Audio Codec WM8960 Porting
Notes for PianoPI Based 3.10.53

i.MX FAE     Sep. 2015
Agenda

- Audio codec driver in L3.10.53
  driver structure
  kcontrol
  audio routing
  control widget

- Porting wm8960 to PianoPI notes
  cpu dai: ssi
  hp detect: JDx pull low
  speaker power enable

- Debugging methods
  I2C read/write log
  alsamixer/tinymix
  routing path check
Audio codec driver in L3.10.53
Audio codec driver in L3.10.53 – driver structure

Platform:
- fsl-sai.c
- fsl-ssi.c
- imx-audmux.h

Machine:
- **imx-wm8960.c**
- config cpu_dai, codec_dai, control and audio routing map

Codec:
- wm8960.c

<table>
<thead>
<tr>
<th>Sound Card</th>
<th>hw_params, hw_free, set_fmt etc</th>
<th>config cpu_dai, codec_dai, control and audio routing map</th>
</tr>
</thead>
<tbody>
<tr>
<td>imx-wm8962.c</td>
<td>imx-wm8960.c</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>snd_soc_dai_ops</th>
<th>snd soc dai ops</th>
<th>hw_params, set sysclk, set_fmt etc</th>
</tr>
</thead>
<tbody>
<tr>
<td>fsl_ssi.c</td>
<td>fsl_sai.c</td>
<td>wm8962.c</td>
</tr>
</tbody>
</table>


Audio codec driver in L3.10.53 – driver structure

```
&i2c2 {
    clock_frequency = <1000000>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_i2c1_2>;
    status = "okay";
    codec: wm8960@1a {
        compatible = "wlf,wm8960";
        reg = <0x1a>;
        clocks = <&clks 201>;
        clock-names = "mclk";
        wlf,shared-lrclk;
    };
}

sound {
    compatible = "fsl,imx6-pianopi-wm8960",
               "fsl,imx-audio-wm8960";
    model = "wm8960-audio";
    cpu-dai = <&ssi2>;
    audio-codc = <&codec>;
    codec-master;
    gpr = <&gpr>;
    audio-routing =
        "Headset Jack", "HP_L",
        "Headset Jack", "HP_R",
        "Ext Spk", "SPK_LP",
        "Ext Spk", "SPK_LN",
        "Ext Spk", "SPK_RP",
        "Ext Spk", "SPK_RN";
    mux-int-port = <2>;
    mux-ext-port = <3>;
    hp-det-gpios = <&gpio3 19 1>;
    spk-en-gpio = <&gpio2 7 0>;
}
```

Codec

Power

Regulator

Machine
Audio codec driver in L3.10.53 – kcontrol

```c
static const struct snd_kcontrol_new wm8690_snd_controls[] = {
    SOC_DOUBLE_R_TLV("Capture Volume", WM8960_LINVOL, WM8960_RINVOL, 0, 63, 0, adc_tlv),
    SOC_DOUBLE_R("Capture Volume ZC Switch", WM8960_LINVOL, WM8960_RINVOL, 6, 1, 0),
    SOC_DOUBLE_R("Capture Switch", WM8960_LINVOL, WM8960_RINVOL, 7, 1, 0),
    SOC_DOUBLE_R_TLV("Playback Volume", WM8960_LDAC, WM8960_RDAC, 0, 255, 0, dac_tlv),
    SOC_DOUBLE_R_TLV("Headphone Playback Volume", WM8960_LOUT1, WM8960_ROUT1, 0, 127, 0, out_tlv),
    SOC_DOUBLE_R("Headphone Playback ZC Switch", WM8960_LOUT1, WM8960_ROUT1, 7, 1, 0),
    SOC_DOUBLE_R_TLV("Speaker Playback Volume", WM8960_LOUT2, WM8960_ROUT2, 0, 127, 0, out_tlv),
    SOC_SINGLE("Speaker DC Volume", WM8960_CLASSD3, 3, 5, 0),
    SOC_SINGLE("Speaker AC Volume", WM8960_CLASSD3, 0, 5, 0),
};
```

