

How to use SPORT AC97 driver

1. Introduction

SPORT in ADSP-21535 DSP works in MCM(Multiple Channel Mode), communicating with AC97 Codec AD1885 via AC-Link.

This driver supplies a typical sound core interface.

2. Configuration of sound driver

We support following ioctl codes:

OSS_GETVERSION
SNDCTL_DSP_RESET
SNDCTL_DSP_SYNC
SNDCTL_DSP_SPEED
SNDCTL_DSP_STEREO
SNDCTL_DSP_GETBLKSIZE
SNDCTL_DSP_SETFMT
SNDCTL_DSP_GETFMTS
SNDCTL_DSP_CHANNELS
SNDCTL_DSP_POST
SNDCTL_DSP_SUBDIVIDE
SNDCTL_DSP_SETFRAGMENT
SNDCTL_DSP_GETOSPACE
SNDCTL_DSP_GETOPTR
SNDCTL_DSP_GETISPACE
SNDCTL_DSP_GETIPTR
SNDCTL_DSP_GETCAPS
SNDCTL_DSP_SETTRIGGER
SNDCTL_DSP_GETTRIGGER
SNDCTL_DSP_GETODELAY
SOUND_PCM_READ_RATE
SOUND_PCM_READ_CHANNELS
SOUND_PCM_READ_BITS

Some ioctl code explanation:

Please refer to <http://www.opensound.com> for more information the Programmer's Guide can be found at <http://www.opensound.com/pguide/index.html>

SNDCTL_DSP_SYNC:

The ioctl SNDCTL_DSP_SYNC can be used when an application needs to wait until the last byte written to the device has been played (it doesn't wait in recording mode). When that occurs, the call resets (stops) the device and returns back to the calling program. Note that this call may take several seconds to execute depending on the amount of data in the buffers. Closing any sound device calls SNDCTL_DSP_SYNC implicitly. It is highly recommended that you close and reopen the device instead of calling SNDCTL_DSP_SYNC.

SNDCTL_DSP_SPEED:

Set sample rate

SNDCTL_DSP_STEREO:

Set stereo or mono channel

`SNDCTL_DSP_GETBLKSIZE`:

Application may ask the buffer size by using this ioctl call

`SNDCTL_DSP_GETFMTS`:

A program can check which formats are supported by the device by calling
ioctl `SNDCTL_DSP_GETFMTS`

`SNDCTL_DSP_SETFMT`:

The sample format can be set using the ioctl call `SNDCTL_DSP_SETFMT`.

`SNDCTL_DSP_CHANNELS`:

Most modern audio devices support stereo mode. The default mode is mono.
An application can select the number of channels calling ioctl
`SNDCTL_DSP_CHANNELS` with an argument specifying the number of channels

`SNDCTL_DSP_POST`:

The ioctl `SNDCTL_DSP_POST` is a lightweight version of `SNDCTL_DSP_SYNC`.
It just tells the driver that there is likely to be a pause in the output.
This makes it possible for the device to handle the pause more intelligently.
This ioctl call doesn't block the application.

`SNDCTL_DSP_SETFRAGMENT`:

Selecting Buffering Parameters (fragment size)

`SNDCTL_DSP_NONBLOCK`:

Select non-block mode

`SNDCTL_DSP_GETCAPS`:

Checking Device Capabilities

3. Receive data

```
fd = open("/dev/dsp", O_RDONLY, 0);  
...  
read(fd, buffer_rx, number_of_bytes);  
...  
close(fd);
```

4. Send data

```
fd = open("/dev/dsp", O_WRONLY, 0);  
...  
write(fd, buffer_tx, number_of_bytes);  
...  
close(fd);
```