

ToRTOS_Config.h 使用文档

该文件集中配置内核功能、规模参数与调试开关。修改后需全量重新编译。

当前文件内容：

```
#define TO_VERSION "1.0.0"
#define TO_LOWER_PRIORITY_NUM_HIGHER_PRIORITY 0
#define TO_THREAD_PRIORITY_MAX 32
#define TO_USING_CPU_FFS 1
#define TO_TIMER_SKIP_LIST_LEVEL 1
#define TO_TICK 1000
#define TO_PRINTF_BUF_SIZE 128
#define TO_IDLE_STACK_SIZE 256
#define TO_USING_STATIC_ALLOCATION 1
#define TO_USING_DYNAMIC_ALLOCATION 0
#define TO_DYNAMIC_MEM_SIZE 10240
#define TO_USING_IPC 1
#define TO_USING_MUTEX 1
#define TO_USING_RECURSIVE_MUTEX 1
#define TO_USING_SEMAPHORE 1
#define TO_USING_QUEUE 1
#define TO_DEBUG 1
```

1. 版本信息

TO_VERSION

- 内核版本字符串
- 用于标识当前内核版本

2. 优先级与调度

TO_LOWER_PRIORITY_NUM_HIGHER_PRIORITY

- 0: 较低数值 = 较低优先级 (0为最低)
- 1: 较低数值 = 较高优先级 (0为最高)
- 影响优先级映射逻辑

TO_THREAD_PRIORITY_MAX

- 最大可用优先级数量 (0 ~ N-1)
- 影响：位图宽度/就绪表大小。增大将增加 RAM 占用 (`t_thread_priority_table`)。
- 建议：32，根据任务数量规划。

TO_USING_CPU_FFS

- 1: 使用内置 **t_ffs** 或 **t_fls** (位扫描) 优化最高优先级查找
 - 0: 可退回软件查找 (需自行实现简易循环)
 - 若架构无 CLZ/汇编支持, 可保持 1 并提供 C 函数。
-