Kcontrol set the codec wm8960 registers via i2c interface
Audio codec driver in L3.10.53 – audio routing

Audio routing for headphone and speakers
Audio codec driver in L3.10.53 – audio routing

static const struct snd_soc_dapm_route audio_paths[] = {
    { "Left Output Mixer", "LINPUT3 Switch", "LINPUT3" },
    { "Left Output Mixer", "Boost Bypass Switch", "Left Boost Mixer" },
    { "Left Output Mixer", "PCM Playback Switch", "Left DAC" },
    { "Right Output Mixer", "RINPUT3 Switch", "RINPUT3" },
    { "Right Output Mixer", "Boost Bypass Switch", "Right Boost Mixer" },
    { "Right Output Mixer", "PCM Playback Switch", "Right DAC" },
    { "LOUT1 PGA", NULL, "Left Output Mixer" },
    { "ROUT1 PGA", NULL, "Right Output Mixer" },
    { "HP_L", NULL, "LOUT1 PGA" },
    { "HP_R", NULL, "ROUT1 PGA" },
    { "Left Speaker PGA", NULL, "Left Output Mixer" },
    { "Right Speaker PGA", NULL, "Right Output Mixer" },
    { "Left Speaker Output", NULL, "Left Speaker PGA" },
    { "Right Speaker Output", NULL, "Right Speaker PGA" },
};

Red marked is the audio path for headphone

wm8960.c
Audio codec driver in L3.10.53 – control widget

```
static const struct snd_soc_dapm_widget wm8960_dapm_widgets[] = {

...  

SND_SOC_DAPM_DAC("Left DAC", "Playback", WM8960_POWER2, 8, 0),
SND_SOC_DAPM_DAC("Right DAC", "Playback", WM8960_POWER2, 7, 0),

SND_SOC_DAPM_MIXER("Left Output Mixer", WM8960_POWER3, 3, 0,&wm8960_loutput_mixer[0],
    ARRAY_SIZE(wm8960_loutput_mixer)),
SND_SOC_DAPM_MIXER("Right Output Mixer", WM8960_POWER3, 2, 0,&wm8960_routput_mixer[0],
    ARRAY_SIZE(wm8960_routput_mixer)),

SND_SOC_DAPM_PGA("LOUT1 PGA", WM8960_POWER2, 6, 0, NULL, 0),
SND_SOC_DAPM_PGA("ROUT1 PGA", WM8960_POWER2, 5, 0, NULL, 0),

SND_SOC_DAPM_OUTPUT("HP_L"),
SND_SOC_DAPM_OUTPUT("HP_R"),
...
};
```

codec driver wm8960:  
widget definition for digital audio power control
Audio codec driver in L3.10.53 – control widget

```c
static const struct snd_soc_dapm_widget imx_wm8960_dapm_widgets[] = {
    SND_SOC_DAPM_HP("Headset Jack", NULL),
    SND_SOC_DAPM_SPK("Ext Spk", NULL),
    SND_SOC_DAPM_MIC("Hp MIC", NULL),
    SND_SOC_DAPM_MIC("Main MIC", NULL),
};
```

```c
data->card.dapm_widgets = imx_wm8960_dapm_widgets;
data->card.num_dapm_widgets = ARRAY_SIZE(imx_wm8960_dapm_widgets);
```

