

HRTOS 使用快速入门

HRTOS V4.0.0 LTS 版本共包含 10 个主要目录，分别为：

- **coop**: HRTOS 协作式调度示例，用于帮助开发者理解基础任务调度机制；
- **Myos**: 自定义调度策略演示，展示如何基于 HRTOS 实现不同调度方案；
- **demo**: 系统功能演示示例；
- **docs**: 项目开发文档及接口说明；
- **inc**: HRTOS 核心头文件；
- **lib**: 系统相关静态库文件；
- **Driver**: HRTOS 驱动库及演示示例（DHT11 和超声波模块因设备原因未验证）；
- **Examples**: 官网实验源代码；
- **shell**: 串口人机交互模块，用于系统调试与功能扩展；
- **modbus_rtu**: 工业通信模块，提供标准 Modbus RTU 通信支持。

为了降低开发者学习和验证成本，本版本对 **demo** 演示工程进行了精简与优化，并提前完成工程配置。开发者无需进行复杂配置，即可使用 **Keil C51** 开发环境直接编译运行。

目前所有演示示例均已在 **STC12C5A60S2 单片机平台** 完成验证，开发者可以直接下载工程进行功能测试，也可以在现有示例基础上进行二次开发，快速构建自己的 8051 嵌入式应用。

演示文件路径：

Demo->01_Hello_HRTOS->Project->Hello_HRTOS.uvproj

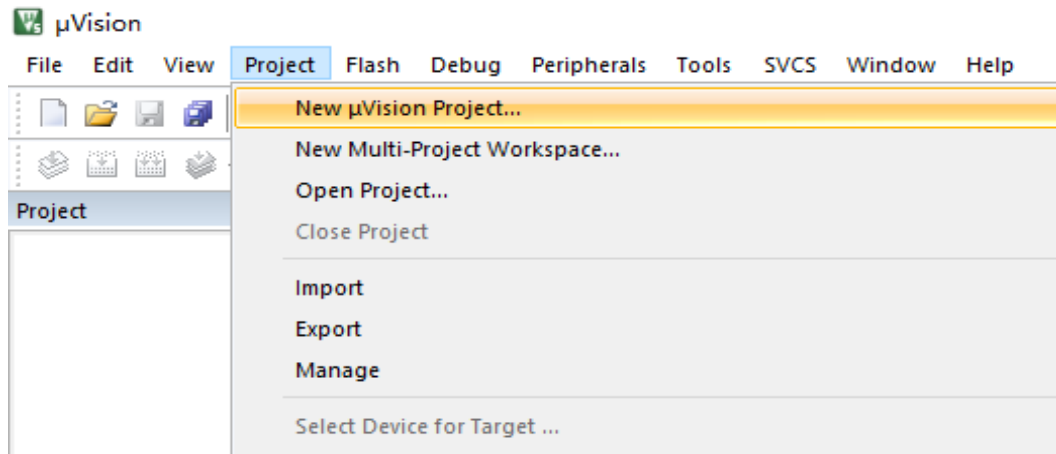
Demo->02_Hello_Event->Project->Hello_Event.uvproj

