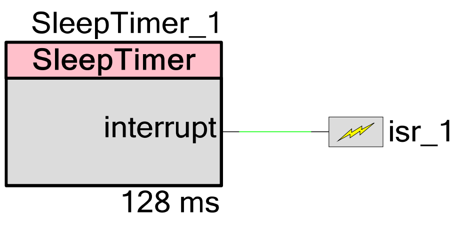
Sleep Timer

1.10

Features



- Wake up PSoC3/5 devices from low power modes: Alternate Active and Sleep
- Configurable option for issuing interrupt
- Generate a periodic interrupts while device is in Active mode
- 12 discrete intervals: 2ms, 4ms, 8ms, 16ms, 32ms, 64ms, 128ms, 256ms, 512ms, 1024ms, 2048ms and 4096ms

General Description

The Sleep Timer component is used to wake the device from Alternate Active and Sleep modes at a configurable interval. It also could be configured to issue an interrupt after wake up or issue an interrupt while being in Active mode at a configurable interval.

When to use a Sleep Timer

The Sleep Timer component can be used to periodically wake a device up from low power modes (Alternate Active and Sleep) at a configurable interval, with or without issuing interrupts. It also can be used to generate periodical interrupts while the device is in Active mode – acting like a counter.

This usage of the Sleep Timer can be implemented by hardware counters. However, this would use hardware resources inefficiently and would require the device to remain in Active Mode.

Input/Output Connections

This section describes output connection for the Sleep Timer. There are no input connections. An asterisk (*) in the list of I/O's states that the I/O may be hidden on the symbol under the conditions listed in the description of that I/O.

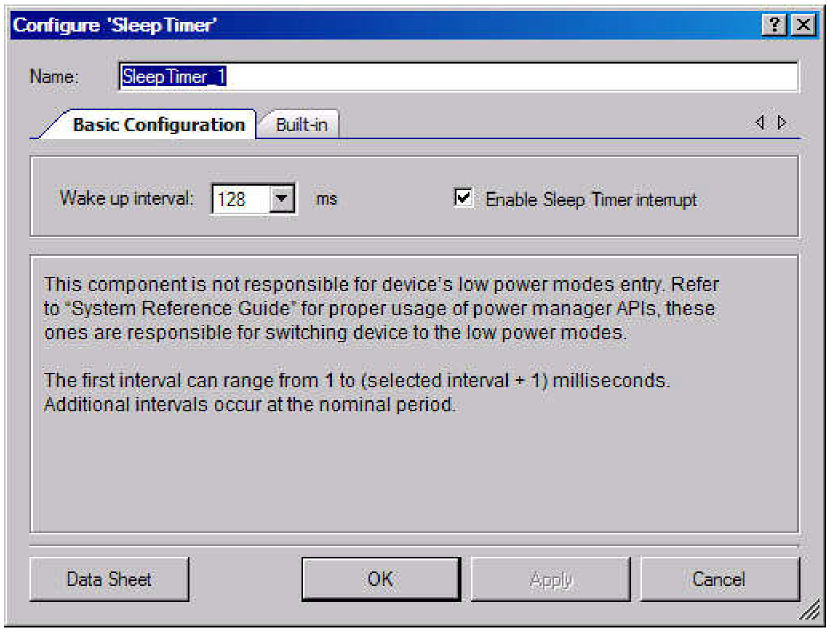
interrupt – Output *

The interrupt output has the central time wheel (CTW) interrupt source. The interrupt is issued when the CTW counter reaches the terminal count, specified in the component customizer. This output is not available when the Sleep Timer's interrupts are disabled.

PRELIMINARY

Parameters and Setup

Drag a Sleep Timer component onto your design and double-click it to open the Configure Sleep Timer dialog.



Name: SleepTimer_1

Wake up interval: 128 ms

[✓] Enable Sleep Timer interrupt

This component is not responsible for device's low power modes entry. Refer to "System Reference Guide" for proper usage of power manager APIs, these ones are responsible for switching device to the low power modes.

