**嵌入式系统安全内容**

**概论**：嵌入式系统安全概论

**可靠性**：80C51架构、PIC架构、ARM体系架构、接口技术、软件设计及代码优化、实时操作系统

**安全性：**COS、旁路攻击及防御、工业控制系统信息安全

**新技术**：TrustZone技术

嵌入式系统安全 = 可靠 + 安全

**嵌入式系统定义：**

IEEE的定义：嵌入式系统是用于控制、监视或辅助操作机器和设备的装置。

一般认为嵌入式系统是以应用为中心，以计算机技术为基础，并且软/硬件可裁剪，可满足应用系统对功能、可靠性、成本、体积和功耗有严格要求的专用计算机系统。

简单地讲：嵌入式系统就是嵌入到对象体系中、用于执行特定功能的专用计算机系统。

**嵌入式系统三要素**（嵌入性、专用性、计算性）

**嵌入式系统特征**（特定应用、软硬件可裁剪、低功耗、低成本、体积受限、实时性、较长的生命周期、本身不具备自主开发能力，需特定开发工具（仿真机、开发器））

**嵌入式系统组成**（微处理器、外围支撑硬件、嵌入式实时操作系统（或调度器）、用户应用软件）

由于嵌入式系统存储空间有限，要求软件代码紧凑、可靠，对实时性有严格要求。

嵌入式微处理器的体系结构

**冯·诺依曼 (von neumann)/普林斯顿(Princeton)体系结构**

程序存储器和数据存储器共用一个存储空间，统一编址；

采用统一的地址及数据总线，指令和数据的宽度相同；

使用灵活（例如 代码远程更新OTA）。

**哈佛体系结构**

程序存储器和数据存储器是独立编址的两个存储空间；

这种分离的程序总线和数据总线可允许在一个机器周期内同时获取指令码（程序存储器）和操作数（数据存储器），从而提高执行速度，提高数据的吞吐率，可靠性高（大部分程序存储器是只读存储器）。

**嵌入式微处理器指令系统**

复杂指令集系统(CISC)

精简指令集系统(RISC)

CISC与RISC之间的主要差异

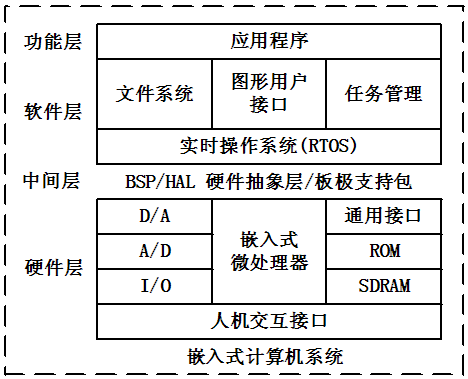
嵌入式系统芯片分类(8/16/32位、MPU/MCU/DSP/SOC)，8位一般是哈佛结构，成本低，可靠性高。32位一般是冯·诺依曼 /普林斯顿体系结构，成本高，灵活性好。

电磁兼容性二要素(能在一定的干扰环境工作、不产生不可容忍干扰)

软件分层

嵌入式操作系统

不是必需的。



**ROM/Flash/OTP差异**



**嵌入式系统软件的要求与PC机有所不同**，其主要特点有：

软件固化存储，修改不易，要有较高的正确率和可靠性；

软件代码要求精简（受成本、体积和功耗存储空间限制）、高效（受主频、功耗限制）、高可靠性（容错）

数据结构简洁（代码优化时，数据结构占80%，编程技巧占20%）

**80C51架构**

**51单片机逻辑结构**：8位、哈佛结构、复杂指令集、有累加器“ 瓶颈”

仿真器开发

**程序存储器**（介质类型、容量、保密及攻击方法、复位及中断入口）

**数据存储器** 通用工作寄存器组R0～R7及应用

**定时/计数器**（数量、功能）

**中断系统**（实时系统通过中断实现、中断种类、数量、优先级、入口模式）

**时钟电路**：振荡周期/节拍、时钟周期/状态、机器周期和指令周期

**指令系统**：位寻址、程序状态字PSW

**扩展**：地址、数据和控制三总线

引脚复用与配置

**51单片机工作方式**： 复位 、程序执行、低功耗（节能、规避干扰） 、编程和校验

**电源**：三类（ A:5V B:3.3V C:1.8V）、误差± 10%，少量芯片± 5%，宽电源芯片适用性好，电压高抗干扰能力强、电压低功耗低）

**振荡器**：内部（不精准）、外部（晶体振荡器、陶瓷振荡器、RC振荡器差异）、工作频率与功耗控制（节能模式）、信号占空比1：1

**复位**：内/外部复位、电平及复位时长要求、特殊功能寄存器赋值与可靠性关系

**PIC架构**

8位、哈佛结构、精简指令集、有累加器“ 瓶颈”

