

# Power Service Overview

© 2014 Sony Computer Entertainment Inc.  
All Rights Reserved.  
SCE Confidential

# Table of Contents

<b>1 Overview .....</b>	<b>3</b>
Purpose and Features.....	3
Files .....	3
<b>2 Obtaining Battery Information.....</b>	<b>4</b>
Obtaining Low Battery Status .....	4
<b>3 Power Button Control.....</b>	<b>5</b>
State Transitions Caused by the Power Button.....	5
<b>4 Automatic Power Control.....</b>	<b>6</b>
Automatic Suspension .....	6
Automatic Standby .....	6
<b>5 Suppression of Power Saving Features.....</b>	<b>7</b>
Suppression of Screen (Touchscreen) Power Saving and Automatic Suspension.....	7
Suppression of Screen (Touchscreen) Power Saving.....	7
Suppression of Automatic Sleep Mode .....	7
<b>6 Power Configuration Control .....</b>	<b>8</b>
Frequency of the GPU Clock of the PlayStation®Vita System Chip.....	8
<b>7 Enabling Power Saving.....</b>	<b>9</b>
CPU Busy Loop.....	9

# 1 Overview

## Purpose and Features

The power service is a kernel mode service that provides the following power management features for the PlayStation®Vita.

- Obtaining the remaining percentage of battery life
- Transition to suspend and resume from suspend
- Power control
- Power button control
- Switching the dynamic operating clock frequency of the PlayStation®Vita system chip

## Files

The following files are required in order to use the power service.

Filename	Description
libScePower_stub.a	Stub library file
power.h	Header file

## 2 Obtaining Battery Information

Use the following functions to obtain the battery information:

Function	Description
<code>scePowerIsPowerOnline()</code>	Gets external power supply connection status
<code>scePowerIsLowBattery()</code>	Gets low battery status

### Obtaining Low Battery Status

The `scePowerIsLowBattery()` function can be used to check for a low battery status.

Low battery status means that the remaining battery life is short. The percentage until a low battery status occurs varies according to the battery capacity.

### 3 Power Button Control

The power service performs transitions between power on, standby, suspend, and resume with the power button of the PlayStation®Vita. Suspend or standby processing due to the power button is automatically handled by the power service.

#### State Transitions Caused by the Power Button

The power button can be used to transition the PlayStation®Vita state between power on, standby, and suspend.

State transitions occur due to the following operations.

##### Power On

If the user holds the power button or PS button for at least two seconds when the power is off, power on (cold boot) processing is performed.

##### Standby

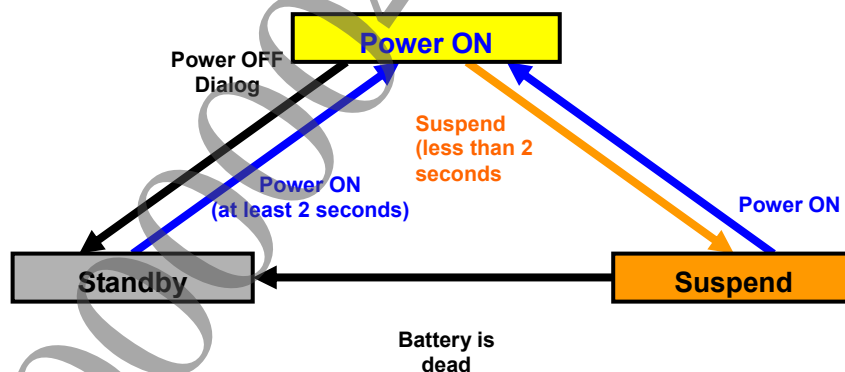
If the user holds the power button for at least two seconds when the power is on, Power Off Confirmation Dialog is displayed. Select **Power OFF** to perform standby processing.

##### Suspend

If the user holds the power button for less than two seconds while the PlayStation®Vita is in power-on state, suspend processing is performed.

##### Resume

If the user pushes the power button while the PlayStation®Vita is in suspend state, resume processing is performed.



---

## 4 Automatic Power Control

---

The power service also performs power control for reasons other than operation of the power button.

### Automatic Suspension

The power service automatically suspends the PlayStation®Vita for the reasons listed below.

- The user has left the PlayStation®Vita unattended for the **Enter Standby Mode Automatically** interval, a power saving setting that the user can set in advance from the Settings application on the home screen, while the PlayStation®Vita was running on battery.
- The remaining battery life of the PlayStation®Vita has become too short
- The PlayStation®Vita has heated up to close to the allowed temperature limit.

