CC32xx Power Management Framework

Introduction

Return to CC31xx & CC32xx Home Page

The low power modes supported in the CC3200 subsystem are: Sleep, DeepSleep, LowPowerDeepSleep (LPDS) and Hibernate (HIB).

The diagram below captures the key considerations that would enable a programmer to choose the lower power mode suitable for a given use case.

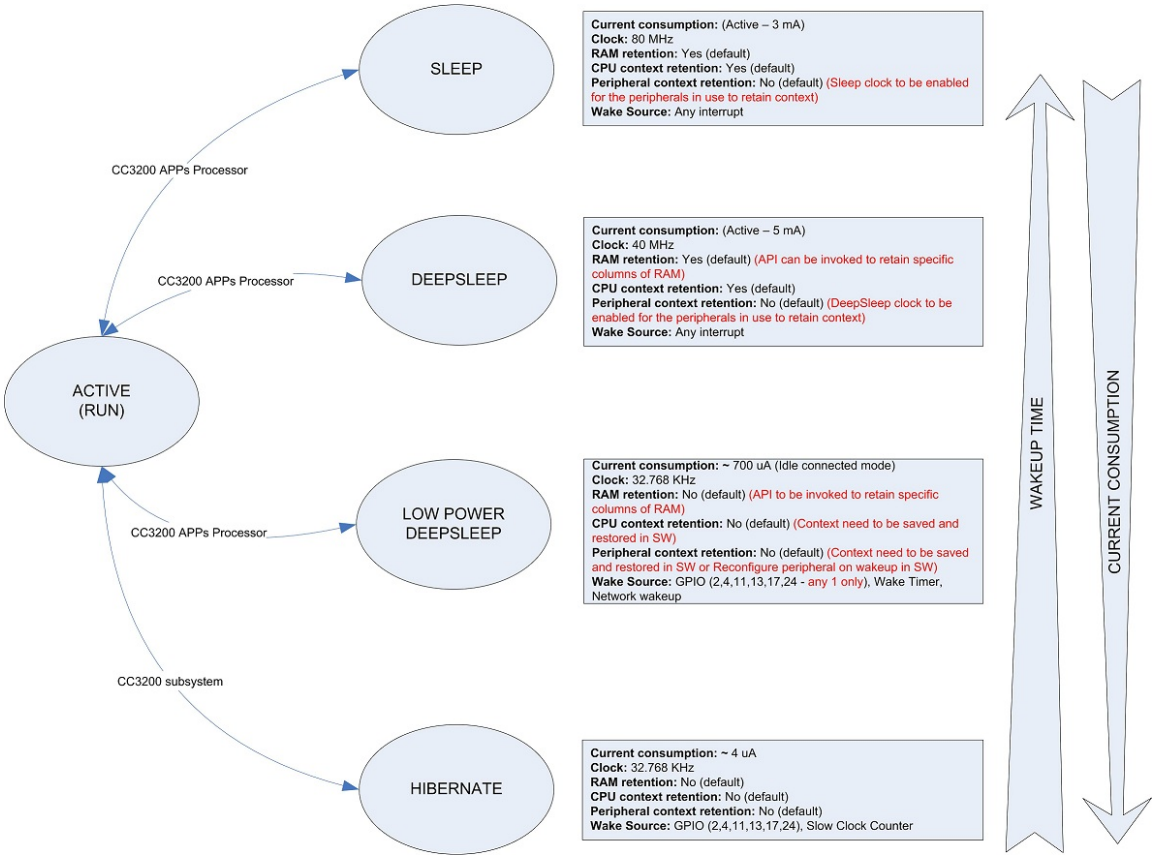
The key points to note are:

1. Sleep: By default the sleep clock to the peripherals are disabled. If the application chooses to enter sleep anytime and requires certain peripherals to be active (as a source of wakeup), the sleep clock to the peripheral has to be enabled. This is an APPs processor control.

2. DeepSleep: It is recommended to use DeepSleep only when using networking services. Using peripherals in conjunction with DeepSleep is not recommended. This is an APPs processor control.

3. LowPowerDeepSleep (LPDS): The CPU and peripheral context needs to be restored on exit from LPDS. Only 1 out of the 6 GPIOs can be used as a wake source from LPDS. This is an APPs processor control.

4. HIBernate (HIB): This is equivalent to a PowerOnReset except for the fact that the SlowClockCOunter continues to tick and a minimal set of registers (two 32 bit registers) are retained across HIB cycles. This is a complete CC3200 subsystem control.

