

基于开源RTOS 的物联网软件设计方法

Design Method of IoT Software Based on Open Source RTOS

何小庆

单片机与嵌入式系统应用杂志社

2018年3月 上海

发言内容

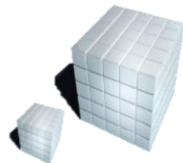
- 物联网设备软件需求
- RTOS从内核向IoT演进
- RTOS的编程方式
- RTOS与IoT应用
- Amazon FreeRTOS简介
- 基于RTOS 的IoT设计

物联网设备的软件需要

2016年风河公司在纽伦堡Embedded World 上分析物联网设备有8大软件需求



模块可升级的架构



设备软件可伸缩



系统安全



虚拟化



性能和实时性



连接性



丰富的UI



安全认证

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	
Zephyr	Linux Foundation	www.zephyrproject.org/	现在Linaro 在主导维护
Nuttx	Gregory Nutt 2007 BSD 授权	http://www.nuttx.org/	POSIX API 无人机应用

OS-9



VRTX
Real-Time Operating System



WIND RIVER

Microsoft



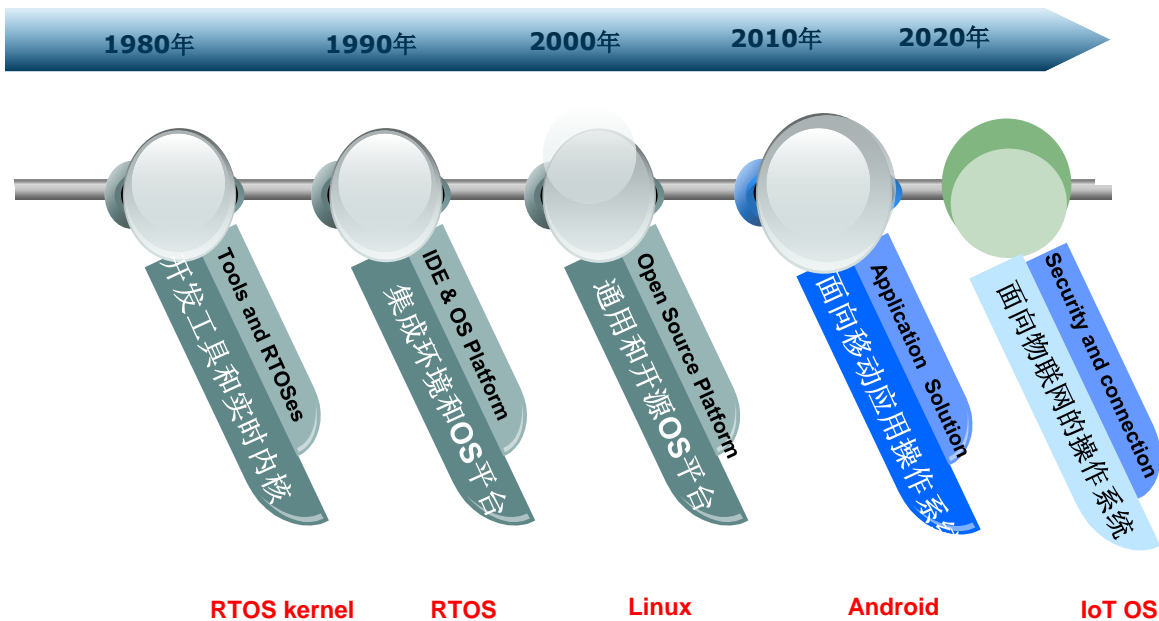
嵌入式OS 始于和发展于RTOS，RTOS 有超过30年历史
全球兴旺的时候有几百家，中国也有几个RTOS



