Software SDK Developer Guide for I2S

VisionFive 2

Version: 1.0

Date: 2022/11/10

Doc ID: JH7110-DGEN-008

Legal Statements

Important legal notice before reading this documentation.

Contact Us

Address: Room 502, Building 2, No. 61 Shengxia Rd., China (Shanghai) Pilot Free Trade Zone, Shanghai, 201203, China

Website: http://www.starfivetech.com

Email:

- Sales: sales@starfivetech.com
- Support: support@starfivetech.com

Preface

About this guide and technical support information.

About this document

This document mainly provides the SDK developers with the programing basics and debugging know-how for the I2S of the StarFive next generation SoC platform - JH7110.

Audience

This document mainly serves the I2S relevant driver developers. If you are developing other modules, place a request to your sales or support consultant for our complete documentation set on JH7110.

Revision History

Table 0-1 Revision History

Version	Released	Revision
1.0		First official release.

Notes and notices

The following notes and notices might appear in this guide:

- **Tip:**
  Suggests how to apply the information in a topic or step.
- **Note:**
  Explains a special case or expands on an important point.
- **Important:**
  Points out critical information concerning a topic or step.
- **CAUTION:**
  Indicates that an action or step can cause loss of data, security problems, or performance issues.
- **Warning:**
  Indicates that an action or step can result in physical harm or cause damage to hardware.

Contents

List of Tables

List of Figures

1. Introduction

Inter-Integrated Circuit for Sound (IIS) is a digital audio transmission standard defined by Philips in 1986 (revised in 1996). The standard is used for the transmission of digital audio data between internal devices of the system, such as codec, DSP, digital input/output interface, ADC, DAC and digital filter. The only relationship between I2S and I2C is that they are both defined by Philips.

I2S is a relatively simple digital interface protocol without address or device selection mechanism. On the I2S bus, only one master device and one source device can exist at a time. The master device may be a source device, or a target device, or other control devices that coordinate the source device and the target device.

1.1. Function Introduction

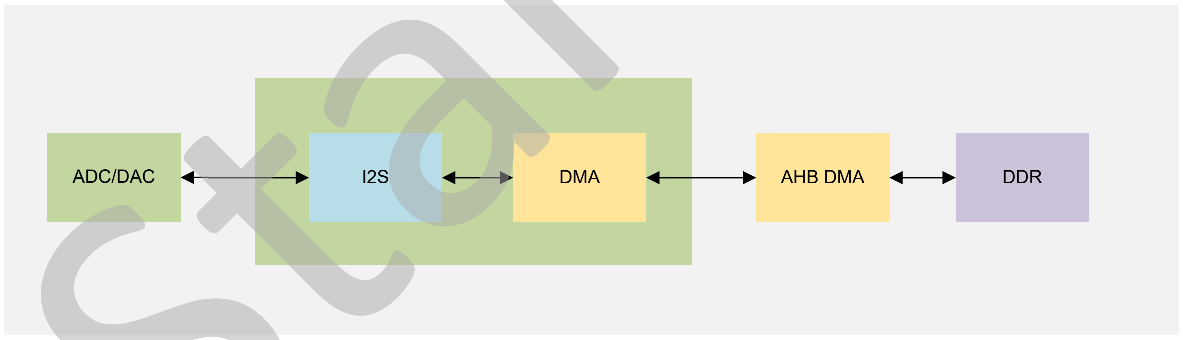
The JH7110 SoC Platform supports the following features and specifications on the audio codec interface.

- Support playback
- Support record
- Support master/slave operation modes
- Support multiple sample rate (with range 8 KHz - 48 KHz) for playback and record
- Support 16-bit/32-bit sample width
- Support up to 8 channels for both playback and record

1.2. Block Diagram

The following figure shows the block diagram of the audio codec interface.

Figure 1-1 Block Diagram



1.3. Device Tree Overview

Since Linux 3.x, device tree is introduced as a data structure and language to describe hardware configuration. It is a system-readable description of hardware settings so that the operating system doesn't have to hard code details of the machine.

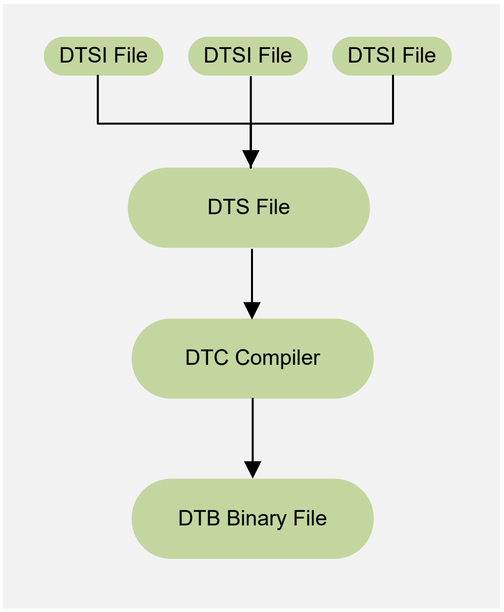
A device tree is primarily represented in the following forms.

- *Device Tree Compiler (DTC)*: The tool used to compile device tree into system-readable binaries.
- *Device Tree Source (DTS)*: The human-readable device tree description file. You can locate the target parameters and modify hardware configuration in this file.

- *Device Tree Source Information (DTSI)*: The human-readable header file which you can include in device tree description. You can locate the target parameters and modify hardware configuration in this file.
- *Device Tree Blob (DTB)*: The system-readable device tree binary blob files which is burned in system for execution.

The following diagram shows the relationship (workflow) of the above forms.