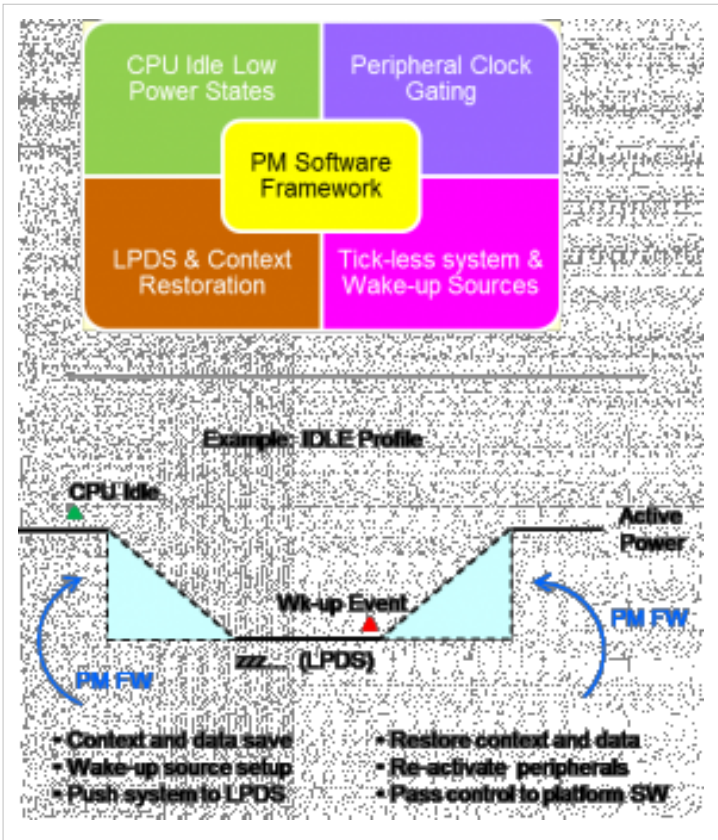


Overview

The objective of the power management framework is to provide an easy to use infrastructure for developers to create a power aware solution. The framework abstracts the inherently complex power management by providing simple services that can be invoked by application developers.

The salient features of the PM framework are:

- Ease of use through intuitive and handful API(s)
- Under the application control selects the lowest power state
- Restores the system context after wake-up
- Clear separation of concerns across apps, peripheral drivers & platform - Enables developers to focus on application and value addition



Software Interface

The programming interface provided by the power management framework is as described in the diagram below.

APIs can be divided into: Core APIs and User hook functions. Key points to note on the APIs are as follows:

1. PM framework initialization APIs - Mandatory

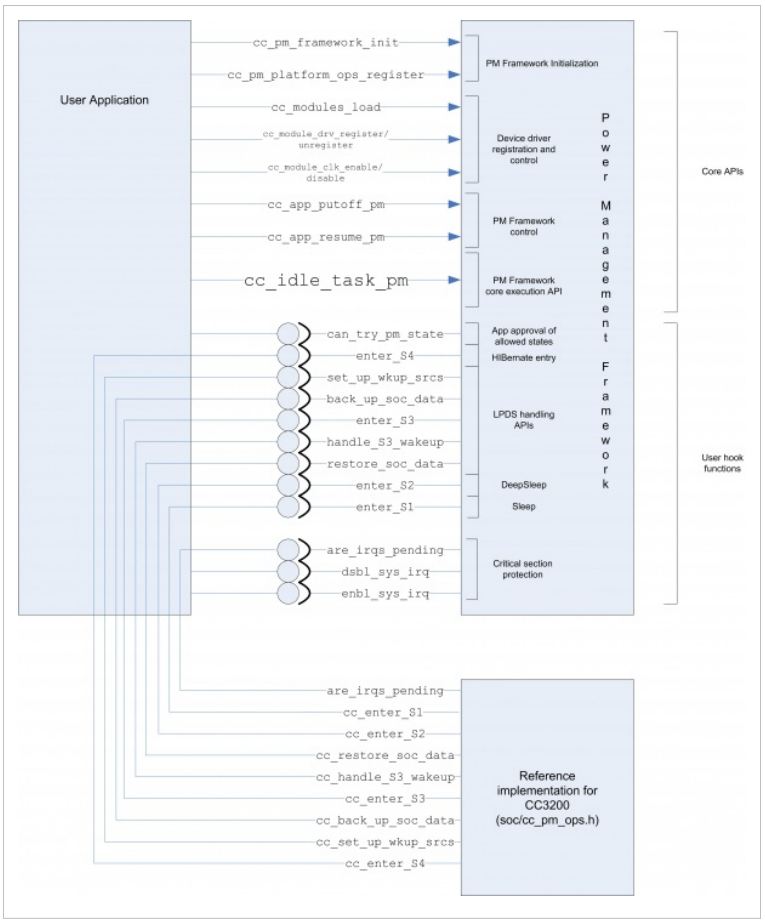
2. Device driver registration and control APIs - These APIs aid the peripheral power state transitions in sync with the system power state transitions.

3. PM framework control: User applications have a choice of controlling the PM framework kicking in (enable/disable) to better control the execution flow. The APIs are reference counted.

4. PM framework core execution API - The API has to be placed in the idle loop of the application. It attempts to enter the low power modes in the order HIB►LPDS►DeepSleep►Sleep. Before entry into HIB, LPDS or DeepSleep, the user callback function "can_try_pm_state" is invoked to get the approval to enter a particular state.