嵌入式系统联谊会
www.esbf.org

嵌入式（设备）操作系统演进之路

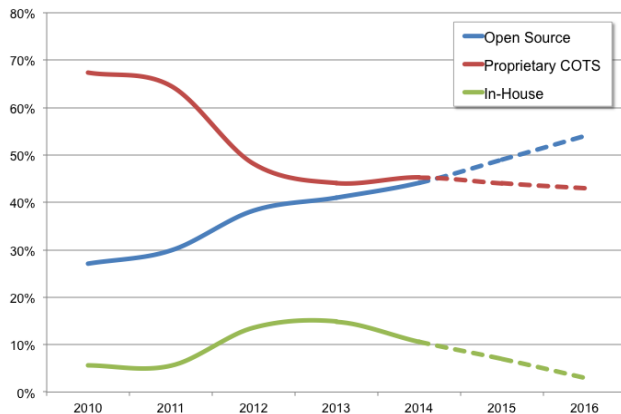
Embedded Device Operating System



开源OS活跃在IoT 市场

- 开源软件优势:

- 应用在数据中心、嵌入式、网络、移动和云计算中
- 各种版本的开源软件可以适合各种应用
- Android - 移动手机和平板
- Linux- 服务器和嵌入式
- FreeRTOS- 嵌入式和传感器
- IoT OS 是开源OS
 - ARM Mbed OS
 - Huawei LiteOS
 - AliOS
 - Amazon FreeRTOS
- Linux等开源软件已经成为事实的工业标准，各种互联标准（HTTP、HTML、SSH/SSL、OSGI、MQTT）在其上实施成为事实标准平台



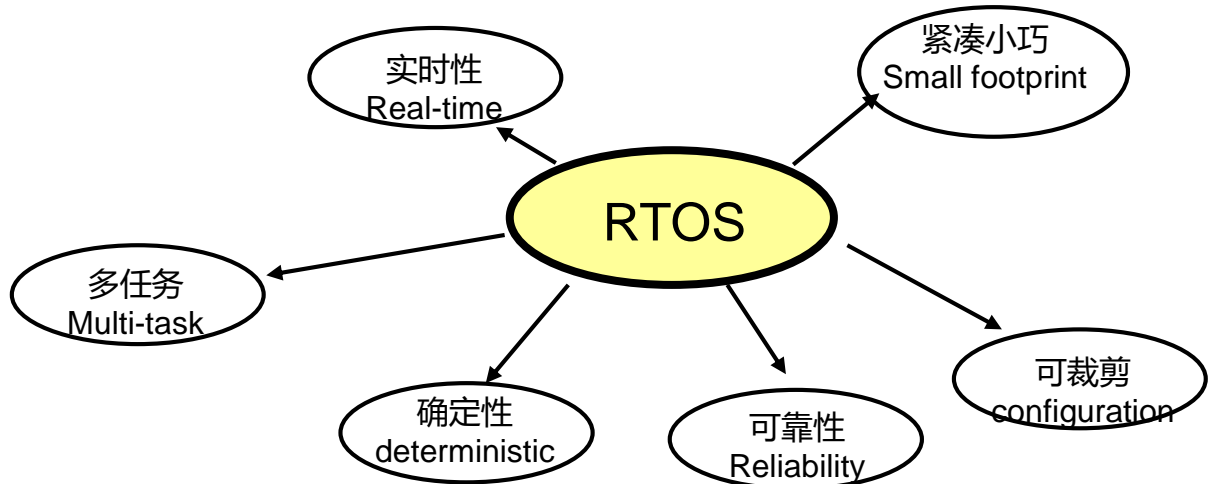
开源的RTOS

- RTEMS：实时多处理器系统，最早运用在美国国防系统
- TOPPERS：日本开源的RTOS，创始人京都大学高田教授，专注在汽车电子
- FreeRTOS：其目标和专注点在支持MCU，开源模式和生态好，但缺乏配套软件，最近加入Amazon，推出Amazon FreeRTOS，授权从GPL改成MIT
- 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基金会宣布了一个微内核项目，Intel 主导，风河提供技术
- NuttX：BSD 授权，POSIX API，无人机应用

