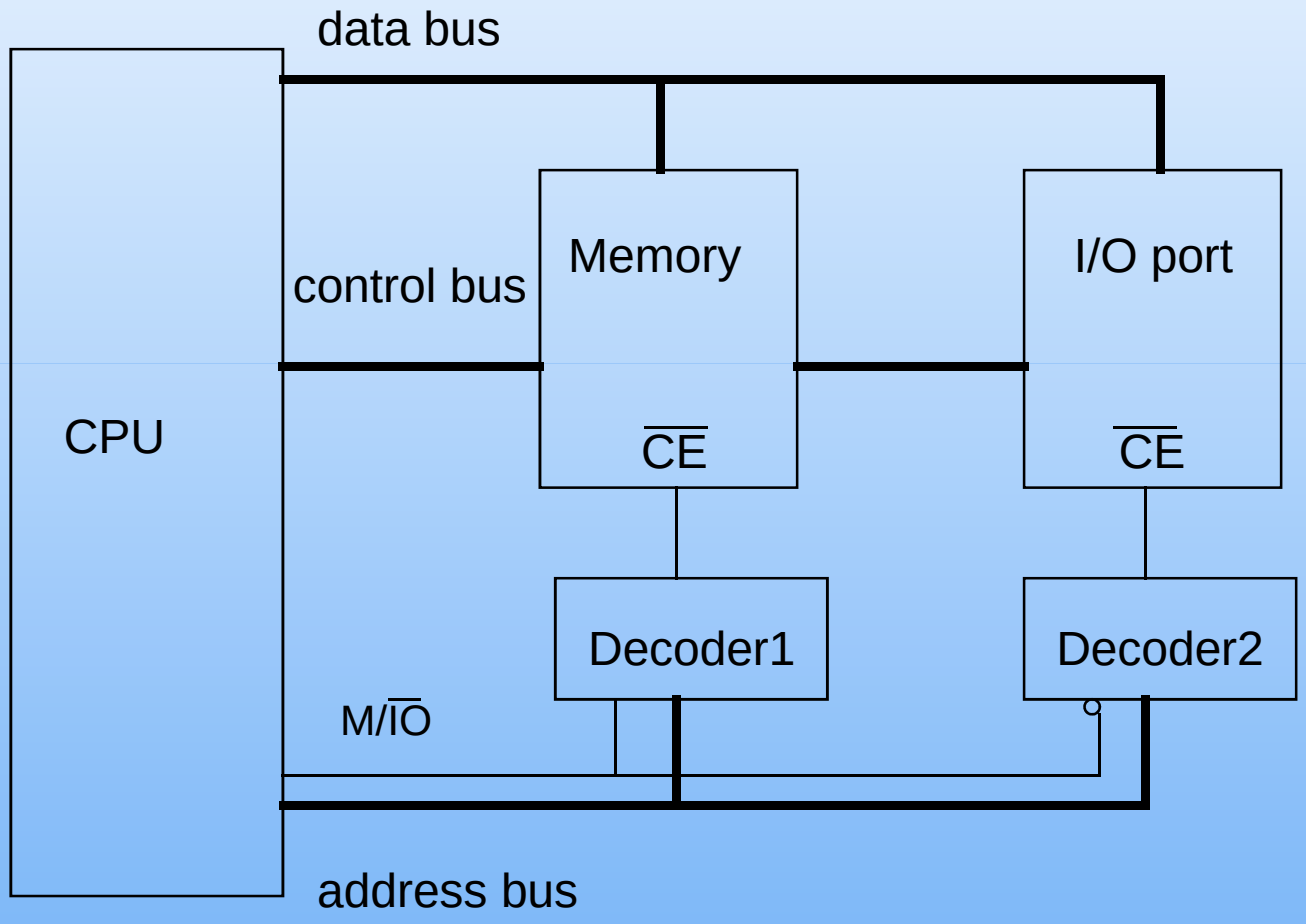
Types of bus-based I/O:memory-mapped I/O and standard I/O

- Processor talks to both memory and peripherals using same bus – two ways to talk to peripherals
  - Memory-mapped I/O
    - Peripheral registers occupy addresses in same address space as memory
    - e.g., Bus has 16-bit address
      - lower 32K addresses may correspond to memory
      - upper 32k addresses may correspond to peripherals
  - Standard I/O (I/O-mapped I/O)
    - Additional pin (*M/IO*) on bus indicates whether a memory or peripheral access
    - e.g., Bus has 16-bit address
      - all 64K addresses correspond to memory when *M/IO* set to 0
      - all 64K addresses correspond to peripherals when *M/IO* set to 1