5. User hook functions - A reference implementation of the user hook functions are provided along with the package (middleware\soc\cc_pm_ops). These APIs can be overridden by the applications as deemed fit. The key user hook function to be implemented are:

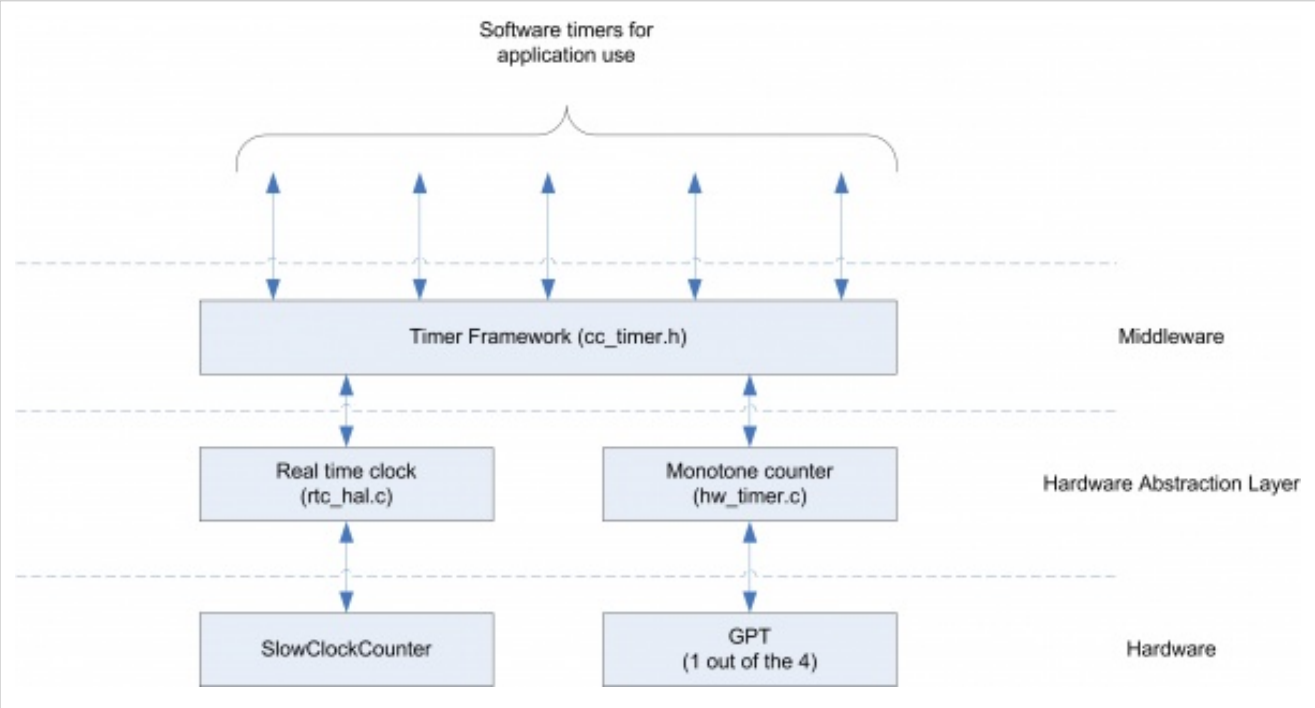
- "can_try_pm_state" - approves the PM framework to enter a particular low power mode.
- "back_up_soc_data" - saves any peripheral context before entering LPDS. Invokes the IO parking as desired (middleware\soc\cc_io_park).
- "restore_soc_data" - restores any peripheral context saved. PinMuxConfig() API has to be reinvoked - to enable clocks to the peripherals.



Timer framework

Overview

Timer framework provides a software timer infrastructure built on top of 2 of the hardware resources (Slow clock counter and GeneralPurposeTimer (GPT)) available in the CC3200 device.



Real Time Clock (RTC) Infrastructure : Low precision, Aids power management

- Clock tick at 32.768Khz.
- Uses a countinuously ticking slow clock counter and configures the match values accordingly to realize multiple software timers.
- Ticks across all low power modes (HIB, LPDS, DeepSleep, Sleep, Active).
- Can be used for usecases that need to work with absolute time. This uses up 2 of the available user HIB registers that are saved across HIB cycles. If there is information that needs to be retained across HIB cycles by the application, the NVMEM (SFlash) can be used alternately.
- Mandatory to be used in usecases that need to excercise low power modes.

Monotone Clock infrastructure : High precision

- Clock tick at 80 Mhz.
- Uses a countinuously ticking GPT and configures the match values accordingly to realize multiple software timers.
- Ticks only in Active and Sleep (by enabling the sleep clock) modes.
- Cannot be used in conjunction with other low power modes (HIB, LPDS, DeepSleep) and hence is the applications resposibility to not enter any of these low power modes when the timer is running.

Exercising Low Power Modes

The low power modes discussed in this section are HIBernate and LowPowerDeepSleep.

Exercising low power modes involves

- **Backing up information**

HIB: There are two 32 bit registers that are retained (These are not available if the RTC is enabled). The NVMEM (SFlash) can also be used.