Figure 1-2 Device Tree Workflow



1.4. Source Code Structure

The following code block displays the source code structure for I2S.

```
/freelight-u-sdk/linux/sound/soc/dwc
| ├ dwc-i2s.c // I2S driver source file for JH7110
| ├ local.h  // I2S driver header file for JH7110
| ├ i2srx-master.c // The source file while I2S works as the master
| ├ i2srx-master.h // The header file while I2S works as the master
/freelight-u-sdk/linux/sound/soc/codecs
| ├ wm8960.c and wm8960.h // external codec (WM8960) driver
| ├ ac108.c, ac108.h, ac101.c, ac101_regs.h, ac10x.h  // external codec (AC108) driver
```

1.5. Device Tree Source Code

Overview Structure

The device tree source code of JH7110 is listed as follows:

```
linux
├── arch
|   ├── riscv
|   |   ├── boot
|   |   |   ├── dts
|   |   |   |   └── starfive
|   |   |   |       ├── codecs
|   |   |   |       |   ├── sf_pdm.dtsi
|   |   |   |       |   ├── sf_pwmdac.dtsi
|   |   |   |       |   ├── sf_spdif.dtsi
|   |   |   |       |   ├── sf_tdm.dtsi
|   |   |   |       |   └── sf_wm8960.dtsi
|   |   |   |       ├── evb-overlay
|   |   |   |       |   ├── jh7110-evb-overlay-can.dts
|   |   |   |       |   ├── jh7110-evb-overlay-rgb2hdmi.dts
|   |   |   |       |   ├── jh7110-evb-overlay-sdio.dts
|   |   |   |       |   ├── jh7110-evb-overlay-spi.dts
|   |   |   |       |   ├── jh7110-evb-overlay-uart4-emmc.dts
|   |   |   |       |   ├── jh7110-evb-overlay-uart5-pwm.dts
|   |   |   |       |   └── Makefile
|   |   |   |       ├── jh7110-clk.dtsi
|   |   |   |       ├── jh7110-common.dtsi
|   |   |   |       ├── jh7110.dtsi
|   |   |   |       ├── jh7110-evb-can-pdm-pwmdac.dts
|   |   |   |       ├── jh7110-evb.dts
|   |   |   |       ├── jh7110-evb.dtsi
|   |   |   |       ├── jh7110-evb-dvp-rgb2hdmi.dts
|   |   |   |       ├── jh7110-evb-pcie-i2s-sd.dts
|   |   |   |       ├── jh7110-evb-pinctrl.dtsi
|   |   |   |       ├── jh7110-evb-spi-uart2.dts
|   |   |   |       ├── jh7110-evb-uart1-rgb2hdmi.dts
|   |   |   |       ├── jh7110-evb-uart4-emmc-spdif.dts
|   |   |   |       ├── jh7110-evb-uart5-pwm-i2c-tdm.dts
|   |   |   |       ├── jh7110-fpga.dts
|   |   |   |       ├── jh7110-visionfive-v2.dts
|   |   |   |       ├── Makefile
|   |   |   |       └── vf2-overlay
|   |   |   |           ├── Makefile
|   |   |   |           └── vf2-overlay-uart3-i2c.dts
```

SoC Platform

The device tree source code of the JH7110 SoC platform is in the following path:

```
freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/jh7110.dtsi
```

VisionFive 2

The device tree source code of the VisionFive 2 *Single Board Computer (SBC)* is in the following path:

```
freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/jh7110-visionfive-v2.dts
-- freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/jh7110-common.dtsi
-- freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/jh7110.dtsi
```

2. Configuration

2.1. Device Tree Configuration

JH7110 platform stores all DMIC configuration in the device tree of the kernel. The device tree is listed as follows:

```
/freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/jh7110.dtsi
```

For example, the device tree configuration of JH7110 is as follows:

```
i2stx: i2stx@100c0000 {
compatible = "snps,designware-i2stx";
reg = <0x0 0x100c0000 0x0 0x1000>;
       interrupt-names = "tx";
       #sound-dai-cells = <0>;
       dmas = <&dma 28 1>;
       dma-names = "rx";
       status = "disabled";
};

i2srx_3ch: i2srx_3ch@100e0000 {
compatible = "snps,designware-i2srx";
       reg = <0x0 0x100e0000 0x0 0x1000>;
       clocks = <&clkgen JH7110_APB0>,
                <&clkgen JH7110_I2SRX0_3CH_CLK_APB>,
                <&clkgen JH7110_AUDIO_ROOT>,
                <&clkgen JH7110_MCLK_INNER>,
                <&clkgen JH7110_I2SRX_3CH_BCLK_MST>,
                <&clkgen JH7110_I2SRX_3CH_LRCK_MST>,
                <&clkgen JH7110_I2SRX0_3CH_BCLK>,
                <&clkgen JH7110_I2SRX0_3CH_LRCK>,
                <&clkgen JH7110_MCLK>,
                <&i2srx_bclk_ext>,
                <&i2srx_lrck_ext>;
       clock-names = "apb0", "3ch-apb",
                     "audioroot", "mclk-inner",
                     "bclk_mst", "3ch-lrck",
                     "rx-bclk", "rx-lrck",
                     "mclk", "bclk-ext",
                     "lrck-ext";
       resets = <&rstgen RSTN_U0_I2SRX_3CH_APB>,
                <&rstgen RSTN_U0_I2SRX_3CH_BCLK>;
       dmas = <&dma 24 1>;
       dma-names = "rx";
       starfive,sys-syscon = <&sys_syscon 0x18 0x34>;
       #sound-dai-cells = <0>;
       status = "disabled";
};

i2stx_4ch0: i2stx_4ch0@120b0000 {
       compatible = "snps,designware-i2stx-4ch0";
       reg = <0x0 0x120b0000 0x0 0x1000>;
       clocks = <&clkgen JH7110_MCLK_INNER>,
                <&clkgen JH7110_I2STX_4CH0_BCLK_MST>,
                <&clkgen JH7110_I2STX_4CH0_LRCK_MST>,
                <&clkgen JH7110_MCLK>,
                <&clkgen JH7110_I2STX0_4CHBCLK>,
                <&clkgen JH7110_I2STX0_4CHLRCK>;
       clock-names = "inner", "bclk-mst",
                     "lrck-mst", "mclk",
                     "bclk0", "lrck0";
       resets = <&rstgen RSTN_U0_I2STX_4CH_APB>,
                <&rstgen RSTN_U0_I2STX_4CH_BCLK>;
       dmas = <&dma 47 1>;
       dma-names = "tx";
       #sound-dai-cells = <0>;
       status = "disabled";
};
```

```
i2stx_4ch1: i2stx_4ch1@120c0000 {
compatible = "snps,designware-i2stx-4ch1";
       reg = <0x0 0x120c0000 0x0 0x1000>;
       clocks = <&clkgen JH7110_AUDIO_ROOT>,
                <&clkgen JH7110_MCLK_INNER>,
                <&clkgen JH7110_I2STX_4CH1_BCLK_MST>,
                <&clkgen JH7110_I2STX_4CH1_LRCK_MST>,
                <&clkgen JH7110_MCLK>,
                <&clkgen JH7110_I2STX1_4CHBCLK>,
                <&clkgen JH7110_I2STX1_4CHLRCK>,
                <&clkgen JH7110_MCLK_OUT>,
                <&clkgen JH7110_APB0>,
                <&clkgen JH7110_I2STX1_4CHCLK_APB>,
                <&mclk_ext>,
                <&i2stx_bclk_ext>,
                <&i2stx_lrck_ext>;
       clock-names = "audroot", "mclk_inner", "bclk_mst",
                     "lrck_mst", "mclk", "4chbclk",
                     "4chlrck", "mclk_out",
                     "apb0", "clk_apb",
                     "mclk_ext", "bclk_ext", "lrck_ext";
       resets = <&rstgen RSTN_U1_I2STX_4CH_APB>,
                <&rstgen RSTN_U1_I2STX_4CH_BCLK>;
       dmas = <&dma 48 1>;
       dma-names = "tx";
       #sound-dai-cells = <0>;
       status = "disabled";
};
```

The following list provides more information for the different nodes in the code block.