Demo->03_Hello_Interrupt->Project->Hello_Interrupt.uvproj

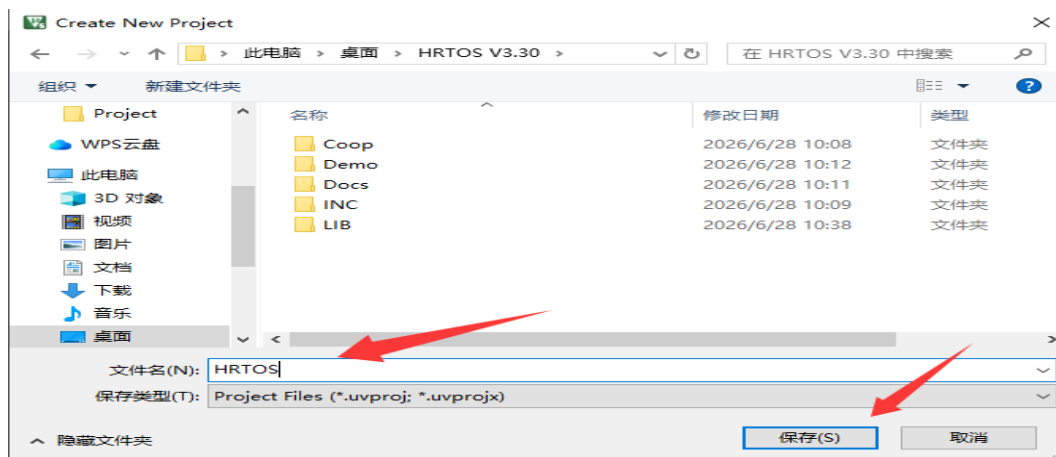
开发板 P1 口连接 LED 灯，部分开发板也有 P0 口连接，只需将 sbit P1 改成 sbit P0 即可