LPDS: CPU and peripheral context need to be saved (in RAM). Also the desired amount of RAM needs to be retained.

- **Setting up the wake sources**

HIB: The wake sources are: PRCM_HIB_SLOW_CLK_CTR, PRCM_HIB_GPIO2, PRCM_HIB_GPIO4, PRCM_HIB_GPIO11, PRCM_HIB_GPIO13, PRCM_

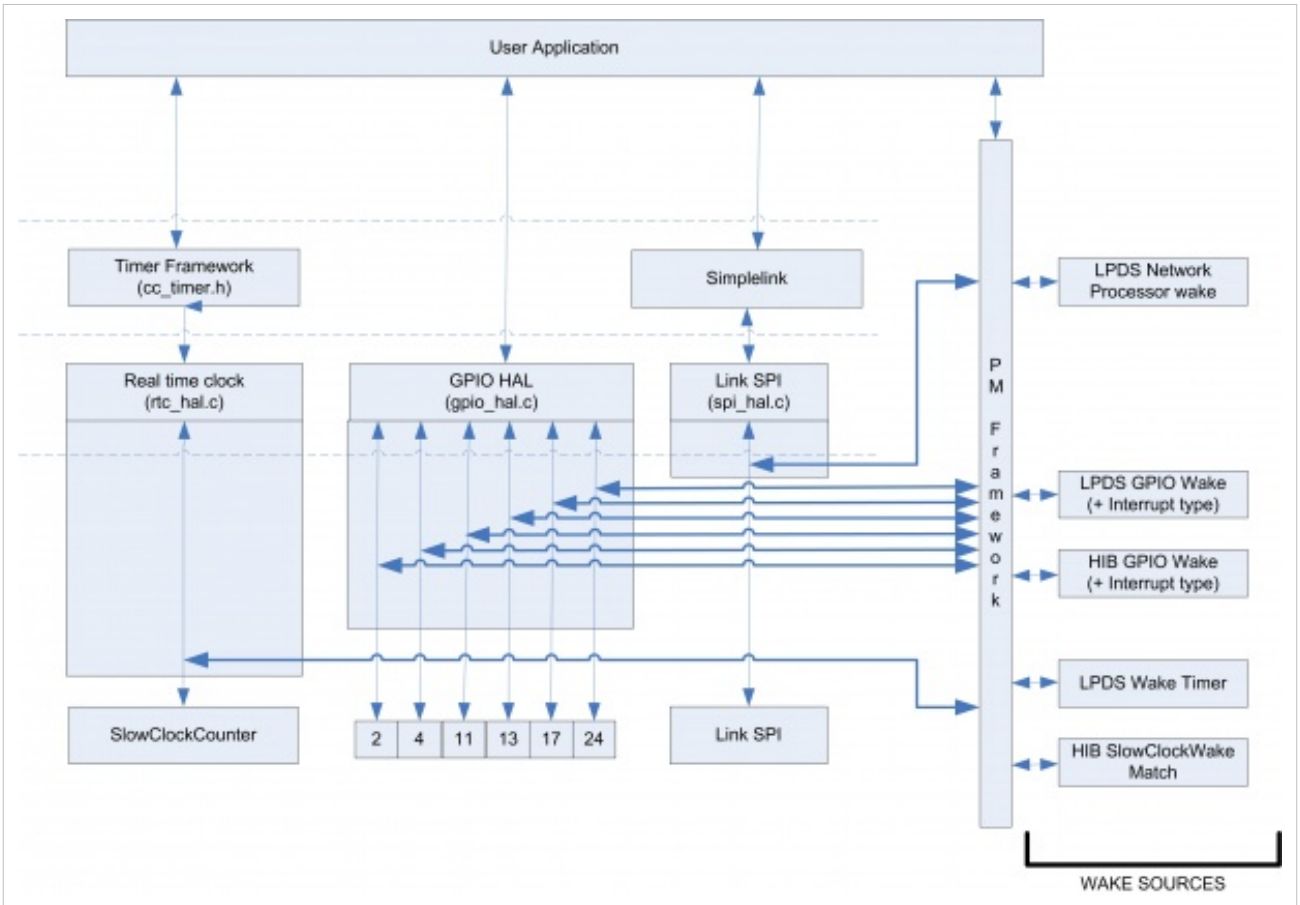
LPDS: The wake sources are: PRCM_HIB_SLOW_CLK_CTR, [PRCM_HIB_GPIO2 or PRCM_HIB_GPIO4 or PRCM_HIB_GPIO11 or PRCM_HIB_GPIO13 or PRCM_HIB_GPIO17 or PRCM_HIB_GPIO24] - any one GPIO at a time only, Network activity based wakeup

- **Entering the low power mode**
- **Restoring the information on wakeup**

HIB: The information can be read back from and restored. PRCMSysResetCauseGet API can be invoked to determine the wake cause.

LPDS: The CPU and peripheral context needs to be restored.

The PM framework abstracts out the information mentioned above and is indirectly derived by the choices the user application. The same are described in the diagram and subsection below.



