Document serial number: 000004892117

librtc Reference

000004892117

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

© 2013 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Table of Contents

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Datatypes

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

SceRtcTick

Time information in ticks

Definition

```
#include <rtc.h>
typedef struct SceRtcTick{
        SceUInt64_t tick;
} SceRtcTick;
```

Members

tick	Cumulative number of Ticks from 0001/01/01 00:00:00

Description

This is a structure for handling time information in a uniform format. It is used for performing operations such as addition or subtraction on time values. This structure can be converted to and from `SceDateTime` by using `sceRtcGetTick()` and `sceRtcSetTick()`.

See Also

`SceDateTime`, `sceRtcGetTick()`, `sceRtcSetTick()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

SceDateTime

Time information

Definition

```
#include <scetypes.h>
typedef struct SceDateTime{
        unsigned short year;
        unsigned short month;
        unsigned short day;
        unsigned short hour;
        unsigned short minute;
        unsigned short second;
        unsigned int microsecond;
} SceDateTime;
```

Members

year	Year (1 to 9999)
month	Month (1 to 12)
day	Day (1 to 31)
hour	Hour (0 to 23)
minute	Minutes (0 to 59)
second	Seconds (0 to 59)
microsecond	Microseconds (0 to 999999)

Description

This structure is used for handling time information in a consistent manner. It is used by various libraries for converting time information.

See Also

`SceRtcTick`, `sceRtcGetMicrosecond()`, `sceRtcGetSecond()`, `sceRtcGetMinute()`, `sceRtcGetHour()`, `sceRtcGetDay()`, `sceRtcGetMonth()`, `sceRtcGetYear()`, `sceRtcSetMicrosecond()`, `sceRtcSetSecond()`, `sceRtcSetMinute()`, `sceRtcSetHour()`, `sceRtcSetDay()`, `sceRtcSetMonth()`, `sceRtcSetYear()`, `sceRtcCheckValid()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Current Time Acquisition Functions

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetCurrentTick

Get current time (UTC) in Tick representation

Definition

```
#include <rtc.h>
int sceRtcGetCurrentTick(
        SceRtcTick *pTick
);
#define sceRtcGetCurrentTickUtc(_tick)
        sceRtcGetCurrentTick(_tick)
```

Calling Conditions

Multithread safe

Arguments

pTick	Pointer to `SceRtcTick` for receiving the current time (UTC)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function gets the current UTC time. Since the time that is obtained is the Tick representation of `SceRtcTick`, it can be converted to `SceDateTime` format by using `sceRtcSetTick()`. To reflect a time zone offset, use `sceRtcTickAddMinutes()` to add the offset in terms of minutes to the Tick representation interval.

Notes

The `sceRtcGetCurrentTick()` function should only be used to display the date and time. It should not be used for thread scheduling.

During a suspend/resume, using the `sceRtcGetCurrentTick()` function will result in a discontinuity in the time that is obtained. Since the real-time clock is resynchronized during suspend/resume, time may appear to move backward. To obtain a continuous system clock for purposes such as thread scheduling, the `sceKernelGetProcessTime()`, `sceKernelGetProcessTimeLow()`, or `sceKernelGetProcessTimeWide()` function should be used instead.

See Also

`sceRtcSetTick()`, `sceRtcTickAddMinutes()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetCurrentClock

Get current time in specified time zone

Definition

```
#include <rtc.h>
int sceRtcGetCurrentClock(
        SceDateTime *pTime,
        int iTimeZone
);
#define sceRtcGetCurrentClockUtc(_p)
        sceRtcGetCurrentClock(_p,0)
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to SceDateTime for receiving the current time
iTimeZone	Time zone offset (in minutes)

Return Values

If an error occurs, a negative value is returned.

Value	Result
SCE_OK	Success
< 0	Error

Description

This function gets the current time in SceDateTime representation based on the specified time zone offset. Since this function is implemented internally by sceRtcGetCurrentTick() and sceRtcSetTick(), it is equivalent to the following code.

```
// Get current UTC in SceDateTime format
SceDateTime *sceRtcGetCurrentClock(SceDateTime *pTime, int iZoneTime)
{
      SceRtcTick tick;
      sceRtcGetCurrentTick(&tick);
      sceRtcTickAddMinutes(&tick, &tick, (SceInt64_t)iTimeZone);
      sceRtcSetTick(pTime, &tick);
      return (pTime);
}
```

If 0 is specified as the *iTimeZone* value, the UTC time is obtained.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Notes

The `sceRtcGetCurrentClock()` function should only be used to display the date and time. It should not be used for thread scheduling.

During a suspend/resume, using the `sceRtcGetCurrentTick()` function will result in a discontinuity in the time that is obtained. Since the real-time clock is resynchronized during suspend/resume, time may appear to move backward. To obtain a continuous system clock for purposes such as thread scheduling, the `sceKernelGetProcessTime()`, `sceKernelGetProcessTimeLow()`, or `sceKernelGetProcessTimeWide()` function should be used instead.

See Also

`sceRtcGetCurrentTick()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetCurrentClockLocalTime

Get current time (local time)

Definition

```
#include <rtc.h>
int sceRtcGetCurrentClockLocalTime(
        SceDateTime *pTime
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` for receiving the current time (local time)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function gets the local time, according to the current time zone setting.