- **reg**: The basic address and the max offset of the module.
- **clocks**: The clocks used by the module. Usually, this parameter is filled with the system clock source or the clock of the module.
- **resets**: The reset items used by the module.
- **dmas**: The DMA channel used by the module.

  For example, in the code `<&dma 48 1>`, `48` indicates channel number; `1` indicates burst length is 4.
- **dma-names**: The names of the DMA channels, including "rx" for the receiver channel and "tx" for the transceiver channel.
- **status**: The status of the module.
  - **okay**: The module is enabled.
  - **disabled**: The module is disabled.
- **other**: Other parameters for sound card registration and association.

  CAUTION:
  Do not change this parameter!

2.2. VisionFive 2 Board Level Configuration

wm8960 Codec

The file `jh7110-visionfive-v2-wm8960.dts` stores the **wm8960** codec.

For example, the board-level configuration for VisionFive 2 SBC is listed in the following code block:

```
/freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/jh7110-visionfive-v2-wm8960.dts
/freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/jh7110-visionfive-v2-ac108.dts
/freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/codecs/sf_wm8960.dtsi
```

Note:
The codecs file `sf_wm8960.dtsi` is used for sound card configuration.

Use the following configuration to connect **wm8960** as the external sound card.

```
&sound{
        /* i2s + wm8960 */
        simple-audio-card,dai-link@1 {
                reg = <0>;
                status = "okay";
                format = "i2s";
                bitclock-master = <&sndcodec1>;
                frame-master = <&sndcodec1>;

                widgets =
                                "Microphone", "Mic Jack",
                                "Line", "Line In",
                                "Line", "Line Out",
                                "Speaker", "Speaker",
                                "Headphone", "Headphone Jack";
                routing =
                                "Headphone Jack", "HP_L",
                                "Headphone Jack", "HP_R",
                                "Speaker", "SPK_LP",
                                "Speaker", "SPK_LN",
                                "LINPUT1", "Mic Jack",
                                "LINPUT3", "Mic Jack",
                                "RINPUT1", "Mic Jack",
                                "RINPUT2", "Mic Jack";
                cpu0 {
                        sound-dai = <&i2srx_3ch>;
                };
                cpu1 {
                        sound-dai = <&i2stx_4ch1>;
                };

                sndcodec1:codec {
                        sound-dai = <&wm8960>;
                        clocks = <&wm8960_mclk>;
                        clock-names = "mclk";
                };
        };
};
```

ac108 Codec

The `jh7110-visionfive-v2-ac108.dts` file stores the ac108 codec.

For example, the board-level configuration for VisionFive 2 is listed in the following code block:

```
/freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/jh7110-visionfive-v2-wm8960.dts
/freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/jh7110-visionfive-v2-ac108.dts
/freelight-u-sdk/linux/arch/riscv/boot/dts/starfive/codecs/sf_ac108.dtsi
```

Note:
The codecs file `sf_ac108.dtsi` is used for sound card configuration.

Use the following configuration to connect ac108 as the external sound card.

```
&sound {
        /* i2s + ac108 */
        simple-audio-card,dai-link@0 {
                reg = <0>;
                format = "i2s";
                bitclock-master = <&sndcodec1>;
                frame-master = <&sndcodec1>;

                widgets =
                                "Microphone", "Mic Jack",
                                "Line", "Line In",
                                "Line", "Line Out",
                                "Speaker", "Speaker",
                                "Headphone", "Headphone Jack";
                routing =
                                "Headphone Jack", "HP_L",
                                "Headphone Jack", "HP_R",
                                "Speaker", "SPK_LP",
                                "Speaker", "SPK_LN",
                                "LINPUT1", "Mic Jack",
                                "LINPUT3", "Mic Jack",
                                "RINPUT1", "Mic Jack",
                                "RINPUT2", "Mic Jack";

                cpu {
                        sound-dai = <&i2srx_3ch>;
                };
                sndcodec1: codec {
                        sound-dai = <&ac108_a>;
                        clocks = <&ac108_mclk>;
                        clock-names = "mclk";
                };
        };
};
```

2.3. Enabling I2S on VisionFive 2

Follow the steps below to enable I2S in the `board.dts` file for VisionFive 2 SBC.

1. Locate the correct board level configuration file (`jh7110-visionfive-v2-wm8960.dts` and `jh7110-visionfive-v2-ac108.dts`).
2. Locate the nodes of **i2srx_3ch** and **i2stx_4ch1**.
3. Modify the **status** parameter to `okay`.

   The following code block shows an example configuration.

   ```
   &i2srx_3ch {
           status = "okay";
   };

   &i2stx_4ch1 {
           status = "okay";
   };
   ```
4. Save and exit.