Memory-mapped I/O vs. Standard I/O

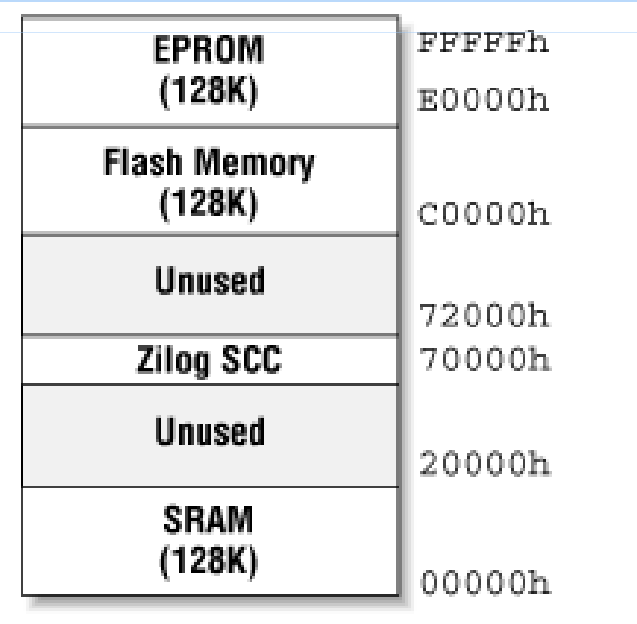
- Memory-mapped I/O
  - Requires no special instructions
    - Assembly instructions involving memory like MOV and ADD work with peripherals as well
    - Standard I/O requires special instructions (e.g., IN, OUT) to move data between peripheral registers and memory
- Standard I/O
  - No loss of memory addresses to peripherals
  - Simpler address decoding logic in peripherals possible
    - When number of peripherals much smaller than address space then high-order address bits can be ignored
      - smaller and/or faster comparators

Standard I/O (80x86)



Memory Map

- A table or diagram containing the name and address range of each *peripheral* addressable by the *processor* within the *memory space*.
- Though the address ranges differs from device to device, overall behavior remains the same.



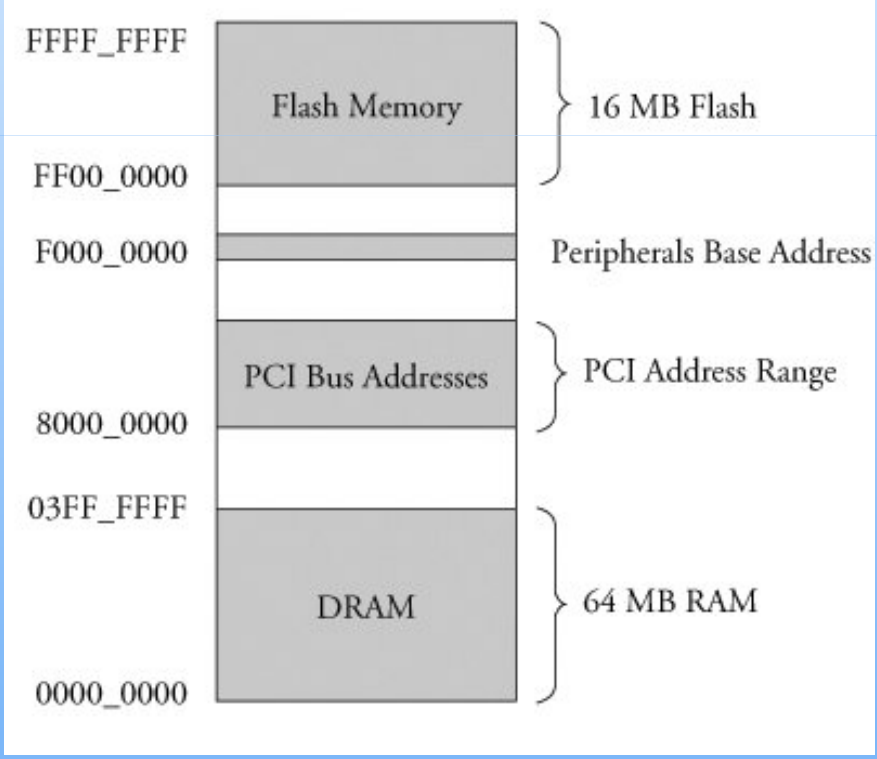
```
/*****************************************************************************
 *
 *  Memory Map
 *
 *            Base Address    Size  Description
 *            --------------  ----- ----------------------------------
 *            0000:0000h      128K  SRAM
 *            2000:0000h      Unused
 *            7000:0000h      Zilog SCC Registers
 *            7000:1000h      Zilog SCC Interrupt Acknowledge
 *            7000:2000h       Unused
 *            C000:E000h      128K  Flash
 *            E000:0000h      128K  EPROM
 *
 *****************************************************************************/

#define SRAM_BASE       (volatile  u8 *)  0x00000000
#define SCC_BASE        (volatile  u16 *) 0x70000000
#define SCC_INTACK      (volatile  u8 *)  0x70001000
#define FLASH_BASE      (volatile  u8 *)  0xC0000000
#define EPROM_BASE      (volatile  u8 *)  0xE0000000
```

```
#define __REG(x)      (*(u32 *)(x))
#define __REGl(x)     (*(u32 *)(x))
#define __REGw(x)     (*(u16 *)(x))
#define __REGb(x)     (*(u8 *)(x))
```

