

实时操作系统课程 核心教学脉络及教学资源分享

苏州大学 王宜怀, yihuaiw@suda.edu.cn

2021年8月17日



内容简介

实时操作系统（RTOS）是嵌入式人工智能与物联网终端的重要工具和运行载体，从应用与原理两个层面梳理RTOS脉络是该课程的核心。这里以国产RT-Thread实时操作系统为蓝本阐述RTOS课程核心教学脉络，并分享《嵌入式实时操作系统—基于RT-Thread的EAI&IoT系统开发》一书的撰写思路、样例源程序、硬件资源、PPT等教学资源；阐述RTOS的BIOS驻留与统一接口；简要给出RTOS的线程、调度、延时函数、事件、消息队列、信号量、互斥量等基本要素的应用方法，给出利用printf函数进行原理剖析的基本方法。

目录

一、实时操作系统课程教学脉络

二、教材脉络梳理

三、RTOS中核心概念导引

四、实践源代码导引

五、实践操作演示

一、实时操作系统课程核心教学脉络

1.1 实时操作系统课程教学目标

学习RTOS有两个可能的出发点：

一是学会在RTOS场景下进行基本应用程序开发；

二是在掌握应用编程的前提下，理解其运行原理，进行深度应用程序开发。

这就是实时操作系统课程**教学目标**：把RTOS作为工具**应用**，**理解**其原理。

进一步说，应用RTOS与理解RTOS可以简单地表述为：

应用就是在基本了解线程、调度等基本概念基础上，正确运用RTOS下的延时函数、事件、消息队列、信号量、互斥量等基本要素，进行应用程序的开发，让RTOS成为嵌入式软件开发的辅助工具。

理解就是要理解RTOS工作的基本原理，也就就是理解调度机制，理解延时函数、事件、消息队列、信号量、互斥量等基本要素工作机制，为应用编程提供更加坚实的基础。

以消息队列为例，**应用编程**就是这样一个场景：有个“装消息”的篮子，有个发送消息处，当发送消息处发出消息，消息就会进入“装消息”的篮子，而只要消息“篮子”中有消息，等待消息的程序就会运行。**理解运行机制**就是为什么发送消息处发出消息，消息就会自动进入消息“篮子”，为什么消息“篮子”中有消息，等待消息的程序就会运行。

1.2 实时操作系统的选型问题

RTOS种类繁多，有国外的，也有国产的；有收费的，也有免费的；有开发者维护升级的，也有依赖爱好者更新升级的，初学者不知如何选型。但是无论如何，学习RTOS，必须以一个具体的RTOS为蓝本。不同RTOS，其应用方法及原理大同小异，掌握其共性是学习的关键，才能达到举一反三的效果。这里给出的国产RT-Thread实时操作系统是上海睿赛德电子科技有限公司于2006年开始推出的开源RTOS，面向嵌入式人工智能与物联网领域，被广泛应用于工业控制等众多行业领域，将其内核用到实际产品中，没有收费陷阱问题。本课程以RT-Thread为蓝本，以通用嵌入式计算机（GEC）为硬件载体，阐述RTOS中的线程、调度、延时函数、事件、消息队列、信号量、互斥量等基本知识要素，给出RTOS下程序设计方法。

1.3 实践问题

RTOS是理论联系实践的典型体系，学习RTOS，不仅要以一个具体的RTOS为蓝本，还要一个具体的微型计算机为蓝本进行实践，否则就有可能成为纸上谈兵。

硬件系统：

对实践器材的基本要求是满足RTOS的基本知识要素实践训练，价格低廉，使用方便，稳定可靠。AHL-STM32L431开发套件分迷你型、扩展型二种型号，更详细的介绍见配套电子资源。迷你型可以完成RTOS基本知识要素实践，扩展型附加了九个外设部件，可用于实践创新训练。

