

# 嵌入式系统综合实践

范捷 西北大学 信息科学与技术学院

# 实践目的

## ●活动简介

- 使学生学习嵌入式系统的相关技术，理解嵌入式系统的设计过程，掌握mbed在线编译器的使用方法。在课程实践中开发一个实用嵌入式系统。

## ●实践目的

- 本次实践旨在在有限的时间内，帮助学生搭建嵌入式开发所需要的知识体系和技能树，通过mbed快速掌握嵌入式系统设计的基本方法，在万物互联时代下，为毕业后从事软件开发、硬件设计、嵌入式产品开发或与嵌入式系统相关的工作打下坚实的基础。

# 提纲

- 嵌入式系统
- 嵌入式系统设计
- Cortex M处理器架构与mbed微控制器
- 实验
- 大作业

# 嵌入式系统

# 什么是嵌入式系统

- 嵌入式系统的定义

- 嵌入式系统是硬件与软件的结合，用来实现特定的功能。

- 嵌入式系统的类型

- 微控制器
- 微处理器
- 系统级芯片（SOC）
- 现场可编程逻辑门阵列（FPGA）

# 应用领域

- 基本上今天生产的任何产品
- 汽车、火车
- 电信
- 制造业
- 智能建筑
- 机器人



# 汽车嵌入式系统

- 今天的高端汽车可能有超过100个微处理器
- 4位微控制器检查安全带
- 16/32位微处理器控制引擎
- 制动系统（ABS）。控制器持续监控每个车轮的转速，并控制制动器的液压。它允许车轮与道路保持接触，防止车轮抱死（停止转动）。
- 主要元件：一个控制器，速度传感器，制动器，水泵

# 嵌入式系统设计



# 嵌入式系统设计的挑战

- 复杂的功能
- 实时操作
- 低制造成本
- 低功率
- 通常要在多个平台上运行复杂的算法
- 通常需要提供复杂的用户界面

# 时间限制

- **实时嵌入式系统**是指系统的正确性不仅取决于逻辑计算结果，还取决于结果产生的时间的系统。
- **硬实时系统**（例如航空电子控制）
  - 硬期限：由于错过期限而受到的惩罚比在期限内得到的奖励要高很多。
- **软实时系统**（例如视频点播）
  - 软期限：因错过期限而受到的惩罚比奖励小。

# 汽车的例子

- 任务：安全抵达目的地
- 受控系统：汽车
- 操作环境：路况
- 控制系统
  - 人类驾驶员：传感器——驾驶员的眼睛、耳朵。
  - 计算机：传感器——摄像机，红外接收器，激光遥测仪。
- 控制：加速器，方向盘，制动踏板
- 制动：车轮，引擎，制动器