Low Power Deep Sleep

GPIO as wake source

Applications have to use the gpio_hal.c APIs for the usage of GPIOs. Notifications have to be enabled for the GPIO to be used as a wake source.

```
//
// setting up GPIO as a wk up source and configuring other related
// parameters
//
tGPIOHndl = cc_gpio_open(GPIO_SRC_WKUP, GPIO_DIR_INPUT);
cc_gpio_enable_notification(tGPIOHndl, GPIO_SRC_WKUP, INT_FALLING_EDGE, (GPIO_TYPE_NORMAL | GPIO_TYPE_WAKE_SOURCE));
```

This selection by the application is used by the PM framework to configure the GPIO as a wake source anytime the system can enter LPDS. On wakeup from LPDS, the registered application callback is invoked.

RTC as wake source

Applications have to use the cc_timer.c, rtc_hal.c APIs for the usage of RTC as wake source.

The PM framework internally manages the setting up of LPDS waketimer value before entering LPDS.

```
//
// setting up Timer as a wk up source
//
/* Setup the RTC initial value - This can be obtained from NTP too */
init_time.secs = 0;
init_time.nsec = 0;
cc_rtc_set(&init_time);

/* Create a real time timer */
realtime_timer.source = HW_REALTIME_CLK;
realtime_timer.timeout_cb = timer_intr_hndlr;
realtime_timer.cb_param = NULL;

timer_hndl = cc_timer_create(&realtime_timer);

/* Configure the timer for a periodic wakeup */
interval_time.secs = LPDS_DUR_SEC;
interval_time.nsec = LPDS_DUR_NSEC;
cc_timer_start(timer_hndl, &interval_time, OPT_TIMER_PERIODIC);
```

In the above sequence, after the timer is setup, the PM framework internally manages the entry into LPDS for (interval_time - WAKEUP_TIME_LPDS). Then the timer fires and the application can perform its desired action. This loop continues periodically.

Network activity as wake source

Once the sl_start API is invoked by the application (and HostIRQ enabled), thereafter anytime the APPs processor enters LPDS, the wake on network activity is always enabled.

Hibernate

GPIO as wake source

Applications have to use the gpio_hal.c APIs for the usage of GPIOs. Notifications have to be enabled for the GPIO to be used as a wake source.

```
// setting up GPIO as a wk up source and configuring other related parameters
tGPIOHndl = cc_gpio_open(GPIO_SRC_WKUP, GPIO_DIR_INPUT);
cc_gpio_enable_notification(tGPIOHndl, GPIO_SRC_WKUP, INT_FALLING_EDGE, (GPIO_TYPE_NORMAL | GPIO_TYPE_WAKE_SOURCE));
```

This selection by the application is used by the PM framework to configure the GPIO as a wake source anytime the system can enter HIB. On wakeup from HIB, the application execution resumes from the reset vector.

RTC as wake source

Applications have to use the cc_timer.c, rtc_hal.c APIs for the usage of RTC as wake source.

The PM framework internally manages the setting up of HIB SlowClockMatch value before entering HIB.

```
//
// setting up Timer as a wk up source and other timer configurations
//
/* Setup the RTC initial value - This can be obtained from NTP too */'''
'''
sInitTime.secs = 0;
sInitTime.nsec = 0;
cc_rtc_set(&sInitTime);

/* Create a real time timer */
sRealTimeTimer.source = HW_REALTIME_CLK;
sRealTimeTimer.timeout_cb = NULL;
sRealTimeTimer.cb_param = NULL;
tTimerHndl = cc_timer_create(&sRealTimeTimer);

/* Configure the timer for a absolute wakeup */
sIntervalTimer.secs = HIB_DUR_SEC;
sIntervalTimer.nsec = HIB_DUR_NSEC;
cc_timer_start(tTimerHndl, &sIntervalTimer, OPT_TIME_ABS_VALUE);
```

In the above sequence, after the timer is setup, the PM framework internally manages the entry into HIB for (sIntervalTimer - WAKEUP_TIME_HIB). On wakeup from HIB, the application execution resumes from the reset vector.

IO Parking while entering LPDS

In order to save power (to avoid any leakage currents) when the system enters LPDS, it is recommended to park the IOs appropriately. The application is expected to specify the IO parking scheme. The various options available are:

- DONT_CARE - No parking of the pin
- NO_PULL_HIZ - No pulls, only HiZ the pin
- WEAK_PULL_UP_STD - Weak pullup as well as HiZ
- WEAK_PULL_DOWN_STD - Weak pulldown as well as HiZ
- WEAK_PULL_UP_OD - Weak pullup, Opendrain as well as HiZ
- WEAK_PULL_DOWN_OD - Weak pulldown, Opendrain as well as HiZ