Typical embedded system memory map

- View and manage system memory as a single large, flat address space.



Register

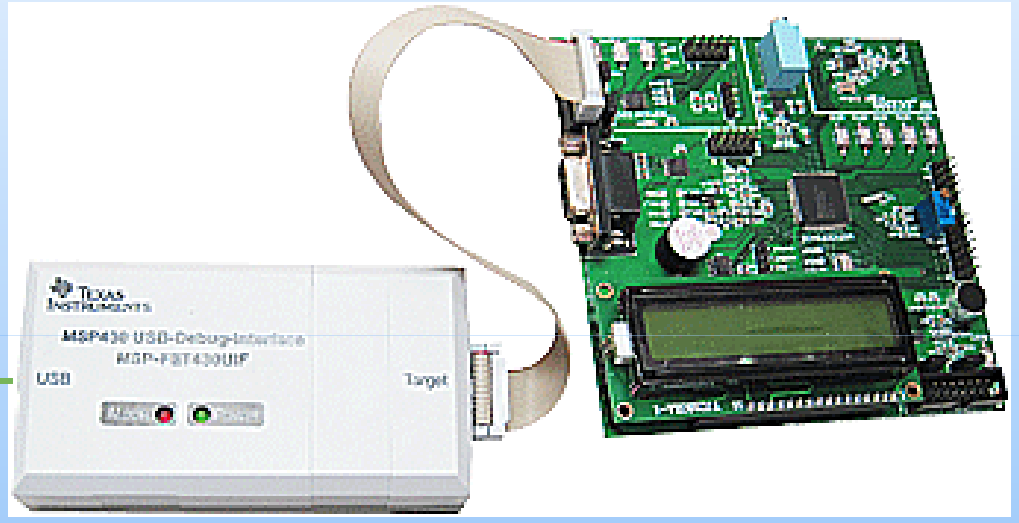
- A memory location that is part of a *processor* or a *peripheral*.
- In other words, it's not normal memory. Generally, each bit or set of bits within the register controls some behavior of the larger device

JTAG based hardware debuggers

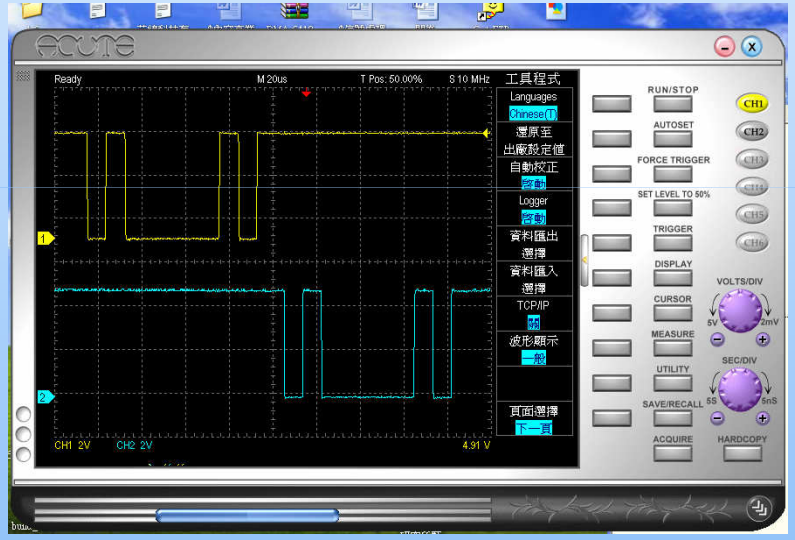
- Using a debugger, you can download software to the target for immediate execution.
- You can also set *breakpoints* and examine the contents of specific memory locations and *registers*