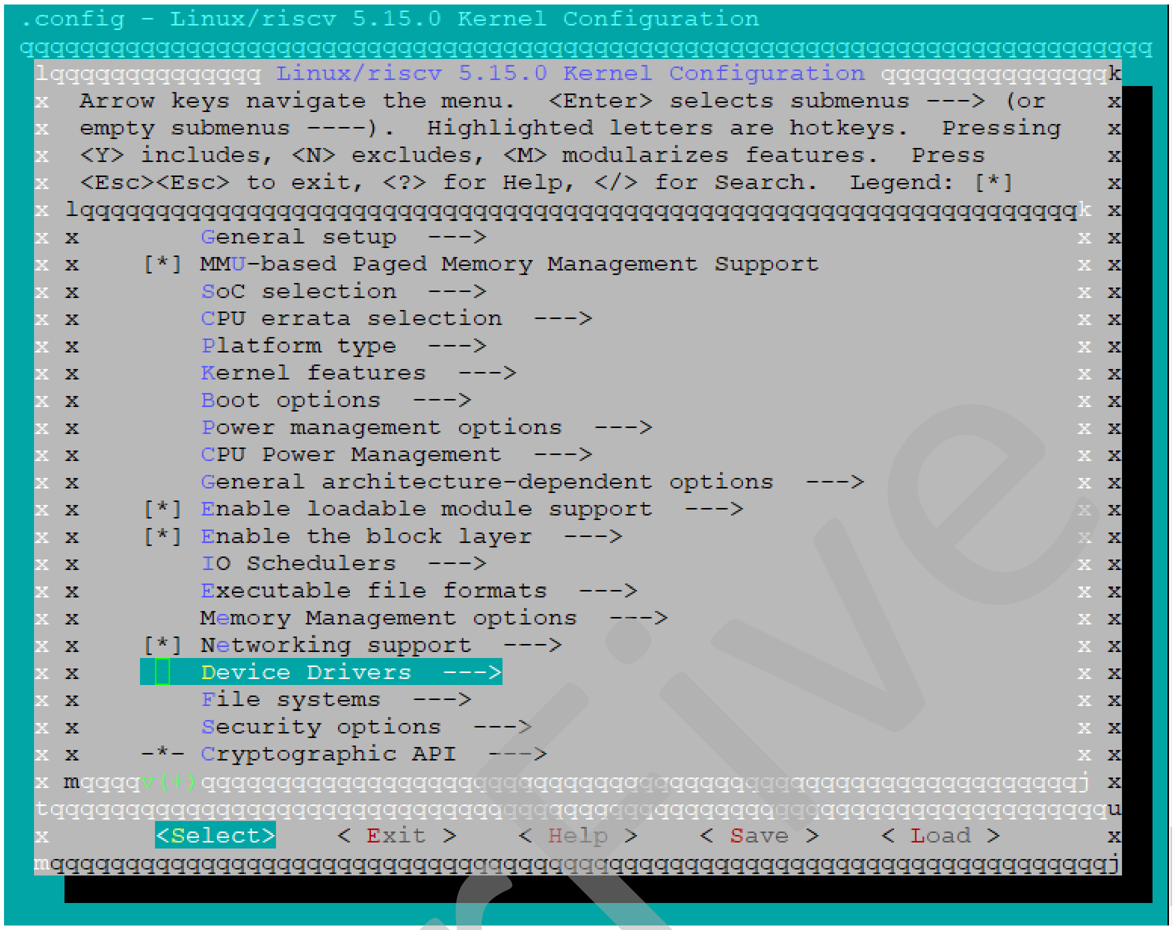
2.4. Kernel Menu Configuration

Follow the steps below to enable the kernel configuration for I2S.

1. Under the root directory of `freelight-u-sdk`, type the following command to enter the kernel menu configuration GUI.

   ```
   make linux-menuconfig
   ```
2. Enter the **Device Drivers** menu.