新项目创建流程以 STC12C5A60S2 为例（使用演示程序，无需操作）：

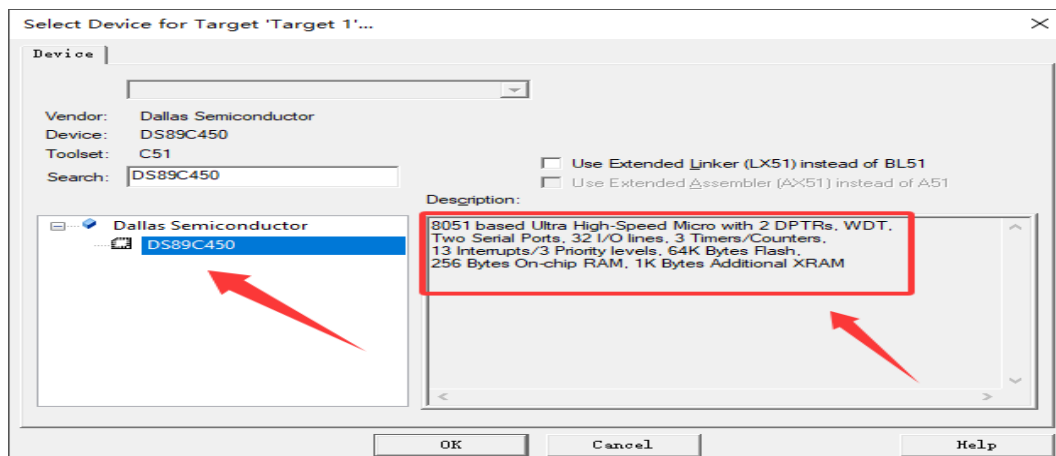
1.运行 keil，按图所示创建工程



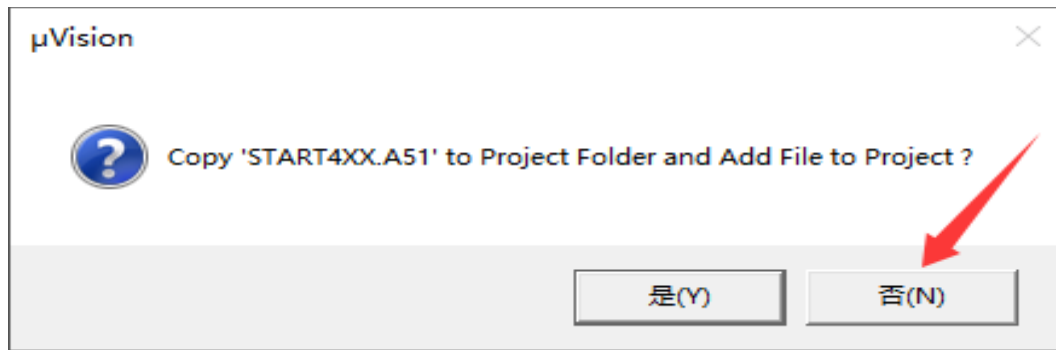
2.选择文件夹，输入工程名称，点击保存



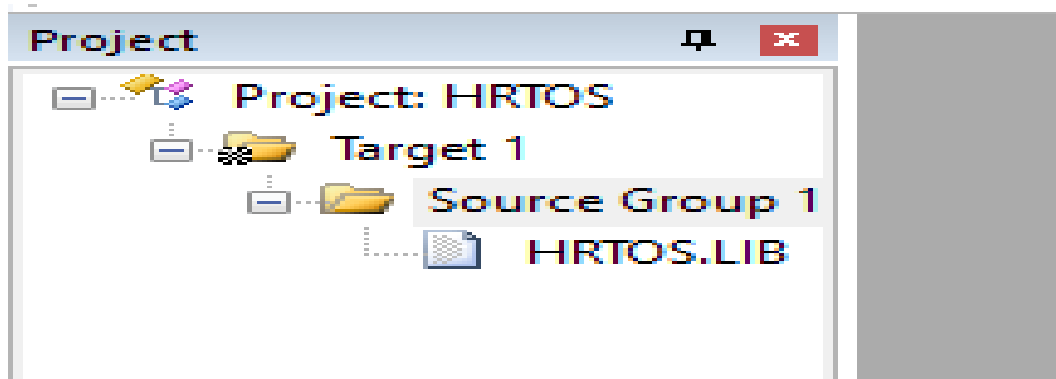
3.选择单片机型号，我这里选择的 DS89C450，这个型号的 ram 与使用的型号内存匹配



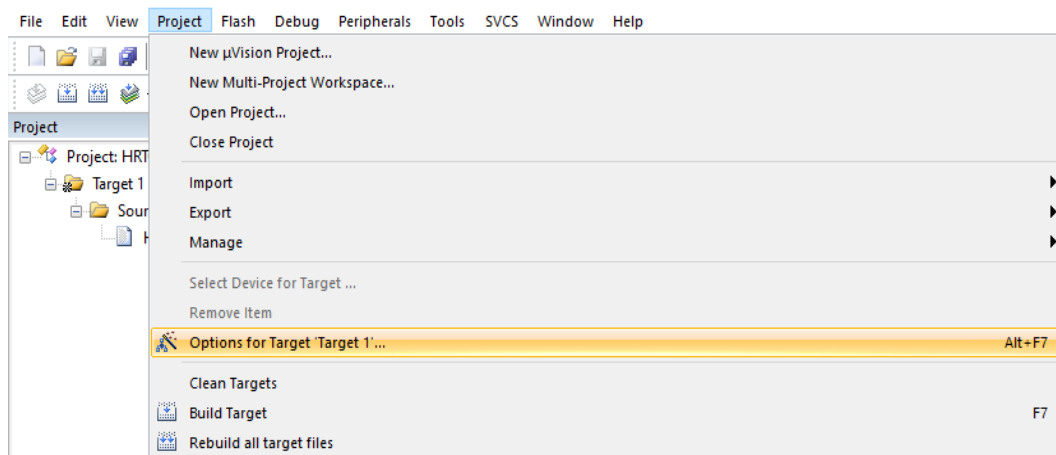
4.编译器询问是否添加通用文件，这里选择“否(N)”