The first interval can range from 1 to (selected interval + 1) milliseconds. Additional intervals occur at the nominal period.

Basic Configuration

Interval

Defines the interval at which the Sleep Timer will wake the device up and/or generate interrupts, if it is configured to do so. Only discrete intervals are accepted from the range: 2ms, 4ms, 8ms, 16ms, 32ms, 64ms, 128ms, 256ms, 512ms, 1024ms, 2048ms and 4096ms.

These parameters define an initial configuration. The software can reconfigure these values only when the Sleep Timer is stopped.

Enable Interrupt

This parameter defines if the Sleep Timer component will issue an interrupt after the selected interval is reached. This parameter does not affect whether the component will wake up the device from low power modes.

This parameter defines an initial configuration. The software can reconfigure this parameter's setting.

Clock Selection

The Sleep Timer component uses the CTW and requires a 1 kHz clock for its operation. This clock is produced by the internal low speed oscillator (ILO). The ILO produces two primary independent output clocks with no external components, and with very low power consumption.

These two outputs operate at nominal frequencies of 1 kHz and 100 kHz. The two clocks run independently, are not synchronized to each other, and can be enabled or disabled together or independently. The API function that starts the Sleep Timer automatically enables the 1 kHz clock and leaves it enabled even after the component is stopped.

Placement

There is no placement specific information.

Resources

The Sleep Timer uses the following device resources:

- 1kHz ILO clock line
- Central Time Wheel (CTW) counter
- Central Time Wheel (CTW) counter's interrupt line

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name "SleepTimer_1" to the first instance of a component in a given design. You can rename the instance to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following table is "SleepTimer".

Function	Description
void SleepTimer_Start(void)	Starts the Sleep Timer operation.
void SleepTimer_Stop(void)	Stops the Sleep Timer operation.
void SleepTimer_EnableInt (void)	Enables Sleep Timer component issuing interrupt on wake up.
void SleepTimer_DisableInt (void)	Disables Sleep Timer component issuing interrupt on wake up.
void SleepTimer_SetInterval (uint8)	Sets interval for Sleep Timer to wake up.

Function	Description
uint8 SleepTimer_GetStatus (void)	Returns value of Power Manager Interrupt Status Register and clears all bits in this register.

void SleepTimer_Start(void)

Description:	Starts the Sleep Timer operation: enables 1 kHz clock, sets default parameters if they were not set previously by corresponding API functions.
Parameters:	Void
Return Value:	Void
Side Effects:	Enables 1 kHz ILO clocks and leaves it enabled after Sleep Time component is stopped.

void SleepTimer_Stop(void)

Description:	Stops the Sleep Timer component’s operation: disables wake up on CTW counter terminal count reached and disables interrupt issuing on foregoing condition.
Parameters:	Void
Return Value:	Void
Side Effects:	None

void SleepTimer_EnableInt (void)

Description:	Enables Sleep Timer component issuing interrupt on wake up.
Parameters:	Void
Return Value:	Void
Side Effects:	None

void SleepTimer_DisableInt (void)

Description:	Disables Sleep Timer component issuing interrupt on wake up.
Parameters:	Void
Return Value:	Void
Side Effects:	None

void SleepTimer_SetInterval (uint8 interval)

Description: Sets interval for Sleep Timer to wake up.

Parameters: uint8 interval: interval's value for Sleep Timer to wake up in.

