



学习和掌握一种RTOS

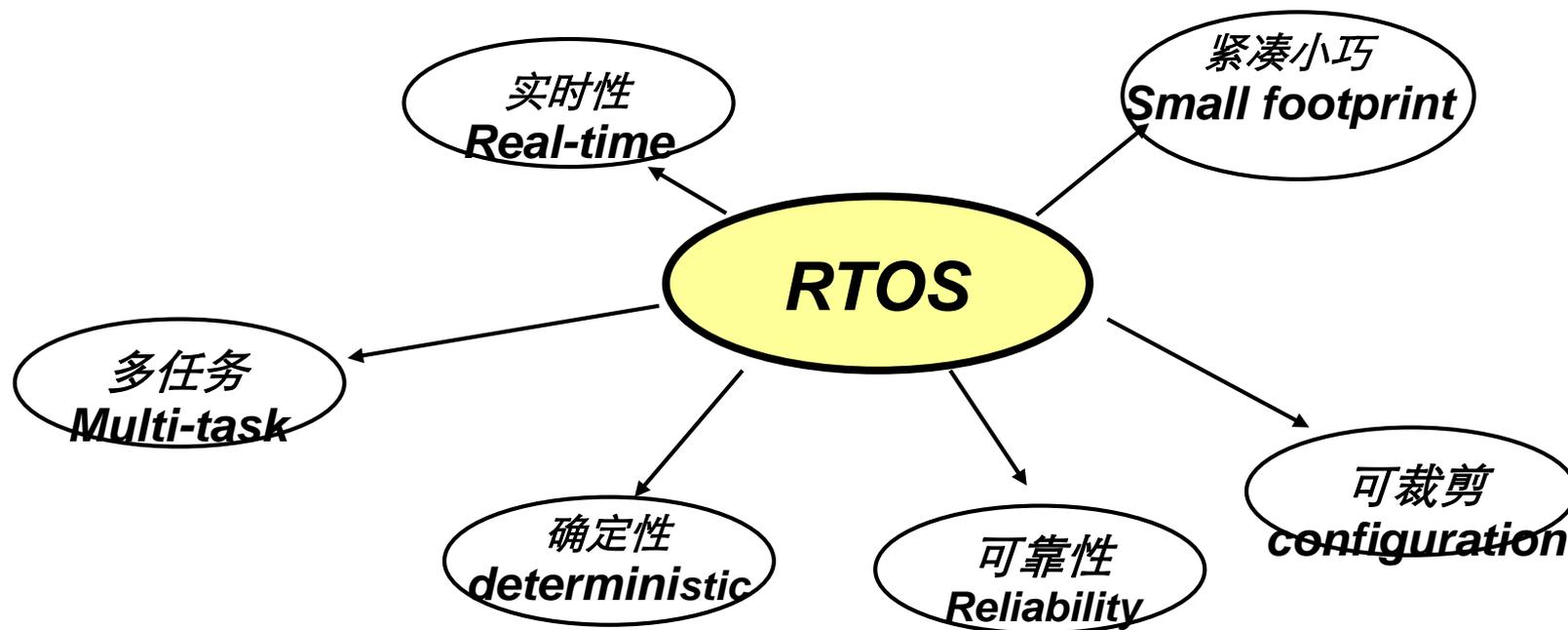
何小庆

北京麦克泰软件技术有限公司

2017年9月

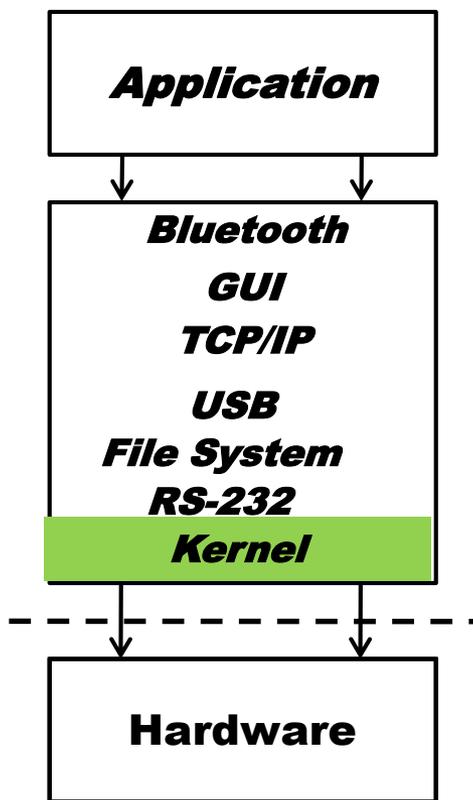
什么是RTOS?