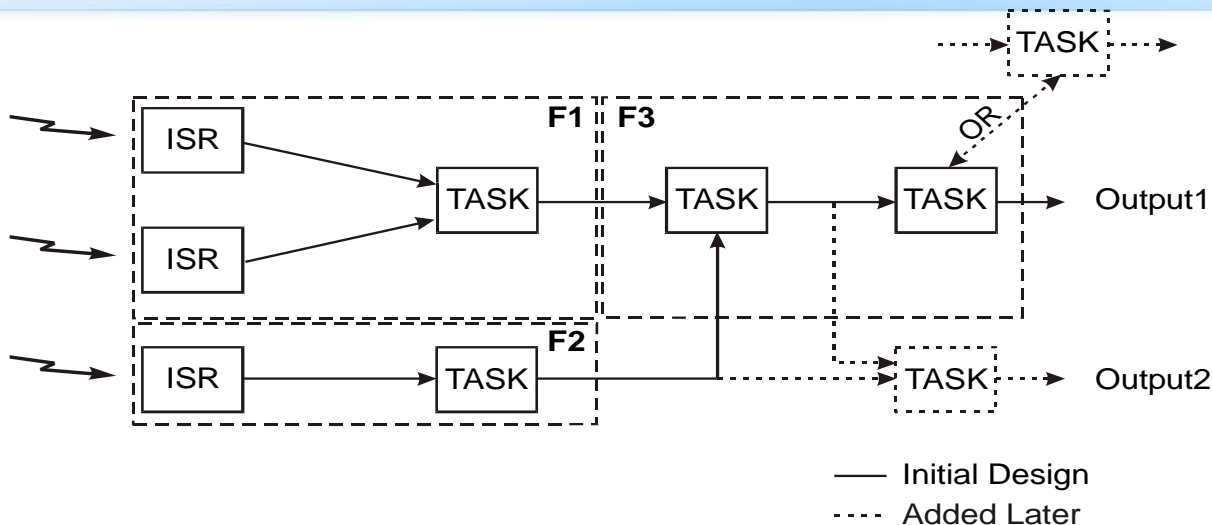


什么是RTOS?

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 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

传统的软件编程方法（轮询）

- 优点：好理解容易实现，适合简单的应用。
 - 一个主程序+ 若干中断。
- 缺点：不适合复杂的应用，响应时间无法确定。

```
void main(void)
{
    init();

    while(1)
    {
        do_task1();

        do_task2();

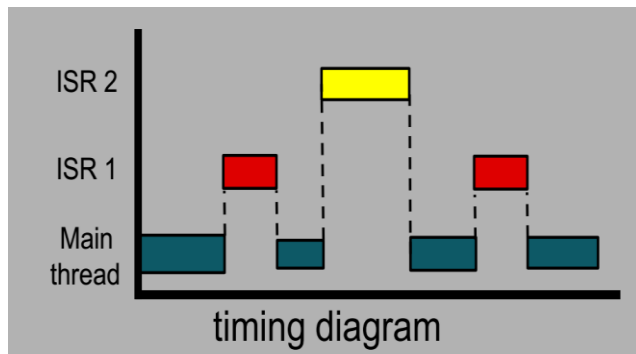
        .....
    }
}
```

```
void ISR_1 (void)
{
    // IRQ 1 handler
    .....
}
```

```
void ISR_2 (void)
{
    // IRQ 2 handler
    .....
}
```

```
.....
```

coding structure



RTOS 多任务编程方法

- 多任务功能，同时支持ISR。
- 容易实现复杂的应用。
- 使用抢占调度机制 (Preemptive)，可以实现响应的确定性

```
void main(void)
{
    BSP_init();

    Task_create(task1);
    Task_create(task2);

    ....

    RTOS_run();
}
```

```
void task1(void)
{
    task1_init();

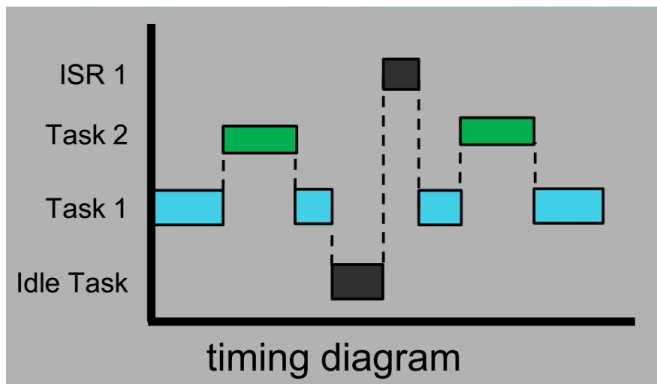
    while(1)
    {
        ....
    }
}
```

```
void ISR_1 (void)
{
    // IRQ 1 handler
    ....
}
```

```
void task2(void)
{
    task2_init();

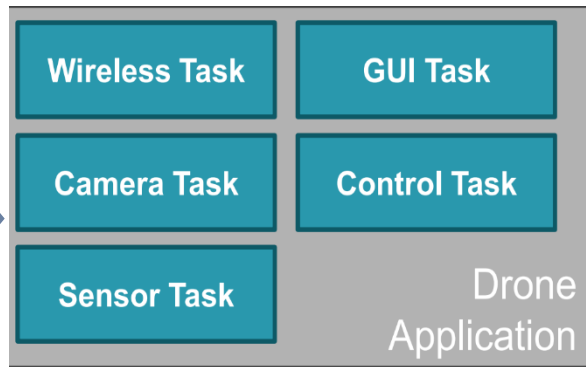
    while(1)
    {
        ....
    }
}
```

