

编码规范和调度器函数说明

Author: 孙关平

0x00 编码规范

- 代码风格

代码编写使用K&R风格，关于代码格式化方法可以参考我的教程：
<https://github.com/ShareCat/CodeFormat>

- 注释风格

使用“/* ... */”风格。不要使用“// ...”风格注释，至于原因，看看这个帖子：
<https://blog.csdn.net/xingqingly/article/details/50833704>

- 函数名字和变量名字全部小写，不同单词之间使用“_”连接，如：

```
volatile uint8_t os_running_task_id = 0;
```

- 宏定义名字全部大写，如：

```
#define OS_SYS_TICK_MS      T_10MS  /* 调度器调度时间10ms */
```

- 结构体定义规范

```
/* 任务控制块结构 */
__packed typedef struct {
    uint8_t exist_flag:1;      /* 任务存在标记，1表示这个任务存在 */
    uint8_t ready_flag:1;     /* 任务就绪标记，可以运行，1表示可以运行（等待调度调用运行） */
    uint8_t suspend_flag:1;   /* 任务挂起标记，1表示挂起，挂起后就不再对任务就绪标记进行置位 */
    uint8_t sleep_flag:1;     /* 任务可以睡眠就绪标记，1表示这个任务就绪，但需要所有任务都是休眠就绪状态才让MCU进入休眠 */
    uint8_t wakeup_flag:1;    /* MCU睡眠后被唤醒，就会置1 */
    uint16_t tick_backup;     /* 任务周期时间备份 */
};
```

```

uint16_t tick_counter;    /* 任务周期时间计算 */
void (*func)(void);       /* 任务入口函数 */

#if OS_TASK_TIMER
    uint16_t timer;        /* 每个任务分配一个定时器，最小定时
                           单位为OS_SYS_TICK_MS的倍数 */
#endif /* OS_TASK_TIMER */
} OS_TASK_S;

```

0x01 创建任务

```

/**
 * @brief 调度器创建任务
 * @param OS_TASK_S*: 任务控制块指针
 * @param func: 任务函数指针
 * @param perid: 任务执行周期
 * @retval None
 */
void os_task_creat(volatile OS_TASK_S *ptask, void (*func)(void), uint16_t
perid)

```

0x02 任务挂起

```

/**
 * @brief 调度器任务挂起
 * @param *task_func: 任务函数指针，也就是要挂起的任务
 * @param suspend_state: ENABLE表示执行任务挂起
 * @retval TASK_NULL 表示任务不存在
                TASK_SUSPEND 表示任务被挂起
                TASK_UNSPEND 表示任务取消被挂起
 */
OS_TASK_STATE_E os_task_suspend(void *task_func, uint8_t suspend_state)

```

0x03 查询任务是否被挂起

```

/**
 * @brief 调度器任务挂起查询
 * @param *task_func: 任务函数指针
 * @retval TASK_NULL 表示任务不存在
                TASK_SUSPEND 表示任务被挂起
                TASK_UNSPEND 表示任务没有被挂起
 */
OS_TASK_STATE_E os_task_suspend_check(void *task_func)

```

0x04 调度器开始运行

```
/**
 * @brief 调度器开始运行
 * @param OS_TASK_S*: 任务控制块指针
 * @retval None
 */
void os_start(volatile OS_TASK_S *ptask)
```

0x05 调度器时间更新

```
/**
 * @brief 调度器所需要的定时更新任务就绪状态函数
 * @param OS_TASK_S*: 任务控制块指针
 * @retval None
 */
void os_interrupt_handle(volatile OS_TASK_S *ptask)
```

0x06 调度器初始化

```
/**
 * @brief 调度器初始化
 * @param OS_TASK_S*: 任务控制块指针
 * @retval None
 */
void os_init(volatile OS_TASK_S *ptask)
```