Audio Output Function Reference

PlayStation®Vita Programmer Tool Runtime Library Release 3.50

Document serial number: 000004892117

Document serial number: 000004892117

Table of Contents

Audio Driver (Output) Functions

sceAudioOutOpenPort

Get and initialize audio port.

Definition

```
#include <audioout.h>
int sceAudioOutOpenPort(
        int portType,
        int len,
        int freq,
        int param
)
```

Calling Conditions

Multithread safe.

Arguments

portType	Type of port (see details below)
len	Number of samples of output data (see details below)
freq	Sampling frequency (see details below)
param	Data format, etc. (see details below)

Specify one of the following values for *portType*.

Macro	Value	Description
SCE_AUDIO_OUT_PORT_TYPE_MAIN	0	Regular audio port
SCE_AUDIO_OUT_PORT_TYPE_BGM	1	BGM audio port
SCE_AUDIO_OUT_PORT_TYPE_VOICE	2	Voice chat audio port

The values that can be specified for *len* are multiples of 64 from 64 to 65472. In the case of stereo, the L and R channels are counted together as 1 sample.

64*1(=64) to 64*1023(=65472)

The values that can be specified for *freq* are the following.

8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100, 48000

These values represent [Hz] as is.

However, if SCE_AUDIO_OUT_PORT_TYPE_MAIN is specified for *portType*, values other than 48000 are not accepted and an error value (SCE_AUDIO_OUT_ERROR_INVALID_SAMPLE_FREQ) is returned.

The values specified for *param* are expressed in bit fields that correspond to the following parameters.

Parameter	Bit Field	Description
format	Bits 0- 7	Specify data format to be handed over.
reserve0	Bits 8-15	Reserved for future use. Specify 0.
reserve1	Bits16-23	Reserved for future use. Specify 0.
reserve2	Bits 24-30	Reserved for future use. Specify 0.
	Bit 31	Unused bit. Specify 0.

Return Values

The format of the return value is not made public, however this function returns a positive value as the port value when successful. Returns a negative value for errors. (Refer to "Error Codes" for details.) When successful, pass this value as is, as the value of argument *port* for `sceAudioOutReleasePort()`, `sceAudioOutOutput()`, `sceAudioOutSetVolume()`, `sceAudioOutSetConfig()`, `sceAudioOutGetConfig()`, and `sceAudioOutGetRestSample()`.

Description

This function obtains and initializes the audio output port value.

The operation to be performed to the port obtained with this function is specified by passing the return value to another function of this audio library.

To release the obtained port, call **`sceAudioOutReleasePort()`**.

Specify either `SCE_AUDIO_OUT_PARAM_FORMAT_S16_MONO` or `SCE_AUDIO_OUT_PARAM_FORMAT_S16_STEREO` in the `format` bit field.

Differences in operation through *portType*

If `SCE_AUDIO_OUT_PORT_TYPE_MAIN` is specified for *portType*, up to 8 ports can be obtained. However, the only sampling frequency that can be set is 48000. (Specifying a value other than 48000 results in an error and `SCE_AUDIO_OUT_ERROR_INVALID_SAMPLE_FREQ` is returned.)

If other than `SCE_AUDIO_OUT_PORT_TYPE_MAIN` is specified for *portType*, multiple ports cannot be obtained. In addition, even during playback using a port other than the `SCE_AUDIO_OUT_PORT_TYPE_MAIN` port, the audio data may be replaced with other BGM or voice chat audio by the system.

Examples

Refer to the example of **`sceAudioOutOutput()`**.

Notes

The output volume is set to the maximum value (`SCE_AUDIO_VOLUME_0DB`) by initialization.

See Also

`sceAudioOutReleasePort()`, `sceAudioOutOutput()`

sceAudioOutReleasePort

End audio output and release port.

Definition

```
#include <audioout.h>
int sceAudioOutReleasePort(
        int port
)
```

Calling Conditions

Multithread safe.

Arguments

port Port value (Return value of **sceAudioOutOpenPort()**)

Be sure to pass the return value of sceAudioOutOpenPort() to *port*.
Passing arbitrary value may result in unintended operation.