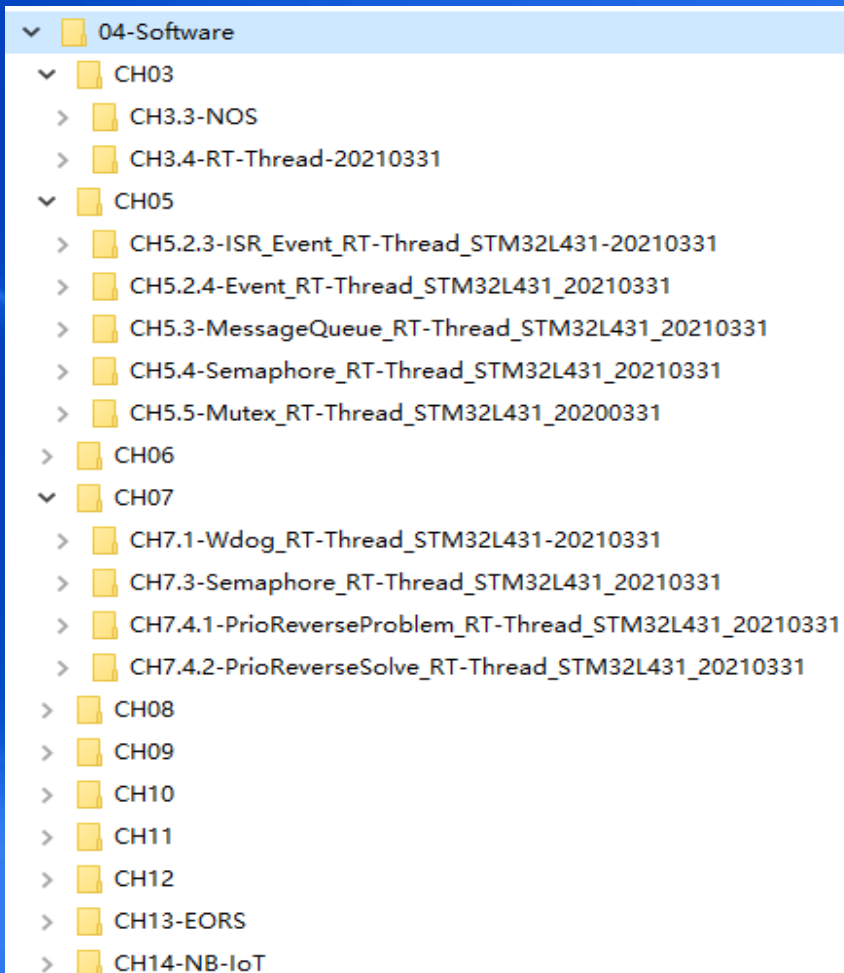


开发环境：苏州大学嵌入式人工智能与物联网实验经过十多年努力研发了AHL-GEC-IDE（<http://sumcu.suda.edu.cn/AHLwGECwIDE/main.htm>），可以方便地用于RTOS实践。该IDE以通用嵌入式计算机GEC为目标对象，开发了AHL-GEC-IDE集成开发环境。它具有编辑、编译、链接等功能，特别是在配合金葫芦硬件后，可直接运行调试程序，兼容常用嵌入式集成开发环境。该环境摒弃了许多烦琐的设置，配合软件最小系统模板，使读者易于入门，简捷实用；特别是配合程序模板中提供的printf输出调试功能，使得打桩调试及运行跟踪显示得以方便实现。



样例程序：规范的体现基本知识要素的样例程序是学习RTOS的基础。样例设计要有统一的考虑，要能够体现对应的知识点。

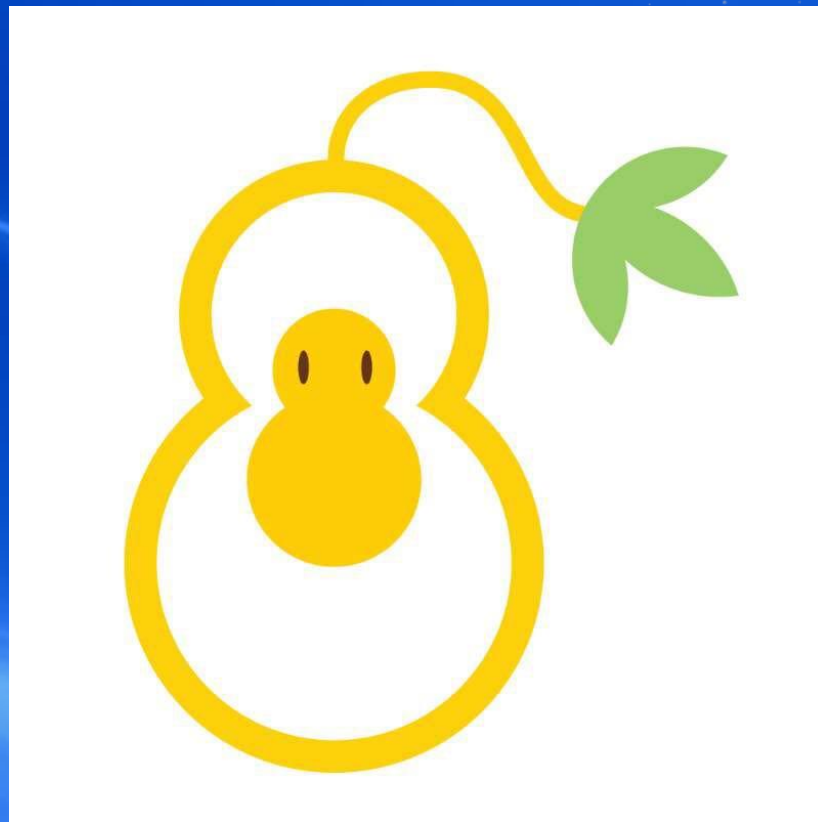
例如，对应消息队列，要能够演示处消息堆积与消息丢失的情况，才能说清楚编程时如何设定最大消息个数。