在线串行编程 (ICSP)

8级深硬件堆栈、芯片配置字

PIC 中断、中断分组与控制、统一中断入口、边沿触发、电平变化中断

LDO、程序闪存(编程保护，擦写1000次，数据保持时间>40年)

端口结构(钳位二极管、模数选择、方向选择、上拉电阻选择)

振荡器 (内部振荡器、软件可调)、配置与选择、指令周期1:4

复位噪声滤波器、上电复位(POR)、上电延时时器 (PWRT)和振荡器起振定时器 (OST)、欠压复位 (BOR)、复位标志

通用端口电平要求：H ≥70%\*Vcc L ≤ 30%\*Vcc

复位电平要求：H ≥80%\*Vcc L ≤ 12%\*Vcc

信号上升时间：10%\*Vcc上升到90%\*Vcc所需时间

信号下降时间：10%\*Vcc上升到90%\*Vcc所需时间

好的信号要求信号上升时间与下降时间 ≤ 信号周期\*4%

功耗控制 P=K\* f² ，分部件控制时钟

死机的本质、WDT看门狗与可靠性

推挽输出与OC（OD）输出（ OC（OD）输出优缺点）

程序存储器的分页(PAGE)

数据存储器的分页(BANK)

**存储器分页的优缺点**：

优点：可靠性高，跨页出错机会小；产品扩展方便。

缺点：处理页切换浪费空间和时间，要合理安排。

**定时/计数器:**预分频和后分频、门控模式、捕捉(捕获)、比较、PWM及死区控制

同步串行端口(SPI 、I2C )

mTouchTM 触摸传感振荡器模块、A/D 转换器

**ARM架构**

Cortex系列（ARM新的命名体系）

Cortex-A: 高性能, 丰富的功能

Cortex-R: 高可靠性, 高实时应用

Cortex-M：低功耗, 代替微控制器（单片机）

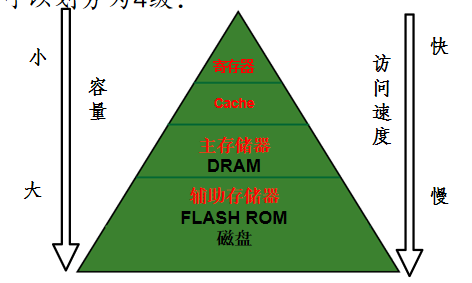
**SecurCore: 安全应用**

ARM 体系结构（Thumb指令优缺点、饱和的带符号数的加减法、ARM工作模式、三总线）

ARM 存储器（统一编址、MMU、寄存器及物理映射、大端/小端存储、主存/辅存种类）

存储系统的层次结构

存储器是用来存储信息的部件，是嵌入式系统硬件中的 重要组成部分。在复杂的嵌入式系统中，存储器系统的 组织结构按作用可以划分为4级：



**存储管理单元**（MMU）是集成在微处理器芯片内部、专 门管理外部存储器总线的一部分硬件。**主要用来完成虚实 地址和物理地址之间的转换**。目前，越来越多的微处理器 芯片均带有存储管理单元（MMU）。MMU完成的主要功能有：

将主存地址从虚拟存储空间映射到物理存储空间。

存储器访问权限控制。

设置虚拟存储空间的缓冲特性等。

ARM 时钟及电源管理(时钟分配、总线时钟FCLK/HCLK/PCLK、锁相环PLL、 S3C2410有4种电源管理模式/电源设计/振荡器设计 )