Return Values

Returns SCE_OK(=0) when successful.
Returns a negative value for errors. (Refer to "Error Codes" for details.)

Description

This function performs the audio output termination processing and releases the port.

Examples

Refer to the example of **sceAudioOutOutput**().

See Also

sceAudioOutOpenPort(), sceAudioOutOutput()

Document serial number: 000004892117

sceAudioOutOutput

Blocking playback of audio

Definition

```
#include <audioout.h>
int sceAudioOutOutput(
        int port,
        const void *ptr
)
```

Calling Conditions

Multithread safe.

Arguments

port Port value (Return value of **sceAudioOutOpenPort()**)
**ptr* Pointer to output data buffer

Be sure to pass the return value of `sceAudioOutOpenPort()` to *port*.

Passing arbitrary value may result in unintended operation.

Return Values

Returns the number of samples (value of 0 or greater) registered to the audio driver for normal termination.

Returns a negative value for errors. (Refer to "Error Codes" for details.)

Description

This is the blocking type function for audio output.

Since the return value of `sceAudioOutOpenPort()` is passed to the *port* argument, pass the size and data format specified with **sceAudioOutOpenPort()** to the *ptr* argument.

Specify the address of the sound data buffer to be output for *ptr*. The data is registered to the audio driver. The audio driver outputs the registered sound data to the audio device.

When NULL is specified for *ptr*, the processing returns from the function after the data passed to the driver but not yet output has been output. If there is no data that has not yet been output, the processing returns immediately.

Examples

```
#define PARAMS16 (
SCE_AUDIO_OUT_PARAM_FORMAT_S16_STEREO << SCE_AUDIO_OUT_PARAM_FORMAT_SHIFT)

#define STEREO (2)

export short g_WavData[];  /* Sound data */
short *pWavData = g_WavData;
int portID;

/* Initialization and port acquisition */
portID = sceAudioOutOpenPort(SCE_AUDIO_OUT_PORT_TYPE_MAIN,
                                          256, 48000, PARAMS16);
if (portID < 0) {
    /* Error processing */
    return;
}
while( /* Sound output interval */ ) {
    int ret;
    short *pBuff;

    /* Sound data update */
    pBuff = g_WavData;
    g_WavData += 256*STEREO;

    /* Audio output */
    if ( sceAudioOutOutput(portID, pBuff) < 0) {
        /* Error processing */
   }
}
/* Last data output wait */
if ( sceAudioOutOutput(portID, NULL) < 0) {
    /* Error processing */
}

/* Port release */
sceAudioOutReleasePort(portID);
```

Notes

When this function is called, the playback of the output data is blocked until the output data has been registered to the audio driver, but during this time, the thread that called this function is in the WAIT state. When registration is accepted, the state changes from READY to RUN.

Do not change the contents of the output data buffer until the audio driver's output processing has ended.

See Also

sceAudioOutOpenPort(),sceAudioOutReleasePort()

sceAudioOutSetVolume

Set volume

Definition

```
#include <audioout.h>
int sceAudioOutSetVolume(
        int port,
        int flag,
        int *vol
)
```

Calling Conditions

Multithread safe.

Arguments

port	Port value (Return value of **sceAudioOutOpenPort()**)
flag	Sets (to 1) the bit corresponding to the channel of the output data to be specified.
**vol*	Address of the array of the volume value corresponding to the *flag* bit Range of values set to each array: 0 to SCE_AUDIO_VOLUME_0DB

Be sure to pass the return value of sceAudioOutOpenPort() to *port*.

Passing arbitrary value may result in unintended operation.

Return Values

Returns SCE_OK(=0) when successful.

Returns a negative value for errors. (Refer to "Error Codes" for details.)

Description

This function sets the volume. It has an argument structure that can support multiple audio channel data formats. Currently it supports up to the two channels of stereo.

If the value of the bit corresponding to the output data channel of the *flag* argument is 1, that value is reflected to the corresponding array of *vol*[] as the volume value. Even if the port type is SCE_AUDIO_OUT_PARAM_FORMAT_S16_STEREO or SCE_AUDIO_OUT_PARAM_FORMAT_S16_MONO, the volume of the L channel output corresponds to bit 0, and that of the R channel to bit 1.