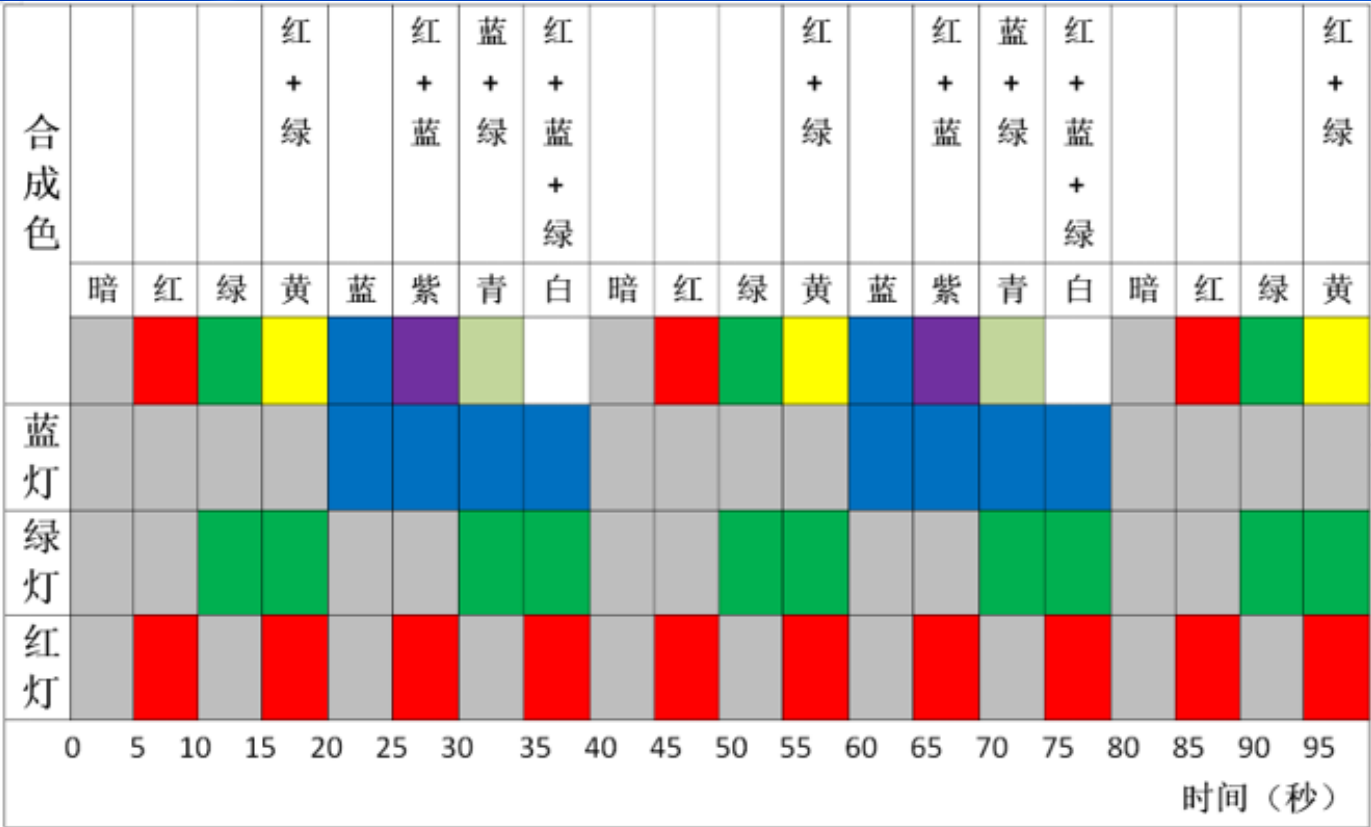
1.4 第一个实践样例

“照葫芦画瓢”基本理念

照葫芦画瓢：出自宋·魏泰《东轩笔录》第一卷，比喻照着样子模仿，表示简单易行之含义。而古希腊哲学家亚里士多德说过：“人从儿童时期起就有模仿本能，他们用模仿而获得了最初的知识，模仿就是学习”。借此，期望通过建立符合软件工程基本原理的“葫芦”，为“照葫芦画瓢”提供基础，达到降低学习难度之目的。



第一个样例功能



第一个样例程序功能

建立标准工程框架：做到“分门别类，各有归处”

| | |
|------------------|--|
| 01_Doc | 文档文件夹：文档作为工程密切相关部分，是软件工程的基本要求 |
| 02_CPU | CPU 文件夹：存放 CPU 相关文件 |
| 03_MCU | MCU 文件夹：含有 linker_file、startup、MCU_drivers 下级文件夹 |
| 04_GEC | GEC 文件夹：引入通用嵌入式计算机（GEC）概念，预留该文件夹 |
| 05_UserBoard | 用户板文件夹：含有硬件接线信息的 User.h 文件及应用驱动 |
| 06_SoftComponent | 软件构件文件夹：含有与硬件无关的软件构件 |
| 07_AppPrg | 应用程序文件夹：应用程序主要在此处编程 |

工程框架

第一个样例工程中的线程

| 归属 | 线程名 | 执行函数 | 优先级 | 线程功能 | 中文含义 |
|----|----------------|-------------------|-----|---------------|------|
| 内核 | main_thread | app_init | 10 | 创建其他线程 | 主线程 |
