ST17H26系统睡眠函数 V1

进入低功耗函数：

(1)**int blt_sleep_wakeup** (**int** deepsleep, **int** wakeup_src, u32 wakeup_tick);

说明：此函数用于进入系统低功耗状态以及设置唤醒源。一旦调用此函数，则系统进入低功耗状态

参数：deepsleep :: 0 → 表示进入suspend mode. 1→表示deepsleep mode.

Wakeup_src :: 表示唤醒源。值选择如下：

PM_WAKEUP_CORE , // 表示数字部分唤醒。（比如gpio）用于suspend

PM_WAKEUP_TIMER ,// 表示定时唤醒,用于suspend mode and deepsleep

PM_WAKEUP_PAD , //用于deepsleep mode下gpio 唤醒。

Wakeup_tick ::用于定时唤醒时，时间设置。该时间为一个时间点。

示例：

a. **blt_sleep_wakeup** (0, *PM_WAKEUP_TIMER*, next_wakeup_tick)

以上调用之后，系统将会进入一个 suspend mode 状态。可以用 timer 唤醒，唤醒的时刻为 next_wakeup_tick。注意：next_wakeup_tick 必须为是未来的时间(当前系统时间之后的一个时间点)。

b. **blt_sleep_wakeup** (0, *PM_WAKEUP_CORE*, next_wakeup_tick)

以上调用之后，系统会进入 suspend mode 状态。此状态只可以通过外部的 io 状态改变唤醒。注意：即使此时 next_wakeup_tick 值不为 0，也不会通过 timer 唤醒。如果使用 IO 唤醒，另有 IO 口的其他配置

c. **blt_sleep_wakeup** (1, *PM_WAKEUP_TIMER*, next_wakeup_tick)

以上调用之后，系统会进入 deepsleep mode 状态。此状态可以通过 timer 唤醒，同 suspend timer 唤醒机制。

d. **blt_sleep_wakeup** (1, *PM_WAKEUP_PAD*, next_wakeup_tick)

以上调用之后，系统会进入 deepsleep mode 状态。此状态可以通过 io 唤醒。如果使用 IO 唤醒，另有 IO 口的其他配置

e. **blt_sleep_wakeup** (1, *PM_WAKEUP_PAD* | *PM_WAKEUP_TIMER*,
next_wakeup_tick)

以上调用之后，系统会进入一个 deepsleep mode 状态。可以通过 io 唤醒，也可以通过 timer 唤醒。