coding structure



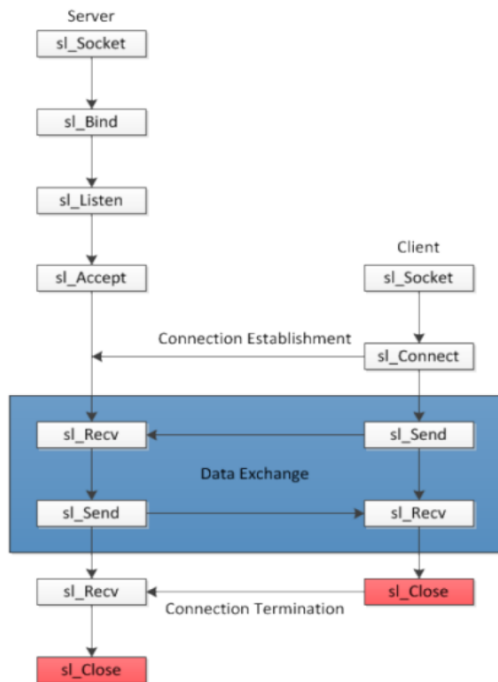
方便的设计方式

- 支持并发处理
- 容易加入新的功能
- 破解应用的复杂性
- 便于系统维护



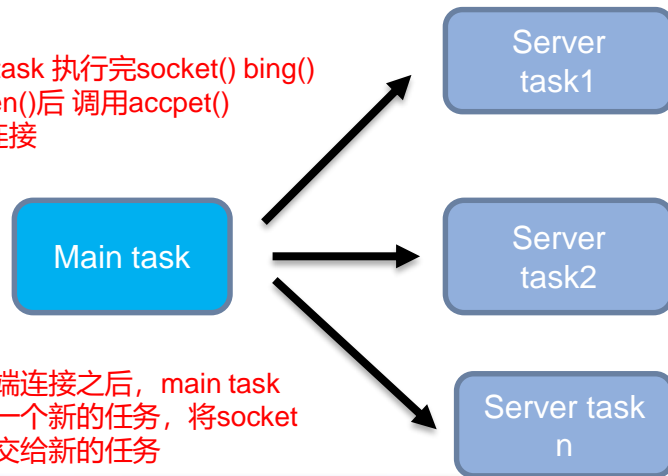
通信服务器软件设计

- 一个通信服务器支持多客户端
- 建立一个单独的任务处理每一个客户端的连接



新建的任务基于接受的socket处理这个连接

Main task 执行完`socket()` `bind()`和`listen()`后 调用`accept()`等待连接



客户端连接之后, main task 建立一个新的任务, 将socket 处理交给新的任务

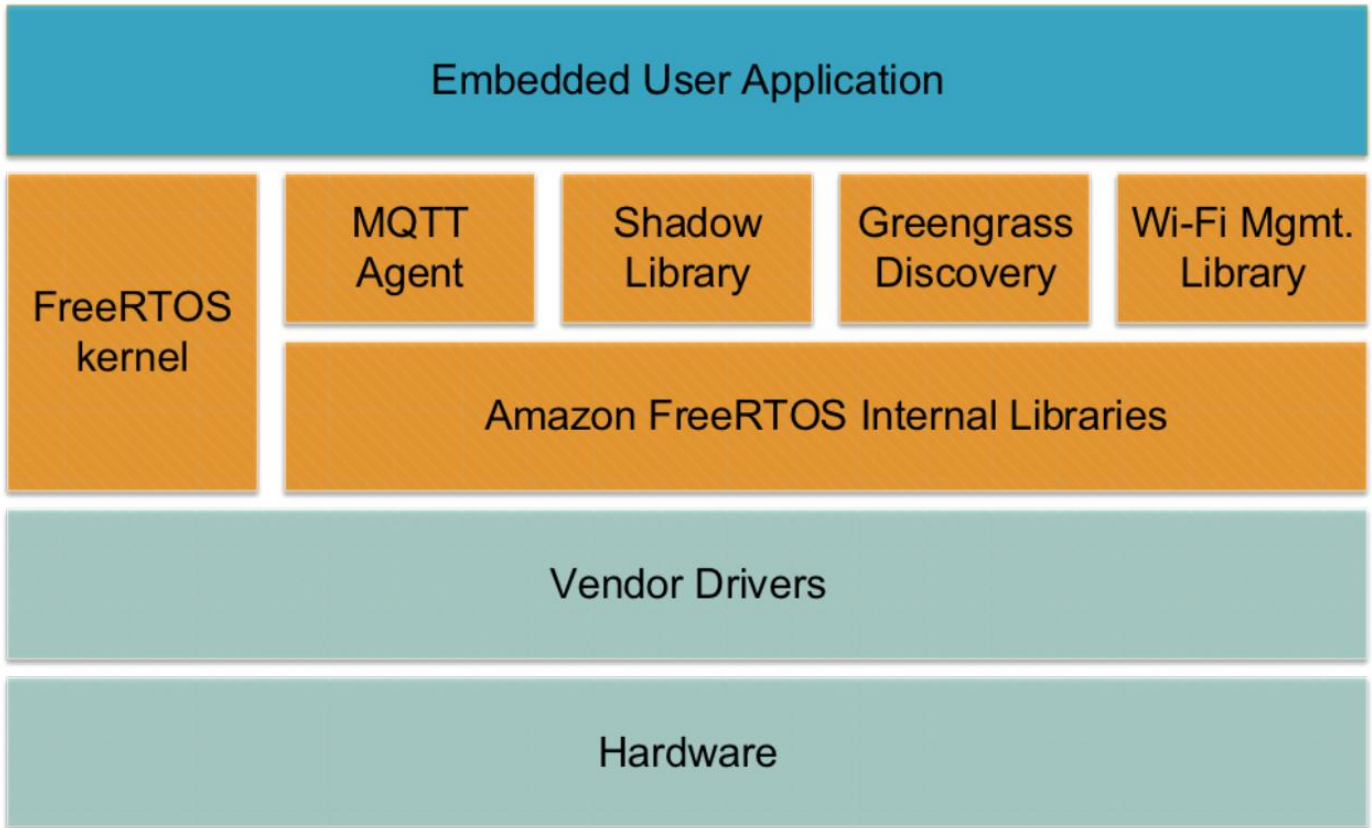
什么是Amazon FreeRTOS?

- IoT operating system for Microcontrollers(MCU)
适用微控制器的IoT操作系统
- Amazon FreeRTOS
 - Amazon FreeRTOS (简称 a:FreeRTOS) 是一款适用于微控制器的操作系统，可让您轻松地对低功耗的小型边缘设备进行编程、部署、安全保护、连接和管理。Amazon FreeRTOS 以 FreeRTOS 内核为基础，并通过软件库对其进行扩展，从而让您可以轻松地将小型低功耗设备安全连接到 AWS 云服务或运行 AWS Greengrass 的功能更强大的边缘设备
- Amazon FreeRTOS 与传统嵌入式开发区别；
 - 致力于提供MCU连接到云端的便捷和安全方案
 - 克服传统的MCU 云连接开发的困难：多平台和缺少安全机制
- Amazon FreeRTOS 是一个开源系统，可以免费使用