The **Time Zone** setting and the **Daylight Saving** setting of the **Date & Time** settings in the system settings are reflected in the local time.

Notes

The `sceRtcGetCurrentClockLocalTime()` function should only be used to display the date and time. It should not be used for thread scheduling.

During a suspend/resume, using the `sceRtcGetCurrentTick()` function will result in a discontinuity in the time that is obtained. Since the real-time clock is resynchronized during suspend/resume, time may appear to move backward. To obtain a continuous system clock for purposes such as thread scheduling, the `sceKernelGetProcessTime()`, `sceKernelGetProcessTimeLow()`, or `sceKernelGetProcessTimeWide()` function should be used instead.

See Also

`sceRtcGetCurrentClock()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetCurrentNetworkTick

Get network time (UTC)

Definition

```
#include <rtc.h>
int sceRtcGetCurrentNetworkTick(
        SceRtcTick *pTick
);
#define sceRtcGetCurrentNetworkTickUTC(_tick) \
        sceRtcGetCurrentNetworkTick(_tick)
```

Calling Conditions

Multithread safe

Arguments

pTick	Pointer to `SceRtcTick` for receiving the network time (UTC)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function gets the network time (UTC).

Since the time that is obtained is the Tick representation of `SceRtcTick`, it can be converted to the `SceDateTime` format by using `sceRtcSetTick()`.

Notes

The `sceRtcGetCurrentNetworkTick()` function should only be used to display the date and time. It should not be used for thread scheduling.

During a suspend/resume, using the `sceRtcGetCurrentTick()` function will result in a discontinuity in the time that is obtained. Since the real-time clock is resynchronized during suspend/resume, time may appear to move backward. To obtain a continuous system clock for purposes such as thread scheduling, the `sceKernelGetProcessTime()`, `sceKernelGetProcessTimeLow()`, or `sceKernelGetProcessTimeWide()` function should be used instead.

See Also

`sceRtcGetCurrentClock()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Accumulative Time Retrieval Functions

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetAccumulativeTime

Get accumulative time in Ticks

Definition

```
#include <rtc.h>
SceULong64 sceRtcGetAccumulativeTime(
        void
);
```

Calling Conditions

Multithread safe.

Arguments

None

Return Values

Returns the accumulative time, in Ticks, from the last time that a time reset occurred due to battery shut-off.

Description

This function returns the accumulative time, in Ticks, from the last time that a time reset occurred due to battery shut-off.

The accumulative time which is obtained by the `sceRtcGetAccumulativeTime()` function is independent of the time which has been set by the user via the PlayStation®Vita system software, and always represents the accumulative absolute time which has passed since last time that a time reset occurred due to battery shut-off.

When drafting specifications which utilize this functionality, be careful that it does not become detrimental to the user.

Notes

The `sceRtcGetAccumulativeTime()` function should only be used to indicate the date and time. It should not be used for thread scheduling, etc.

To obtain a continuous system clock for purposes such as thread scheduling, etc., the `sceKernelGetProcessTime()`, `sceKernelGetProcessTimeLow()`, or `sceKernelGetProcessTimeWide()` function should be used instead.

See Also

`sceRtcGetLastReincarnatedTick()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetLastReincarnatedTick

Get time, in Ticks, recovered after the last battery shut-off

Definition

```
#include <rtc.h>
int sceRtcGetLastReincarnatedTick(
        SceRtcTick *pTick
);
#define sceRtcGetLastReincarnatedTickUtc(_tick)
        sceRtcGetLastReincarnatedTick(_tick)
```

Calling Conditions

Multithread safe.

Arguments

pTick	Pointer to the `SceRtcTick` to receive the time (UTC) recovered after the last battery shut-off

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
<0	Error

Description

Returns the time, in Ticks, from the last time that a time reset occurred due to battery shut-off.

Since the time obtained is the Tick representation of `SceRtcTick`, it can be converted to the `SceDateTime` format by using `sceRtcSetTick()`.

The time obtained by the `sceRtcGetLastReincarnatedTick()` function is based on the time which was set by the user via the PlayStation®Vita system software. In other words, the sum of the time obtained by the `sceRtcGetLastReincarnatedTick()` function and the accumulative time obtained by the `sceRtcGetAccumulativeTime()` function matches `sceRtcGetCurrentTick()`.

Note that regardless of whether the battery shut-off event represented a historical time in the past, the result of the `sceRtcGetLastReincarnatedTick()` function will change in relative terms if the user has manipulated the clock. By storing the value of the `sceRtcGetLastReincarnatedTick()` function and determining whether or not it has changed, it is possible to detect whether or not the battery shut off, the time was manipulated by the user, the application moved to another PlayStation®Vita, etc. However, because the PlayStation®Vita system software has a feature to automatically align the clock upon connecting to a network, the discrepancy between the time counted within the PlayStation®Vita system at this time and the time obtained from a network server may accumulate.

