

《嵌入式系统》开卷资料提纲与目录索引

1 嵌入式系统概述 P1~P26

嵌入式系统基本概念 P2

- 引言
 - 计算机发展的三大阶段 P2
 - CPU - 计算机的核心 P3
- 嵌入式系统定义
 - def 嵌入到对象体中的专用计算机系统; P4
 - 三要素: 嵌入、专用、计算机 P4
- 嵌入式系统分类 P4
 - 按表现形式分: 芯片级嵌入、模块级嵌入、系统级嵌入
 - 按处理位数分: 8、16、32、64位
 - 按确定性要求分: 硬实时系统、软实时系统、强实时系统、一般实时系统、弱实时系统
 - 按嵌入式系统软件复杂程度分: 循环轮询、前后台、有限状态机、单处理器多任务、多处理器多任务系统
 - 按应用来分

嵌入式系统特点 P5

- 专用性强、系统精简、高可靠性、高实时性、操作系统内核小, 可裁剪, 可固化, 实时可靠、开发工具与开发环境的专用性、关注成本

嵌入式系统发展 P6

- 无操作系统阶段、简单操作系统阶段、实时操作系统 (RTOS) 阶段、面向Internet阶段

嵌入式系统应用 P7

- 系统的典型应用、未来嵌入式系统的发展趋势

嵌入式系统组成 P14

- 三层四个部分 P14
 - 嵌入式处理器、嵌入式外围设备、嵌入式操作系统、应用程序
- 四层 P14
 - 嵌入式硬件平台、硬件抽象层 (HAL)、嵌入式操作系统、应用程序
- HAL = hardware abstraction layer P14 (又称: BSP)
 - 面向硬件的抽象
- 中间件 = middleware P15
 - 应用程序与操作系统的接口

嵌入式处理器 P15

- MCU、eMPU、eDSP、SoC
- 嵌入式外围设备：分为存储设备、通信设备、显示设备
- 四个特点：支持RTOS，存储区保护，处理器结构可扩展，低功耗

ARM公司和ARM处理器简介 P20

嵌入式操作系统 P23

- 分类
 - 非实时操作系统、实时操作系统
- 实时系统：能够在指定或确定时间内完成系统功能，并且对外部和内部事件在同步或异步时间内能及时响应的系统
- 特点：系统实时高效性、硬件相关依赖性、软件固化、应用专用性

2 嵌入式系统开发流程 P27~P31

开发层次 P28

- 硬件开发
 - 包括：电路设计 / 固件开发
- 系统软件开发
 - 包括：驱动开发、Linux开发、μC/OS开发、.....
- 应用软件开发
 - 包括：应用程序、图形界面、.....

开发分工 P28

- 功能与需求分析 P28
 - 包括：系统功能部分、系统非功能部分、系统架构设计、硬件与软件协同设计
 - 负责人：系统分析师
 - 任务：根据项目要求进行功能需求分析、系统总体设计
- 硬件设计与开发 P29
 - 任务：根据项目总体设计，基于开发板/评估板进行添加、裁剪或修改
- Bootloader开发 P29
 - 负责人：驱动工程师 / 硬件工程师
 - 任务：基于硬件工程师设计的硬件平台，进行配置和移植等
- 驱动程序开发 P29
 - 负责人：驱动工程师 / 硬件工程师
 - 任务：基于硬件工程师设计的硬件平台，对硬件平台修改和新增的硬件进行驱动程序开发

- OS内核 P30
 - 负责人：驱动工程师
 - 任务：根据项目总体设计和驱动程序，对OS内核进行配置与移植
- 应用程序 P30
 - 负责人：应用工程师
 - 任务：根据项目总体设计与需求，开发和项目相关的应用程序
- 文件系统及其他选件 P30
 - 负责人：驱动工程师
 - 任务：将应用工程师开发的APP和其他系统软件等，一起制成root file system
- 嵌入式系统测试 P30
 - 任务：系统功能测试、系统性能测试、系统集成测试、样机实验、部署系统

系统设计方法 P31

- 瀑布设计方法、自顶向下、自下向上、螺旋设计、逐步细化设计、并行设计方法等

3 嵌入式系统硬件平台 P32~P65

概述 P32

- 硬件：除了核心部件嵌入式处理器外，还包括存储器系统、外围接口部件、总线系统
- 存储器、外围设备
- 嵌入式处理器在功能与通用处理器的区别：嵌入式处理器上集成了大量的IO电路