FreeRTOS Kernel

- FreeRTOS 内核是RTOS，支持各种处理器和MCU 架构，非常适合基于MCU的嵌入式应用，内核提供以下基本功能：
 - FreeRTOS Kernel 调度
 - 存储管理
 - Kernel Memory Allocation
 - Application Memory Management
 - 任务通信协调
 - Queues
 - Semaphores 和 Mutexes
 - Direct-to-Task Notifications
 - Stream Buffers(定长) 和Message Buffers (变长)
 - Software Timer
 - Lower Power
 - Amazon FreeRTOS 基于FreeRTOS 10.0

Amazon FreeRTOS 架构

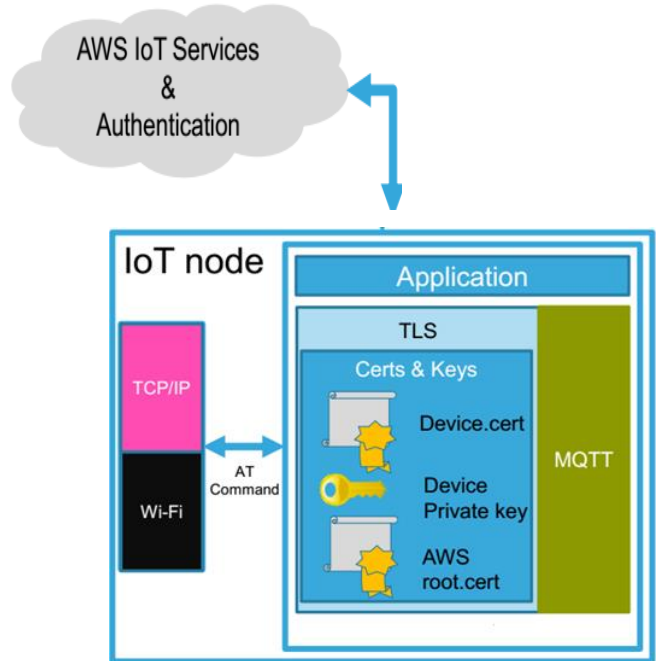


Amazon FreeRTOS 软件库

- Amazon FreeRTOS 包括了一系列软件库：
 - 安全的将设备连接到 AWS IoT, 使用了 MQTT 协议和 device shadows
 - Device shadows是一种JSON 文档用来存储和检索Things 当前状态信息
 - 发现和连接到AWS Greengrass cores
 - AWS Greengrass 是一种允许你以安全方式为互联网设备执行本地计、消息收发和数据缓存的软件
 - 管理 Wi-Fi 连接
 - OTA 系统升级和维护功能

Amazon FreeRTOS 的安全机制

- Amazon FreeRTOS 安全 API 支持开发者建立安全通信的嵌入式应用
 - Secure Sockets
 - Transport Layer Security (TLS)
 - 公钥加密标准 (PKCS #11)
 - 支持Asymmetric Cryptosystem
 - OTA 安全机制
 - 连接安全
 - OTA 更新的认证和完整性
 - 操作者的安全



Amazon FreeRTOS文件结构

- D:\freeRTOS\amazon-freertos-master\
 - Demo
 - Common
 - Vendor
 - Nxp/ST/TI/Microchip
 - MQTT
 - Shadow
 - Greengrass
 - Lib
 - MQTT
 - shadow
 - Greengrass
 - Wi-Fi
 - TLS
 - PKCLS11
 - OTA