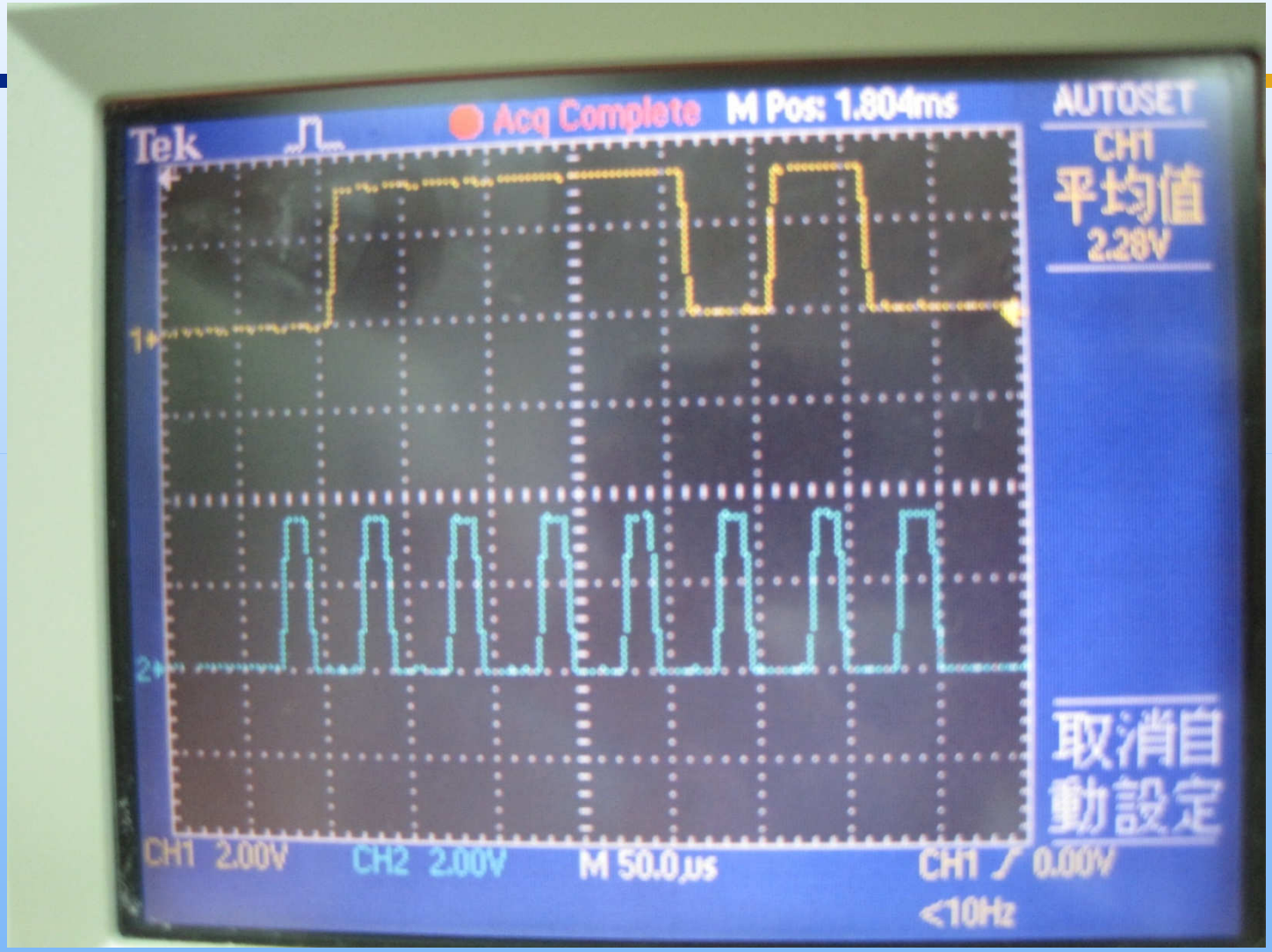
- **Emulators emulate hardware; Simulators mimic software**

Example



數位儲存示波器DSO


Ready
M 20us
T Pos: 50.00%
S 10 MHz
工具程式
Languages
Chinese(T)
還原至
出廠設定值
自動校正
Logger
資料匯出
資料匯入
TCP/IP
波形顯示
一般
頁面選擇
下一頁
CH1 2V
CH2 2V
4.91 V
RUN/STOP
AUTOSET
FORCE TRIGGER
SET LEVEL TO 50%
TRIGGER
DISPLAY
CURSOR
MEASURE
UTILITY
SAVE/RECALL
ACQUIRE
HARDCOPY
VOLTS/DIV
SEC/DIV
CH1
CH2


Tek
Acq Complete
M Pos: 1.804ms
AUTOSET
CH1
平均值
2.28V
1+
2+
取消自
動設定
CH1 2.00V
CH2 2.00V
M 50.0µs
CH1 ∫ 0.00V
<10Hz