R(real) T(time) OS 实时多任务操作系统、RTOS是一种操作系统，属于嵌入式操作系统，RTOS种类很多；有商业的、DIY和开源的。



RTOS vs. RTOS Kernel

- 操作系统(OS)是一系列软件的集合, 提供资源管理和应用代码服务的能力
- RTOS 已经包含了一系列的软件库 (中间件, 比如vxwork, QNX)
- RTOS kernel 只包含 OS 基本服务 (比如FreeRTOS, uc/OS-III)



RTOS发展历史

RTOS 名称	公司名称	网站	近况
VRTX	Ready System/Microte		被mentor 收购
pSoS	ISI		被Windriver 收购
OS-9	Microware		被Metorwork 收购
SMX	Mico Digital	www.smxrtos.com	
vxwork	Wind River	www.wrs.com	被 intel 收购
LynxOS	Lynuxwork	www.lynx.com	
QNX	QNX	www.qnx.com	被黑莓公司收购
CMX	CMX system	www.cmx.com	
Nucleus	ATI/Mentor	www.mentor.com	被Mentor 收购
ThreadX	Expresslogic	www.rtos.com	
uc/OS	Micrium	www.micrium.com	被Silicon Lab 收购
Integrity	Gree Hill	www.ghs.com	
OSE	Enea	www.enea.com	

OS-9



VRTX
Real-Time Operating System



WIND RIVER

Microsoft®



全球超过100多种，中国几种，更有许多用户自己设计RTOS
RTOS 有超过30年历史，全球兴旺的时候有几百家，中国也有几个RTOS

开源的RTOS

■ RTEMS

- 实时多处理器系统，最早运用在美国国防系统

■ TOPPERS

-日本著名开源的RTOS，创始人是京都大学高天教授，现专注在汽车电子



■ FreeRTOS

- 其目标在支持MCU，开源模式和生态好，但缺乏配套软件

■ eCOS

- 基于GNU的RTOS，含TCP/IP和文件系统，在消费电子产品有许多应用



■ Contiki

- 起源于无线传感网络的的RTOS，有超低功耗管理和IPV6支持。

■ Zephyr

- Linux基金会有一个微内核项目，由Intel主导，风河提供技术。



■ Nutt

- 它的设计目的还在于完全符合POSIX标准，完全实时，配套丰富



今天为什么要学习RTOS ?

- 物联网的大潮，MCU 迎来一个新的发展机遇：
 - 物联网系统表现为是一个**分布式的嵌入式计算系统**，大量的MCU替代过去运行Linux 的嵌入式处理器的中心计算模式。
 - 物联网应用需要**动态功耗效率**，以使系统能够在一定的频率范围（50 到 300+ MHz）内，以最低的功耗水平运行。高功耗效率（DMIPS/mW）意味着应用可以在较低的频率下运行，从而降低有效功耗。
 - 物联网系统是互联系统，需要高度的**安全性**。互联和安全设计是物联网设备区别传统的嵌入式系统，需要RTOS 的支撑。
 - 物联网时代产生了**物联网OS**，RTOS是许多物联网OS基础和核心。