**machine driver imx-wm8960:**
define widgets which will be linked to codec driver power management widget in last slide
see device tree audio-routing on the right

```c
imx6qdl-pianopi.dtsi
audio-routing =
    "Headset Jack", "HP_L",
    "Headset Jack", "HP_R",
    "Ext Spk", "SPK_LP",
    "Ext Spk", "SPK_LN",
```
Porting wm8960 to PianoPI notes
Porting notes – cpu dai: ssi

i.MX6UL EVK Audio Codec CPU DAI:
SAI

PianoPI Audio Codec CPU DAI:
SSI

imx-wm8960.c is from imx6UL evk audio driver in L3.14.38-6ul-ga.
In imx6UL evk, wm8960 is connected to SAI while in PianoPI board it is SSI by AUDMUX.
Porting notes – cpu dai: ssi

```c
#include "../codecs/wm8960.h"
#include "fsl_sai.h"
#include "imx-audmux.h"

static int imx_hifi_startup(struct snd_pcm_substream *substream)
{
    struct snd_soc_pcm_runtime *rtd = substream->private_data;
    struct snd_soc_dai *codec_dai = rtd->codec_dai;
    //struct snd_soc_dai *cpu_dai = rtd->cpu_dai;
    struct snd_soc_card *card = codec_dai->codec->card;
    struct imx_wm8960_data *data = snd_soc_card_get_drvdata(card);
    bool tx = substream->stream == SNDRV_PCM_STREAM_PLAYBACK;
    // struct fsl_sai *sai = dev_get_drvdata(cpu_dai->dev);
    int ret = 0;

    data->is_stream_opened[tx] = true;
    /* if (data->is_stream_opened[tx] != sai->is_stream_opened[tx] ||
        data->is_stream_opened[!tx] != sai->is_stream_opened[!tx]) { 
        data->is_stream_opened[tx] = false;
        return -EBUSY;
    } */

    sound {
        compatible = "fsl, imx6-pianopi-wm8960",
                   "fsl,imx-audio-wm8960";
        model = "wm8960-audio";
        cpu-dai = <&ssi2>;
        audio-codec = <&codec>;
        codec-master;
        ...
    }
}
```

-imx-wm8960.c is from imx6UL evk audio driver in L3.14.38-6ul-ga.
It should remove sai related codes as left.
And set device tree as above.
Porting notes – cpu dai: ssi

Port 1, 2 & 7 connects to internal SSI, while port 3, 4, 5, 6 to external pins

The PianoPI schematic is connected to external port 3

sound {
  compatible = "fsl,imx6-pianopi-wm8960",
  "fsl,imx-audio-wm8960";
  model = "wm8960-audio";
  cpu-dai = &ssi2;
  audio-codec = &codec;
  codec-master;
  ...
}

imx6qdl-pianopi.dtsi
if (!strstr(cpu_np->name, "ssi"))
    goto audmux_bypass;

ret = of_property_read_u32(np, "mux-int-port", &int_port); …
ret = of_property_read_u32(np, "mux-ext-port", &ext_port); …

... int_port--;
ext_port--;
ret = imx_audmux_v2_configure_port(int_port,
    IMX_AUDMUX_V2_PTCR_SYN |
    IMX_AUDMUX_V2_PTCR_TFSEL(ext_port) |
    IMX_AUDMUX_V2_PTCR_TCSEL(ext_port) |
    IMX_AUDMUX_V2_PTCR_TFSDIR |
    IMX_AUDMUX_V2_PTCR_TCLKDIR,
    IMX_AUDMUX_V2_PDCR_RXDSEL(ext_port)); …
imx_audmux_v2_configure_port(ext_port,
    IMX_AUDMUX_V2_PTCR_SYN,
    IMX_AUDMUX_V2_PDCR_RXDSEL(int_port));
if (ret) {
    dev_err(&pdev->dev, "audmux external port setup failed\n");
    return ret;
}
audmux_bypass:

imx-wm8960.c should add audmux support.
The device tree ports configurations see the above.
Please note ssi should be corresponding to mux-int-port as listed in last slide:
ssi1<->1, ssi2<->2, ssi3<->7.
Porting notes – hp detect: i.MX6UL EVK

**Codec**

i.MX6UL EVK Audio Codec Headphone Detect:

HP_JD signal is connected to wm8960 JD3 pin.
Porting notes – hp detect: i.MX6UL EVK

update R48(30h) with hp-det = <3 0>;

update R24(18h) with hp-det = <3 0> ;
### Porting notes – hp detect: i.MX6UL EVK

**imx6ul-14x14-evk.dts**

```c
sound {
    compatible = "fsl,imx6ul-evk-wm8960",
        "fsl,imx-audio-wm8960";
    model = "wm8960-audio";
    cpu-dai = <&sai2>;
    audio-codec = <&codec>;
    asrc-controller = <&asrc>;
    codec-master;
    gpr = <&gpr>;
    hp-det = <3 0>;
    hp-det-gpios = <&gpio5 4 0>;
};
```

**wm8960_init():**

```c
/* Enable headphone jack detect */
snd_soc_update_bits(codec, WM8960_ADDCTL2, 1<<6, 1<<6);
snd_soc_update_bits(codec, WM8960_ADDCTL2, 1<<5, data->hp_det[1]<<5);
snd_soc_update_bits(codec, WM8960_ADDCTL4, 3<<2, data->hp_det[0]<<2);
snd_soc_update_bits(codec, WM8960_ADDCTL1, 3, 3);
```

- Update R24(18h) with `hp-det = <3 0>;
- Update R48(30h) with `hp-det = <3 0>;`
Porting notes – hp detect: PianoPI

PianoPI Audio Codec Headphone Detect:

Headphone detect signal is connected to i.MX6 GPIO3_19, while wm8960 input signals JD3 is pulled low.

So the headphone detection is done by MPU GPIO triggered event, not by wm8960 HP detection function.
Porting notes – hp detect: PianoPI

static void wm8960_init(struct snd_soc_dai *codec_dai)
{
    ...
    // snd_soc_update_bits(codec, WM8960_ADDCTL4, 7<<4, 3<<4);
    // snd_soc_update_bits(codec, WM8960_ADDCTL2, 1<<6, 1<<6);
    // snd_soc_update_bits(codec, WM8960_ADDCTL2, 1<<5, data->hp_det[1]<<5);
    // snd_soc_update_bits(codec, WM8960_ADDCTL4, 1<<2, data->hp_det[0]<<2);
    // snd_soc_update_bits(codec, WM8960_ADDCTL1, 3, 3);
    ...
}

hp_set_status_check():
...
if (hp_status != priv->hp_active_low) { //headphone plugged in
    snd_soc_dapm_enable_pin(&priv->codec->dapm, "Headset Jack");
    snd_soc_dapm_disable_pin(&priv->codec->dapm, "Ext Spk");
    ...
} else {
    snd_soc_dapm_disable_pin(&priv->codec->dapm, "Headset Jack");
    snd_soc_dapm_enable_pin(&priv->codec->dapm, "Ext Spk");
    ...
}

We should disable wm8960 headphone detection function and change dapm as left.
The device tree hp-det configuration is not supported anymore. We delete it as above.
Porting notes – speaker power enable

**PianoPI** speaker power enable:

In Documentation/devicetree/bindings/regulator/fixed-regulator.txt

- **enable-active-high**: Polarity of GPIO is Active high

If this property is missing, the default assumed is active low.

Please note modify GPIO polarity gpio = <&gpio2 7 1> has no effect.

```
reg_audio: wm8960_supply {
    compatible = "regulator-fixed";
    regulator-name = "wm8960-supply";
    gpio = <&gpio2 7 0>;
    /* enable-active-high; */
};
```
Porting notes – speaker power enable