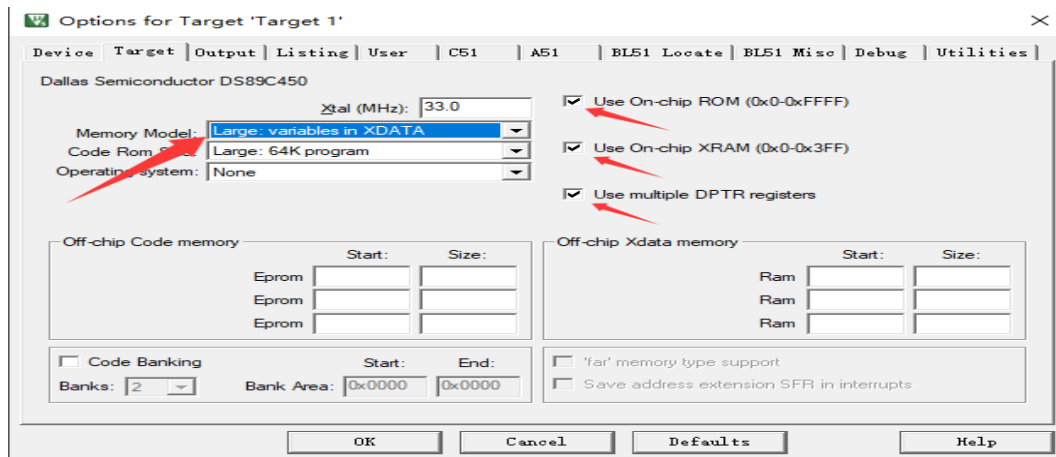
5.将 HRTOS.LIB 添加到工程目录，并添加用户程序



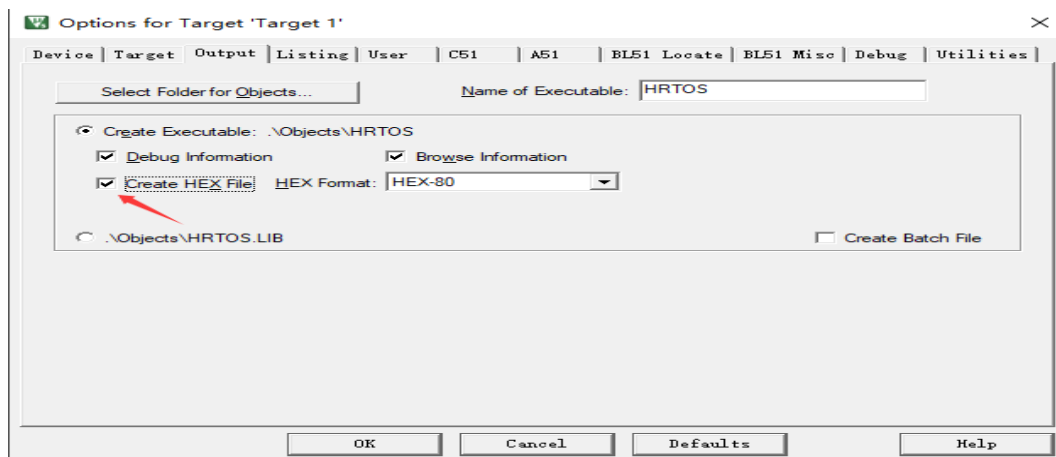
6.配置工程参数



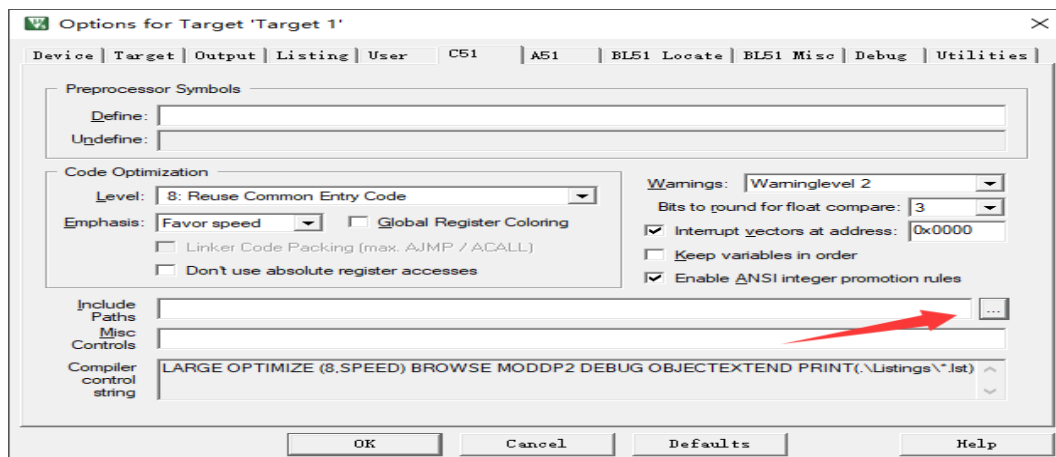
7.内存选择 xdata，默认 data 必须改

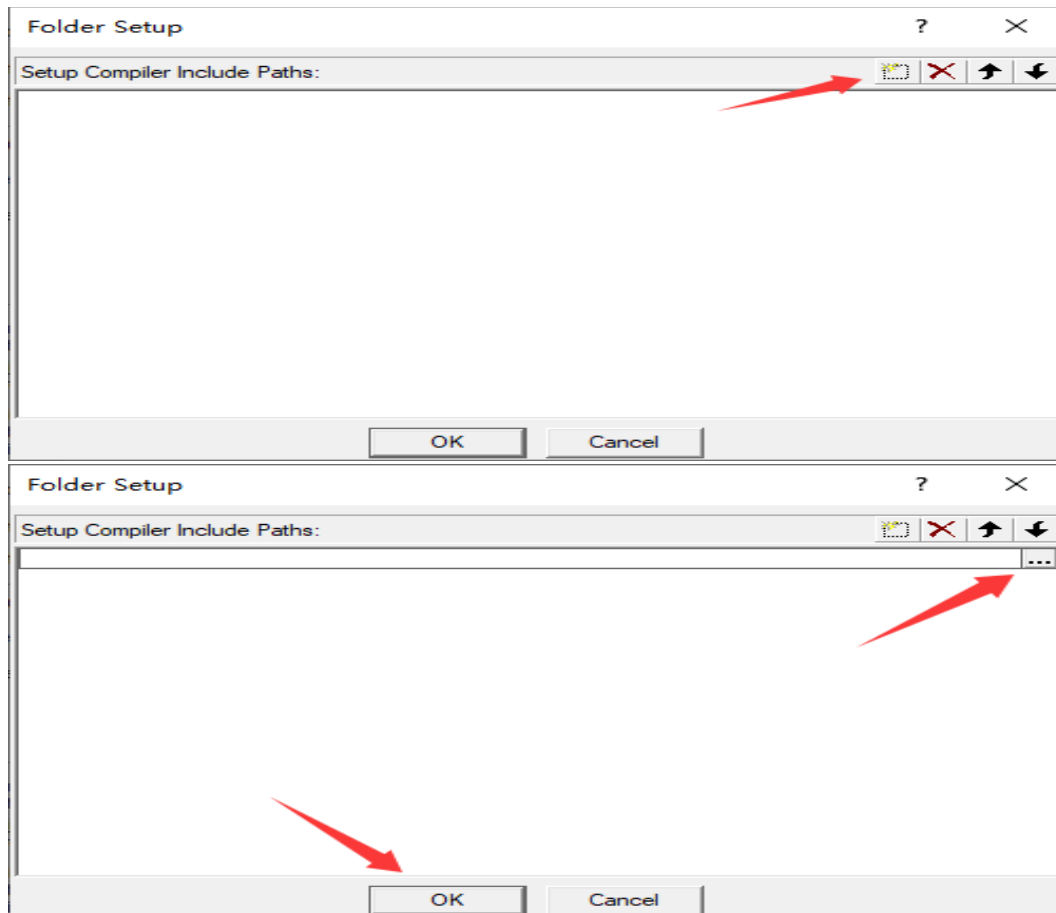


8.选择输出 hex 文件，下载单片机

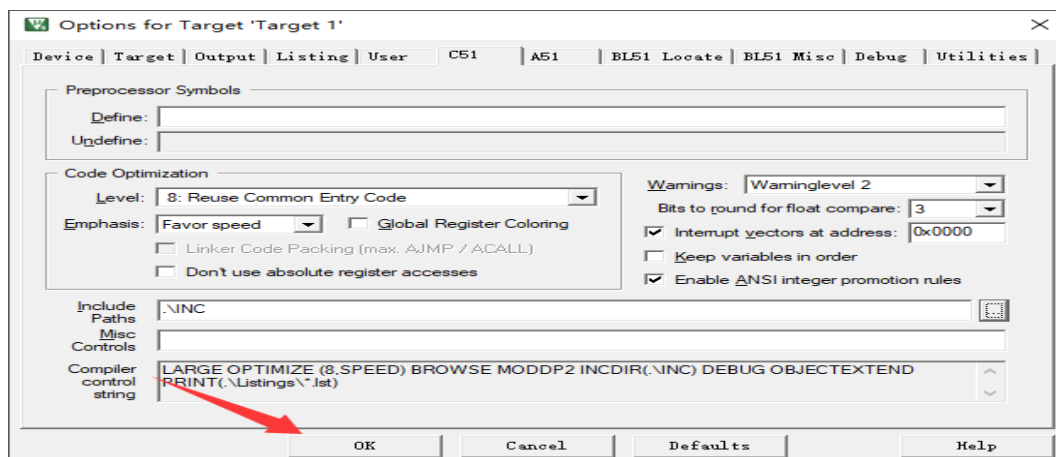


9.在 C51 页签，添加头文件目录





10.配置完成后，点击 OK 保存



(注意：MYOS 工程，代码优化等级请选择 0)

Options for Target 'myos'

Device | Target | Output | Listing | User | **C51** | A51 | BL51 Locate | BL51 Misc | Debug | Utilities