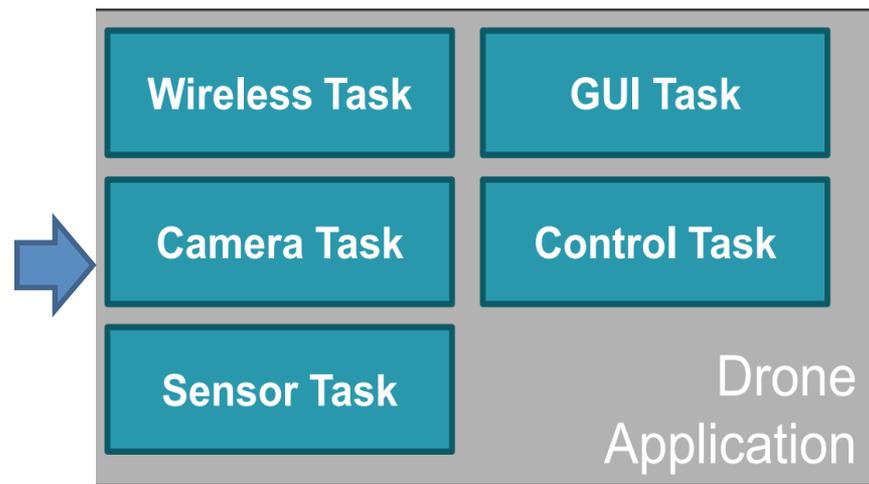
RTOS 的应用

- 工业控制装置
- 通信设备
- 消费电子产品
- 仪器仪表
- 军事电子设备
- 航空航天系统
- 计算机外设
- 医疗电子产品
-

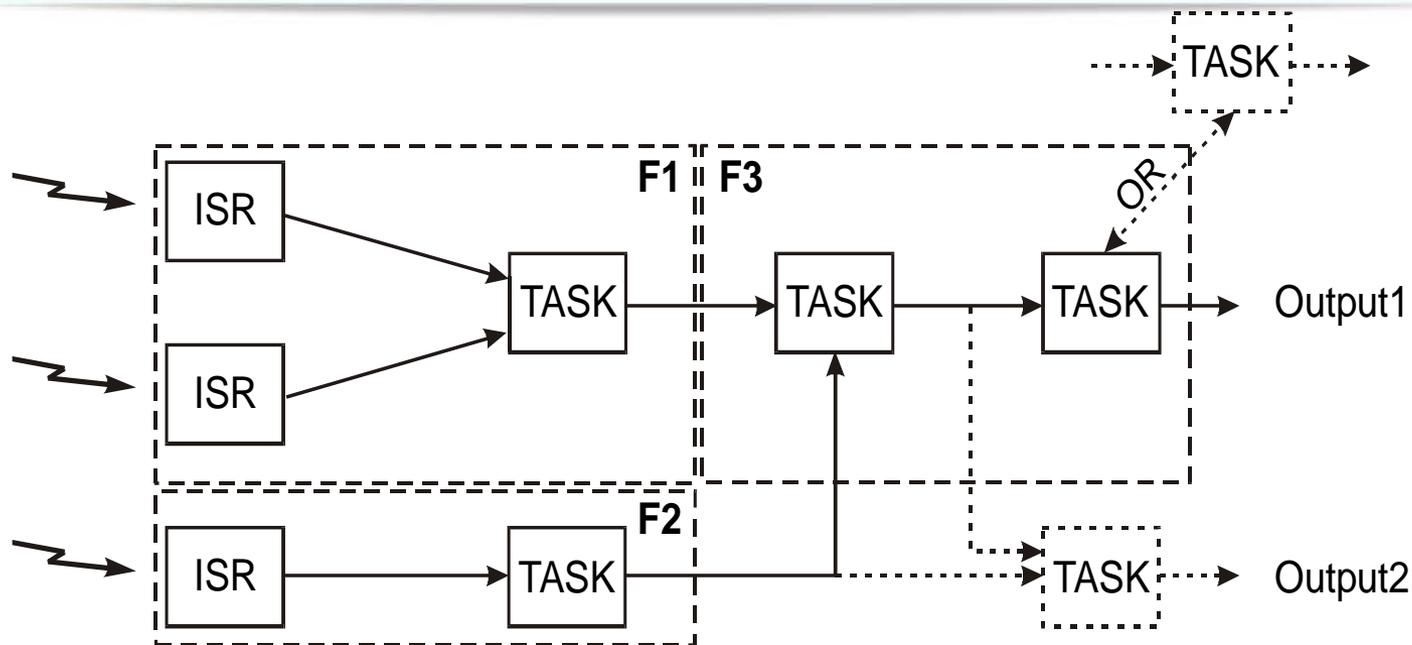


RTOS 给软件设计带来的好处

- 支持并发处理
- 容易加入新的功能
- 破解应用的复杂性



RTOS 的精髓—多任务系统



- 任务独立，基于优先级任务调度
- 任务间通信，异步处理
- F1, F2, F3三个功能模块接口清晰
- 易于加入任务

— Initial Design
- - - Added Later

掌握RTOS内核三大要素

■ 事件驱动

- 中断机制和多任务
- 优先级抢占和时间片轮转调度

■ 资源共享

- 任务间通信和同步互斥
- 提供的机制有信号量、邮箱、消息队列、事件标志、互斥

■ 开发和分析工具

- 动态调试
- 可与应用结合



RTOS 应用的可视分析

- **RTOS 在开发和学习中经常会遇到这样一些问题：**
 - 我的任务是在运行吗？
 - 这些任务使用了多少堆栈空间？
 - 每个任务占用了多少 **CPU** 时间？
 - 我应该如何优化我的代码？
 - 中断关闭多少时间？
 - 有优先级反转的问题？
 - 有死锁的问题？
 - 查看变量、数据结构、队列和 **I/O** 设备
- 静态调试帮不上忙，因为嵌入式系统都是动态系统
- 有专门的工具可以帮助基于**RTOS**系统的可视化分析