### Automatic Standby

The power service automatically performs standby processing for the following reason.

- The battery ran out while the PlayStation®Vita was in suspend state

## 5 Suppression of Power Saving Features

The PlayStation®Vita system software has a power saving setting for automatically dimming the screen (touchscreen) or automatically causing a transition to sleep (suspend) state if the controller, touch panel etc. are not operated for a certain period of time.

The power service can suppress the transition to power save mode in the following cases by canceling the timer until power save mode begins in the power save configuration.

- A movie is playing and you do not want to turn off the screen (touchscreen) even though there has been no key input for a long time
- You want to turn on the screen (touchscreen), which had been previously turned off, when there is a connection from another PlayStation®Vita via the WLAN

The `sceKernelPowerTick()` function is used to suppress power save mode. However, this function only cancels a timer. To maintain a condition in which power save mode is suppressed continuously, call the `sceKernelPowerTick()` function periodically during the desired period (for example, at 1V intervals).

Note that the controller service and the touch service will automatically call the `sceKernelPowerTick()` function when the button input state changes and touch panel operation is performed. This processing is also performed automatically when headphones are attached or removed. Whether or not the `sceKernelPowerTick()` function is to be explicitly called from the application for any other cause is a decision that is left to the individual application.

### Suppression of Screen (Touchscreen) Power Saving and Automatic Suspension

`SCE_KERNEL_POWER_TICK_DEFAULT` is used for normal power save suppression.

```
// Prevent power save mode
sceKernelPowerTick(SCE_KERNEL_POWER_TICK_DEFAULT);
```

### Suppression of Screen (Touchscreen) Power Saving

`SCE_KERNEL_POWER_TICK_DISABLE_OLED_DIMMING` is used to suppress only power saving for the screen (touchscreen). Only the timer for turning off the screen (touchscreen) during a continuous idle state is canceled, and the timer for performing automatic suspension is not canceled.

```
// Prevent power saving for the screen (touchscreen)
sceKernelPowerTick(SCE_KERNEL_POWER_TICK_DISABLE_OLED_DIMMING);
```

### Suppression of Automatic Sleep Mode

`SCE_KERNEL_POWER_TICK_DISABLE_AUTO_SUSPEND` is used to suppress suspension only, caused by automatic sleep mode. Only the timer for performing automatic suspension during a continuous idle state is canceled, and the timer for turning off the screen (touchscreen) is not canceled.

```
// Prevent power saving for automatic suspension
sceKernelPowerTick(SCE_KERNEL_POWER_TICK_DISABLE_AUTO_SUSPEND);
```

## 6 Power Configuration Control

### Frequency of the GPU Clock of the PlayStation®Vita System Chip

We have not released any specific frequency for the GPU clock of the PlayStation®Vita system chip. However, frequency can be changed from the initial state of the process. In order to avoid releasing specific GPU frequencies, the API was made so as to allow choosing among three power configurations.

- Mode A  
This is the normal mode at process start-up. The clock frequency of the GPU core is the "normal" clock frequency. The WLAN/COM can be used.
- Mode B  
This mode accelerates the GPU clock frequency. The clock frequency of the GPU core is the "high" clock frequency. The WLAN/COM cannot be used.
- Mode C  
This mode accelerates the GPU clock frequency, and also uses the WLAN/COM. The clock frequency of the GPU core is the "high" clock frequency, and use of the WLAN/COM is possible. The screen (touchscreen) brightness, however, is limited. Also, camera cannot be used.

**Note**

Display brightness control is not applied with PlayStation®TV.



## 7 Enabling Power Saving

### CPU Busy Loop

If there is no thread in READY state in any of the cores, the PlayStation®Vita kernel transitions to an idle thread. The idle thread then executes an instruction (WFI, WFE) for stopping the CPU clock.

If coding does not allow threads to transition to WAIT state and uses busy loops for timing, the kernel will not transition to the idle thread. Therefore, the CPU core clock will not be stopped, resulting in increased power consumption. Also, inadequate scheduling of device driver threads running with a lower priority than the application may result in delays in suspend/resume processing, or deadlocks.

When scheduling threads, make sure to call a waiting function to change the thread to WAIT state during periods in which there is no work to be processed, without busy-loop polling, etc.