3. 定时器与 Tick

TO_TIMER_SKIP_LIST_LEVEL

- 目前实现仅使用 Level=1 (有序链表)
- 未来可扩展跳表加速插入。

TO_TICK

- 每秒 Tick 数 (Hz)
 - 用于: 时间片、sleep、信号量超时
 - 取值注意: 过大增加中断负载, 过小降低时间分辨率 (典型 1000)
-

4. 打印

TO_PRINTF_BUF_SIZE

- t_printf 内部缓冲区大小
 - 格式化字符串长度超过此值将被截断
 - 增大意味着消耗更多栈 (在 t_printf 调用栈帧中)
-

5. 空闲线程

TO_IDLE_STACK_SIZE

- Idle 线程栈大小
 - Idle 中仅执行清理与可选低功耗, 通常较小即可 (128~512)
-

6. 内存分配

TO_USING_STATIC_ALLOCATION

- 1: 启用静态分配 (线程、IPC对象预分配)
- 0: 禁用静态分配

TO_USING_DYNAMIC_ALLOCATION

- 1: 启用动态分配 (使用堆内存)
- 0: 禁用动态分配
- 注意: 至少一个分配方式必须启用

TO_DYNAMIC_MEM_SIZE

- 动态内存池大小（字节）
- 仅当 TO_USING_DYNAMIC_ALLOCATION=1 时有效

7. IPC 功能开关

TO_USING_IPC

- 总控开关：为 0 时所有 IPC 模块（信号量/互斥量/消息队列）编译剔除

TO_USING_SEMAPHORE

- 信号量支持（依赖 TO_USING_IPC=1）
- 关闭：相关结构与 API 不编译，节省代码空间

TO_USING_MUTEX

- 互斥量支持（依赖 TO_USING_IPC=1）

TO_USING_RECURSIVE_MUTEX

- 递归互斥量支持（依赖 TO_USING_IPC=1）

TO_USING_QUEUE

- 消息队列支持（依赖 TO_USING_IPC=1）

8. 调试

TO_DEBUG

- 1：启用调试日志输出
- 0：调试宏为空，不产生代码
- 输出级别宏（信息/警告/错误）由调用处传入 level

9. 典型裁剪配置示例

最小精简（仅线程 + 睡眠）

```
#define TO_THREAD_PRIORITY_MAX    32
#define TO_USING_CPU_FFS          1
#define TO_TIMER_SKIP_LIST_LEVEL 1
#define TO_TICK                   1000
#define TO_PRINTF_BUF_SIZE        64
#define TO_IDLE_STACK_SIZE        128
#define TO_USING_STATIC_ALLOCATION 1
#define TO_USING_DYNAMIC_ALLOCATION 0
#define TO_USING_IPC               0
#define TO_DEBUG                   0
```

完整开发调试

```
#define TO_THREAD_PRIORITY_MAX    32
#define TO_USING_CPU_FFS          1
#define TO_TIMER_SKIP_LIST_LEVEL 1
#define TO_TICK                   1000
#define TO_PRINTF_BUF_SIZE        256
#define TO_IDLE_STACK_SIZE        512
#define TO_USING_STATIC_ALLOCATION 1
#define TO_USING_DYNAMIC_ALLOCATION 1
#define TO_DYNAMIC_MEM_SIZE        20480
#define TO_USING_IPC               1
#define TO_USING_SEMAPHORE         1
#define TO_USING_MUTEX             1
#define TO_USING_RECURSIVE_MUTEX  1
#define TO_USING_QUEUE             1
#define TO_DEBUG                   1
```

10. 依赖关系

宏	依赖
TO_USING_SEMAPHORE	TO_USING_IPC
TO_USING_MUTEX	TO_USING_IPC
TO_USING_RECURSIVE_MUTEX	TO_USING_IPC
TO_USING_QUEUE	TO_USING_IPC
TO_USING_CPU_FFS	提供 <code>t_ffs</code> 或 <code>t_fls</code> 实现
TO_TICK	SysTick 配置
TO_DYNAMIC_MEM_SIZE	TO_USING_DYNAMIC_ALLOCATION

11. 修改注意

- 修改 `TO_THREAD_PRIORITY_MAX` 后需清理/重建工程（防止旧对象文件里数组尺寸不匹配）。
- 降低 `TO_TICK` 需要调整与毫秒换算：`t_tick_from_ms = (ms * TO_TICK)/1000`。
- 禁用 IPC 后，无法再调用 `t_sema_xxx` 等IPC函数。
- 至少启用一种内存分配方式。

12. 运行时验证建议

配置项	验证
TO_THREAD_PRIORITY_MAX	创建多个不同优先级线程并确认调度顺序
TO_TICK	使用延时 100ms 测试 tick 频率精度（串口时间戳）
TO_PRINTF_BUF_SIZE	输出长字符串确认截断行为
TO_DEBUG	确认调试日志有/无输出
TO_USING_SEMAPHORE	测试信号量阻塞 & 释放
TO_IDLE_STACK_SIZE	压测 idle（增加打印/栈水位检查）
TO_USING_STATIC_ALLOCATION	测试静态创建线程和IPC对象
TO_USING_DYNAMIC_ALLOCATION	测试动态分配和释放

13. 扩展建议

可新增配置：

- TO_DEBUG_LEVEL（过滤日志等级）
- TO_ENABLE_ASSERT（启用断言）
- TO_STACK_PATTERN（栈填充用于溢出检测）
- TO_TICKLESS_ENABLE（未来 Tickless 支持）

14. 常见配置错误

现象	排查
创建线程失败无输出	priority>=MAX 或 tick=0
信号量 API 未定义	TO_USING_SEMAPHORE 未开启
调度不工作	TO_TICK 未配置 SysTick，或 t_ffs/t_flx 实现错误
日志花屏/交叉	多线程同时调用 t_printf，无锁属正常竞争
内存分配失败	TO_USING_DYNAMIC_ALLOCATION=0 时尝试动态分配
IPC 创建失败	TO_USING_IPC=0 或未启用具体IPC类型