Micrium 的 μ C/Probe

■ 基于Windows软件

■ 通过 JTAG、USB和ETH 连接目标板

■ 不需要 CPU 介入

■ 在运行时显示RTOS 状态

■ 每个任务运行计算

■ 每个任务的 CPU占用时间

■ 每个任务中断关闭时间

■ 堆栈使用情况

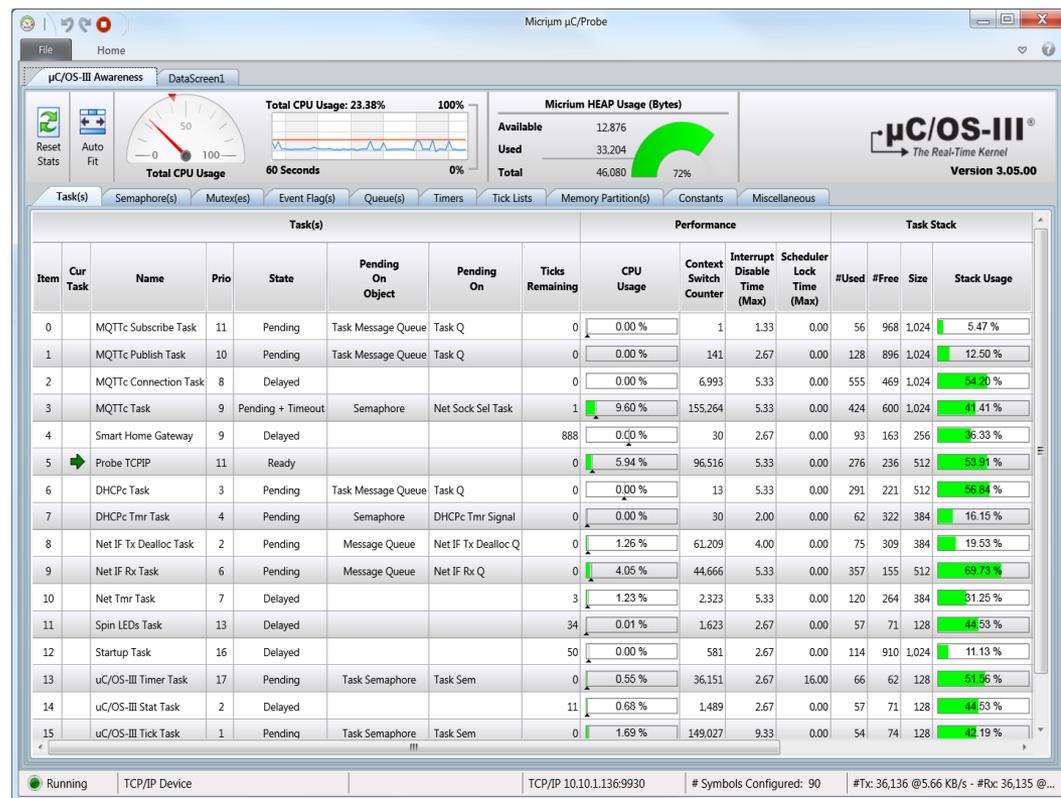
■ 任务相应时间

■ 任务状态

■

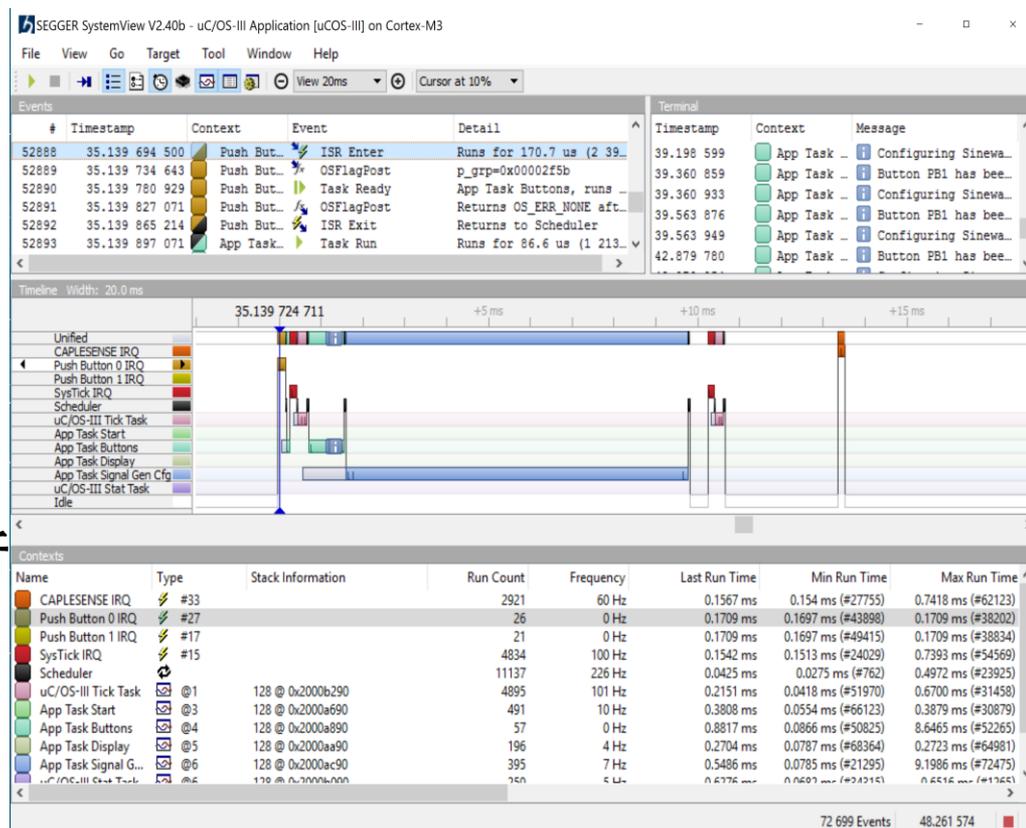
■ 支持RTOS

■ μ C/OS-II, μ C/OS-III 和 μ C/OS 5



SEGGER 的 SystemView

- 在时间轴上显示RTOS 事件
 - 动态实现
 - 中断 (ISR)
 - 任务
- 以优先级为序
- 触发
 - 以任务和ISR结束为开始
 - 以用户事件为开始
- 跟踪信息可保存下来用作后分析
- 需要有 SEGGER J-Link
- 支持RTOS
 - FreeRTOS
 - μ C/OS-III 和 μ C/OS 5



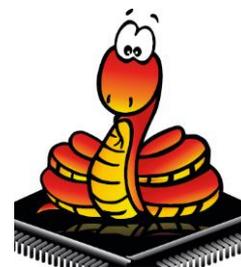
Percepio 的 Tracealyzer

- 以时间轴记录 RTOS 事件
 - ISRs
 - 任务
- 跟踪信息可保存下来用作后分析
- 超过25种不同的数据展示方式
 - 展示的内容有相关性
- 支持:
 - SEGGER J-Link
 - TCP/IP
 - USB-CDC
- 支持 RTOS
 - FreeRTOS
 - $\mu\text{C}/\text{OS-III}$ 和 $\mu\text{C}/\text{OS 5}$
 - Linux和vxwork...



学习RTOS 方法

- 任务管理是重点
 - 掌握任务建立，调度，通信和互斥等机制
 - 掌握内存管理方式（静态作为重点）
 - 学习RTOS 内核和硬件相关部分—中断和时钟管理
- 简单的驱动编写
 - 比如串口、GPIO 等基本外设
 - 移植已经不是是重点
- RTOS编程接口
 - CMSIS-RTOS-ARM 制定Cortex MCU RTOS接口标准
 - POSIX标准
 - uITRON 日本标准
 - OSEK/VDX—汽车电子和交通标准
- RTOS 编程语言和开发工具
 - C/C++， IAR/KEIL/GCC， 未来Python和JS