| | idle | idle | 31 | 空闲线程 | 空闲线程 |
| 用户 | thd_redlight | thread_redlight | 10 | 红灯以 5s 为周期闪烁 | 红灯线程 |
| | thd_greenlight | thread_greenlight | 10 | 绿灯以 10s 为周期闪烁 | 绿灯线程 |
| | thd_bluelight | hread_bluelight | 10 | 蓝灯以 20s 为周期闪烁 | 蓝灯线程 |



1.5 为RTOS教学所做的准备工作总结

- (1) 实践硬件: AHL-STM32L431
- (2) 开发环境: AHL-GEC-IDE
- (3) 样例程序编制: 针对各基本要素, 按照统一模板编制了样例程序
- (4) 撰写书籍: 嵌入式实时操作系统--基于RT-Thread的EAI&IoT系统开发, 机械工业出版社, 2021年
- (5) 课件: 需要时可以获取, 正在进行其他教学资源建设
- (6) 科研拓展: AHL-EORS应用、NB-IoT应用开发
- (7) 其他努力: CE-Python实践

二、教材脉络梳理

2.1 内容简介

嵌入式实时操作系统是嵌入式人工智能与物联网终端的重要工具和运行载体。本书以国产RT-Thread实时操作系统为背景阐述实时操作系统的线程、调度、延时函数、事件、消息队列、信号量、互斥量等基本知识要素，给出实时操作系统下程序设计方法。全书分为基础应用篇（1~7章）、原理剖析篇（8~12章）及综合实践篇（13~14章）三大部分，如果读者只做实时操作系统下的应用开发，可只阅读基础应用篇与综合实践篇；如果希望理解实时操作系统原理，则建议通读全书。