Make sure to allow several minutes as a margin of error when determining whether the return value of the `sceRtcGetLastReincarnatedTick()` function changed.

When drafting specifications which utilize this functionality, be careful that it does not become detrimental to the user.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

See Also

`sceRtcGetAccumulativeTime()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetLastAdjustedTick

Get time, in Ticks, to which the clock was last set by the user

Definition

```
#include <rtc.h>
int sceRtcGetLastAdjustedTick(
        SceRtcTick *pTick
);
#define sceRtcGetLastAdjustedTickUtc(_tick)
        sceRtcGetLastAdjustedTick(_tick)
```

Calling Conditions

Multithread safe.

Arguments

pTick	Pointer to the `SceRtcTick` to receive the time (UTC) to which the user last set the clock

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
<0	Error

Description

Returns the time, in Ticks, to which the clock was last set by the user.

Since the time obtained is the Tick representation of `SceRtcTick`, it can be converted to the `SceDateTime` format by using `sceRtcSetTick()`.

The time obtained by the `sceRtcGetLastAdjustedTick()` function is the time at which the user last set the clock via the PlayStation®Vita system software, and is returned in Tick representation. Note that even if the user has set the clock, there will be no change in the result of the `sceRtcGetLastAdjustedTick()` function if the clock was merely set again to the same time. In these cases as well, the value of the accumulative time obtained by the `sceRtcGetAccumulativeTime()` function will still increase uniformly, so it is possible to detect that the clock has been turned back by the user.

When drafting specifications which utilize this functionality, be careful that it does not become detrimental to the user.

See Also

`sceRtcGetAccumulativeTime()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Formatting Functions

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcFormatRFC2822

Format Tick-representation UTC time in RFC2822 format

Definition

```
#include <rtc.h>
int sceRtcFormatRFC2822(
        char *pszDateTime,
        const SceRtcTick *pUtc,
        int iTimeZoneMinutes
);
```

Calling Conditions

Multithread safe

Arguments

pszDateTime	Buffer for receiving formatted string
pUtc	Tick representation of current time (UTC)
iTimeZoneMinutes	Time zone offset (minutes)

Return Values

If an error occurs, a negative value is returned.

Value	Result
SCE_OK	Success
< 0	Error

Description

This function formats the specified time (UTC) according to RFC2822. The time is formatted after it is converted to a local time in which the **Time Zone** setting and **Daylight Saving** setting of the **Date & Time** settings in the system settings are reflected.

For example, December 3, 1995 13:23 in London in RFC2822 format would be

```
Sun, 03 Dec 1995 13:23:00 +0000
```

When NULL is specified to *pUtc*, the current time will be formatted.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcFormatRFC2822LocalTime

Format Tick-representation UTC time in RFC2822 format

Definition

```
#include <rtc.h>
int sceRtcFormatRFC2822LocalTime(
        char *pszDateTime,
        const SceRtcTick *pUtc
);
```

Calling Conditions

Multithread safe

Arguments

pszDateTime	Buffer for receiving formatted string
pUtc	Tick representation of current time (UTC)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function formats the specified time (UTC) according to RFC2822. The time is formatted after it is converted to a local time in which the **Time Zone** setting and **Daylight Saving** setting of the **Date & Time** settings in the system settings are reflected.

For example, December 3, 1995 13:23 in London in RFC2822 format would be

```
Sun, 03 Dec 1995 13:23:00 +0000
```

When NULL is specified to *pUtc*, the current time will be formatted.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcFormatRFC3339

Format Tick-representation UTC time in RFC3339 (ISO8601) format

Definition

```
#include <rtc.h>
int sceRtcFormatRFC3339(
        char *pszDateTime,
        const SceRtcTick *pUtc,
        int iTimeZoneMinutes
);
```

Calling Conditions

Multithread safe

Arguments

pszDateTime	Buffer for receiving formatted string
pUtc	Tick representation of current time (UTC)
iTimeZoneMinutes	Time zone offset (minutes)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function formats the specified time (UTC) according to RFC3339 (ISO8601) format. The time is formatted after it is converted to a local time in which the **Time Zone** setting and **Daylight Saving** setting of the **Date & Time** settings in the system settings are reflected.

For example, December 3, 1995 13:23 in London in RFC3339 format would be

```
1995-12-03T13:23:00.00Z
```

When NULL is specified to *pUtc*, the current time will be formatted.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcFormatRFC3339LocalTime

Format Tick-representation UTC time in RFC3339 (ISO8601) format

Definition

```
#include <rtc.h>
int sceRtcFormatRFC3339LocalTime(
        char *pszDateTime,
        const SceRtcTick *pUtc
);
```

Calling Conditions

Multithread safe

Arguments

pszDateTime	Buffer for receiving formatted string
pUtc	Tick representation of current time (UTC)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function formats the specified time (UTC) according to RFC3339 (ISO8601). The time is formatted after it is converted to a local time in which the **Time Zone** setting and **Daylight Saving** setting of the **Date & Time** settings in the system settings are reflected.