# 汽车的例子

- 关键任务：转向和制动
- 非关键任务：打开收音机
- 完成任务的成本→有效的解决方案
- 驾驶员的可靠性→需要考虑容错

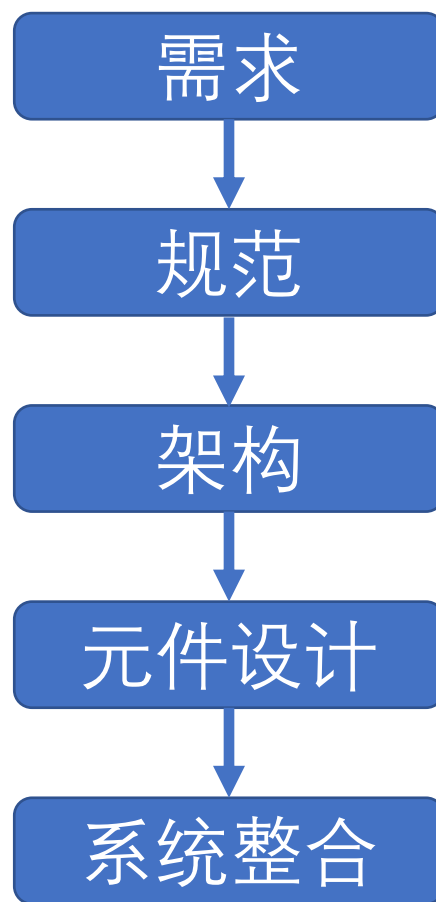
# 节能

- 现代微处理器提供有助于控制功耗的功能，如关闭未使用的电路和动态调整时钟速度。
- 软件设计技术能够减小功耗。许多软件电源管理算法通过结合硬件支持实现某种形式的电压或频率缩放来实现这一设计目标。
- 动态电压缩放是一种有效的低能耗技术，它能够在程序运行时动态地改变处理器的电压和频率，在不影响处理器峰值性能的情况下有效的减少处理器的能耗。

# 嵌入式系统设计方法

- 自上而下的设计：从最抽象的描述开始；工作到最详细。
- 自下而上的设计：从小部件到大系统的工作。
- 实际的嵌入式系统设计同时使用这两种技术。

# 嵌入式系统设计方法



# 需求

- 使用简单的语言对客户需求进行非正式描述。它可以在采访客户、与营销代表交谈或从客户那里获得反馈时开发。
- 需求阶段将顾客与设计者连接起来。



# 创建需求

- 客户访谈。
- 与竞争者对比。
- 销售反馈，与营销代表交谈。
- 原型，向用户提供原型以供评论。
- 影响需求生成的五个因素：
  - 利益相关者（最终用户、客户、经理、工程师、维护专家和认证机构）
  - 技术限制
  - 行业标准
  - 质量保证
  - 销售和营销

# 好的需求

- 正确
- 毫不含糊
- 完整（包括所有的需求）
- 一致性：需求不相互矛盾。
- 可验证：系统中是否满足每个需求？
- 可修改：应结构化，并能轻松更新需求。

# 需求表

- 知道为什么每个需求会存在
- 从源文件到需求
- 从需求到应用
- 从应用回到需求

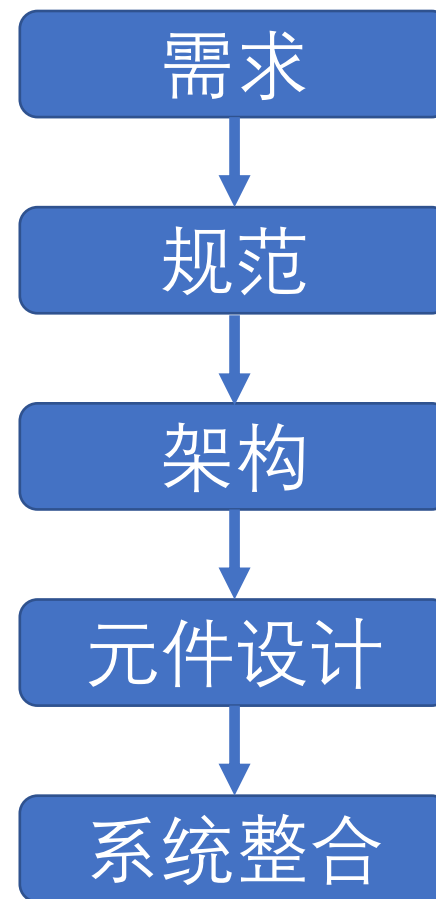
名称	
目标	
输入	
输出	
功能	
性能	
制造成本	
物理大小/重量	
功率	

# 规范

- 系统的更精确描述应为架构设计过程提供输入。
- 捕获功能性和非功能性属性
  - 功能性：输入/输出关系
  - 非功能性：时间，功耗，制造成本，物理大小、重量等，上市时间，可靠性
- 许多规范类型：
  - 面向控制与面向数据
  - 文字与图形