本书面向应用开发工程师、高等学校研究生及高年级本科生，也可作为实时操作系统技术培训用书。



2.2 一级目录

第1章 RTOS的基本概念与线程基础知识

第2章 相关基础知识

第3章 RT-Thread第一个样例工程

第4章 RTOS下应用程序的基本要素

第5章 同步与通信的应用方法

第6章 底层硬件驱动构件

第7章 RTOS下程序设计方法

第8章 理解RT-Thread的启动过程

第9章 理解时间嘀嗒

第10章 理解调度机制

第11章 理解事件与消息队列

第12章 理解信号量与互斥量

第13章 基于RT-Thread的AHL-EORS应用

第14章 基于RT-Thread的NB-IoT应用开发

第1篇 基础应用篇

第2篇 原理剖析篇

第3篇 综合实践篇

2.3 配套开发环境AHL-GEC-IDE

苏州大学嵌入式团队在早期研发集成开发环境的基础上，经过三年多的努力，以通用嵌入式计算机GEC为目标对象，开发了AHL-GEC-IDE集成开发环境。它具有编辑、编译、链接等功能，特别是在配合金葫芦硬件后，可直接运行调试程序，兼容常用嵌入式集成开发环境。该环境摒弃了许多烦琐的设置，配合软件最小系统模板，使读者易于入门，简捷实用；特别是配合程序模板中提供的printf输出调试功能，使得打桩调试及运行跟踪显示得以方便实现。

三、RTOS中核心概念导引

3.1 线程

线程是RTOS中最重要的概念之一。在RTOS下，把一个复杂的嵌入式应用工程按一定规则分解成一个个功能清晰的小工程，然后设定各个小工程的运行规则，交给RTOS管理，这就是基于RTOS编程的基本思想。这一个个小工程被称为线程（Thread），RTOS管理这些线程，被称为调度（Scheduling）。

补充说明：多进程与多线程

RTOS一般是单进程的，所以不再出现进程概念，这是RTOS由RTOS应用场合决定的，用于连续运行的实时性要求较高的工业场景，而台式计算机、笔记本电脑、平板电脑、手机等是多进程的，如看似同时打开WORD、播放音乐、发邮件等，每个正在运行的程序均是一个进程，每个进程一般会有若干线程

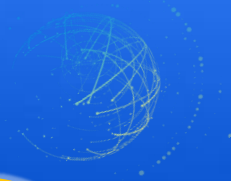
要给RTOS中的线程下一个准确而完整的定义并不十分容易，可以从不同视角理解线程。**从线程调度视角理解**，可以认为，RTOS中的线程是一个功能清晰的小程序，是RTOS调度的基本单元；**从软件设计视角来理解**，就是在软件设计时，需要根据具体应用，划分出独立的、相互作用的程序集合，这样的程序集合就被称为线程，每个线程都被赋予一定的优先级；**从CPU视角理解**，在单CPU下，某一时刻CPU只会处理（执行）一个线程，或者说只有一个线程占用CPU。RTOS内核的关键功能就是以合理的方式为系统中的每个线程分配时间（即调度），使之得以运行。

实际上，根据特定RTOS，线程可能被称为任务(Task)，也可能使用其他名词，含义或许稍有差异，但本质不变，也不必花过多精力追究其精确语义，因为**学习RTOS的关键在于理解调度过程、掌握线程设计方法、提高编程鲁棒性、理解底层驱动原理、提高可移植性与可复用性、程序规范性、提高实际开发能力等。**

3.2 调度

调度 (Scheduling) 就是决定该轮到哪个线程运行了，它是内核最重要的职责。每个线程根据其重要程度不同，被赋予一定的优先级。不同的调度算法对RTOS的性能有较大影响，基于优先级的调度算法是RTOS常用的调度算法，其核心思想是，总是让处于就绪态的、优先级最高的线程先运行。然而何时高优先级线程掌握CPU的使用权，由使用的内核类型确定，基于优先级的内核有不可抢占型和可抢占型两种类型。

3.3 线程的三要素：线程函数、线程堆栈、线程描述符



线程函数。一个线程，对应一段函数代码，完成一定功能。从代码上看，线程函数与一般函数并无区别，但是从线程自身视角来看，它认为CPU就是属于它自己的，并不知道还有其他线程存在。线程函数也不是用来被其他函数直接调用的，而是由RTOS内核调度运行。要使线程函数能够被RTOS内核调度运行，必须将线程函数进行“登记”，要给线程设定优先级、设置线程堆栈大小、给线程编号等，不然有几个线程都要运行起来，RTOS内核如何知道哪个该先运行呢？由于任何时刻只能有一个线程在运行（处于激活态），当RTOS内核使一个线程运行时，之前的运行线程就会退出激活态。CPU被处于激活态的线程所独占，从这个角度看，线程函数与无操作系统（NOS）中的“main”函数性质相近，一般被设计为“永久循环”，认为线程一直在执行，永远独占处理器。