Name	Value	Period
SleepTimer__CTW_2_MS	4'b0001	2 ms
SleepTimer__CTW_4_MS	4'b0010	4 ms
SleepTimer__CTW_8_MS	4'b0011	8 ms
SleepTimer__CTW_16_MS	4'b0100	16 ms
SleepTimer__CTW_32_MS	4'b0101	32 ms
SleepTimer__CTW_64_MS	4'b0110	64 ms
SleepTimer__CTW_128_MS	4'b0111	128 ms
SleepTimer__CTW_256_MS	4'b1000	256 ms
SleepTimer__CTW_512_MS	4'b1001	512 ms
SleepTimer__CTW_1024_MS	4'b1010	1024 ms
SleepTimer__CTW_2048_MS	4'b1011	2048 ms
SleepTimer__CTW_4096_MS	4'b1100	4096 ms

Return Value: Void

Side Effects: Interval value can be only changed when Sleep Timer is stopped. The first interval can range from 1 to (period + 1) ms. Additional intervals occur at the nominal period.

uint8 SleepTimer_GetStatus (void)

Description: Clears all interrupt status bits (react_int, limact_int, onepps_int, ctw_int, and ftw_int) in Power Manager Interrupt Status Register. This function should always be called (when Sleep Timer's interrupt is disabled or enabled) after wake up to clear ctw_int status bit.

Parameters: Void

Return Value: Returns 8 bits value (uint8) with bits set if corresponding event has occurred.

Name	Description
SleepTimer_PM_INT_SR_REACT	Device has moved from limited active to active mode
SleepTimer_PM_INT_SR_LIMACT	A limited active ready event has occurred
SleepTimer_PM_INT_SR_ONEPPSP	A onepps event has occurred
SleepTimer_PM_INT_SR_CTW	A central timewheel event has occured
SleepTimer_PM_INT_SR_FTW	A fast timewheel event has occured

Side Effects: If an interrupt gets generated at the same time as a clear, the bit will remain set (which causes another interrupt).

Sample Firmware Source Code

The following C language example demonstrates the basic functionality of the Sleep Timer component.

This example demonstrates device’s wake up from the Alternate Active Mode every 16 ms without issuing interrupt. It assumes the component has been placed in the schematic with the default name of "SleepTimer".

```
#include <device.h>

void main()
{
    /* Enable all interrupts by the processor. */
    CYGlobalIntEnable;
    /* Configure and start SleepTimer */
    SleepTimer_DisableInt();
    SleepTimer_SetInterval(SleepTimer__CTW_16_MS);
    SleepTimer_Start();

    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    for(;;)
    {
        /* Place your code here to be executed in Active Mode.*/

        /* Switch to the Standby Mode */
        CyWait();
       /* Clear interrupt status bit */
        SleepTimer_GetStatus();
    }
}
```

Functional Description

The Sleep Timer component is not responsible for the device’s low power modes entry. Refer to the Power Management APIs section of the System Reference Guide. The guide is available in PSoC Creator's Help menu.

Refer to the device data sheet to understand the relationship between the Sleep Timer and Watch Dog Timer.

As described previously, the Sleep Timer can be configured to the following intervals: 2ms, 4ms, 8ms, 16ms, 32ms, 64ms, 128ms, 256ms, 512ms, 1024ms, 2048ms and 4096ms. However, it is important to remember that it can have up to 20% deviation, because the sleep timer is derived from the device’s internal low speed oscillator.

For the proper switching to the low power modes, the MHz crystal oscillator and PLL have to be disabled. Before disabling those clock sources, you must switch to the lower values of IMO.

For the proper operation of Sleep Timer component, user should execute SleepTimer_GetStatus function after every wake up and/or interrupt being issued.

PRELIMINARY

References

See also the RTC component.

DC and AC Electrical Characteristics

The following values are indicative of expected performance and based on initial characterization data.

5.0V/3.3V DC and AC Electrical Characteristics

Parameter	Typical	Min	Max	Units	Conditions and Notes
Input					
Input Voltage Range	---		Vss to Vdd	V	
Input Capacitance	---		---	pF	
Input Impedance	---		---	Ω	
Maximum Clock Rate	---		67	MHz	

Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes	Reason for Changes / Impact
1.10	Removed SleepTimer_Reset() function and added SleepTimer_GetStatus() function. The interrupt output terminal is connected to an interrupt component by default when the component is placed in a design.	Various changes were made to fix issues with the previous version, which was not fully functional.