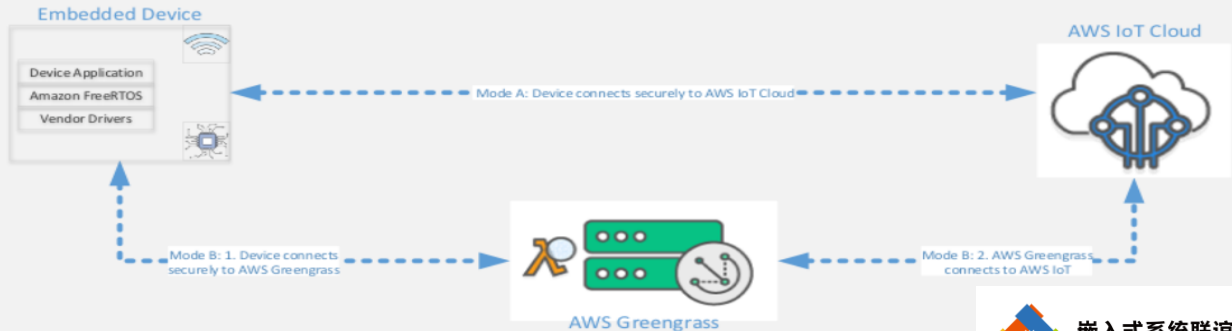
Amazon FreeRTOS 开发过程

Embedded Software Development Workflow



Build, Test
and Deploy
Device
Application

Embedded Device Runtime Workflow



开发要点

▪ AWS 云侧

- 创建AWS 账户，第一个12月是免费的。
- 进入AWS IoT Console
- 管理-创建事物-注册-证书-下载证书
 - xxxxxxxxxxxx-certificate.pem.crt
 - xxxxxxxxxxxx-private.pem.key
- 管理-安全-创建策略
- 关联证书和策略

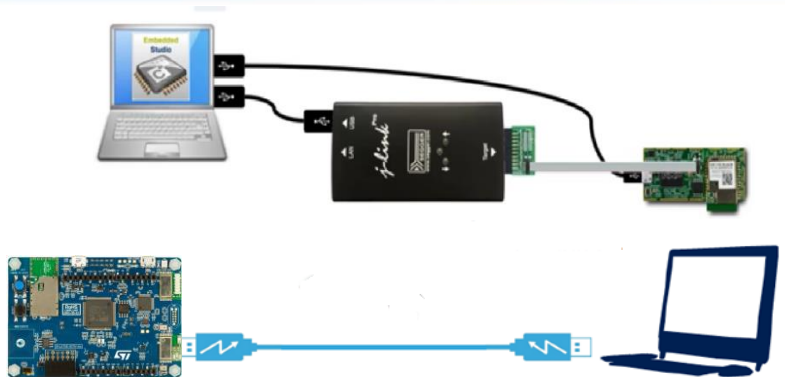
▪ 嵌入式侧

- IAR Embedded workbench 和 Segger J-link
- 进入Amazon FreeRTOS console
- 选择NXP/ST/TI/Microchip 目标板代码下载（或者直接到github 下载）
- 配置你的 证书-
/demo/common/include/aws_clientcredential.h
- 配置AWS IoT 端点信息: IOT thing 地址、thing 名字、SSID/PW
- 格式化 key
demo/common/devmode_key_provisioning/
- 运行CertificateConfigurator.html
- IAR 构建-下载-运行