For example, December 3, 1995 13:23 in London in RFC3339 format would be

```
1995-12-03T13:23:00.00Z
```

When NULL is specified to *pUtc*, the current time will be formatted.

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcParseDateTime

Parse time information represented as a string

Definition

```
#include <rtc.h>
int sceRtcParseDateTime(
        SceRtcTick *pUtc,
        const char *pszDateTime
);
```

Calling Conditions

Multithread safe

Arguments

pUtc	Pointer to `SceRtcTick` for receiving the time (UTC)
pszDateTime	Formatted string

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function parses a formatted string to convert it to a time having `SceRtcTick` representation. If a string indicating a time zone is matched within the string, it is reflected when converting to the UTC time.

The supported formats are RFC2822, RFC3339 (ISO8601), and the format in which time is converted to a string by the libc function `asctime()`.

For example, December 3, 1995 13:23 in London in each format would be as follows

Format	Example
RFC2822	`Sun, 03 Dec 1995 13:23:00 +0000`
RFC3339	`1995-12-03T13:23:00.00Z`
`asctime()`	`Sun Dec 03 13:23:00 1995`

See Also

`asctime()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcParseRFC3339

Parse time information represented in RFC3339 format

Definition

```
#include <rtc.h>
int sceRtcParseRFC3339(
        SceRtcTick *pUtc,
        const char *pszDateTime
);
```

Calling Conditions

Multithread safe

Arguments

pUtc	Pointer to `SceRtcTick` for receiving the time (UTC)
pszDateTime	String formatted according to RFC3339 format

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function parses a string that was formatted according to RFC3339, to convert it to a time having `SceRtcTick` representation. The string that is input must strictly follow RFC3339 format.

For example, December 3, 1995 13:23:00 in London in RFC3339 format would be

```
1995-12-03T13:23:00.00Z
```

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Tick Manipulation Functions

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetTickResolution

Get Tick corresponding to 1 second

Definition

```
#include <rtc.h>
int sceRtcGetTickResolution(
        void
);
```

Calling Conditions

Multithread safe

Arguments

None

Return Values

Value	Result
`Tick`	Tick corresponding to 1 second
<0	Error

Description

Returns the Tick corresponding to 1 second.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetTick

Get time information in Tick format

Definition

```
#include <rtc.h>
int sceRtcGetTick(
        const SceDateTime *pTime,
        SceRtcTick *pTick
);
#define sceRtcConvertDateTimeToTick(_pdatetime, _ptick)  \
                     sceRtcGetTick(_pdatetime, _ptick)
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` to be converted
pTick	Pointer to `SceRtcTick` for receiving converted time information

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function converts `SceDateTime` time representation to `SceRtcTick` - as a cumulative time in terms of 1 microsecond units starting from 0001/01/01 00:00:00.

See Also

`sceRtcSetTick()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetTick

Set time information in Tick format

Definition

```
#include <rtc.h>
int sceRtcSetTick(
        SceDateTime *pTime,
        const SceRtcTick *pTick
);
#define sceRtcConvertTickToDateTime(_ptick, _pdatetime)  \
                    sceRtcSetTick(_pdatetime, _ptick)
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` for receiving converted time information
pTick	Pointer to `SceRtcTick` to be converted

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function converts `SceRtcTick` to `SceDateTime` time representation - as a cumulative time in terms of 1 microsecond units starting from 0001/01/01 00:00:00.

See Also

`sceRtcGetTick()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcTickAddTicks

Add specified number of Ticks to time having Tick representation

Definition

```
#include <rtc.h>
int sceRtcTickAddTicks(
        SceRtcTick *pTick0,
        const SceRtcTick *pTick1,
        SceLong64 lAdd
);
```

Calling Conditions

Multithread safe

Arguments

pTick0	Pointer to `SceRtcTick` for receiving result
pTick1	Pointer to `SceRtcTick` to which to add
lAdd	Amount to add (in Ticks)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function adds an interval, in Ticks, to the time information represented as `SceRtcTick`. The same address may be specified for *pTick0* and *pTick1*. To perform subtraction, specify a negative value for *lAdd*.

See Also

`sceRtcTickAddMicroseconds()`, `sceRtcTickAddSeconds()`, `sceRtcTickAddMinutes()`, `sceRtcTickAddHours()`, `sceRtcTickAddDays()`, `sceRtcTickAddWeeks()`, `sceRtcTickAddMonths()`, `sceRtcTickAddYears()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcTickAddMicroseconds

Add specified number of microseconds to time having Tick representation

Definition

```
#include <rtc.h>
int sceRtcTickAddMicroseconds(
        SceRtcTick *pTick0,
        const SceRtcTick *pTick1,
        SceLong64 lAdd
);
```

Calling Conditions

Multithread safe

Arguments

pTick0	Pointer to `SceRtcTick` for receiving result
pTick1	Pointer to `SceRtcTick` to which to add
lAdd	Amount to add (in microseconds)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function adds an interval, in microseconds, to the time information represented as `SceRtcTick`. The same address may be specified for *pTick0* and *pTick1*. To perform subtraction, specify a negative value for *lAdd*.