线程堆栈。它是独立于线程函数之外的RAM，按照“先进后出”策略组织的一段连续存储空间，是RTOS中线程概念的重要组成部分。在RTOS中被创建的每个线程都有自己私有的堆栈空间，在线程的运行过程中，堆栈用于保存线程程序运行过程中的局部变量、线程调用普通函数时会为线程保存返回地址等参数变量、保存线程的上下文等。

线程描述符。线程被创建时，系统会为每个线程创建一个唯一的线程描述符（Task Descriptor, TD），它相当于线程在RTOS中的一个“身份证”，RTOS就是通过这些“身份证”来管理线程和查询线程信息的。

3.4 线程的四种状态：终止态、阻塞态、就绪态和激活态



RTOS中的线程一般有四种状态，分别为终止态、阻塞态、就绪态和激活态。在任一时刻，线程被创建后所处的状态一定是四种状态之一。

终止态：线程已经完成或被删除，不再需要使用CPU。

阻塞态：又可称为“挂起态”。线程未准备好，不能被激活，因为该线程需要等待一段时间或某些情况发生；当等待时间到或等待的情况发生时，该线程才变为就绪态，处于阻塞态的线程描述符存放于等待列表或延时列表中。

就绪态：线程已经准备好可以被激活，但未进入激活态，因为其优先级等于或低于当前的激活线程，一旦获取CPU的使用权就可以进入激活态，处于就绪态的线程描述符存放于就绪列表中。

激活态：又称“运行态”，该线程在运行中，线程拥有CPU使用权。

3.5 RTOS下的时间嘀嗒

时钟嘀嗒 (Time Tick) ，是RTOS中时间的最小度量单位,是线程调度的基本时间单元。主要用于系统计时、线程调度等。也就是说，要进行线程切换，至少等一个时间嘀嗒。时钟嘀嗒由硬件定时器产生，一般以毫秒 (ms) 为单位。
本系统使用1ms。

3.6 RTOS下的延时函数

RTOS中，线程一般不采用原地空跑（空循环）的方式进行延时（该方式线程仍然占用CPU的使用权），而往往会使用到延时函数（该方式线程会让出CPU使用权），通过使用延时列表管理延时线程，从而实现对线程的延时。在RTOS下使用延时函数，内核把暂时不需执行的线程，插入到延时列表中，让出CPU的使用权，并对线程进行调度。

3.7 事件与消息队列

在RTOS中，当为了协调中断与线程之间或者线程与线程之间同步，但不需要传送数据时，常采用事件作为手段。

在RTOS中，如果需要在线程间或线程与中断间传送数据，那就需要采用消息队列作为同步与通信手段。

3.8 信号量与互斥量

当共享资源有限时，可以采用信号量来表达资源可使用的次数，当线程获得信号量时就可以访问该共享资源了。

当共享资源只有一个时，为了确保在某个时刻只有一个线程能够访问该共享资源，可以考虑采用互斥量来实现。








四、实践源代码导引

电子资源下载：

<http://sumcu.suda.edu.cn/RTwThread/list.htm>



获得途径：百度搜索“苏州大学嵌入式学习社区”官网，在“著作”→“RTOS”→“RT-Thread”。

| | |
|--|----------------------------------|
| ▼  RT-Thread | 电子资源根文件夹名 |
|  01-Information | 资料文件夹（存放 RT-Thread 编程指南、芯片原始资料等） |
|  02-Document | 文档文件夹（存放实践平台快速指南、习题、辅助阅读材料等） |
|  03-Hardware | 硬件文件夹（存放硬件资源电子文档） |
| >  04-Software | 软件文件夹（存放各章样例源程序，按照章进行编号） |
|  05-Tool | 工具文件夹（存放实践中可能使用的软件工具） |
| >  06-Other | 其他（硬件测试程序等） |

五、实践操作演示

- ▼ CH03
 - > CH3.3-NOS
 - > CH3.4-RT-Thread-20210331
- ▼ CH05
 - > CH5.2.3-ISR_Event_RT-Thread_STM32L431-20210331
 - > CH5.2.4-Event_RT-Thread_STM32L431_20210331
 - > CH5.3-MessageQueue_RT-Thread_STM32L431_20210331
 - > CH5.4-Semaphore_RT-Thread_STM32L431_20210331
 - > CH5.5-Mutex_RT-Thread_STM32L431_20200331



Thank You!