RTOS的组件

- OS 组件越来越多、越来越重要
 - 协议—TCP/IP
 - 开源LWIP
 - 商业的USB 协议, 蓝牙协议和CANOPEN的价格相对要贵
 - 文件系统, Flash NAND, SD/MMC, USB 盘支持优化
 - 图形模块, uC/GUI (emWin) 和TouchFX, 纯软件模块对于MCU 消耗大, 软硬结合方案, 多点触屏和2D/3D图形是未来趋势。
 - 芯片公司开始提供RTOS 的组件(源代码和二进制)
- 大型的RTOS 基本包括了基本组件
 - 比如 Vxwork, QNX 包含TCP/IP, File和GUI
- 小型的RTOS 组件是外加的
 - uC/OS-III 有uC/GUI, uC/FS, uC/TCP-IP等

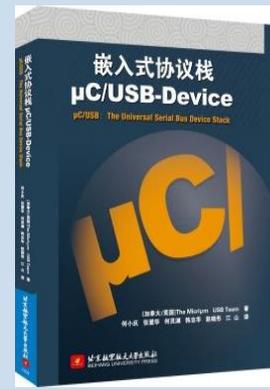
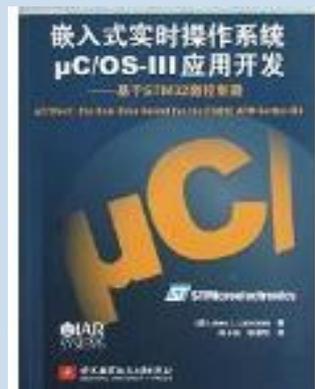
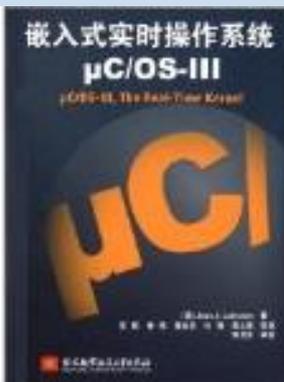
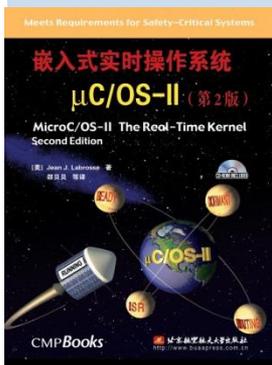
应用决定需要那些组件, 组件也决定了使用 and 选择哪种RTOS

如何选择一个RTOS?

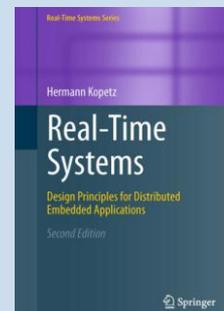
- 你需要 **RTOS kernel 还是RTOS?**
 - FreeRTOS 自己只有kernel , 其他第三方, uc/OS-II/III 相当完整组件。
- **RTOS 还是Linux (Android) ?**
 - 你的硬件设计使用的是MCU还是AP
 - 两者都可以使用RTOS , 但是后者可以支持Linux 或者Android
- 嵌入式安全的需求
 - 借助MCU/AP 的 MPU和MMU 可以实现系统内存保护, 从而获得安全认证和预认证的安全产品, 比如 SaftRTOS
- 芯片和硬件平台的支持
 - 每家芯片公司的平台都会支持1-2 种RTOS 或者Linux/Android
- **价格、技术支持和服务升级**
 - 开源? 一次性版式, 还是unit/per CPU/side , 技术服务?

*Comparing microcontroller real-time operating systems
Sergey Kolesnik, Telecard-Pribor Ltd. - December 08, 2013*

■ uC/OS-II 和 III 有非常好中英文文图书（官方）



- FreeRTOS 资料比较少，官方只有英文的PDF手册
- FreeRTOS 芯片公司的BSP 比较多
- 网上有些视频



嵌入式操作系统风云录：历史的演进与物联网未来

- 本书全面回顾了嵌入式操作系统演进历史，主流的嵌入式操作系统技术特点、成长历程以及背后的商业故事，展望了嵌入式操作系统未来的技术路径、市场发展趋势和物联网时代的新机遇。本书以时间轴讲述了从RTOS、开源嵌入式操作系统到物联网操作系统发展历程，以技术为视角剖析了嵌入式操作系统的实时性、安全性和云计算等重要技术，从手机、通信、汽车和可穿戴几个市场讨论了嵌入式操作系统的应用，从嵌入式操作系统知识产权讨论了商业模式的问题。
- 共计**15章 20万字**，历史**3年**。





入门培训

涉及前后台系统的设计，使用RTOS的好处。详细分析RTOS的基本功能，包括内核调度机制，任务管理，中断管理，时间管理，资源管理，任务间的同步与通信机制等。了解RTOS的启动过程、学习如何创建任务及使用RTOS的系统服务。



