

ASoC Machine Driver

The ASoC machine (or board) driver is the code that glues together the platform and codec drivers.

The machine driver can contain codec and platform specific code. It registers the audio subsystem with the kernel as a platform device and is represented by the following struct:-

```
/* SoC machine */
struct snd_soc_card {
    char *name;

    int (*probe)(struct platform_device *pdev);
    int (*remove)(struct platform_device *pdev);

    /* the pre and post PM functions are used to do any PM work before and
     * after the codec and DAIs do any PM work. */
    int (*suspend_pre)(struct platform_device *pdev, pm_message_t state);
    int (*suspend_post)(struct platform_device *pdev, pm_message_t state);
    int (*resume_pre)(struct platform_device *pdev);
    int (*resume_post)(struct platform_device *pdev);

    /* machine stream operations */
    struct snd_soc_ops *ops;

    /* CPU <--> Codec DAI links */
    struct snd_soc_dai_link *dai_link;
    int num_links;
};
```

probe()/remove()

probe/remove are optional. Do any machine specific probe here.

suspend()/resume()

The machine driver has pre and post versions of suspend and resume to take care of any machine audio tasks that have to be done before or after the codec, DAIs and DMA is suspended and resumed. Optional.

Machine operations

The machine specific audio operations can be set here. Again this is optional.

Machine DAI Configuration

The machine DAI configuration glues all the codec and CPU DAIs together. It can also be used to set up the DAI system clock and for any machine related DAI initialisation e.g. the machine audio map can be connected to the codec audio map, unconnected codec pins can be set as such. Please see `corgi.c`, `spitz.c` for examples.

machine.txt

struct snd_soc_dai_link is used to set up each DAI in your machine. e.g.

```
/* corgi digital audio interface glue - connects codec <--> CPU */
static struct snd_soc_dai_link corgi_dai = {
    .name = "WM8731",
    .stream_name = "WM8731",
    .cpu_dai = &pxa_i2s_dai,
    .codec_dai = &wm8731_dai,
    .init = corgi_wm8731_init,
    .ops = &corgi_ops,
};
```

struct snd_soc_card then sets up the machine with its DAIs. e.g.

```
/* corgi audio machine driver */
static struct snd_soc_card snd_soc_corgi = {
    .name = "Corgi",
    .dai_link = &corgi_dai,
    .num_links = 1,
};
```

Machine Audio Subsystem

The machine soc device glues the platform, machine and codec driver together. Private data can also be set here. e.g.

```
/* corgi audio private data */
static struct wm8731_setup_data corgi_wm8731_setup = {
    .i2c_address = 0x1b,
};

/* corgi audio subsystem */
static struct snd_soc_device corgi_snd_devdata = {
    .machine = &snd_soc_corgi,
    .platform = &pxa2xx_soc_platform,
    .codec_dev = &soc_codec_dev_wm8731,
    .codec_data = &corgi_wm8731_setup,
};
```

Machine Power Map

The machine driver can optionally extend the codec power map and to become an audio power map of the audio subsystem. This allows for automatic power up/down of speaker/HP amplifiers, etc. Codec pins can be connected to the machines jack sockets in the machine init function. See soc/pxa/spitz.c and dapm.txt for details.

Machine Controls

machine.txt

Machine specific audio mixer controls can be added in the DAI init function.