Why modify GPIO polarity gpio = <&gpio 7 1> has no effect

```c
config->gpio = of_get_named_gpio(np, "gpio", 0);
if (of_find_property(np, "enable-active-high", NULL))
    config->enable_high = true;

if (config->gpio >= 0)
    cfg.gpio = config->gpio;
    cfg.gpio_invert = !config->enable_high;
    if (config->enabled_at_boot){
        if (config->enable_high){
            cfg.gpio_flags |= GPIOF_OUT_INIT_HIGH;
        } else {
            cfg.gpio_flags |= GPIOF_OUT_INIT_LOW;
        }
    } else {
        if (config->enable_high){
            cfg.gpio_flags |= GPIOF_OUT_INIT_LOW;
        } else {
            cfg.gpio_flags |= GPIOF_OUT_INIT_HIGH;
        }
    }
```

drivers/regulator/fixed.c

Set GPIO init voltage here, the configuration of GPIO polarity is overwritten.
Debugging methods
Debugging methods: I2C read/write log

```c
unsigned int snd_soc_read(struct snd_soc_codec *codec, unsigned int reg) {
    unsigned int ret;
    ret = codec->read(codec, reg);
    dev_dbg(codec->dev, "read %x => %x\n", reg, ret);
    trace_snd_soc_reg_read(codec, reg, ret);
    return ret;
}

unsigned int snd_soc_write(struct snd_soc_codec *codec, unsigned int reg, unsigned int val) {
    dev_dbg(codec->dev, "write %x = %x\n", reg, val);
    trace_snd_soc_reg_write(codec, reg, val);
    return codec->write(codec, reg, val);
}
```

To get I2C read/write to the audio codec wm8960, we could add printk functions in the above i2c read/write functions or enable dev_dbg in this module.
Debugging methods: I2C read/write log

The I2C read/write to wm8960 log looks like the above, which helps us to analyze how the driver configure the codec.
Debugging methods: alsamixer&tinymix

Alsamixer and tinymix are useful utilities for audio codec driver debugging. They could modify the Kcontrol properties thus change the audio codec registers. Below is the alsamixer utility, which is one of the alsa-utils project tools and the source code could be obtained in http://www.alsa-project.org/main/index.php/Download.

After modify the kcontrol properties, we could use command “alsactl store” to store the values to /var/lib/alsa/asound.state.
Debugging methods: alsamixer&tinymix

Below is the **tinymix** utility, which is the default alsalib in android.
The source code locates in android/external/tinyalsa.
Run command “mmm external/tinyalsa/” to compile the tools.

```
root@imx6qsabresd:~# tinymix
Mixer name: 'wm8960-audio'

Number of controls: 53

<table>
<thead>
<tr>
<th>ctl</th>
<th>type</th>
<th>num</th>
<th>name</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>INT</td>
<td>2</td>
<td>Capture Volume</td>
<td>63 63</td>
</tr>
<tr>
<td>1</td>
<td>INT</td>
<td>2</td>
<td>Capture Volume ZC Switch</td>
<td>0 0</td>
</tr>
<tr>
<td>2</td>
<td>BOOL</td>
<td>2</td>
<td>Capture Switch</td>
<td>On On</td>
</tr>
<tr>
<td>3</td>
<td>INT</td>
<td>1</td>
<td>Right Input Boost Mixer RINPUT3 Volume</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>INT</td>
<td>1</td>
<td>Right Input Boost Mixer RINPUT2 Volume</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>INT</td>
<td>1</td>
<td>Left Input Boost Mixer LINPUT3 Volume</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>INT</td>
<td>1</td>
<td>Left Input Boost Mixer LINPUT2 Volume</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>INT</td>
<td>2</td>
<td>Playback Volume</td>
<td>255 255</td>
</tr>
<tr>
<td>8</td>
<td>INT</td>
<td>2</td>
<td>Headphone Playback Volume</td>
<td>127 127</td>
</tr>
<tr>
<td>9</td>
<td>BOOL</td>
<td>2</td>
<td>Headphone Playback ZC Switch</td>
<td>Off Off</td>
</tr>
<tr>
<td>10</td>
<td>INT</td>
<td>2</td>
<td>Speaker Playback Volume</td>
<td>101 101</td>
</tr>
<tr>
<td>11</td>
<td>BOOL</td>
<td>2</td>
<td>Speaker Playback ZC Switch</td>
<td>Off Off</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
```
## Debugging methods: alsamixer&tinymix

```markdown
... 7  INT  2  Playback Volume       255 255
 8  INT  2  Headphone Playback Volume  127 127
 9  BOOL 2  Headphone Playback ZC Switch  Off Off
10  INT  2  Speaker Playback Volume   101 101
11  BOOL 2  Speaker Playback ZC Switch  Off Off
12  INT  1  Speaker DC Volume         0 ...
```