# 架构设计

- 描述满足规范的主要元件
- 硬件元件：微控制器，外围设备，等
- 软件元件：主程序及其操作
- 需要考虑功能性和非功能性规范



# Cortex M处理器架构与mbed微控制器

# Cortex M处理器

- Cortex M处理器是哈佛结构，有单独的数据与指令总线。
- 使用ICode总线从闪存ROM获取指令。数据通过系统总线接口与内存和I/O交换。
- 嵌套矢量中断控制器（NVIC）管理中断，中断是硬件触发的软件功能。一些内部外围设备，如NVIC，通过专用外围总线（PPB）直接与处理器通信。处理器和中断控制器的紧密集成提供了中断服务例程（ISR）的快速执行，大大减少了中断延迟。

# mbed NXP LPC1768

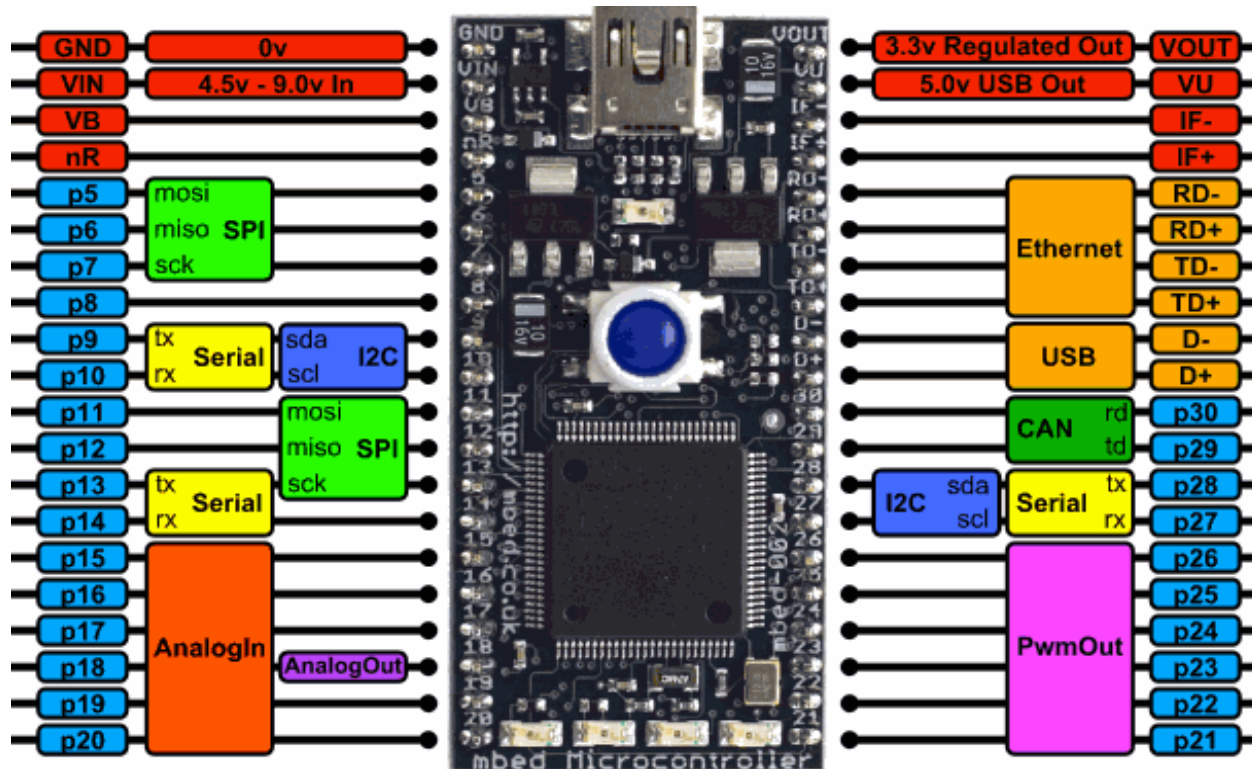
- NXP LPC1768是一个基于ARM Cortex-M3的微控制器。
- 设计用于各种设备的原型开发，包括以太网、USB，许多外围接口的灵活性。
- 包括内置USB闪存编程器。





