English | **中文**

# StaRT RTOS 简介

StaRT 是一个轻量实时内核，面向 Cortex-M。目标：用尽量少的代码和资源展示优先级调度、时间片、睡眠定时器与最基本的 IPC 框架。

## 1. 已具备功能

- 固定优先级 + 位图快速查找（0 为最高）
- 同优先级时间片轮转（round-robin）
- 线程生命周期：初始化 / 启动 / 睡眠 / 让出 / 删除 / 重启 / 退出
- 每线程软件定时器（用于 sleep 与超时）
- 有序定时器链表
- 轻量格式化输出 `s_printf`
- 信号量初步（互斥量、消息队列结构占位）
- 平台相关代码独立（汇编上下文切换 + 栈初始化）

## 2. 目录结构

```
include/    公共 API 与类型
libcpu/     架构移植 (CM3)
src/        内核源码
readme/     文档
bsp/        板级支持包
```

## 3. 配置

编辑 `StaRT_Config.h`（参考 StaRT_CONFIG.md）：

- `START_THREAD_PRIORITY_MAX`
- `START_TICK`
- `START_IDLE_STACK_SIZE`
- `START_DEBUG`
- `START_USING_IPC` 及子开关

修改后全量重编。

## 4. 快速上手

**c**

```c
s_thread t1;
static s_uint8_t t1_stack[512];

static void t1_entry(void)
{
    while (1) {
        s_printf("tick=%d\n", (int)s_tick_get());
        s_mdelay(500);
    }
}

int main(void)
{
    /* 硬件初始化 + SysTick_Handler 调用 s_tick_increase() */
    s_start_init();

    s_thread_init(&t1, t1_entry, t1_stack, sizeof(t1_stack), 5, 10);
    s_thread_startup(&t1);
```

```
    s_sched_start();
}
```

SysTick：

c

```
void SysTick_Handler(void)
{
    s_tick_increase();
}
```

自定义输出：

c

```
void s_putc(char c) { /* UART 发送 */ }
```

# 5. 核心 API（节选）

| 模块 | 主要函数 |
| --- | --- |
| 线程 | `s_thread_init`, `s_thread_startup`, `s_thread_sleep`, `s_thread_yield`, `s_thread_exit` |
| 调度 | `s_sched_start`, `s_sched_switch` (内部) |
| 定时 | `s_mdelay`, `s_timer_start`, `s_tick_get` |

| | |
|---|---|
| 同步 | `s_sem_init`, `s_sem_take`, `s_sem_release` |
| 输出 | `s_printf`, `S_DEBUG_LOG` |

返回值统一 `s_status`。

## 6. 端口要点

- 汇编：PendSV 上下文保存/恢复
- 栈初始化：`s_stack_init` 填入 xPSR / PC / LR (`s_thread_exit`)
- 关/开中断：`s_irq_disable / s_irq_enable`
- SysTick 调用 `s_tick_increase`
- 可选 `__s_ffs` 加速位图扫描（否则使用内建/回退）

移植相关详见 StaRT_TRANS.md。

## 7. 已知限制

- 互斥量、消息队列未完成
- 无优先级继承
- 无堆/内存管理
- 无栈溢出检测
- 定时器插入 O(n)

- `s_printf` 非线程安全

## 8. 调试建议

| 现象 | 排查 |
| --- | --- |
| 不切换 | 检查 SysTick 与位图 |
| 睡眠不醒 | 定时器回调未执行（中断未调） |
| HardFault | 栈未 8 字节对齐或返回地址异常 |
| 日志交叉 | 正常竞争，可忽略或加锁 |

## 9. 许可证

```
SPDX-License-Identifier: MIT
Copyright (c) 2025
```

## 10. 贡献

欢迎提交 Issue / PR：

- 保持单一职责
- 补充简要 Doxygen 注释
- 说明测试步骤

祝使用愉快！