See Also

`sceRtcTickAddTicks()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcTickAddSeconds

Add specified number of seconds to time having Tick representation

Definition

```
#include <rtc.h>
int sceRtcTickAddSeconds(
        SceRtcTick *pTick0,
        const SceRtcTick *pTick1,
        SceLong64 lAdd
);
```

Calling Conditions

Multithread safe

Arguments

pTick0	Pointer to `SceRtcTick` for receiving result
pTick1	Pointer to `SceRtcTick` to which to add
lAdd	Amount to add (in seconds)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function adds an interval, in seconds, to the time information represented by `SceRtcTick`. The same address may be specified for *pTick0* and *pTick1*. To perform subtraction, specify a negative value for *lAdd*.

See Also

`sceRtcTickAddTicks()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcTickAddMinutes

Add specified number of minutes to time having Tick representation

Definition

```
#include <rtc.h>
int sceRtcTickAddMinutes(
        SceRtcTick *pTick0,
        const SceRtcTick *pTick1,
        SceLong64 lAdd
);
```

Calling Conditions

Multithread safe

Arguments

pTick0	Pointer to `SceRtcTick` for receiving result
pTick1	Pointer to `SceRtcTick` to which to add
lAdd	Amount to add (in minutes)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function adds an interval, in minutes, to the time information represented as `SceRtcTick`. The same address may be specified for *pTick0* and *pTick1*. To perform subtraction, specify a negative value for *lAdd*.

See Also

`sceRtcTickAddTicks()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcTickAddHours

Add specified number of hours to time having Tick representation

Definition

```
#include <rtc.h>
int sceRtcTickAddHours(
        SceRtcTick *pTick0,
        const SceRtcTick *pTick1,
        int iAdd
);
```

Calling Conditions

Multithread safe

Arguments

pTick0	Pointer to `SceRtcTick` for receiving result
pTick1	Pointer to `SceRtcTick` to which to add
iAdd	Amount to add (in hours)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function adds an interval, in hours, to the time information represented as `SceRtcTick`. The same address may be specified for *pTick0* and *pTick1*. To perform subtraction, specify a negative value for *iAdd*

See Also

`sceRtcTickAddTicks()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcTickAddDays

Add specified number of days to time having Tick representation

Definition

```
#include <rtc.h>
int sceRtcTickAddDays(
        SceRtcTick *pTick0,
        const SceRtcTick *pTick1,
        int iAdd
);
```

Calling Conditions

Multithread safe

Arguments

pTick0	Pointer to `SceRtcTick` for receiving result
pTick1	Pointer to `SceRtcTick` to which to add
iAdd	Amount to add (in days)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function adds an interval, in days, to the time information represented as `SceRtcTick`. The same address may be specified for *pTick0* and *pTick1*. To perform subtraction, specify a negative value for *iAdd*.

See Also

`sceRtcTickAddTicks()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcTickAddWeeks

Add specified number of weeks to time having Tick representation

Definition

```
#include <rtc.h>
int sceRtcTickAddWeeks(
        SceRtcTick *pTick0,
        const SceRtcTick *pTick1,
        int iAdd
);
```

Calling Conditions

Multithread safe

Arguments

pTick0	Pointer to `SceRtcTick` for receiving result
pTick1	Pointer to `SceRtcTick` to which to add
iAdd	Amount to add (in weeks)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function adds an interval, in weeks, to the time information represented as `SceRtcTick`. The same address may be specified for *pTick0* and *pTick1*. To perform subtraction, specify a negative value for *iAdd*

See Also

`sceRtcTickAddTicks()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcTickAddMonths

Add specified number of months to time having Tick representation

Definition

```
#include <rtc.h>
int sceRtcTickAddMonths(
        SceRtcTick *pTick0,
        const SceRtcTick *pTick1,
        int iAdd
);
```

Calling Conditions

Multithread safe

Arguments

pTick0	Pointer to `SceRtcTick` for receiving result
pTick1	Pointer to `SceRtcTick` to which to add
iAdd	Amount to add (in months)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function adds an interval, in months, to the time information represented as `SceRtcTick`. The same address may be specified for *pTick0* and *pTick1*. To perform subtraction, specify a negative value for *iAdd*.

See Also

`sceRtcTickAddTicks()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcTickAddYears

Add specified number of years to time having Tick representation

Definition

```
#include <rtc.h>
int sceRtcTickAddYears(
        SceRtcTick *pTick0,
        const SceRtcTick *pTick1,
        int iAdd
);
```

Calling Conditions

Multithread safe

Arguments

pTick0	Pointer to `SceRtcTick` for receiving result
pTick1	Pointer to `SceRtcTick` to which to add
iAdd	Amount to add (in years)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function adds an interval, in years, to the time information represented as `SceRtcTick`. The same address may be specified for *pTick0* and *pTick1*. To perform subtraction, specify a negative value for *iAdd*.

