8.2 时间管理

8.2.1 OSTimeDly()函数

当我们需要对一个任务进行延时操作的时候就可以使用这个函数，函数原型如下。

```
void  OSTimeDly (OS_TICK    dly,OS_OPT     opt,OS_ERR    *p_err)
```

dly： 指定延时的时间长度，这里单位为时间节拍数。

opt： 指定延迟使用的选项，有四种选项。

OS_OPT_TIME_DLY	相对模式
OS_OPT_TIME_TIMEOUT	和 OS_OPT_TIME_DLY 一样
OS_OPT_TIME_MATCH	绝对模式
OS_OPT_TIME_PERIODIC	周期模式

p_err： 指向调用该函数后返回的错误码

“相对模式”在系统负荷较重时有可能延时会少一个节拍，甚至偶尔差多个节拍，在周期模式下，任务仍然可能会被推迟执行，但它总会和预期的“匹配值”同步。因此，推荐使用“周期模式”来实现长时间运行的周期性延时。

“绝对模式”可以用来在上电后指定的时间执行具体的动作，比如可以规定，上电 N 秒后关闭某个外设。

8.2.2 OSTimeDlyHMSM()函数

我们也可调用 OSTimeDlyHMSM()函数来更加直观的来对某个任务延时，OSTimeDlyHMSM()函数原型如下：

```
void  OSTimeDlyHMSM (CPU_INT16U    hours,          //需要延时的小时数
                     CPU_INT16U    minutes,        //需要延时的分钟数
                     CPU_INT16U    seconds,        //需要延时的秒数
                     CPU_INT32U    milli,          //需要延时的毫秒数
                     OS_OPT        opt,            //选项
                     OS_ERR       *p_err)
```

hours

minutes

seconds

milli： 前面这四个参数用来设置需要延时的时间，使用的是：小时、分钟、秒和毫秒这种格式，这个就比较直观了，这个延时最小单位和我们设置的时钟节拍频率有关，比如我们设置时钟节拍频率 OS_CFG_TICK_RATE_HZ 为 200 的话，那么最小延时单位就是 5ms。

opt： 相比 OSTimeDly()函数多了两个选项 OS_OPT_TIME_HMSM_STRICT 和 OS_OPT_TIME_HMSM_NON_STRICT，其他四个选项都一样的。

使用 OS_OPT_TIME_HMSM_NON_STRICT 选项的话将会检查延时参数，hours 的范围应该是 0~99，minutes 的范围应该是 0~59，seconds 的范围为 0~59，milli 的范围为 0~999。

使用 OS_OPT_TIME_HMSM_NON_STRICT 选项的话，hours 的范围为 0~999，minutes 的范围为 0~9999，seconds 的范围为 0~65535，mili 的范围为 0~4294967259。

p_err： 调用此函数后返回的错误码

8.2.3 其他有关时间函数

1、OSTimeDlyResume()函数

一个任务可以通过调用这个函数来“解救”那些因为调用了 OSTimeDly()或者OSTimeDlyHMSM()函数而进入等待态的任务，函数原型如下：

```
void    OSTimeDlyResume (OS_TCB    *p_tcb,OS_ERR    *p_err)
```

p_tcb： 需要恢复的任务的任务控制块。

p_err： 指向调用这个函数后返回的错误码。

2、OSTimeGet()和 OSTimeSet()函数

OSTimeGet()函数用来获取当前时钟节拍计数器的值。OSTimeSet()函数可以设置当前时钟节拍计数器的值，这个函数谨慎使用。