Examples

```
#define PARAMS16 (
SCE_AUDIO_OUT_PARAM_FORMAT_S16_STEREO << SCE_AUDIO_OUT_PARAM_FORMAT_SHIFT)

int portID;
int volume[2];

/* Initialization and port obtainment */
portID = sceAudioOutOpenPort(SCE_AUDIO_OUT_PORT_TYPE_MAIN,
                                          256, 48000, PARAMS16);

/* Set value of maximum volume/2 to both L and R channels */
volume[0] = volume[1] = (SCE_AUDIO_VOLUME_0DB / 2);
sceAudioOutSetVolume(portID, 0x0003, volume);

while( /* Playback continuation condition */ ){
    /* Sound playback processing */
}

/* Release port */
sceAudioOutReleasePort(portID);
```

Notes

The effect of the set volume is reflected from the next mixer processing unit in the driver.

Since all the volumes are initialized to the maximum value by the initialization function **`sceAudioOutOpenPort()`**, perform volume setting after calling **`sceAudioOutOpenPort()`**.

See Also

`sceAudioOutOutput()`

sceAudioOutSetConfig

Set parameters again

Definition

```
#include <audioout.h>
int sceAudioOutSetConfig(
        int port,
        int len,
        int freq,
        int param
)
```

Calling Conditions

Multithread safe.

Arguments

port	Port value (Return value of **sceAudioOutOpenPort()**)
len	Number of samples of output data (refer to **sceAudioOutOpenPort()**)
freq	Sampling frequency (refer to **sceAudioOutOpenPort()**)
param	Data format, etc. (refer to **sceAudioOutOpenPort()**)

Be sure to pass the return value of `sceAudioOutOpenPort()` to *port*.

Passing arbitrary value may result in unintended operation.

Return Values

Returns `SCE_OK(=0)` when successful.

Returns a negative value for errors. (Refer to "Error Codes" for details.)

Description

This function sets again the sound parameters that were set during initialization.

The parameters can be changed without releasing the port or stopping audio playback.

Specifying -1 for arguments *len*, *freq*, and *param* holds the current value, so only the value of the parameter to be changed can be set for the argument, and -1 can be specified for the other parameters.

Examples

```
#define PARAMS16 (
SCE_AUDIO_OUT_PARAM_FORMAT_S16_STEREO << SCE_AUDIO_OUT_PARAM_FORMAT_SHIFT)

int portID;

/* Initialization and port obtainment */
portID = sceAudioOutOpenPort(SCE_AUDIO_OUT_PORT_TYPE_BGM,
                                          256, 48000, PARAMS16);

while( /* Playback continuation condition */ ){
   /* Sound playback processing */

   if( /* Parameter change condition 1 */ ) {
      sceAudioOutSetConfig(portID, 1024, 44100, PARAMS16);
   }
   if( /* Parameter change condition 2 */ ) {
      sceAudioOutSetConfig(portID, 256, 48000, PARAMS16);
   }
   if( /* Parameter change condition 3  Change only sampling frequency */ ) {
      sceAudioOutSetConfig(portID, -1, 24000, -1);
   }

}

/* Release port */
sceAudioOutReleasePort(portID);
```

Notes

If the settings are changed with this function during audio output processing, the playback sound becomes discontinuous and noise occurs. Therefore, prior to using this function, reduce the volume with `sceAudioOutSetVolume()`.

The changed settings are reflected from the next time **`sceAudioOutOutput()`** is called.

The settings done with this function are overwritten with the **`sceAudioOutOpenPort()`** initialization function.

See Also

`sceAudioOutOpenPort()`

sceAudioOutSetAlcMode

Set ALC

Definition

```
#include <audioout.h>
int sceAudioOutSetAlcMode(
        int mode
)
```

Calling Conditions

Multithread safe.

Arguments

mode ALC mode.
Specify `SCE_AUDIO_ALC_OFF` (disabled) or `SCE_AUDIO_ALC_MODE1` (enabled).

Return Values

Returns `SCE_OK(=0)` when successful.

Returns a negative value for errors. (Refer to "Error Codes" for details.)