See Also

`sceRtcTickAddTicks()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcConvertUtcToLocalTime

Convert Tick-representation UTC time into local time

Definition

```
#include <rtc.h>
int sceRtcConvertUtcToLocalTime(
        const SceRtcTick *pUtc,
        SceRtcTick *pLocalTime
);
```

Calling Conditions

Multithread safe

Arguments

pUtc	Pointer to `SceRtcTick` representing the UTC time to be converted
pLocalTime	Pointer to `SceRtcTick` for receiving the converted time (local time)

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
<0	Error

Description

This function converts UTC time information having `SceRtcTick` representation into local time based on the current system settings.

See Also

`sceRtcConvertLocalTimeToUtc()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcConvertLocalTimeToUtc

Convert Tick-representation local time into UTC time

Definition

```
#include <rtc.h>
int sceRtcConvertLocalTimeToUtc(
        const SceRtcTick *pLocalTime,
        SceRtcTick *pUtc
);
```

Calling Conditions

Multithread safe

Arguments

pLocalTime	Pointer to SceRtcTick representing the local time to be converted
pUtc	Pointer to SceRtcTick for receiving the converted time (UTC time)

Return Values

If an error occurs, a negative value is returned.

Value	Result
SCE_OK	Success
<0	Error

Description

This function converts local time information having SceRtcTick representation into UTC time based on the current system settings.

See Also

sceRtcConvertUtcToLocalTime()

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Time Information Manipulation Functions

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetMicrosecond

Get microsecond field

Definition

```
#include <rtc.h>
int sceRtcGetMicrosecond(
        const SceDateTime *pTime
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed

Return Values

A microsecond value is returned.

Description

This function gets microsecond information.

See Also

`sceRtcSetMicrosecond()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetSecond

Get second field

Definition

```
#include <rtc.h>
int sceRtcGetSecond(
        const SceDateTime *pTime
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed

Return Values

A second value is returned.

Description

This function gets second information.

See Also

`sceRtcSetSecond()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetMinute

Get minute field

Definition

```
#include <rtc.h>
int sceRtcGetMinute(
        const SceDateTime *pTime
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed

Return Values

A minute value is returned.

Description

This function gets minute information.

See Also

`sceRtcSetMinute()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetHour

Get hour field

Definition

```
#include <rtc.h>
int sceRtcGetHour(
        const SceDateTime *pTime
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed

Return Values

An hour value is returned.

Description

This function gets hour information.

See Also

`sceRtcSetHour()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetDay

Get day field

Definition

```
#include <rtc.h>
int sceRtcGetDay(
        const SceDateTime *pTime
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed

Return Values

A day value is returned.

Description

This function gets day information.

See Also

`sceRtcSetDay()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetMonth

Get month field

Definition

```
#include <rtc.h>
int sceRtcGetMonth(
        const SceDateTime *pTime
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed

Return Values

A month value is returned.

Description

This function gets month information.

See Also

`sceRtcSetMonth()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetYear

Get year field

Definition

```
#include <rtc.h>
int sceRtcGetYear(
        const SceDateTime *pTime
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed

Return Values

A year value is returned.

Description

This function gets year information.

See Also

`sceRtcSetYear()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetMicrosecond

Set microsecond field

Definition

```
#include <rtc.h>
int sceRtcSetMicrosecond(
        SceDateTime *pTime,
        int microsecond
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed
microsecond	Microsecond value to be set

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function sets microsecond information.

See Also

`sceRtcGetMicrosecond()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetSecond

Set second field

Definition

```
#include <rtc.h>
int sceRtcSetSecond(
        SceDateTime *pTime,
        int second
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed
second	Second value to be set

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function sets second information.

See Also

`sceRtcGetSecond()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetMinute

Set minute field

Definition

```
#include <rtc.h>
int sceRtcSetMinute(
        SceDateTime *pTime,
        int minute
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to SceDateTime on which operation is to be performed
minute	Minute value to be set

Return Values

If an error occurs, a negative value is returned.

Value	Result
SCE_OK	Success
< 0	Error

Description

This function sets minute information.

See Also

sceRtcGetMinute()

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetHour

Set hour field

Definition

```
#include <rtc.h>
int sceRtcSetHour(
        SceDateTime *pTime,
        int hour
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed
hour	Hour value to be set

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function sets hour information.

See Also

`sceRtcGetHour()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetDay

Set day field

Definition

```
#include <rtc.h>
int sceRtcSetDay(
        SceDateTime *pTime,
        int day
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed
day	Day value to be set

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function sets day information.

See Also

`sceRtcGetDay()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetMonth

Set month field

Definition

```
#include <rtc.h>
int sceRtcSetMonth(
        SceDateTime *pTime,
        int month
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed
month	Month value to be set

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function sets month information.

See Also

`sceRtcGetMonth()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetYear

Set year field

Definition

```
#include <rtc.h>
int sceRtcSetYear(
        SceDateTime *pTime,
        int year
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` on which operation is to be performed
year	Year value to be set

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function sets year information.