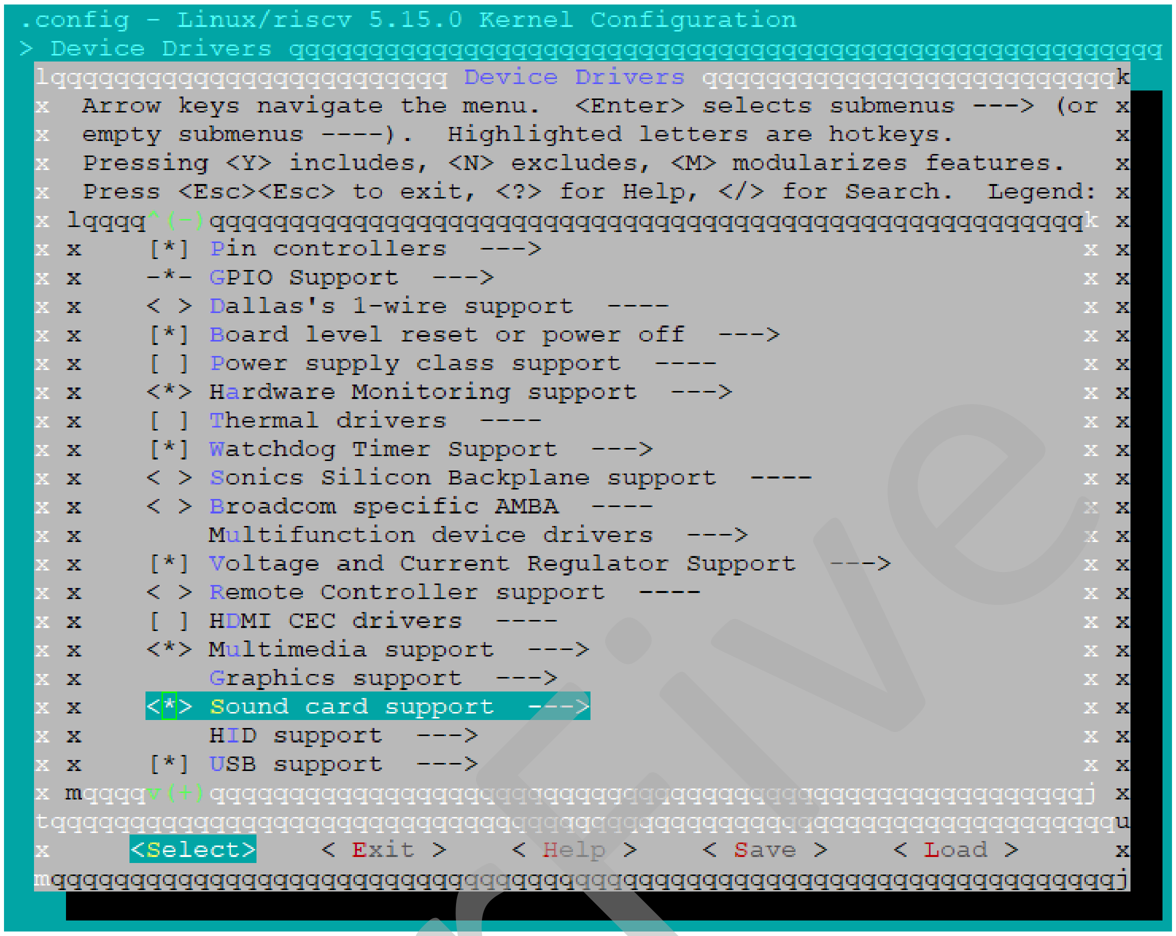
Figure 2-1 Device Drivers



```
.config - Linux/riscv 5.15.0 Kernel Configuration
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
 lqqqqqqqqqqqqqqq Linux/riscv 5.15.0 Kernel Configuration qqqqqqqqqqqqqqqk
 x  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or     x
 x  empty submenus ----).  Highlighted letters are hotkeys.  Pressing    x
 x  <Y> includes, <N> excludes, <M> modularizes features.  Press         x
 x  <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*]       x
 x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
 x x        General setup  --->                                        x x
 x x    [*] MMU-based Paged Memory Management Support                  x x
 x x        SoC selection  --->                                        x x
 x x        CPU errata selection  --->                                 x x
 x x        Platform type  --->                                        x x
 x x        Kernel features  --->                                      x x
 x x        Boot options  --->                                         x x
 x x        Power management options  --->                             x x
 x x        CPU Power Management  --->                                 x x
 x x        General architecture-dependent options  --->               x x
 x x    [*] Enable loadable module support  --->                       x x
 x x    [*] Enable the block layer  --->                               x x
 x x        IO Schedulers  --->                                        x x
 x x        Executable file formats  --->                              x x
 x x        Memory Management options  --->                            x x
 x x    [*] Networking support  --->                                   x x
 x x        Device Drivers  --->                                       x x
 x x        File systems  --->                                         x x
 x x        Security options  --->                                     x x
 x x    -*- Cryptographic API  --->                                    x x
 x mqqqqv(+)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
 tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
 x       <Select>    < Exit >    < Help >    < Save >    < Load >        x
 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

3. Select the **Sound Card Support** option and enter the next level.

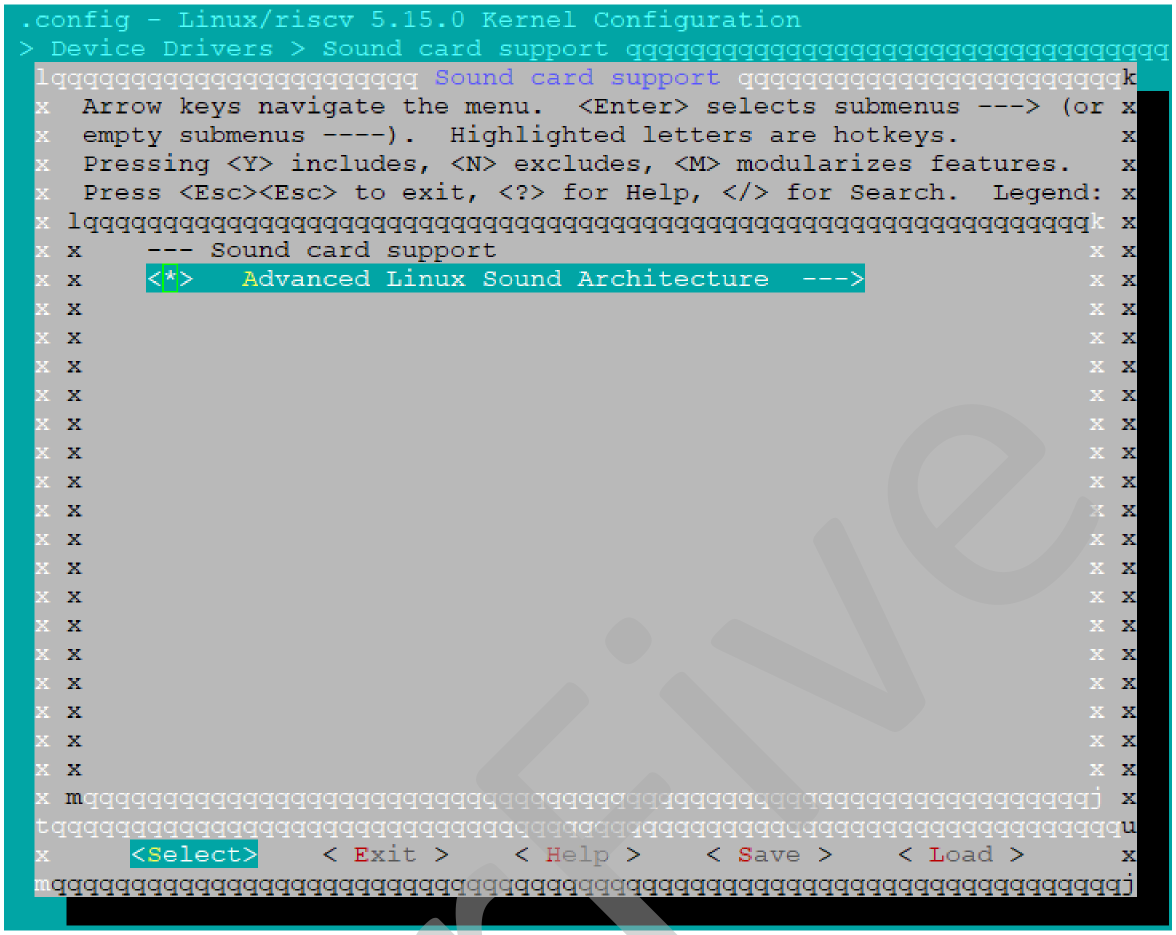
Figure 2-2 Sound Card Support



```
.config - Linux/riscv 5.15.0 Kernel Configuration
> Device Drivers qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
 lqqqqqqqqqqqqqqqqqqqqqqqqqq Device Drivers qqqqqqqqqqqqqqqqqqqqqqqqqqk
 x  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or x
 x  empty submenus ----).  Highlighted letters are hotkeys.          x
 x  Pressing <Y> includes, <N> excludes, <M> modularizes features.   x
 x  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: x
 x lqqqq^(-)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
 x x     [*] Pin controllers  --->                                 x x
 x x     -*- GPIO Support  --->                                    x x
 x x     < > Dallas's 1-wire support  ----                         x x
 x x     [*] Board level reset or power off  --->                  x x
 x x     [ ] Power supply class support  ----                      x x
 x x     <*> Hardware Monitoring support  --->                     x x
 x x     [ ] Thermal drivers  ----                                 x x
 x x     [*] Watchdog Timer Support  --->                          x x
 x x     < > Sonics Silicon Backplane support  ----                x x
 x x     < > Broadcom specific AMBA  ----                          x x
 x x         Multifunction device drivers  --->                    x x
 x x     [*] Voltage and Current Regulator Support  --->           x x
 x x     < > Remote Controller support  ----                       x x
 x x     [ ] HDMI CEC drivers  ----                                x x
 x x     <*> Multimedia support  --->                              x x
 x x         Graphics support  --->                                x x
 x x     <*> Sound card support  --->                              x x
 x x         HID support  --->                                     x x
 x x     [*] USB support  --->                                     x x
 x mqqqqv(+)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
 tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
 x     <Select>    < Exit >    < Help >    < Save >    < Load >      x
 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