An example of IO parking scheme table is as below:

```
struct soc_io_park cc32xx_io_park[] = {
{PIN_01, "GPIO_10", WEAK_PULL_DOWN_STD},
{PIN_02, "GPIO_11", WEAK_PULL_DOWN_STD},
{PIN_03, "GPIO_12", WEAK_PULL_DOWN_STD},
{PIN_04, "GPIO_13", WEAK_PULL_DOWN_STD},
{PIN_05, "GPIO_14", WEAK_PULL_DOWN_STD},
{PIN_06, "GPIO_15", WEAK_PULL_DOWN_STD},
{PIN_07, "GPIO_16", WEAK_PULL_DOWN_STD},
{PIN_08, "GPIO_17", WEAK_PULL_DOWN_STD},
{PIN_15, "GPIO_22", WEAK_PULL_DOWN_STD},
{PIN_16, "GPIO_23/JTAG_TDI", WEAK_PULL_DOWN_STD},
{PIN_17, "GPIO_24/JTAG_TDO", WEAK_PULL_DOWN_STD},
{PIN_18, "GPIO_28", WEAK_PULL_DOWN_STD},
{PIN_19, "GPIO_28//JTAG_TCK", WEAK_PULL_DOWN_STD},
{PIN_20, "GPIO_29/JTAG_TMS", WEAK_PULL_DOWN_STD},
{PIN_21, "GPIO_25/SOP2", WEAK_PULL_DOWN_STD},
{PIN_45, "DCDC_FLASH_SW_P", WEAK_PULL_DOWN_STD},
{PIN_50, "GPIO_00", WEAK_PULL_DOWN_STD},
{PIN_53, "GPIO_30", WEAK_PULL_DOWN_STD},
{PIN_55, "GPIO_01", WEAK_PULL_DOWN_STD},
{PIN_57, "GPIO_02", WEAK_PULL_DOWN_STD},
{PIN_58, "GPIO_03", WEAK_PULL_DOWN_STD},
{PIN_59, "GPIO_04", WEAK_PULL_DOWN_STD},
{PIN_60, "GPIO_05", WEAK_PULL_DOWN_STD},
{PIN_61, "GPIO_06", WEAK_PULL_DOWN_STD},
{PIN_62, "GPIO_07", WEAK_PULL_DOWN_STD},
{PIN_63, "GPIO_08", WEAK_PULL_DOWN_STD},
{PIN_64, "GPIO_09", WEAK_PULL_DOWN_STD}
};
```

Recommendations for use of PM framework

Choice of low power mode

Some of the key considerations for the choice of the low power mode can be the following (These are generic suggestions only):

- No activity duration

No activity duration - Order of minutes - Choose HIB

No activity duration - Order of seconds - Choose LPDS

No activity duration - Order of milliseconds - Choose Deepsleep/Sleep

- Acceptable lead time on wakeup

HIB - The application is reloaded from the SFLASH. Connection to the AP needs to be re-established. All desired actions must be performed again.

LPDS - On using the PM framework, the CPU context is restored. Reinitialization of peripherals is required. The connection to the AP is retained.

Sleep - Fastest response time with maximum power penalty.

- Retention of context

HIB - All context lost. All desired actions must be performed again. NVMEM or the two 32 bit registers can be used to store information across HIB cycles.

LPDS - Using the power management framework the CPU and registered peripheral context can be retained. All peripheral context need to be restored on exit from LPDS. Desired

amount of RAM can be retained (in multiples of 64 KB).

Choice of wake source

The choice of wake source is dependent on the application usecase. Summarizing the wake sources below:

1. Sleep - Any interrupt

2. Deepsleep - Any interrupt

3. LowPowerDeepSleep - Waketimer (specific counter that ticks only during LPDS), Any 1 GPIO (2,4,11,13,17,24), Network activity based wakeup - HostIRQ (This is in response to an API issued by the application before entering LPDS (for ex., sl_recv()))

4. HIBernate - SlowClockCounter (match value is set, the slow clock counter once started in PRCMCC3200MCUInit, continues to tick), Any combination of GPIOs (2,4,11,13,17,24)

Note: In case of HIBernate, the source of the wakeup cannot be determined.

Working with time of day (real time)

The time of day information can be obtained using NTP. This information can be passed to the timer framework and then can be used as the basis of having timed events based on time of day. Many usecases can be realized using this feature (alarms...).

Working with time of day requires setting up of the initial RTC time. Thereafter, the framework maintains the synchronization between the Real time and the free running SlowClockCounter.

The code could be like below:

```
//
// setting up Timer as a wk up source
```

```
//
/* Setup the RTC initial value obtained from NTP */
init_time.secs = NTP_TIME_SECS;
init_time.nsec = NTP_TIME_NSECS;
cc_rtc_set(&init_time);

/* Create a real time timer */
realtime_timer.source = HW_REALTIME_CLK;
realtime_timer.timeout_cb = timer_intr_hndlr;
realtime_timer.cb_param = NULL;
timer_hndl = cc_timer_create(&realtime_timer);

/* Configure the timer for a periodic wakeup */
interval_time.secs = WAKEUP_SEC_TIMEOFDAY;
interval_time.nsec = WAKEUP_NSEC_TIMEOFDAY;
cc_timer_start(timer_hndl, &interval_time, OPT_TIMER_PERIODIC);
```

There precision of the timer is only in the order of milliseconds. The input of nanoseconds is retained to be able to pass the values obtained over NTP to the API directly.