### List specific Kcontrol (#7) property:

```yaml
tinymix 7
```

### Modify specific Kcontrol (#7) property:

```yaml
tinymix 7 127 127
```
Debugging methods: routing path check

```
mount -t debugfs none /sys/kernel/debug
```

After run the command, we could get running system **audio routing path** as below:

```
root@imx6qsabresd:/sys/kernel/debug/asoc/wm8960-audio/wm8960.0-001a/dapm# ls
Capture            Left DAC       RINPUT1       Right Speaker Output
HP_L               Left Input Mixer RINPUT2       Right Speaker PGA
HP_R               Left Output Mixer RINPUT3       SPK_LN
LINPUT1             Left Speaker Output ROUT1 PGA       SPK_LP
LINPUT2             Left Speaker PGA  Right ADC        SPK_RN
LINPUT3             MICB              Right Boost Mixer SPK_RP
LOUT1 PGA           Mono Output Mixer Right DAC        bias_level
Left ADC            OUT3             Right Input Mixer
Left Boost Mixer    Playback          Right Output Mixer
```

```
root@imx6qsabresd:/sys/kernel/debug/asoc/wm8960-audio/wm8960.0-001a/dapm# cat Left\ DAC
Left DAC: On  in 2 out 1 - R26(0x1a) bit 8
    stream Playback inactive
    in "static" "Playback"
    in "static" "Playback"
    out "PCM Playback Switch" "Left Output Mixer"
```
Debugging methods: routing path check

root@imx6qsabresd:/sys/kernel/debug/asoc/wm8960-audio/wm8960.0-001a/dapm# cat Left\ Output\ Mixer
Left Output Mixer: On  in 2 out 1 - R47(0x2f) bit 3
  in "PCM Playback Switch" "Left DAC"
  out "static" "Left Speaker PGA"
  out "static" "LOUT1 PGA"

root@imx6qsabresd:/sys/kernel/debug/asoc/wm8960-audio/wm8960.0-001a/dapm# cat LOUT1\ PGA
LOUT1 PGA: On  in 2 out 1 - R26(0x1a) bit 6
  in "static" "Left Output Mixer"
  out "static" "HP_L"

root@imx6qsabresd:/sys/kernel/debug/asoc/wm8960-audio/wm8960.0-001a/dapm# cat HP_L
HP_L: On  in 2 out 1
  in "static" "LOUT1 PGA"
  out "static" "Headset Jack"

The routing path is:

Left DAC  -> Left Output Mixer  ->  LOUT1 PGA  ->  HP_L  ->  Headset Jack
Debugging methods: some debug commands

aplay /unit_tests/audio8k16s.wav  Play a wav file under /unit_tests directory

aplay -l  List of PLAYBACK Hardware Devices
card 0: wm8960audio [wm8960-audio], device 0: HiFi wm8960-0 []
card 1: imxhdmisoc [imx-hdmi-soc], device 0: i.MX HDMI Audio Tx hdmi-hifi-0 []

speaker-test -t sine -f 3000  Play beeps with specific frequency 3kHz

root@imx6qdsolo:~# speaker-test --help
Usage: speaker-test [OPTION]...
-D,--device playback device
-r,--rate stream rate in Hz
-c,--channels count of channels in stream
-f,--frequency sine wave frequency in Hz
-F,--format sample format
-b,--buffer ring buffer size in us
-p,--period period size in us
...