4. Select the **Advanced Linux Sound Architecture** option and enter the next level.

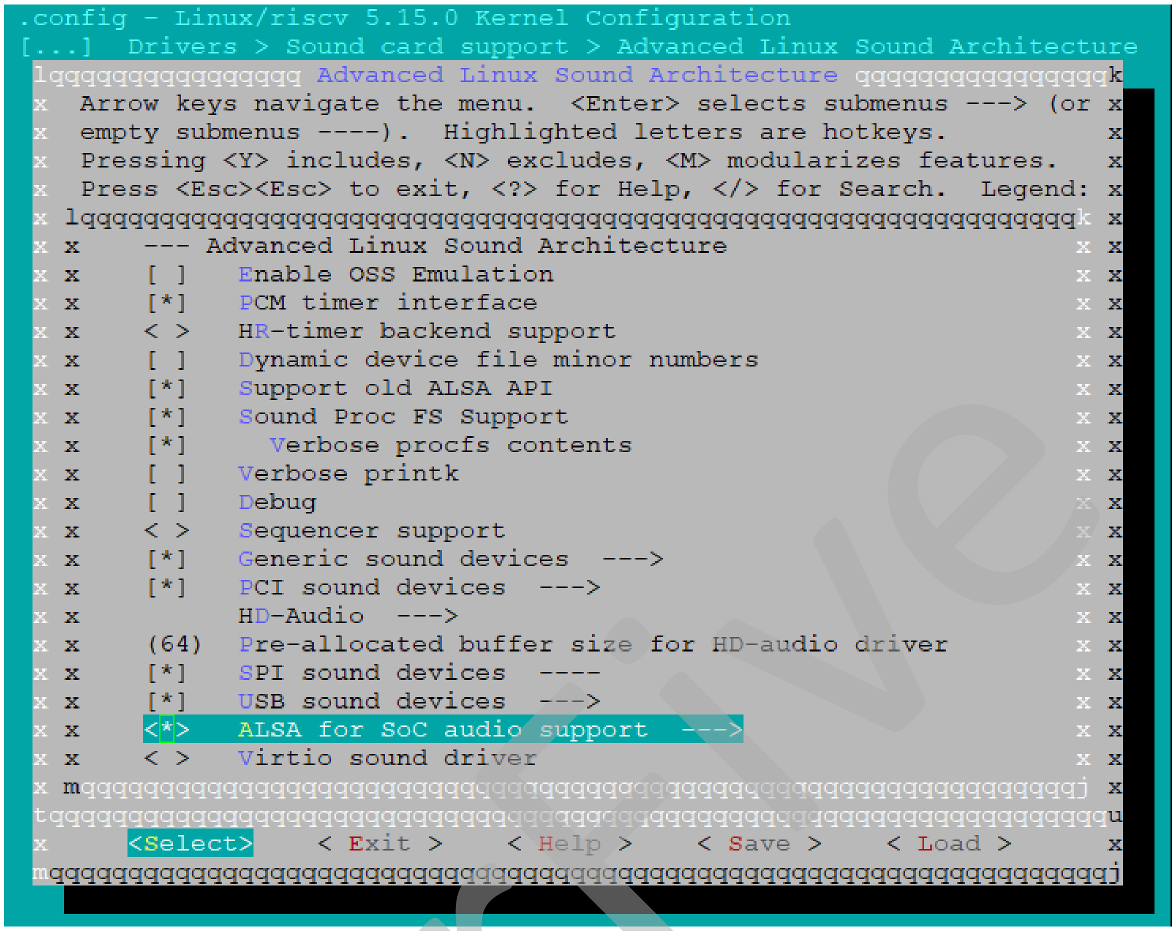
Figure 2-3 Advanced Linux Sound Architecture



```
.config - Linux/riscv 5.15.0 Kernel Configuration
> Device Drivers > Sound card support qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
 lqqqqqqqqqqqqqqqqqqqqqqqq Sound card support qqqqqqqqqqqqqqqqqqqqqqqqk
 x  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or x
 x  empty submenus ----).  Highlighted letters are hotkeys.          x
 x  Pressing <Y> includes, <N> excludes, <M> modularizes features.   x
 x  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: x
 x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
 x x     --- Sound card support                                    x x
 x x     <*>   Advanced Linux Sound Architecture  --->             x x
 x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
 tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
 x      <Select>    < Exit >    < Help >    < Save >    < Load >     x
 mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

5. Select the **ALAS for SoC audio support** option and enter the next level.

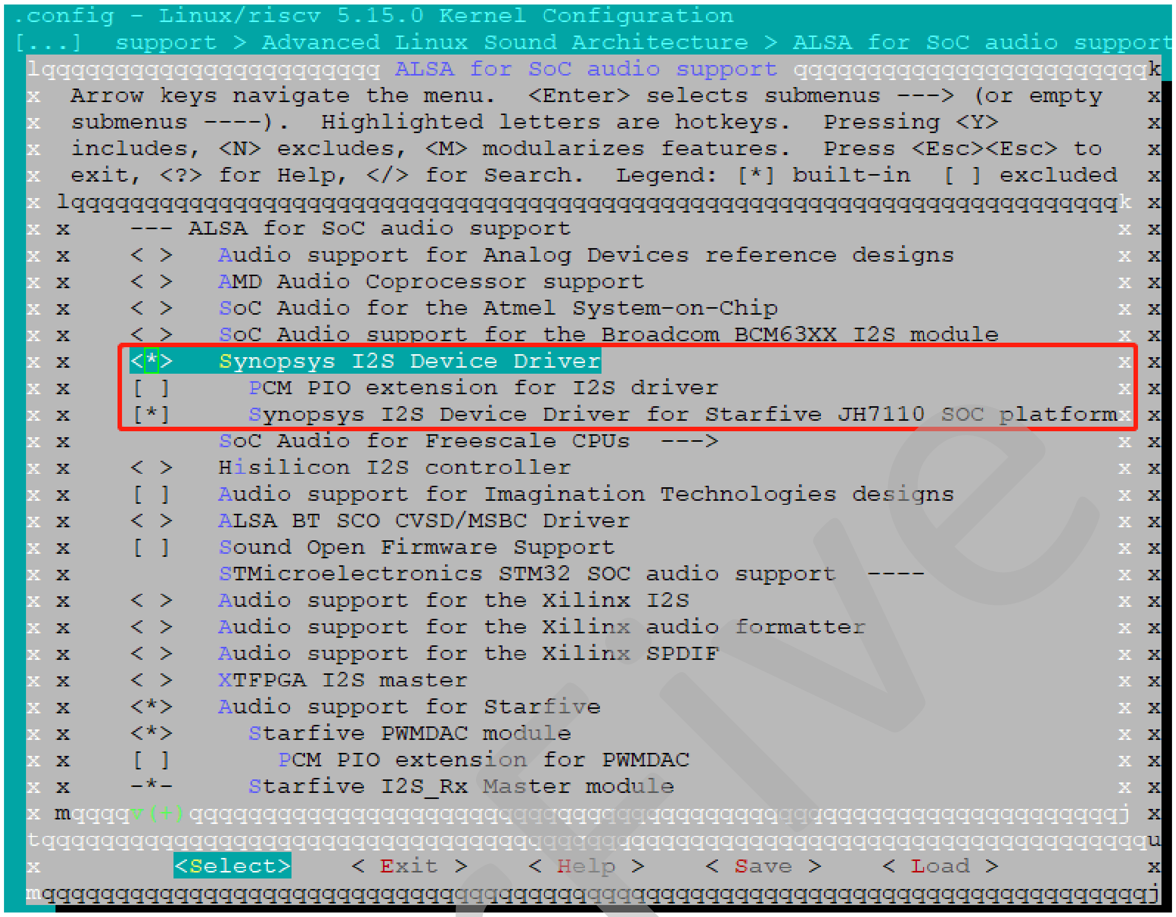
Figure 2-4 ALSA for SoC Audio Support