高阶培训

详细分析RTOS内核调度机制、资源管理、配置和裁剪以及移植要点和外设驱动编写，课程将介绍RTOS低功耗技术，RTOS系统设计方法以及RTOS图形应用编程和应用知识。



工具培训

介绍IAR EWARM环境及工程设置，编译器与优化技术，链接器与存储空间管理和Cortex-M调试技巧。通过实验掌握代码优化，Bootloader应用和SWO Trace调试技巧。

FreeRTOS高阶培训

FreeRTOS是一个开源和免费的RTOS内核，已经被广泛应用到各种嵌入式和物联网产品开发中。BMR（麦克泰）与ST合作，为具有嵌入式软件开发基础，已经有一些FreeRTOS 经验的开发人员、高校学生安排了为期两天的FreeRTOS高阶培训课程。

课程内容

- 1.实时嵌入式系统设计和编程方法
- 2.深入FreeRTOS内核-原理和编程
- 3.FreeRTOS 移植技术和外设驱动
- 4.FreeRTOS Tickles的低功耗技术
- 5.FreeRTOS 内核调试和分析技术
- 6.基于FreeRTOS 应用 - TouchGFX开发应用

实验部分

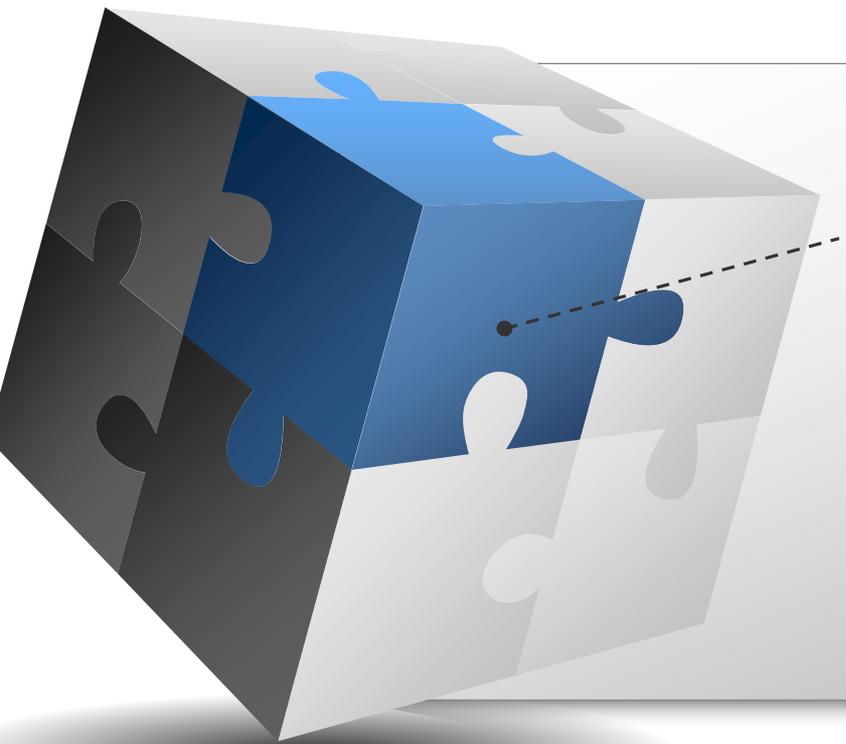
- 1.实验环境介绍和软件安装
- 2.FreeRTOS内核高阶实验
- 3.FreeRTOS低功耗实验
- 4.FreeRTOS 分析工具实验
- 5.TouchGFX 图形应用实验
- 6.串口资源共享实验

实验平台和开发环境

本课程的实验是硬件基于基于STM32F769 Discovery Kit硬件平台，开发环境是IAR EWARM，软件工具Tracealyzer 3.1 和SystemView 2.5，实验案例使用最新的FreeRTOS 9.00 版本和STM32Cube 固件库1.7.0 版本。

北京站：12月1-2 日

感谢您的关注RTOS技术，欢迎交流！



公司网址: www.bmrtech.com

邮箱: info@bmrtech.com

北京: 010-62975900

上海: 021-62127690

深圳: 0755-82977971

版本信息: 2010年 第一次撰写 v1.0
2016年第二次撰写 v2.0
2017年第三次修改v2.2-2.3-2.4