Preprocessor Symbols

Define:

Undefine:

Code Optimization

Level:

Emphasis:

☐ Linker Code Packing (max. AJMP / AJMP.L)

☐ Don't use absolute register accesses

Warnings:

Bits to round for float compare:

☒ Interrupt vectors at address:

☐ Keep variables in order

☒ Enable ANSI integer promotion rules

Include Paths: ...

Misc Controls:

Compiler control string:

OK Cancel Defaults Help

HRTOS V3.30 新版特性：

1. 统一 API 接口命名规范
2. 将 Myos 体系移除 HRTOS 内核，后续将在演进版本中，公开其实现方式
3. 增加 wait 统一阻塞机制
4. 重构事件、邮箱、信号量、消息队列、互斥锁机制，完善中断 API 机制
5. 增加 idle 机制，实现资源统一回收、挂载用户程序
6. 优化任务响应链路，提升系统实时响应水平
7. 创建统一测试框架，对包括基础功能、压力检测、内存泄露、性能分析展开长时间（20h+）测试，测试节点 1600 余个，调度触发超 100 万次、中断触发超 700 万次，运行稳定性、CPU 占比、断言输出均无异常，由于测试框架编译后文件体积较大（rom 原 60kb+, 精简后 40kb+; ram 2kb+），测试平台选用 STC8H8U64K
8. 构建文档自动生成体系
9. 配套开源协作式系统 HRTOS Coop 方便大家理解

HRTOS V3.31 新版特性：

1. 修复同优先级无法正常切换
2. 整理开源 Myos 自调度策略（基于 HRTOS 实现），用户可自行参考实现属于自己的调度策略，该模块仅作为演示实现。（网址：[HRTOS Myos Kernel | HRTOS 项目实战](#)）
3. 回归测试，调度超 400 万次，运行超 8 小时