Note:

- The two 32bit HIB registers are used to maintain the mapping of initial timeofday with the slow clock counter ticks and hence are not available for application use.
- The APIs donot take care of the timezone correction (to be applied to the GMT time obtained over NTP). The timezone correction is expected to be handled by the application before initializing the RTC.

Disabling PM framework

The APIs to control the enabling and disabling of PM framework are: cc_app_resume_pm and cc_app_putoff_pm. These APIs are reference counted and the resumption of PM framework happens accordingly.

The intervention from the PM framework can be controlled by using these APIs. An example is as below:

```
SELECT_LPDS_LOW_POWER_MODE();


while(FOREVER){
      /* Resume the PM framwork operations before entering the Idle wait in the application */
        cc_app_resume_pm();
        /* Enter the Wait for registered event (for ex., sl_recv) */
        WAIT_HERE();  /* --------- Application will enter LPDS here on executing IDLE loop ----*/
        /* Put off the PM framework the process the event */
        cc_app_putoff_pm();

        /* --------- Application will remain ACTIVE here ----*/
        ..... /* Processing */
}
```

Extending/Customizing PM framework

A reference implementation of the user hook functions have been provided in the file cc3200-sdk\middleware\soc\cc_pm_ops.c. While this may suffice for most of the implementations, the applications can choose to override this implementation by hooking an alternate function.

The default user hook registration function will be as below:

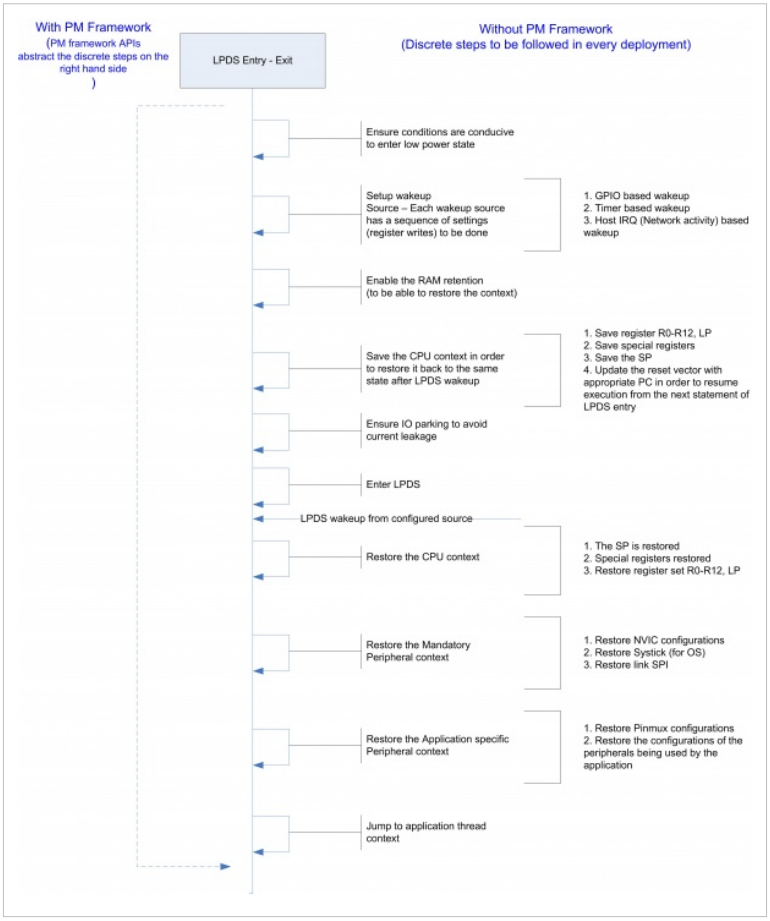
```
lp3p0_pm_ops->set_up_wkup_srcs = cc_set_up_wkup_srcs;
lp3p0_pm_ops->handle_S3_wakeup = cc_handle_S3_wakeup;
lp3p0_pm_ops->are_irqs_pending = cc_are_irqs_pending;
lp3p0_pm_ops->can_try_pm_state = lp3p0_can_try_pm_state;
lp3p0_pm_ops->enter_S4 = cc_enter_S4;
lp3p0_pm_ops->enter_S3 = cc_enter_S3;
lp3p0_pm_ops->enter_S2 = cc_enter_S2;
lp3p0_pm_ops->enter_S1 = cc_enter_S1;
lp3p0_pm_ops->back_up_soc_data = lp3p0_back_up_soc_data;
lp3p0_pm_ops->restore_soc_data = lp3p0_restore_soc_data;
lp3p0_pm_ops->dsbl_sys_irq = osi_EnterCritical;
lp3p0_pm_ops->enbl_sys_irq = osi_ExitCritical;
```

The application is expected to implement the following functions consciously:

- can_try_pm_state - approval of the entry into low power state
- back_up_soc_data - save any information (peripheral) if required before entering LPDS
- restore_soc_data - restore peripheral context (or reinitialize peripherals) on exiting LPDS