```
.config - Linux/riscv 5.15.0 Kernel Configuration
[...]  Drivers > Sound card support > Advanced Linux Sound Architecture
lqqqqqqqqqqqqqqqqq Advanced Linux Sound Architecture qqqqqqqqqqqqqqqqk
x  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or x
x  empty submenus ----).  Highlighted letters are hotkeys.          x
x  Pressing <Y> includes, <N> excludes, <M> modularizes features.   x
x  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x     --- Advanced Linux Sound Architecture                     x x
x x     [ ]   Enable OSS Emulation                                x x
x x     [*]   PCM timer interface                                 x x
x x     < >   HR-timer backend support                            x x
x x     [ ]   Dynamic device file minor numbers                   x x
x x     [*]   Support old ALSA API                                x x
x x     [*]   Sound Proc FS Support                               x x
x x     [*]     Verbose procfs contents                           x x
x x     [ ]   Verbose printk                                      x x
x x     [ ]   Debug                                               x x
x x     < >   Sequencer support                                   x x
x x     [*]   Generic sound devices  --->                         x x
x x     [*]   PCI sound devices  --->                             x x
x x           HD-Audio  --->                                      x x
x x     (64)  Pre-allocated buffer size for HD-audio driver       x x
x x     [*]   SPI sound devices  ----                             x x
x x     [*]   USB sound devices  --->                             x x
x x     <*>   ALSA for SoC audio support  --->                    x x
x x     < >   Virtio sound driver                                 x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
x      <Select>    < Exit >    < Help >    < Save >    < Load >     x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