总线 P33

- 总线概念 P33
 - Bus：计算机各种功能部件间传输信息的公共通信干线，由导线组成的传输线束
 - 划分1：数据总线、地址总线、控制总线
 - 划分2：并行总线、串行总线
- 总线结构 P34
 - 经典的两种总线结构：冯·诺伊曼、哈佛
 - 改进型哈佛总线结构：2条而不是4条：合并了两个存储器的地址总线(程序地址总线、数据地址总线)，以及合并了2条数据总线(程序数据总线、纯数据总线)
 - 数据总线：宽度又称计算机“位数”，代表计算机系统处理数据的能力
 - 程序总线：决定CPU每次能够从程序存储器取出多少位指令，宽度决定计算机系统指令集的效率
 - 数据地址总线、程序地址总线：对数据存储器或程序存储器进行寻址
- 总线协议 P35
 - 握手协议
 - 基本形式：四次握手协议
 - 总线读写

- 总线行为常用时序图说明，时序图表示总线上的信号如何随时间变化
 - 总线事务的状态机 P36
- DMA P37
 - 使用DMA控制器这一专用硬件完成存储器/外设之间的高速数据传送
 - 能像CPU那样提供存储地址和必要的读写控制信号，实现直接在存储器/外设之间执行读写操作
 - 总线主设备、总线从设备、仲裁器
- 总线配置：一个微处理器系统常使用多条总线来连接设备 P38

存储设备 P38

- 基本概念 P38
 - 存储器：用于存放计算机工作所必须的数据和程序
 - 嵌入式微处理器运行时大部分总线周期都是用于对处理器的读/写访问
 - 追求存储器的高性能：1. 存储单元的设计、制造；2. 存储器系统结构的探索、优化
 - 存储器子系统设计首要目标：使存储器在工作速度上很好地与处理器匹配，并满足各种存取需要
- 存储器子系统分级 P39
 - 寄存器组 P39
 - CPU内含，待使用数据或运算的中间结果可以暂存在这些寄存器中，为尽可能减少CPU从外部取数的次数
 - 高速缓存 P40
 - 小型、快速的存储器，其存取速度足以与CPU匹配；保存部分内存内容的拷贝，这部分内容被高频率使用
 - 内存 P40
 - 运行的程序和数据存放在内存中。内存除主要使用RAM外，还要使用一定量的ROM
 - 外存 P40
 - 最低一级存储器，容量大，但在存取速度比内存慢得多
- 存储单元设计与制造 P40
 - RAM 随机存取存储器 P40
 - 用于数据临时存储或缓冲存储；读写方便、使用灵活 P41
 - 分类：静态RAM、动态RAM(DRAM) P41~P42
 - 如何选择RAM P43
 - SRAM：系统随机存储器容量不大、特别高速度应用、嵌入式系统对功耗敏感
 - ROM 只读存储器 P43
 - 已经写入，在工作过程只能读出不能重写
 - 用途：存放系统软件、应用程序等不随时间改变的代码，或常量数据
 - 划分依据：工厂可编程ROM、现场可编程ROM
 - Flash P44
 - 快闪存储器，广泛使用于内存和外存 P44
 - 闪存与EEPROM的区别 P44
 - Flash：使用单一供电电压擦除编程，而EEPROM擦除、编程时加高电压

- EEPROM可单独擦写、编程某个存储单元，而Flash必须擦除一个存储块
- 大部分Flash允许某些块被保护
- Flash可以有非常大的容量
- 分类：NOR Flash, NAND Flash P45
- 编程：在线编程、离线编程；一次至少有一个扇区或块被擦除，而不可能只是擦除一个单个的字节 P45
- Flash的嵌入式应用：Flash文件系统、便携设备的存储装置、内存中的ROM P46

I/O设备 P46

- 分类 P46
 - 输入设备、输出设备、数据控制与转换设备
- 定时器/计数器 P46
 - 组成：带有保存当前值的寄存器和向当前寄存器值加1或减1的加法器/减法器逻辑电路组成
 - 定时器与计数器的区别 P46
 - 嵌入式处理器上的定时器/计数器的作用 P47
 - 扩展 P47
 - PWM、捕获
- ADC / DAC P47
 - 注意：两者均需要一个参考电压，决定可数字化的最大模拟输入电压值，或者可转换输出的最大模拟电压值
 - DAC关键技术指标 P48
 - ADC关键技术指标 P48
 - ADC主要类型：并行比较型、逐次比较型、 Σ - Δ 型、积分型、电压-频率型
 - 电路构造：采样/保持电路
- 人机接口设备 P48~P50
 - 键盘、LCD显示屏、触摸屏