PM framework details

LPDS entry-exit sequence



Folder structure and API listing

Folder structure

The folder structure of the middleware component is as below:

+middleware

+ccs --- CCS project files

+ewarm --- IAR project files

+driver --- driver implementation (synchronouns drivers are spi_drv and uart_drv)

+hal --- Hardware abstraction layer for the peripherals

+framework

+pm --- Power management framework

+timer --- Timer framework

+soc --- Reference implementation of the PM framework user hook functions

Core PM framework APIs

API	Description
cc_pm_framework_init	Initialize the power management framework
cc_pm_platform_ops_register	Register the user hook functions with the PM framework
cc_modules_load	Load the registered modules
cc_module_drv_register	Register a new module driver with the PM framework
cc_module_drv_unregister	UnRegister a module driver from the PM framework
cc_module_clk_enable	TBD
cc_module_clk_disable	TBD
cc_app_putoff_pm	Put off the PM framework
cc_app_resume_pm	Resume the PM fra

Platform Service APIs

API	Description
cc_set_up_wkup_srcs	Sets up wake-up sources for indicated power mode
cc_handle_S3_wakeup	Process events that have woken up system from S3 (LPDS) by triggering a software interrupt
cc_are_irqs_pending	Check if any interrupts are pending in the system
cc_enter_S4	Enter HIBernate
cc_enter_S3	Enter LPDS
cc_enter_S2	Enter DeepSleep
cc_enter_S1	Enter Sleep
cc_back_up_soc_data	Enables the SRAM retention, Saves the NVIC registers
cc_restore_soc_data	Restores the NVIC registers, Acquires the I2C and GPIO semaphore
wake_interrupt_handler	Handles the Power and Reset Control Module (PRCM) wake events

Reference Implementation

Reference implementations using the power management framework is available in the SDK package. The example apps are the following:

1. Idle profile:

The CC3200 device is connected to an AP and is continuously waiting for a packet over the air (UDP Rx) to wakeup. For all the times when the application is idling, it enters LPDS. In order to showcase all wake sources, GPIO13 and timer based wakeup from LPDS is also supported. Please refer to the link for more details: CC32xx Idle Profile Application

2. Sensor profile:

The CC3200 device periodically wakes up from HIBernate, broadcasts a message and then enters HIBernate. The application also allows for wakeup from HIBernate based on GPIO13. Please refer to the link for more details: CC32xx Sensor Profile Application

Limitations and known issues

1. This is a feature complete version of the PM framework. System tests have been performed successfully. The production version of the framework would be released after system test cycles on ES 1.33 device.
2. ARM CM-4 Clock gating (WFI/WFE instructions) cannot be invoked if Low power deep sleep is invoked in the user application. Above restriction would be relaxed with production devices.
3. Usage of Deepsleep while using peripherals other than GPIO is not recommended due to the variation in clock frequency during the transition to the power state.
4. The drivers have not been tested for all possible input configurations.
5. Power Management framework would not work with ES 1.21 devices.

Article Sources and Contributors

CC32xx Power Management Framework *Source*: http://processors.wiki.ti.com/index.php?oldid=178654 *Contributors*: A0221015, Codycooke, Jitgupta

Image Sources, Licenses and Contributors

File:Cc31xx cc32xx return home.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Cc31xx_cc32xx_return_home.png *License*: unknown *Contributors*: A0221015

Image:CC3200 low power modes.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:CC3200_low_power_modes.jpg *License*: unknown *Contributors*: Codycooke, Jitgupta

Image:CC3200 PMFramework overview.png *Source*: http://processors.wiki.ti.com/index.php?title=File:CC3200_PMFramework_overview.png *License*: unknown *Contributors*: Codycooke

Image:CC3200 PMF SoftwareInterface.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:CC3200_PMF_SoftwareInterface.jpg *License*: unknown *Contributors*: Codycooke

Image:CC3200 timer fmwk.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:CC3200_timer_fmwk.jpg *License*: unknown *Contributors*: Codycooke

Image:CC3200 low power wakesources.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:CC3200_low_power_wakesources.jpg *License*: unknown *Contributors*: Codycooke

Image:CC3200 LPDS entryexit.jpg *Source*: http://processors.wiki.ti.com/index.php?title=File:CC3200_LPDS_entryexit.jpg *License*: unknown *Contributors*: Codycooke

License

License

1. Definitions

a. "**Adaptation**" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.
b. "**Collection**" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.
c. "**Creative Commons Compatible License**" means a license that is listed at http://creativecommons.org/compatiblelicenses that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.
d. "**Distribute**" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.
e. "**License Elements**" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.
f. "**Licensor**" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.
g. "**Original Author**" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.
h. "**Work**" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.
i. "**You**" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.
j. "**Publicly Perform**" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.
k. "**Reproduce**" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;
b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";
c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,
d. to Distribute and Publicly Perform Adaptations.
e. For the avoidance of doubt:

i. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;
ii. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,
iii. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.
b. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US)); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.
c. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Ssection 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.
d. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.