ARM 定时技术（定时器结构、死区发生器作用、RTC类型与读取）

ARM 中断（独立向量、分组向量、统一向量的异同与管理、普通中断、快速异常中断）

ARM DMA（关注DMA源与目的、块与字符、中断性能、对CPU时序的影响）

**接口技术**

**硬件**:通信(相互通道)、输入(前向通道)、输出(后向通道)

**部件**:传感器、执行器、人机界面、相互通信、电源

UART:

单工通信、半双工通信、全双工通信

串行异步协议（起始位、数据、校验、停止、波特率、发送/接收缓存数量、中断种类）

RS232：距离≤ 15m，数据速率≤20Kbps

RS485：半双工差分双绞线传输，有极性、距离≤ 1219M（ 100Kbps ），最大传输速率为10Mbps。

RS485应用注意事项：极性、收发切换时间、最远端端匹配电阻（各110欧）、T型线长度（<3.4M）、线路保护

**串行通信协议应用注意事项**

1、接收与死机（超时处理、重新初始化、接收缓冲区防溢出）

2、停止位（又叫数据保护时间）位数与可靠通信（有效通信速率、保障接收方的数据处理时间）

3、波特率精度（ ±2.5%、外置振荡器、配置选择与通信精度）与可靠通信

4、通信协议、帧格式、收发切换时间及校验

5、帧间隔（字符帧、通信协议帧）、帧长度

6、波特率与通信距离

I2C总线最主要的优点是其简单性和有效性总线的长度可高达25英尺(约7.6m)，并且能够以100Kbps的最大传输速率支持40个组件。支持多主控。每个设备有唯一地址。

I2C总线是由数据线SDA和时钟SCL构成，两根信号线必须OC并接上拉电阻。同步总线。

起始、数据、停止、确认

标准模式位速率最大100Kb/s

快速模式位速率最大400Kb/s

高速模式位速率最大3.4Mb/s

**I2C应用注意事项：**

1、ACK与死机（标准为永久等待，实际要加超时判错）

2、帧间隔与正常通信（保护时间）

3、不是所有芯片都支持广播地址

4、7位与10位地址

5、SCL、SDA 一定要开路输出（注意配置端口）

6、时钟速率与通信距离、EMC

7、结束与复位

8、方向切换时间

**SPI 串行外设接口总线**

5Mb/S 、CS/MOSI/MISO/SCK、具体时序由从设备决定、适用于大数据量（FLASH显示高速通信）

**1-WIRE(单总线)**

数据表示、复位与初始化、ROM 功能命令、功能命令、数据

**人机接口**

按键：去抖动、独立与矩阵（延时、OC）、AD键盘

LED显示：独立、扫描、消隐、散热、寿命、伽马校正、呼吸灯

蜂鸣器:分类、驱动、保护。蜂鸣器的电磁干扰。

功率晶体管（电流型、可靠性高）

场效应管（电压型、速度快）

晶闸管（可控硅、交流、单/双向、隔离（脉冲变压器、光耦）、过压/流、过零、周期/导通角控制、电压电流同步）

电磁继电器（动作频率： ≤5次/秒、驱动及保护（保护二极管、灭弧））

固态继电器（输入、隔离、输出、保护、类型）

步进电机（时序（单四拍、双四拍、八拍）、失步、注意事项（缓升缓降、齿轮间隙、皮带张紧））

BLDC马达

Vm（电机动力电源）、Vp（电机控制电源）、Vsp（PWW占空比调速、频率）、FG（转速脉冲，n个/转）

加载时序： Vp加载→ Vm加载→ Vsp加载

卸载时序： Vsp卸载→ Vm卸载→ Vp卸载

**嵌入式软件设计及操作系统**

嵌入式软件设计的演变（顺序设计、状态机程序设计、简易任务调度器程序设计、操作系统的程序设计）

任务切换（栈指针切换/堆栈搬移、公栈/私栈）

调度器种类（合作式调度器、抢占式调度器、混合式调度器）

嵌入式软件架构与层次（硬件层、驱动层、系统层、应用层）分层便于在不同的CPU、不同OS间移植，节省开发时间，提高软件可靠性。

实时多任务与分时多任务操作系统 (实时、时间确定性)

实时操作系统中的重要概念（时钟节拍、系统响应时间、任务切换时间 、中断延迟时间）

嵌入式Linux(种类、启动、存储及文件系统、写均衡)