Description

This function sets whether to enable or disable the dynamic normalizer (ALC: Automatic Level Control) effect of the BGM port. The setting is memorized for each process, and the enabled/disabled status keeps track of the process switching. When a process is generated, the setting is reset to the ALC disabled status. The ALC enabled/disabled status is stored regardless of the opened/closed status of the BGM port.

Examples

```
sceAudioOutSetAlcMode(SCE_AUDIO_ALC_MODE1);
```

Notes

Calling this function when a sound is produced creates a discontinuous wave and noise, so call this function when the BGM port is not producing a sound.

ALC processing is performed on the main processor, so take into consideration the effects of an increased processing load when ALC is enabled.

sceAudioOutGetConfig

Get parameter values

Definition

```
#include <audioout.h>
int sceAudioOutGetConfig(
        int port,
        int configType
)
```

Calling Conditions

Multithread safe.

Arguments

port	Port value (Return value of **sceAudioOutOpenPort()**)
configType	Type of setting value to be obtained (see details below)

Be sure to pass the return value of `sceAudioOutOpenPort()` to *port*.

Passing arbitrary value may result in unintended operation.

Specify one of the following values for *configType*

Macro	Value	Description
SCE_AUDIO_OUT_CONFIG_TYPE_LEN	0	Obtains granularity.
SCE_AUDIO_OUT_CONFIG_TYPE_FREQ	1	Obtains sampling frequency.
SCE_AUDIO_OUT_CONFIG_TYPE_PARAM	2	Obtains *param*.

Return Values

The setting value (positive value) specified for the *configType* argument is returned when successful.

Returns a negative value for errors. (Refer to "Error Codes" for details.)

Description

This function obtains the current setting values. The setting values are the values set with **sceAudioOutOpenPort()** or **sceAudioOutSetConfig()**.

See Also

sceAudioOutOpenPort(),sceAudioOutSetConfig()

sceAudioOutGetRestSample

Get number of samples that have not been played back

Definition

```
#include <audioout.h>
int sceAudioOutGetRestSample(
        int port
)
```

Calling Conditions

Multithread safe.

Arguments

port Port value (Return value of **sceAudioOutOpenPort()**)

Be sure to pass the return value of `sceAudioOutOpenPort()` to *port*.

Passing arbitrary value may result in unintended operation.

Return Values

Returns a value of 0 or greater as the number of samples that have not been played back during audio output.

Returns a negative value for errors. (Refer to "Error Codes" for details.)

Description

This function returns the number of samples of sound data that have not been played back whose output is planned in the audio driver. If the port type is other than the MAIN port, either the granularity value set for the *len* argument or 0 is returned.

Examples

```
#define PARAMS16 (
SCE_AUDIO_OUT_PARAM_FORMAT_S16_STEREO << SCE_AUDIO_OUT_PARAM_FORMAT_SHIFT)

int portID;
int restSample;

/* Initialization and port obtainment */
portID = sceAudioOutOpenPort(SCE_AUDIO_OUT_PORT_TYPE_BGM,
                                    256, 48000, PARAMS16);

while( /* Playback continuation condition */ ){

    /* Sound playback processing */

    restSample = sceAudioOutGetRestSample(portID);
}

/* Port release */
sceAudioOutReleasePort(portID);
```

sceAudioOutGetAdopt

Get status of whether port is selected as target for sound generation

Definition

```
#include <audioout.h>
int sceAudioOutGetAdopt(
        int portType
)
```

Calling Conditions

Multithread safe.

Arguments

portType Type of port (see details below)

Specify one of the following values for *portType*.

Macro	Value	Description
SCE_AUDIO_OUT_PORT_TYPE_MAIN	0	Regular audio port
SCE_AUDIO_OUT_PORT_TYPE_BGM	1	BGM audio port
SCE_AUDIO_OUT_PORT_TYPE_VOICE	2	Voice chat audio port

Return Values

If the port is the target for sound generation, 1 is returned. Otherwise, 0 is returned.

Returns a negative value for errors. (Refer to "Error Codes" for details.)

Description

The system selects the port for generating sound with the audio port type used between processes as the unit. Audio driver functions can be called as normal simply by not generating sound even when the port is not selected. This function can be used to learn whether the audio from the port of the type specified with the current process is the target for sound generation. Although the status can be known before the port is opened, the moment at which the status changes cannot be known. The status may have changed since the time it was obtained.

