

Audio Input Function Overview

© 2012 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

Table of Contents

1 Library Overview..... 3

 Features3

 Files3

 Sample Programs.....3

 Reference Materials3

2 Usage Procedure 4

 Basic Processing Procedure4

 Examples.....5

 Notes5

000004892117

1 Library Overview

Features

The audio input library is used to retrieve audio data in the PCM format. It allows retrieving the signal of the audio input device (internal microphone, headset microphone, etc.) selected by the system. Currently 2 port types are provided.

Port Type	Sampling Frequency	Description
VOICE	16KHz	Used for chat and other audio applications. Functions as an echo canceller and ALC (automatic level control) on the system.
RAW	16KHz, 48KHz	Provides a signal without performing signal processing.

Files

The following files are required in order to use the audio input library.

Filename	Description
audioin.h	Header file
libSceAudioIn_stub.a	Stub library file

Sample Programs

The following program using the audio input library is provided for reference purposes.

- samples/sample_code/audio_video/api_audioin/simple/

Reference Materials

For details on the various functions, refer to the "Audio Input Function Reference" document.

2 Usage Procedure

Basic Processing Procedure

The basic flow of the processing procedure is described below.

List of Functions

Function Name	Description
<code>sceAudioInOpenPort ()</code>	Acquires audio input port
<code>sceAudioInInput ()</code>	Captures audio input data
<code>sceAudioInReleasePort ()</code>	Releases audio input port

These functions are used with the following procedure.

(1) Initialization

The input port is generated with the `sceAudioInOpenPort ()` function.

The port type, input sample granularity, sampling frequency rate of the captured data, and number of channels are set. The setting value has restrictions depending on the port type. For details, refer to the "Audio Input Function Reference" document.

(2) Audio input

Audio input is done with `sceAudioInInput ()`.

`sceAudioInInput ()` is a function to perform blocking read. When the audio input signal is captured, the read timing jitter does exist and must be taken into account.

- Set a buffering margin with the premise that jitter exists.
- In addition, discrepancies in the data volume over a given period of time may occur as a result of discontinuous signals due to device change, etc. Therefore check buffering status and perform any necessary corrections every time you retrieve data with this function.

Since these are blocking functions, creation of dedicated threads and the use of these functions from these threads are recommended.

(3) Terminating processing

The input port generated with the `sceAudioInOpenPort ()` function is released with the `sceAudioInReleasePort ()` function.

Examples

The following is an example of performing sound input (capture every 32 msec) while the sound input conditions are met.

```
Int portID;

/* Acquire port */
portID = sceAudioInOpenPort(
    SCE_AUDIO_IN_PORT_TYPE_VOICE , 512, 16000,
    SCE_AUDIO_IN_PARAM_FORMAT_S16_MONO);
if( portID < 0 ) {
    /* Error. The following processing is not possible */
    /* To required processing such as user notification */
}

/* Sound input loop */
While( /* Sound input condition */ ) {
    /* Sound input */
    Res = sceAudioInInput( portID, buffer);
    /* This is a blocking function. In the case of this setting,
    * the function captures data every 512*(1/16000) = 32 msec and returns
    * on an average. precautions must be taken. Refer to the document.
    */
    If( Res < 0 ) {
        /* Anomaly occurrence. Error processing */
    }
    /* Process input sound */
    /* For example, write to ring buffer */
}

/* Release input port */
sceAudioInReleasePort(portID);
```

Notes

With regard to PlayStation®Vita, only one application can obtain the usage right of the audio input signal.

Furthermore, a system chat running simultaneously with applications is scheduled to be implemented on the PlayStation®Vita. Since the system chat will run simultaneously with applications, it will hold the usage right of audio input signal, and applications might be unable to obtain this.

In the case that an application is unable to obtain the usage right of the audio input signal, that application's audio input will continue running, but the audio input signal will be muted, entering silent state.

For instance, in the case of an application that requires microphone input, microphone input will, in practice, become inactive for the application if the application is unable to obtain the usage right of audio input signal, because the audio input signal will be silent.

If necessary, check audio input status as appropriate with functions for detecting audio input status, such as `sceAudioInGetAdopt()` and `sceAudioInGetStatus()`. Polling is envisaged for both functions. For details, refer to the "Audio Input Function Reference" document.