**RTOS优缺点**(优点：实时、多人协作开发、程序结构简化；缺点：更多资源、成本、时效、能耗)

什么时候该使用RTOS？（任务耗时、复杂、资源充足、低功耗与RTOS）

实时系统概念（软实时系统、硬实时系统）

代码的临界段（关中断、影响中断响应时间、防数据撕裂）

资源（可以是硬件设备，也可以是软件、存储空间；共享资源、互斥）

任务（优先级（静态/动态、反转、）、5种状态、切换）

调度 (不可剥夺型、可剥夺型、可重入型函数、时间片调度、死锁)

中断响应时间 、中断处理时间、中断恢复时间、非屏蔽中断

影响中断时间的处理方法（任务外移与通知、时间轮、查表、前导0指令）

相对时间、绝对时间

存储器的需求（局部变量；函数嵌套 ；中断及嵌套 ；库函数需要的栈空间 ）

**实时内核的优缺点**

COS特点（快、省、安全、防数据撕裂、存储器管理）

COS任务（数据交换、命令解释与执行、文件管理和数据安全、密码算法管理）

COS要解决的问题（文件操作、鉴别与核实(鉴别设备,核实持卡人)、安全体系(状态,属性,机制)）

COS的四个模块（传递、安全、应用、文件）

COS的程序结构（完工操作、跳转表、硬件识别、自检、掩膜与软掩膜）

文件系统（主文件MF、专用文件DF( DDF/ADF）、基本文件EF、文件结构、访问条件、防拔、文件属性、文件头(不变)、文件体(可变)、页面管理、）

COS数据传输（正向逻辑、负向逻辑、ETU、 ATR、 PPS、命令APDU、 T=0 、 T=1）

COS命令集

**嵌入式软件设计**

嵌入式软件架构与层次（硬件层、驱动层、系统层和应用层）

代码优化（平衡、利用硬件中断/DMA）

小变量类型/寄存器变量、利用结构体和联合、局部/全局变量

算法优化(查表、求余、平方、公共表达式、数学方法)

宏代码、内嵌汇编

循环语句的效率（分解、指针、判0、 ≥ 、循环展开/集中）

switch （跳转表和比较链/树、嵌套）

自增/减和复合赋值、优化赋值

减少浮点

函数优化（减少参数/返回值、原型定义、本地函数、尽量避免使用标准库））

代码可靠性：

程序存储器（复位区分/恢复/延时、 WDT、参数/执行检查、陷阱与错误捕获、 SLEEP 、变频、自检、校正）

RAM（数据单元、备份与空间距离、内存清理）、

EEPROM/FLASH（数据单元、多编码备份、密文存储 、防拔、寿命与页面管理）

输入（周期性配置/输入/滤波、寿命管理）、

输出(周期性刷新内容/配置、寿命管理)）

**旁路攻击 SCA及防御**

密码攻击(数学攻击、实体攻击、实现攻击(主动式、被动式(旁路攻击)))

主动攻击((逃避检查、减少密码运算轮数、非法指令、 破坏RNG)(降/升时钟、异常电压/时钟/复位、光/射线/温度)

被动式攻击(提取泄漏物理量(温度/声波/能量/时间/电磁波/光)分析出密钥)

旁路攻击的必要条件(足够多样本/与密钥相关性)

旁路攻击分类(时间/能耗/声音/光/电磁波)

旁路攻击分析技术(故障分析/侵入分析/时间分析/简单功耗分析/差分功耗分析/电磁辐射分析/高阶差分分析/汉明差分分析/模板分析)

防御立足点(消除泄漏/增加难度(增加噪声/减小有效信息量)

电路级抗攻击(互补电路/电路级Masking/预充电/随机化)、安全芯片

软件防御技术：

数据冗余(数据单元/备份)

控制冗余(前序/入口检查)

执行冗余(随机顺序/等价实现/随机延时功耗)

