**嵌入式系统的定义，特点。分类。典型应用**

* 嵌入式系统的定义

1. “嵌入式系统”实际上是“嵌入式计算机系统”的简称。
2. **IEEE的定义：**嵌入式系统是“用于控制、监视或者辅助操作机器和设备的装置。
3. **国内普遍被认同的定义：**嵌入式系统是“以应用为中心，以计算机技术为基础，软硬件可裁减，适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统”。

* 特点

1. 嵌入式系统通常是形式多样、面向特定应用的；
2. 嵌入式系统得到多种类型的处理器和处理器体系结构的⽀持；
3. 嵌入式系统通常极其关注成本；
4. 嵌入式系统有实时性和可靠性的要求；
5. 嵌入式系统使用的操作系统一般是适应多种处理器、可剪裁、轻量型、实时可靠、可固化的嵌入式操作系统；
6. 嵌入式系统开发需要专门工具和特殊方法。

* 分类

1. **按嵌入式处理器的位数来分类**：4位、8位、16位、32位、64位
2. **按应用来分类**：通信类、信息家电类、汽车电子类、移动终端类、工业控制类
3. **按速度分类**：强实时系统（系统响应时间在毫秒或微秒级）、强实时系统,（其系统响应时间在毫秒或微秒级）、弱实时系统（其系统响应时间约为数⼗秒或更长）
4. **按确定性来分类**：根据确定性的强弱，可将嵌⼊式系统分为硬实时、软实时系统：硬实时系统对系统响应时间有严格的要求，如果系统响应时间不能满⾜，就要引起系统崩溃或致命的错误。软实时系统对系统响应时间有要求，但是如果系统响应时间不能满⾜，不会导致系统出现致命的错误或崩溃。
5. **按嵌⼊式系统软件复杂程度来分类**：循环轮询系统、有限状态机系统、前后台系统、单处理器多任务系统、多处理器多任务系统。

* 典型应用 ？
* 软硬件划分

1. 嵌⼊式系统的设计涉及硬件与软件部件，设计中必须决定什么功能由硬件实现，什么功能由软件实现。
2. 硬件和软件具有双重性
3. 软硬件变动对系统的决策造成影响
4. 划分和选择需要考虑多种因素
5. 硬件和软件的双重性是划分决策的前提

操作系统功能、协议栈、应⽤软件框架等除基本系统、物理接⼝、基本逻辑电路都可以由软件实现。双重部分：算法（加密／解密 编码／解码 压缩／解压）数学运算（浮点运算，FFT）

* 流水线技术

现代cpu大多设计成流水线型的机器,在这种类型机器中几个指令可以并行执行，大大地提高了CPU的效率。ARM和SHARC都有一个三段的流水线：取址、译码、执行。在每个时钟周期流水线都有一条指令的吞吐量。

分类：3-stage：取址、译码、执行

5-stage：取址、译码、执行、Buffer/data、write-back（数据与指令分离，CPI降低）

6-stage

但是一些指令序列可能会打断流水线内的信息流，或者至少暂时降低CPU的执行速度。根据指令特征减小跳跃来获得最佳的流水线过程。

* ARM

ARM是一个已经被开发了多年的RISC(reduced instruction set computer)体系结构族。

1. 工作状态：

第一种是arm状态，处理器执行32位的字对齐的arm指令

第二种是thumb状态，此时处理器执行16位的半字对齐thumb指令

随时切换，不影响处理器工作模式和相应寄存器内容

1. 运行模式：

User（用户模式）：正常执行模式，系统资源不可达，只能通过异常转换模式

Privileged（特权模式）：在异常中进入、系统资源完全可达、模式转换自由

System、FIQ（快速中断请求）、IRQ（中断请求）、Abort、Undefined、Supervisor

1. 寄存器结构：

32位长

r0-r15共16个用户可用，一共37个寄存器

一个PC，一个CPSR，5个SPSR，30个普通寄存器

r14：LR 存储分支指令的返回地址

r15：PC program counter

r16：SP stack pointer

CPSR：Current Program Status Register

SPSR：Saved Program Status Register



* Flash的种类与选型

NOR:

• Word-accessible read.

• Erase by blocks

NAND:

• Read by pages

(512-4K bytes)

• Erase by blocks

* 输入输出编程

忙等IO：等待前一个操作结束再进行下一个操作

#define OUT\_CHAR 0x1000

#define OUT\_STATUS 0x1001

Char \*mystring = “hello,world.”

Char \* current\_char;

Current\_char = mystring;

While(\*current\_char !=’\0’){

Poke(OUT\_CHAR, \*current\_char);

While(peek(OUT\_STATUS)!=0);

Current\_char++;

}

中断IO：I/O设备决定什么时候中断

Main(){

While(TRUE){

If(gotchar){

Poke(OUT\_DATA,achar);

Poke(OUT\_STATUS,1)

Gotchar=FALSE;

}

}

}

Void input\_handler(){

Achar = peek(IN\_DATA);

Gotchar = TRUE;

Poke(IN\_STATUS,0);

}

* 嵌入式软件体系结构

1. 驱动层
2. 操作系统层
3. 中间层
4. 应用层

* RTOS概念、特点、选型原则

1. 概念：实时[操作系统](http://baike.baidu.com/view/880.htm)（RTOS）是指当外界事件或数据产生时，能够接受并以足够快的速度予以处理，其处理的结果又能在规定的时间之内来控制生产过程或对处理系统作出快速响应，并控制所有实时任务协调一致运行的操作系统。
2. 特点：1）高精度计时系统

2）多级中断机制

3）实时调度机制

1. 选型原则：

市场进入时间

可移植性

可利用资源

系统定制能力

成本

中文内核支持

* 实时系统及任务调度（RMS、EDF算法）

RMS：任务按单调[速率](http://baike.baidu.com/view/65192.htm)优先级分配（RMPA）的调度算法，称为单调速率调度（RMS）。根据任务的执行周期的长短来决定调度优先级，那些具有小的执行周期的任务具有较高的优先级，周期长的任务优先级低。

EDF：全称Earliest Deadline First。最早截止时间优先算法（EDF）也称为截止时间驱动调度算法（DDS），是一种动态调度算法。EDF在调度时，任务的优先级根据任务的截止时间动态分配。截止时间越短，优先级越高。ucOS-II任务调度

* 嵌入式软件运行流程