通信设备 P51

- UART = 通用异步收发器 P51
 - UART：典型的用于计算机与异步串行设备的接口 P51
 - 通信相关概念
 - 数据通信模式：单工和双工通信、串行和并行通信、同步和异步通信 P51~52
 - 异步与同步通信的差别：各数据线传输位的定位是否由收发双方共有的时钟信号来定位 P52
 - 异步通信：需要约定好波特率 P52
 - 电气接口 P53
 - RS-232C, RS-422, RS-485 P53
 - UART设备：提供对RS-232C等数据终端设备的接口，提供大量功能 P55
- USB设备 P55

- USB = universal serial bus, 一种计算机外部串行通信标准
- USB总线: 基于令牌的总线 P56
 - USB总线的硬件结构: Vbus, D+, D-, GND
- USB系统 P57
 - 采用级联星形拓扑, 由三个基本部分组成: 主机、集线器 (hub)、功能设备
- USB总线的软件结构 P58
 - 包括USB总线接口、USB设备层、功能层三层结构
 - 主机端: USB总线接口、USB系统软件、USB客户软件
- USB总线的数据传输方式 P58
 - 管道传输方式, 包括以下四种: 同步、控制、中断、批量
- USB设备实现热插拔
- USB器件的选择 P59
- USB系统的开发流程
- 蓝牙设备 P59
 - 分类
 - 单模蓝牙 BLE: 低成本、低功耗、快连接、长距离 P59
 - 工作流程: 主机、从机、观察者、广播者 P60
 - 广播: 从机每经过一个时间间隔发送一次广播数据包, 该时间间隔称为广播间隔, 该动作称为广播事件 P60
 - 扫描: 主机监听从机广播数据包、发送扫描请求过程 P60
 - 扫描动作的时间参数: 扫描窗口、扫描间隔
 - 扫描的两种类型: 被动扫描、主动扫描
 - 连接 P61
 - 连接事件: 用于收发数据的相遇事件
 - 通信 P61
 - 主机主动向从估计Write或Read数据
 - 从机主动向主机Notify通知数据
 - "Profile": 从机所具有的数据或属性特征
 - 断开 P61
 - 双模蓝牙: 支持BLE的同时还兼容经典蓝牙, 支持音频、数据两大类协议
 - 蓝牙的体系结构
 - 控制器、主机、应用

其他 (电源、时钟、复位、中断) P62

- 电源 P62
 - 开关电源: 不同于线性电源, 所利用的切换晶体管多半在全开模式和全闭模式间切换
 - 现代开关电源的种类: 直流、交流开关电源
 - 直流开关电源核心: DC/DC转换器

- 隔离式DC/DC转换器
- 非隔离式DC/DC转换器
 - 按照有源功率的个数，可以分为：单管、双管、四管三类
- 充电泵
- 时钟 P63
 - 时钟电路的几种形式
 - RC振荡器、石英晶体振荡器、外部时钟输入、锁相倍频时钟
- 复位 P63
 - 指使得器件进入到一个确定的初始状态，并且使处理器从起始地址开始执行指令
 - 常用的复位电路
 - 阻容复位电路、专用复位电路、手动复位电路、看门狗定时器超时复位、软件复位
 - 看门狗定时器：一种特殊用途的定时器。当定时器计数值达到所设定时长时，引起处理器复位；需要及时喂狗
- 中断 P64
 - 中断机制的提出
 - 定时查询方式、中断方式的区别
 - 中断的三个过程 P64
 - 请求过程、响应过程、服务过程
 - 中断的基本概念 P65
 - 中断源、中断断点（地址）、中断服务程序、中断向量/中断矢量
 - 中断源的分类、中断系统的功能
 - 使用中断的优点 P65
 - 同步操作、实现实时处理、故障处理