Amazon FreeRTOS 的应用例子

四个DEMO应用

- MQTT Echo demo
- Shadow Demo
- Greengrass Discovery Demo
- OTA demo



```
We cannot display the message as JSON, and are instead displaying it as UTF-8 String.
Hello World 7 ACK
freertos/demos/echo Jan 26, 2018 8:50:35 PM +0800 Export Hide

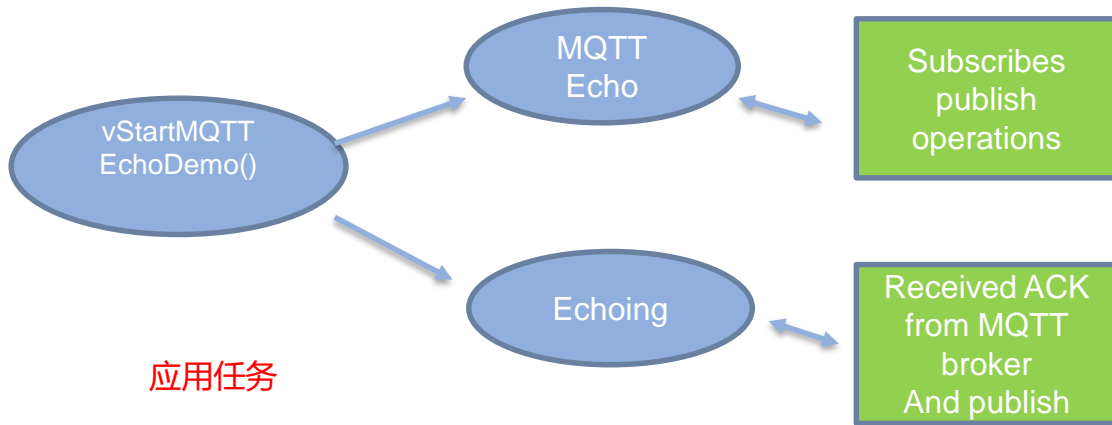
We cannot display the message as JSON, and are instead displaying it as UTF-8 String.
Hello World 7
freertos/demos/echo Jan 26, 2018 8:50:30 PM +0800 Export Hide

We cannot display the message as JSON, and are instead displaying it as UTF-8 String.
Hello World 6 ACK
freertos/demos/echo Jan 26, 2018 8:50:26 PM +0800 Export Hide

We cannot display the message as JSON, and are instead displaying it as UTF-8 String.
Hello World 6
```

```
COM9 - Tera Term VT
File Edit Setup Control Window Help
8 34422 [MQTTEcho] Echo successfully published 'Hello World 4'
9 34524 [Echoing] Sending command to MQTT task.
10 34525 [MQTT] Received message c0000 from queue.
11 34830 [MQTT] MQTT Publish was successful.
12 34830 [MQTT] Notifying task.
13 34830 [Echoing] Command sent to MQTT task passed.
14 34830 [Echoing] Message returned with ACK: 'Hello World 4 ACK'
15 39422 [MQTTEcho] Sending command to MQTT task.
16 39422 [MQTT] Received message d0000 from queue.
17 39727 [MQTT] MQTT Publish was successful.
18 39727 [MQTT] Notifying task.
19 39727 [MQTTEcho] Command sent to MQTT task passed.
20 39727 [MQTTEcho] Echo successfully published 'Hello World 5'
21 39829 [Echoing] Sending command to MQTT task.
22 39830 [MQTT] Received message e0000 from queue.
23 40236 [MQTT] MQTT Publish was successful.
24 40236 [MQTT] Notifying task.
25 40236 [Echoing] Command sent to MQTT task passed.
26 40237 [Echoing] Message returned with ACK: 'Hello World 5 ACK'
27 44727 [MQTTEcho] Sending command to MQTT task.
28 44727 [MQTT] Received message f0000 from queue.
29 45032 [MQTT] MQTT Publish was successful.
```

MQTT Echo demo 任务流程



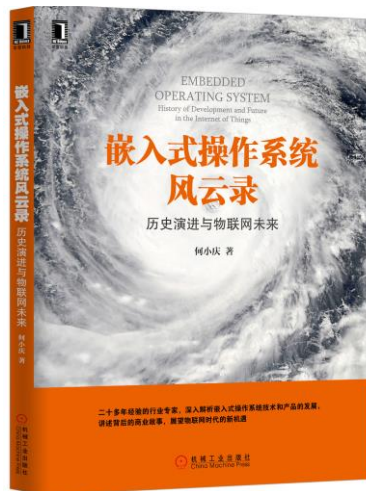
应用任务



系统任务

嵌入式操作系统风云录：历史的演进与物联网未来

- 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15章 20万字，历史3年,2016年底出版
- 何小庆出版有其他四本书籍和60余篇论文



RTOS和IoT培训课程

- 参与麦克泰提供的培训课程
 - 基于uC/OS 和 FreeRTOS 入门课程
 - 主讲 “学习和掌握一种RTOS ”
 - 基于FreeRTOS 高阶课程
 - 主讲 “基于RTOS实时软件设计方法”
 - 主讲 “Amazon FreeRTOS 简介”
 - IoT 课程 (2018年暑期培训)



高阶培训 3月13-14 上海, 3月21-22日北京
入门培训 4月10日上海、12日 北京、17日深圳, 19日广州

Thank you !

Any question please feel free to contact:

xiaoqinghe@live.com

www.hexiaoqing.net

@何小庆微博



嵌入式资讯精选
微信号: mcuworld