See Also

`sceRtcGetYear()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Format Conversion Functions

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetDosTime

Get time information in MS-DOS format

Definition

```
#include <rtc.h>
int sceRtcGetDosTime(
        const SceDateTime *pDateTime,
        unsigned int *puiDosTime
);
#define sceRtcConvertDateTimeToDosTime(_pdatetime, _pdostime)   \
                    sceRtcGetDosTime(_pdatetime, _pdostime)
```

Calling Conditions

Multithread safe

Arguments

pDateTime	Pointer to `SceDateTime` format to be converted
puiDosTime	Pointer to `unsigned int` for receiving converted MS-DOS format time information

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function converts from the librtc time representation `SceDateTime` to a 32-bit unsigned int value that is the MS-DOS format time representation.

See Also

`sceRtcSetDosTime()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetTime_t

Get time information in POSIX time_t format

Definition

```
#include <rtc.h>
int sceRtcGetTime_t(
        const SceDateTime *pDateTime,
        time_t *piTime
);
#define sceRtcConvertDateTimeToTime_t(_pdatetime, _ptimet)      \
                     sceRtcGetTime_t(_pdatetime, _ptimet)
```

Calling Conditions

Multithread safe

Arguments

pDateTime	Pointer to `SceDateTime` to be converted
piTime	Pointer to time_t for receiving converted POSIX time_t format time information

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function converts from the librtc time representation, `SceDateTime`, to POSIX time_t format.

The time_t format is represented as a signed 32-bit value and is affected by the year 2038 problem.

See Also

`sceRtcSetTime_t()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetTime64_t

Get time information in time64_t format

Definition

```
#include <rtc.h>
int sceRtcGetTime64_t(
        const SceDateTime *pTime,
        SceUInt64 *pullTime
);
#define sceRtcConvertDateTimeToTime64_t(_pdatetime, _ptimet)    \
                     sceRtcGetTime64_t(_pdatetime, _ptimet)
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` to be converted
pullTime	Pointer to SceUInt64 for receiving converted time64_t format time information

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function converts from the librtc time representation, `SceDateTime`, to time64_t format.

See Also

`sceRtcSetTime64_t()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetWin32FileTime

Get time information in Win32 FILETIME format

Definition

```
#include <rtc.h>
int sceRtcGetWin32FileTime(
        const SceDateTime *pTime,
        SceUInt64 *ulWin32Time
);
#define sceRtcConvertDateTimeToWin32Time(_pdatetime, _pw32time) \
                        sceRtcGetWin32FileTime(_pdatetime, _pw32time)
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` to be converted
ulWin32Time	Pointer to `SceUInt64` for receiving converted Win32 FILETIME format time information

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function converts from the librtc time representation, `SceDateTime`, to an `SceUInt64` value having Win32 FILETIME format.

See Also

`sceRtcSetWin32FileTime()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetDosTime

Set time information in MS-DOS format

Definition

```
#include <rtc.h>
int sceRtcSetDosTime(
        SceDateTime *pDateTime,
        unsigned int uiDosTime
);
#define sceRtcConvertDosTimeToDateTime(_dostime, _pdatetime)    \
                    sceRtcSetDosTime(_pdatetime, _dostime)
```

Calling Conditions

Multithread safe

Arguments

pDateTime	Pointer to `SceDateTime` for receiving converted time information
uiDosTime	MS-DOS format time information to be converted

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function converts an unsigned int value having MS-DOS format time representation to the librtc time representation, `SceDateTime`.

See Also

`sceRtcGetDosTime()`

000004892117

Document serial number: 000004892117

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetTime_t

Set time information in POSIX time_t format

Definition

```
#include <rtc.h>
int sceRtcSetTime_t(
        SceDateTime *pDateTime,
        time_t iTime
);
#define sceRtcConvertTime_tToDateTime(_timet, _pdatetime)       \
                    sceRtcSetTime_t(_pdatetime, _timet)
```

Calling Conditions

Multithread safe

Arguments

pDateTime	Pointer to `SceDateTime` for receiving converted time information
iTime	POSIX time_t format time information to be converted

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function converts from POSIX time_t format to the librtc time representation, `SceDateTime`.

See Also

`sceRtcGetTime_t()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetTime64_t

Set time information in time64_t format

Definition

```
#include <rtc.h>
int sceRtcSetTime64_t(
        SceDateTime *pTime,
        SceUInt64 ullTime
);
#define sceRtcConvertTime64_tToDateTime(_timet, _pdatetime)     \
                    sceRtcSetTime64_t(_pdatetime, _timet)
```

Calling Conditions

Multithread safe

Arguments

pTime	Pointer to `SceDateTime` for receiving converted time information
ullTime	time64_t format time information to be converted

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function converts from time64_t format to the librtc time representation, `SceDateTime`.

See Also

`sceRtcGetTime64_t()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcSetWin32FileTime

Set time information in Win32 FILETIME-compatible format

Definition

```
#include <rtc.h>
int sceRtcSetWin32FileTime(
        SceDateTime *pDateTime,
        SceUInt64 ulWin32Time
);
#define sceRtcConvertWin32TimeToDateTime(_pw32time, _pdatetime) \
                    sceRtcSetWin32FileTime(_pdatetime, _pw32time)
```

Calling Conditions

Multithread safe

Arguments

pDateTime	Pointer to `SceDateTime` for receiving converted time information
ulWin32Time	Win32 FILETIME format time information to be converted

Return Values

If an error occurs, a negative value is returned.