特定算法

软件陷阱

SLEEP躲避

软件防御技术关键是代码分块的粒度粗细和等价实现的数量

**工业控制系统信息安全**

**工控系统安全介绍**(工业控制系统(ICS) 、可编程逻辑控制器(PLC)、监控和数据采集(SCADA)系统、分布式控制系统（DCS))

**工业控制系统特点**(实时性、可用性、长寿命、可靠性、私有协议、封闭性)

**工业控制系统功能层次**(ERP(商务)、MES(制造管理)、SCADA/DCS、PLC、I/O）工控系统安全威胁(震网病毒、工控系统安全事件;“两化”融合,系统漏洞已知与无法修复、无法杀毒、非工控应用软件带来未知风险、移动介质和笔记本随意接入、工业无线网络边界不可见,网络安全风险急速增加)

**工控系统的脆弱性**(由于兼容性的问题，系统补丁和杀毒等安全措施不到位；设计主要考虑可用性/实时性，对安全性考虑不足，缺乏安全机制；工控系统用户信息安全意识薄弱，缺少完善管理)

**工控系统安全理念**(白名单、层次化、边缘化、透明化)

**工控系统安全策略**(纵向分层、横向分域进行安全隔离防护，可用性监控、行为监控、指令监控)

国际上有两种不同的工控系统信息安全解决方案:

主动隔离式解决方案、被动检测式解决方案

安全不是一个结果，而是一个过程。

**TrustZone技术**

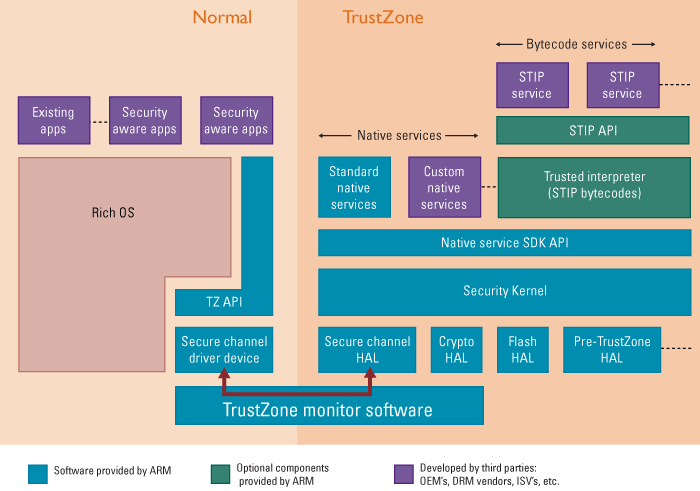
尽量不影响原有处理器设计的情况下提高了系统的安全性

ARM TrustZone 技术是系统范围的安全方法

从ARMv6架构的处理器开始就已经集成了TrustZone技术，主要应用于Cortex-A 和 Cortex-M 系列处理器

TrustZone技术的硬件架构（特殊的机制-监控模式）

它将CPU内核隔离成安全和普通两个区域，即单个的物理处理器包含了两个虚拟处理器核：安全处理器核和普通处理器核



普通环境想要进入监控模式是严格被控制的，仅能通过以下的方式：中断、外部中断或直接调用SMC（Secure Monitor Call）指令。

安全环境进入监控模式则更加灵活些，可以直接通过写程序状态寄存器（Current Program Status Register，CPSR），另外也可通过异常机制切换到普通环境。

为了实现安全性的要求，TrustZone形成了TOS和ROS同时运行在CPU上的两套操作系统。TrustZone启动时，安全引导程序从SoC的Rom中运行，进入TEE执行环境初始化，并启动TOS，逐级检查TOS各阶段的关键代码以保证TOS完整性；随后运行REE的引导程序，并启动ROS，至此完成整个系统的安全引导过程。

**TrustZone３种方式的完整性安全策略：**

首先，它会先从片内执行引导程序完成系统安全状态的配置才启动操作系统，只有通过安全验证的模块才允许被加载；

其次，在系统运行期间，由TrustZone技术提供的安全代码区会处理普通代码区的安全请求，在处理之前把安全请求保存在共享内存中，当安全检测通过后请求会被处理；

最后，一组受限的、可信的进程可以在远离ROS的私有空间内安全地执行。

**TrustZone架构存在问题**

TrustZone镜像未加密存储

物理指针使用频繁

缺少严格的IO输入输出控制与检测

缺少ASLR, DEP机制

**TrustZone采取的安全机制不完善**

在安全功能实现中加入部分验证

部分厂商实现时使用位掩码屏蔽不再使用的功能