HRTOS 快速入门指南

版本：HRTOS V4.0.0 LTS

---

1. HRTOS 简介

HRTOS（Hard Real-Time Operating System）是一款面向资源受限 MCU 平台设计的轻量级实时操作系统。

HRTOS 主要面向 8051 系列单片机，在有限硬件资源条件下提供完整的实时系统基础能力，包括：

- 任务调度
- 时间管理
- 同步通信
- 中断管理
- 消息通信
- Shell 调试
- Modbus RTU 通信

HRTOS 的设计目标：

- 轻量化
- 高可靠
- 易学习
- 易扩展

当前版本：

HRTOS V4.0.0 LTS

---

2. 五分钟运行 HRTOS

如果你只是希望快速体验 HRTOS，可以直接运行官方 Demo。

第一步：打开示例工程

进入：

```
Demo/01_Hello_HRTOS/Project/
```

打开：

```
Hello_HRTOS.uvproj
```

第二步：编译工程

使用：

- Keil C51

完成工程编译。

编译成功后生成 HEX 文件。

第三步：下载程序

使用 STC ISP 等下载工具，将 HEX 文件下载到开发板。

推荐测试平台：

- STC12C5A60S2

第四步：查看运行效果

程序运行后，即可观察 HRTOS 基础任务调度效果。

该示例用于展示：

- 系统初始化
- 任务创建
- 基础任务调度

这是学习 HRTOS 推荐的第一个示例。

---

3. 开发环境准备

推荐开发环境

工具	说明
Keil C51	8051 编译环境
STC ISP	STC 单片机下载工具
STC12C5A60S2	推荐测试平台

HRTOS 已提供完整工程配置。

用户无需从零创建工程，可以直接打开官方示例进行编译运行。

---

4. 工程目录说明

HRTOS V4.0.0 包含多个功能模块：

目录	说明
Demo	官方功能演示工程
INC	HRTOS 公共头文件
LIB	HRTOS 静态库文件
Docs	用户手册及系统文档
Shell	串口 Shell 人机交互模块
Modbus_RTU	Modbus RTU 工业通信模块
Debug	系统运行监测与故障排查
Myos	自定义调度策略演示
Driver	HRTOS 驱动库及演示
Examples	官网实验示例

5. Demo 示例说明

HRTOS 提供多个官方示例，帮助开发者逐步理解系统功能。

5.1 Hello_HRTOS

路径：

`Demo/01_Hello_HRTOS`

用途：

- HRTOS 初始化
- 创建任务
- 基础任务调度

推荐第一次使用 HRTOS 时运行。

5.2 Hello_Event

路径：

`Demo/02_Hello_Event`

用途：

- 学习事件机制
- 理解任务同步

5.3 Hello_Interrupt

路径：

`Demo/03_Hello_Interrupt`

用途：

- 学习中断处理
- 理解中断任务协同运行

---

6. 创建第一个 HRTOS 任务

HRTOS 应用程序主要由任务组成。

用户通过创建任务实现不同功能模块，系统负责任务调度。

一个简单任务示例：

```
void app_task(void)
{
    while(1)
    {
        /* 用户任务代码 */

        os_delay(100);
    }
}
```

任务特点：

- 独立运行
- 由系统统一调度
- 可以使用 HRTOS 提供的同步和通信机制

---

7. HRTOS 核心模块介绍

HRTOS 提供以下核心功能模块：

模块	功能
task	任务管理
kernel	内核接口
time	时间管理
semaphore	信号量
mutex	互斥锁
msgq	消息队列
mailbox	邮箱
event	事件机制
interrupt	中断管理
wait	统一等待机制
config	系统配置

详细 API 使用方法请参考：

`Docs/HRTOS API Reference Manual.pdf`

---

8. 扩展模块

8.1 Shell

路径：

`Shell`

Shell 提供串口人机交互能力，用于系统调试和功能扩展。

主要用途：

- 查看系统状态
- 调试任务运行情况
- 查询系统版本
- 测试系统功能

8.2 Modbus RTU

路径：

`Modbus_RTU`

用于展示 HRTOS 在工业通信场景中的应用。

主要功能：

- Modbus RTU 协议支持
- 通信框架示例
- 工业设备通信应用参考

---

9. 高级示例

9.1 Coop

路径：

`Coop`

Coop 是 HRTOS 协作式调度示例。

用于帮助开发者理解：

- 基础任务调度思想
- 协作式运行机制

9.2 Myos

路径：

`Myos`

Myos 是基于 HRTOS 实现的自定义调度策略演示。

用于展示：

- 调度机制设计
- 任务切换实现方式

注意：

Myos 不属于 HRTOS 核心模块，而是独立的调度策略演示。

---

10. 推荐学习路线

建议按照以下顺序学习：

第一步：运行 Demo

运行：

`Demo/01_Hello_HRTOS`

理解系统基本运行流程。

第二步：阅读使用手册

查看：

`Docs/HRTOS 使用入门.pdf`

学习 HRTOS 使用方法。

第三步：创建自己的任务

掌握：

- 任务创建
- 延时控制
- 优先级设置

第四步：学习同步通信

学习：

- 信号量
- 互斥锁
- 消息队列
- 事件

第五步：开发实际应用

结合：

- Shell
- Modbus RTU

开发完整嵌入式应用。

---

11. 常见问题

如果开发过程中遇到问题，请优先参考：

用户手册：

`Docs/HRTOS 使用入门.pdf`

包含：

- 开发环境配置
- 工程创建方法
- 基础使用流程

系统手册：

`Docs/HRTOS 4.0 System Manual.pdf`

包含：

- 系统架构
- 内核设计
- 调度机制
- 底层实现说明

同时可以查看：

- FAQ 文档
- Demo 工程
- 示例代码

---

12. 系统配置概览

默认配置：

项目	配置
普通任务数量	16
高速任务数量	2
优先级范围	0~10
普通任务优先级	0~7
高速任务优先级	8

项目	配置
中断任务优先级	9
中断嵌套优先级	10

13. 结语

HRTOS 致力于在资源受限 MCU 平台上提供简单、可靠、易学习的实时操作系统体验。

通过完整的：

- 内核
- 示例
- 文档
- 调试工具
- 通信模块

开发者可以快速掌握 RTOS 基础，并构建自己的 8051 嵌入式应用。

HRTOS Project

https://hrtos.com