ALSA PCM channel-mapping API

Takashi Iwai <tiwai@suse.de>

General

The channel mapping API allows user to query the possible channel maps and the current channel map, also optionally to modify the channel map of the current stream.

A channel map is an array of position for each PCM channel. Typically, a stereo PCM stream has a channel map of `{ front_left, front_right }` while a 4.0 surround PCM stream has a channel map of `{ front left, front right, rear left, rear right }`.

The problem, so far, was that we had no standard channel map explicitly, and applications had no way to know which channel corresponds to which (speaker) position. Thus, applications applied wrong channels for 5.1 outputs, and you hear suddenly strange sound from rear. Or, some devices secretly assume that center/LFE is the third/fourth channels while others that C/LFE as 5th/6th channels.

Also, some devices such as HDMI are configurable for different speaker positions even with the same number of total channels. However, there was no way to specify this because of lack of channel map specification. These are the main motivations for the new channel mapping API.

Design

Actually, "the channel mapping API" doesn't introduce anything new in the kernel/user-space ABI perspective. It uses only the existing control element features.

As a ground design, each PCM substream may contain a control element providing the channel mapping information and configuration. This element is specified by:

- iface = SNDRV_CTL_ELEM_IFACE_PCM
- name = "Playback Channel Map" or "Capture Channel Map"
- device = the same device number for the assigned PCM substream
- index = the same index number for the assigned PCM substream

Note the name is different depending on the PCM substream direction.

Each control element provides at least the TLV read operation and the read operation. Optionally, the write operation can be provided to allow user to change the channel map dynamically.

TLV

The TLV operation gives the list of available channel maps. A list item of a channel map is usually a TLV of `type data-bytes ch0 ch1 ch2...` where type is the TLV type value, the second argument is the total bytes (not the numbers) of channel values, and the rest are the position value for each channel.

As a TLV type, either `SNDRV_CTL_TLVT_CHMAP_FIXED`, `SNDRV_CTL_TLV_CHMAP_VAR` or `SNDRV_CTL_TLVT_CHMAP_PAIRED` can be used. The `_FIXED` type is for a channel map with the fixed channel position while the latter two are for flexible channel positions. `_VAR` type is for a channel map where all channels are freely swappable and `_PAIRED` type is where pair-wise channels are swappable. For example, when you have {FL/FR/RL/RR} channel map, `_PAIRED` type would allow you to swap only {RL/RR/FL/FR} while `_VAR` type would allow even swapping FL and RR.

These new TLV types are defined in `sound/tlv.h`.

The available channel position values are defined in `sound/asound.h`, here is a cut:

```
/* channel positions */
enum {
        SNDRV_CHMAP_UNKNOWN = 0,
        SNDRV_CHMAP_NA,         /* N/A, silent */
        SNDRV_CHMAP_MONO,       /* mono stream */
        /* this follows the alsa-lib mixer channel value + 3 */
        SNDRV_CHMAP_FL,         /* front left */
        SNDRV_CHMAP_FR,         /* front right */
        SNDRV_CHMAP_RL,         /* rear left */
        SNDRV_CHMAP_RR,         /* rear right */
        SNDRV_CHMAP_FC,         /* front center */
        SNDRV_CHMAP_LFE,        /* LFE */
        SNDRV_CHMAP_SL,         /* side left */
        SNDRV_CHMAP_SR,         /* side right */
        SNDRV_CHMAP_RC,         /* rear center */
        /* new definitions */
        SNDRV_CHMAP_FLC,        /* front left center */
```

```
        SNDRV_CHMAP_FRC,        /* front right center */
        SNDRV_CHMAP_RLC,        /* rear left center */
        SNDRV_CHMAP_RRC,        /* rear right center */
        SNDRV_CHMAP_FLW,        /* front left wide */
        SNDRV_CHMAP_FRW,        /* front right wide */
        SNDRV_CHMAP_FLH,        /* front left high */
        SNDRV_CHMAP_FCH,        /* front center high */
        SNDRV_CHMAP_FRH,        /* front right high */
        SNDRV_CHMAP_TC,         /* top center */
        SNDRV_CHMAP_TFL,        /* top front left */
        SNDRV_CHMAP_TFR,        /* top front right */
        SNDRV_CHMAP_TFC,        /* top front center */
        SNDRV_CHMAP_TRL,        /* top rear left */
        SNDRV_CHMAP_TRR,        /* top rear right */
        SNDRV_CHMAP_TRC,        /* top rear center */
        SNDRV_CHMAP_LAST = SNDRV_CHMAP_TRC,
};
```

When a PCM stream can provide more than one channel map, you can provide multiple channel maps in a TLV container type. The TLV data to be returned will contain such as:

```
SNDRV_CTL_TLVT_CONTAINER 96
    SNDRV_CTL_TLVT_CHMAP_FIXED 4 SNDRV_CHMAP_FC
    SNDRV_CTL_TLVT_CHMAP_FIXED 8 SNDRV_CHMAP_FL SNDRV_CHMAP_FR
    SNDRV_CTL_TLVT_CHMAP_FIXED 16 NDRV_CHMAP_FL SNDRV_CHMAP_FR \
        SNDRV_CHMAP_RL SNDRV_CHMAP_RR
```

The channel position is provided in LSB 16bits. The upper bits are used for bit flags.

```
#define SNDRV_CHMAP_POSITION_MASK       0xffff
#define SNDRV_CHMAP_PHASE_INVERSE       (0x01 << 16)
#define SNDRV_CHMAP_DRIVER_SPEC         (0x02 << 16)
```

`SNDRV_CHMAP_PHASE_INVERSE` indicates the channel is phase inverted, (thus summing left and right channels would result in almost silence). Some digital mic devices have this.

When `SNDRV_CHMAP_DRIVER_SPEC` is set, all the channel position values don't follow the standard definition above but driver-specific.

Read Operation

The control read operation is for providing the current channel map of the given stream. The control element returns an integer array containing the position of each channel.

When this is performed before the number of the channel is specified (i.e. hw_params is set), it should return all channels set to `UNKNOWN`.

Write Operation

The control write operation is optional, and only for devices that can change the channel configuration on the fly, such as HDMI. User needs to pass an integer value containing the valid channel positions for all channels of the assigned PCM substream.

This operation is allowed only at PCM PREPARED state. When called in other states, it shall return an error.