HRTOS V3.32 新版特性：

1. 增加 Modbus RTU 通信模块，提供标准化工业通信支持，完善嵌入式设备通信应用场景
2. 增加 Shell 命令交互模块，支持用户通过串口进行系统调试、状态查看及功能扩展
3. 完善时间管理及文档体系，修复压力测试过程中发现的潜在资源异常问题
4. 优化系统调度流程，提升任务切换稳定性，完善高优先级任务、普通任务及中断任务协同运行机制
5. 完善 Modbus、Shell 等组件与 HRTOS 内核的适配接口，统一模块化设计规范
6. 完成长期稳定性测试，系统连续运行 72 小时以上，调度触发超过 2000 万次，节点检测超过 60 万个，运行过程中无异常复位、任务异常及资源泄露问题
7. 完善开发文档及示例代码，补充通信模块、系统调试及应用开发相关说明
8. 完成 HRTOS V3.32 版本功能冻结，后续版本主要进行问题修复、生态完善及商业化定制支持

HRTOS V4.0.0 LTS 新版特性：

1. 完善系统功能演示示例，Demo 从原本 3 个增加到 13 个；
2. 完成周级压力测试（2026 年 8 月 22 日 17 点 51 分至 29 日 18 点 02 分），累计完成 8842 轮压力测试，每轮含 1611 个测试节点。CPU 占比稳定在 71%（由 IDLE 空闲任务测得），系统定时器触发中断 57510118 次，任务自行触发调度次数本次未记录。
3. 正式支持中断嵌套，高优先级中断可打断低优先级中断，也可以打断内核调度，确保系统的实时性。
4. 修复 V3 系列任务唤醒机制，避免因唤醒已删除任务造成的系统异常重启问题；
5. 完成 HRTOS 驱动库编写验证，目前除超声波和 DHT11 超声波模块外，其他部分已完成真实硬件验证，欢迎测试，如遇问题可通过 QQ 群和邮箱反馈。

作者寄语：

HRTOS 最初规划中新版本计划支持 STM32 等 32 位微控制器平台，但经过长期开发与调研后发现，目前 32 位 MCU 领域已经拥有较为成熟的 RTOS 生态。由于本人并非嵌入式行业从业者，HRTOS 的持续开发与维护主要来源于个人兴趣与对自研系统的责任感，受限于个人时间与精力，暂未开展 STM32 等 32 位平台适配工作。

早期部分介绍中提及支持非 51 系列单片机，是基于当时对未来发展方向的判断。随着项目不断演进，HRTOS 当前正式定位调整为：**面向资源受限 8051 单片机平台的轻量级硬实时操作系统**。对于此前信息可能造成的理解偏差，在此表示歉意。

HRTOS 核心内核针对 8051 系列单片机设计，支持多任务调度机制，包括最多 16 个标准任务、2 个高速任务以及中断任务支持；支持多级优先级调度和中断嵌套机制。系统通过合理控制资源占用，在有限硬件资源下提供稳定、可靠的实时任务运行能力。

HRTOS 初代版本最早完成于 2017 年 7 月 29 日，同年 11 月 18 日注册 hrtos.com 域名，并于 2026 年注册 hrtos.cn 作为项目介绍网站。

HRTOS 名称来源于 **Hard Real-Time Operating System**（硬实时操作系统），该缩写形式在嵌入式系统和实时计算领域具有明确含义，并非个人创造词汇。项

目采用这一名称代表开发目标与技术方向，并不意味着当前系统已经覆盖所有理论意义上的硬实时应用场景，而是希望不断向轻量、高可靠、高实时性的方向发展。

HRTOS 核心内核坚持个人开发者免费使用原则，同时禁止未经授权的逆向破解、商业复制及破坏系统完整性的行为。一个自主设计的实时操作系统凝聚了大量时间、精力与实践积累，希望在为开发者提供便利的同时，也能够获得对原创技术成果的尊重。

如果在使用过程中遇到问题，欢迎通过 admin@hrtos.com 进行反馈。由于 HRTOS 仍处于持续发展阶段，目前主要通过邮件方式进行交流，后续将根据项目发展情况建设更加完善的社区交流体系。

未来，HRTOS 将继续长期维护与优化，不断完善内核稳定性、开发体验和应用生态，努力成为一个轻量、可靠、值得信赖的实时操作系统。

—— HRTOS 作者