4 嵌入式系统软件开发工具 P66~P81

嵌入式系统软件开发工具分类 P66

- 开发过程：需求分析、设计、编码、测试、发布与维护 P66

嵌入式软件交叉开发环境 P67

- 交叉开发方式 P67
 - 交叉开发环境：文本编辑器、交叉编译器/链接器、交叉调试器、仿真器/下载器等配套软硬件工具
 - 宿主机、目标机 P67
 - 宿主机与目标机的区别
 - 为什么要交叉编译？目标机上往往无法进行有效率的本地（native）编译 P68

- *3

- 编程语言

嵌入式软件实现阶段的开发过程 P68

- 总体三步骤：生成、调试、固化运行
- 生成阶段 P69
 - 步骤
 - 源代码程序编写 P69
 - 编译成各个目标模块
 - 链接成可供下载调试或固化的目标程序
 - 生成：直接将源代码文件转换成可执行文件的过程 P69
 - 定址和打包：将可重定位执行文件整合为一个可运行在嵌入式系统中的二进制映像文件 P70
 - 可执行文件的可重定位和不可重定位 P70
- 调试阶段 P70
 - 最大问题：缺少监控程序实时运行的手段
 - 交叉调试器 P71
 - Crash and Burn
 - 最早的嵌入式软件调试方法
 - ROM monitor P72
 - ROM emulator：目标机的ROM芯片被外部替代 P73
 - ICE = in-circuit emulator，一种用于替代目标机上CPU的设备 P73
 - OCD = on-chip debugging P74
 - CPU运行模式分为一般模式和调试模式
 - 各种OCD的实现：BDM、OnCE、JTAG
 - 纯软件Simulator P76
 - 测试工具 P77
 - 内存分析工具、性能分析工具、覆盖分析工具、缺陷跟踪工具等 P77~P79
- 固化运行阶段 P79
 - 调试环境与固化环境的主要区别：代码定位不同、初始化部分不同 P79
 - Boot模块、主要功能：应用程序在真实目标环境下运行时，首先执行，至少由系统加电时执行的代码组成 P80

嵌入式软件开发工具的发展趋势 P81

- 向着开放的、集成化的方向发展
 - 具有系统设计、可视化建模、仿真与验证功能
 - 自动生成代码和文档
 - 具有更高的灵活性

5 嵌入式系统软件设计基础 P82

单片机的C语言简述 P82

- 为什么使用C语言？大幅减少编程劳动量、代码通用性
- 具体：变量定义、特殊寄存器操作、位操作、中断、内部函数、函数的可重入性 P83~85

程序设计规范 P85

- 编程风格、可移植性、版本管理与可配置性 P86~89

前后台多任务程序设计 P90

- 前后台程序的基本概念 P90
 - 任务、后台程序、前台程序
- 前后台程序的编写基本原则 P90
 - 任何一个任务都不能阻塞CPU
 - 关注函数重入问题
 - 临界代码保护
 - 依靠软件产生时间严格时序的程序段、共享资源互斥性造成的临界代码、函数重入造成的临界代码、CPU字长造成的临界代码
 - 具体的保护方法：开关中断、硬件栈保存中断使能状态、利用变量保存中断使能状态、利用模拟栈保存中断使能状态

FSM：状态机建模 P92

- FSM = finite state machine，表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型 P92
- 为什么要引入状态机 P92
 - 能够描述并发结构的软件，能从行为的角度描述软件，且能根据模型生成代码，也能够对软件进行完整测试的新手段
 - 优势：能够处理并发（大量独立事件）、能够消除阻塞、能够降低系统复杂度、提高可测性
- 表示法：状态转移图 P93
- 由状态转移图生成代码 P94
- 状态机建模举例 P94~95

模块化程序设计 P96

- 模块化程序的原则：非阻塞性、硬件隔离、模块独立性、时间独立性、可移植性、100%注释、开放性

事件触发多任务程序设计 P97~98

- 什么是事件触发程序？任务全部在中断内完成，主程序休眠。可以视作前后台程序中，只有前台任务。
 - 特点：1. 实时性较好，事件响应较快；2. 低功耗
- 程序架构：事件 → 中断 → 事件引擎 → 事件处理程序

时间触发系统 P99

- 解决：使用调度器，可以视作一个简单的操作系统，允许以周期性或单次方式来调用任务 P99
- 从底层角度看，调度器可以视作是一个由许多不同任务共享的定时器中断服务程序
- 调度器分类：合作式调度器、抢占式调度器、混合式调度器 P100

end