

ASoC Platform Driver

An ASoC platform driver class can be divided into audio DMA drivers, SoC DAI drivers and DSP drivers. The platform drivers only target the SoC CPU and must have no board specific code.

Audio DMA

The platform DMA driver optionally supports the following ALSA operations:-

```
/* SoC audio ops */
struct snd_soc_ops {
    int (*startup)(struct snd_pcm_substream *);
    void (*shutdown)(struct snd_pcm_substream *);
    int (*hw_params)(struct snd_pcm_substream *, struct snd_pcm_hw_params *);
    int (*hw_free)(struct snd_pcm_substream *);
    int (*prepare)(struct snd_pcm_substream *);
    int (*trigger)(struct snd_pcm_substream *, int);
};
```

The platform driver exports its DMA functionality via struct `snd_soc_component_driver`:-

```
struct snd_soc_component_driver {
    const char *name;

    ...
    int (*probe)(struct snd_soc_component *);
    void (*remove)(struct snd_soc_component *);
    int (*suspend)(struct snd_soc_component *);
    int (*resume)(struct snd_soc_component *);

    /* pcm creation and destruction */
    int (*pcm_new)(struct snd_soc_pcm_runtime *);
    void (*pcm_free)(struct snd_pcm *);

    ...
    const struct snd_pcm_ops *ops;
    const struct snd_compr_ops *compr_ops;
    ...
};
```

Please refer to the ALSA driver documentation for details of audio DMA. <http://www.alsa-project.org/~iwai/writing-an-alsa-driver/>

An example DMA driver is `soc/pxa/pxa2xx-pcm.c`

SoC DAI Drivers

Each SoC DAI driver must provide the following features:-

1. Digital audio interface (DAI) description
2. Digital audio interface configuration
3. PCM's description
4. SYSCLK configuration
5. Suspend and resume (optional)

Please see `codec.rst` for a description of items 1 - 4.

SoC DSP Drivers

Each SoC DSP driver usually supplies the following features :-

1. DAPM graph
2. Mixer controls
3. DMA IO to/from DSP buffers (if applicable)
4. Definition of DSP front end (FE) PCM devices.

Please see `DPCM.txt` for a description of item 4.