Value	Result
`SCE_OK`	Success
< 0	Error

Description

This function converts an `SceUInt64` value having Win32 FILETIME format to the librtc time representation, `SceDateTime`.

See Also

`sceRtcGetWin32FileTime()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Miscellaneous Functions

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcIsLeapYear

Check leap year

Definition

```
#include <rtc.h>
int sceRtcIsLeapYear(
        int year
);
```

Calling Conditions

Multithread safe

Arguments

year	Year to be checked

Return Values

If an error occurs, a negative value is returned and, depending on the type of error, the low-order 16 bits of the return value will be one of the following values.

Value	Result
1	Leap year
0	Normal year
< 0	Year specified is invalid

Description

This function checks whether or not the year specified by *year* is a leap year.

If the year is a multiple of 4, it is a leap year; however, if it is also a multiple of 100, it is a normal year, unless it is also a multiple of 400, in which case it is a leap year.

See Also

`sceRtcGetDaysInMonth()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetDaysInMonth

Get number of days in specified month

Definition

```
#include <rtc.h>
int sceRtcGetDaysInMonth(
        int year,
        int month
);
```

Calling Conditions

Multithread safe

Arguments

year	Year to be checked
month	Month to be checked

Return Values

The specified number of days is returned. If an error occurs, a negative value is returned.

Value	Result
31	Month having 31 days
30	Month having 30 days
29	Month having 29 days
28	Month having 28 days
< 0	Year or month specified is invalid

Description

This function obtains the number of days in the month specified by *year* and *month*.

See Also

`sceRtcIsLeapYear()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcGetDayOfWeek

Calculate day of the week

Definition

```
#include <rtc.h>
int sceRtcGetDayOfWeek(
        int year,
        int month,
        int day
);
```

Calling Conditions

Multithread safe

Arguments

year	Year for which day of the week is to be calculated
month	Month for which day of the week is to be calculated
day	Day for which day of the week is to be calculated

Return Values

The day of the week for the specified day is returned.

Value	Result
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
< 0	Year, month, or day specified is invalid

Description

This function calculates the day of the week for the date specified by *year*, *month*, and *day*.

See Also

`sceRtcGetDaysInMonth()`

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

sceRtcCheckValid

Check range of each field

Definition

```
#include <rtc.h>
int sceRtcCheckValid(
        const SceDateTime *pTime
);
```

Calling Conditions

Multithread safe

Arguments

pTime	Time information to be checked

Return Values

If an error occurs, a negative value is returned.

Macro	Value	Result
SCE_OK	0	Valid time information
SCE_RTC_ERROR_INVALID_YEAR	0x80251081	Year field value is invalid
SCE_RTC_ERROR_INVALID_MONTH	0x80251082	Month field value is invalid
SCE_RTC_ERROR_INVALID_DAY	0x80251083	Day field value is invalid
SCE_RTC_ERROR_INVALID_HOUR	0x80251084	Hour field value is invalid
SCE_RTC_ERROR_INVALID_MINUTE	0x80251085	Minute field value is invalid
SCE_RTC_ERROR_INVALID_SECOND	0x80251086	Second field value is invalid
SCE_RTC_ERROR_INVALID_MICROSECOND	0x80251087	Microsecond field value is invalid

Description

This function checks whether or not each field of the time information specified by `pTime` has a valid value.

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Constants

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117

Document serial number: 000004892117

Return Codes

List of return codes returned by librtc

Definition

Macro	Value	Description
`SCE_RTC_ERROR_INVALID_VALUE`	0x80251000	A value in the arguments is invalid
`SCE_RTC_ERROR_INVALID_POINTER`	0x80251001	The pointer passed in is invalid.
`SCE_RTC_ERROR_NOT_INITIALIZED`	0x80251002	Library has not yet been initialized. Please initialize before use.
`SCE_RTC_ERROR_ALREADY_REGISTERD`	0x80251003	Already registered
`SCE_RTC_ERROR_NOT_FOUND`	0x80251004	Not registered
`SCE_RTC_ERROR_BAD_PARSE`	0x80251080	Error occurred in parsing, perhaps an invalid format
`SCE_RTC_ERROR_INVALID_YEAR`	0x80251081	The year value is invalid
`SCE_RTC_ERROR_INVALID_MONTH`	0x80251082	The month value is invalid
`SCE_RTC_ERROR_INVALID_DAY`	0x80251083	The day value is invalid
`SCE_RTC_ERROR_INVALID_HOUR`	0x80251084	The hour value is invalid
`SCE_RTC_ERROR_INVALID_MINUTE`	0x80251085	The minute value is invalid
`SCE_RTC_ERROR_INVALID_SECOND`	0x80251086	The second value is invalid
`SCE_RTC_ERROR_INVALID_MICROSECOND`	0x80251087	The microsecond value is invalid

000004892117

Document serial number: 000004892117

©SCEI

Document serial number: 000004892117