Examples

```
int adopt;

/* Initialization and port obtainment */
adopt = sceAudioOutGetAdopt(SCE_AUDIO_OUT_PORT_TYPE_BGM);

if (adopt) {
        /* Sound is generated from BGM port */
} else {
        /* Sound is not generated from BGM port */
}
```

Constants

Port Type

Constants indicating the audio port type

Definition

Macro	Value	Description
SCE_AUDIO_OUT_PORT_TYPE_MAIN	0	MAIN port
SCE_AUDIO_OUT_PORT_TYPE_BGM	1	BGM port
SCE_AUDIO_OUT_PORT_TYPE_VOICE	2	VOICE port

Description

These constants specify the type of the audio output port with the *portType* argument of the **sceAudioOutOpenPort()** function.

Format

Constants indicating the audio format

Definition

Macro	Value	Description
SCE_AUDIO_OUT_PARAM_FORMAT_S16_MONO	0	16-bit integer monaural
SCE_AUDIO_OUT_PARAM_FORMAT_S16_STEREO	1	16-bit integer stereo

Macro	Value	Description
SCE_AUDIO_OUT_PARAM_FORMAT_MASK	0x000000ff	Mask value in *param*
SCE_AUDIO_OUT_PARAM_FORMAT_SHIFT	0	Shift value in *param*

Description

These constants specify the format of audio output with the `format` bit field of the *param* argument of the **`sceAudioOutOpenPort()`** function. Both consist of 16-bit integer little endian data.

`SCE_AUDIO_OUT_PARAM_FORMAT_S16_STEREO` is the L/R interleave format.

`SCE_AUDIO_OUT_PARAM_FORMAT_MASK` and `SCE_AUDIO_OUT_PARAM_FORMAT_SHIFT` are the mask value and the shift value for handling the bit field in *param*.

Type of Status to Be Obtained

Constant specifying the type of status to be obtained

Definition

Macro	Value	Description
SCE_AUDIO_OUT_CONFIG_TYPE_LEN	0	Granularity obtainment
SCE_AUDIO_OUT_CONFIG_TYPE_FREQ	1	Sampling frequency obtainment
SCE_AUDIO_OUT_CONFIG_TYPE_PARAM	2	*param* obtainment

Description

This constant is specified in the *configType* argument of the **sceAudioOutGetConfig()** function. It specifies the type of the status to be obtained.

Volume Value

Volume value constant

Definition

Macro	Value	Description
SCE_AUDIO_VOLUME_0DB	32768	Maximum value of volume

Description

This is the maximum value of the volume to be specified with the **sceAudioOutSetVolume()** function.

Error Codes

List of error codes returned by audio output function

Definition

Macro	Value	Description
SCE_AUDIO_OUT_ERROR_NOT_OPENED	0x80260001	Specified port is not opened
SCE_AUDIO_OUT_ERROR_BUSY	0x80260002	Request invalid because audio output is in progress
SCE_AUDIO_OUT_ERROR_INVALID_PORT	0x80260003	ID value of port is invalid
SCE_AUDIO_OUT_ERROR_INVALID_POINTER	0x80260004	Invalid pointer value
SCE_AUDIO_OUT_ERROR_PORT_FULL	0x80260005	No more ports can be opened
SCE_AUDIO_OUT_ERROR_INVALID_SIZE	0x80260006	Specified sample length is invalid
SCE_AUDIO_OUT_ERROR_INVALID_FORMAT	0x80260007	Format value is invalid
SCE_AUDIO_OUT_ERROR_INVALID_SAMPLE_FREQ	0x80260008	Sampling frequency value is invalid
SCE_AUDIO_OUT_ERROR_INVALID_VOLUME	0x80260009	Volume value is invalid
SCE_AUDIO_OUT_ERROR_INVALID_PORT_TYPE	0x8026000a	Port type value is invalid
SCE_AUDIO_OUT_ERROR_OUT_OF_MEMORY	0x8026000d	Not enough memory
SCE_AUDIO_OUT_ERROR_INVALID_FX_TYPE	0x80260200	Value for the ALC mode is invalid