# mbed NXP LPC1768

- 以96MHz运行的32位ARM Cortex-M3内核。
- 512KB闪存, 64KB RAM。
- 内置以太网、USB主机和设备、CAN、SPI、I2C、ADC、DAC、PWM等I/O接口。
- 为经验丰富的嵌入式开发人员提供一个强大且高效的平台，用于构建概念验证。



os.mbed.com

# 总结

- 嵌入式系统无处不在。
- 许多系统都有复杂的嵌入式硬件和软件。嵌入式系统带来了许多设计挑战：设计时间、截止日期、电源等。
- 设计方法有助于管理设计过程。
- 嵌入式系统设计过程有五个抽象层次。

# 实验

# 实验1

- 1-1 使LED以0.5s的速率闪烁。
- 1-2 在串行终端中显示 “Hello World” 。
- 1-3 使4个LED以不同的频率闪烁（0.2秒、0.5秒、1.0秒和2.0秒） 。  
在串行终端中打印每个LED的状态。

# 实验2

- 2-1 定时器中断。
- 2-2 在LCD中显示 “Hello World” 消息。
- 2-3 使用定时器中断设置LED闪烁频率，当LED亮起时显示“XXXX”，当LED熄灭时显示“OOOO”。可以选择四种闪烁频率：0.2秒、0.5秒、1.0秒和2.0秒。

# 实验3

- 3-1 将8个不同的LED连接到同一个引脚，并使所有LED闪烁。
- 3-2 在3个LED亮起时使其余5个LED闪烁（只需使用不同的引脚）。
- 3-3 将1个LED连接到开关。
- 3-4 将8个不同的开关连接至8个LED，并在LCD上显示传感器的激活情况。

# 实验4

- 4-1 激活触摸检测并显示在屏幕上绘制的内容。
- 4-2 在触摸屏上绘制4×4键盘，接收来自虚拟键盘的输入，并将其显示在LCD上。

# 大作业



# 作业——嵌入式系统设计开发

- 让学生使用mbed在线编译器设计开发一个嵌入式系统（密码锁、报警系统、计算器、智能台灯等）。（20学时）
- 以下为几个实用嵌入式系统及其基本功能。根据所学的嵌入式系统设计的方法，学生自行选择设计一个嵌入式系统，不必局限于下面列出的系统及功能，并在mbed在线编译器中进行实现。

# 密码锁

- 用户输入正确的6位数字密码，然后输入“B”，密码锁打开，LED灯点亮。如果输入的密码不正确，LED灯闪烁。若三次输入错误密码，锁定1小时，锁定期间无法开锁。输入密码时，屏幕显示“Code: \_ \_ \_ \_ \_”，每输入一个数字，“\_”被“\*”代替，当按下“C”键，删除最后一个输入的数字并用“\_”代替。

# 报警系统

- 划分进出区域与其他区域。未设置状态下，如果用户输入正确的密码，变为退出状态。退出状态下，用户有一段时间用来离开，时间到期后进入设置状态。设置状态下，如果进出区域被激活，变为进入状态；如果其他区域被激活，变为报警状态。进入状态下，如果用户输入正确的密码，变为未设置状态；如果用户没能输入正确的密码，变为报警状态。

# 智能台灯

- 台灯能够通过触控打开、关闭和变色，有循环彩灯模式，有复位功能，可以定时打开、关闭和调节光线。晚上18点，台灯自动打开为白色弱光（用白色LED表示）；晚上20点，随着天色变暗，台灯光线变亮（用蓝色LED表示）；晚上22点，即将进入睡眠，台灯光线变为暖黄色（用黄色LED表示）。

# 报告

- 报告（命名为：姓名-报告）
  - 嵌入式系统综合实践
  - 姓名，学号，年级专业
  - 嵌入式系统名称
  - 嵌入式系统的功能说明
  - mbed在线编译器截图
  - 源代码作为附录
- 演示视频（命名为：姓名-演示视频）
- 7月15日前发送至fanjie@nwu.edu.cn