6. Select the **Synopsys I2S Device Driver** option and in the next level, select the I2S driver for JH7110.

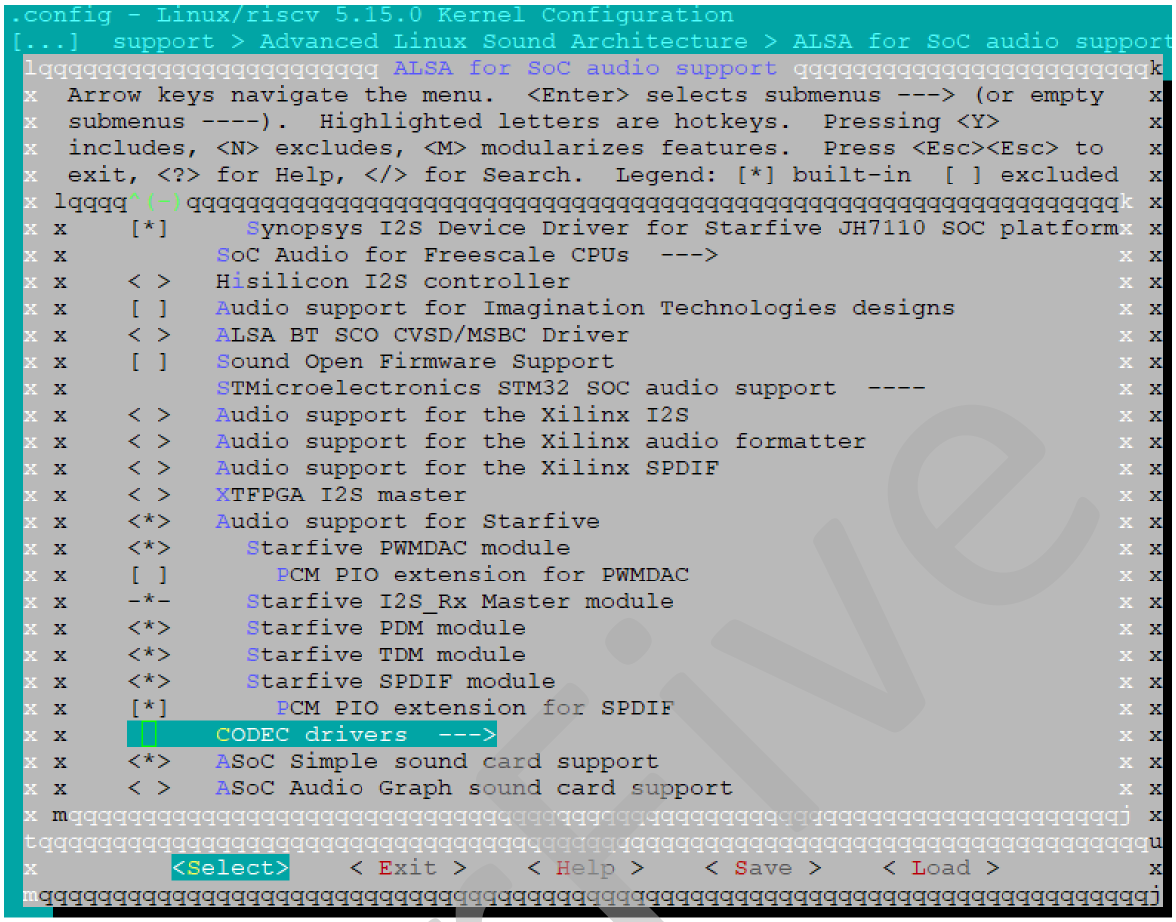
Figure 2-5 Synopsys I2S Device Drivers



```
.config - Linux/riscv 5.15.0 Kernel Configuration
[...]  support > Advanced Linux Sound Architecture > ALSA for SoC audio support
lqqqqqqqqqqqqqqqqqqqqqqqq ALSA for SoC audio support qqqqqqqqqqqqqqqqqqqqqqqqk
x  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty   x
x  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>          x
x  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to   x
x  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x     --- ALSA for SoC audio support                                    x x
x x     < >   Audio support for Analog Devices reference designs          x x
x x     < >   AMD Audio Coprocessor support                               x x
x x     < >   SoC Audio for the Atmel System-on-Chip                      x x
x x     < >   SoC Audio support for the Broadcom BCM63XX I2S module       x x
x x     <*>   Synopsys I2S Device Driver                                  x x
x x     [ ]     PCM PIO extension for I2S driver                          x x
x x     [*]     Synopsys I2S Device Driver for Starfive JH7110 SOC platformx x
x x           SoC Audio for Freescale CPUs  --->                          x x
x x     < >   Hisilicon I2S controller                                    x x
x x     [ ]   Audio support for Imagination Technologies designs          x x
x x     < >   ALSA BT SCO CVSD/MSBC Driver                                x x
x x     [ ]   Sound Open Firmware Support                                 x x
x x           STMicroelectronics STM32 SOC audio support  ----            x x
x x     < >   Audio support for the Xilinx I2S                            x x
x x     < >   Audio support for the Xilinx audio formatter                x x
x x     < >   Audio support for the Xilinx SPDIF                          x x
x x     < >   XTFPGA I2S master                                           x x
x x     <*>   Audio support for Starfive                                  x x
x x     <*>     Starfive PWMDAC module                                    x x
x x     [ ]       PCM PIO extension for PWMDAC                            x x
x x     -*-     Starfive I2S_Rx Master module                             x x
x mqqqqv(+)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
x         <Select>    < Exit >    < Help >    < Save >    < Load >         x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

7. Enter the **CODEC Drivers** menu.

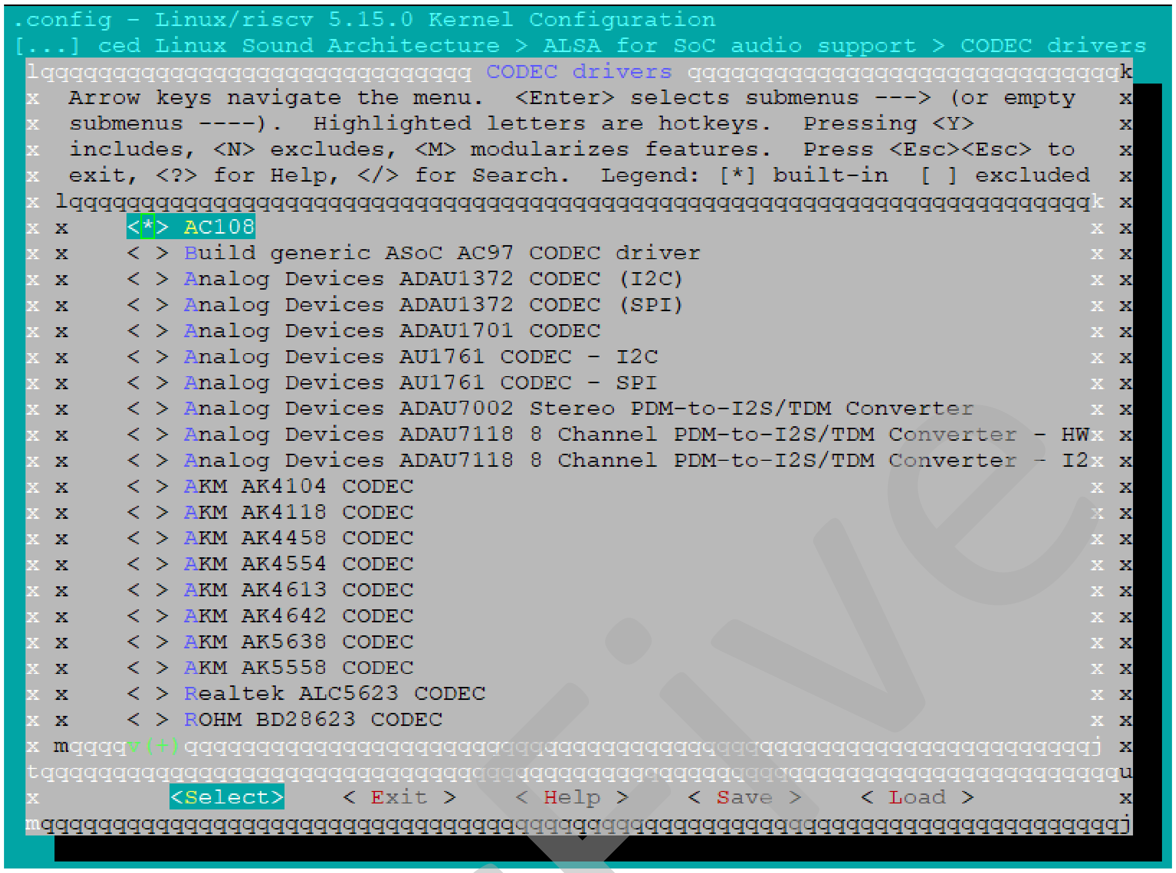
Figure 2-6 CODEC Drivers



```
.config - Linux/riscv 5.15.0 Kernel Configuration
[...]  support > Advanced Linux Sound Architecture > ALSA for SoC audio support
lqqqqqqqqqqqqqqqqqqqqqqqq ALSA for SoC audio support qqqqqqqqqqqqqqqqqqqqqqqqqk
x  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty   x
x  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>          x
x  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to   x
x  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  x
x lqqqq^(-)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x    [*]     Synopsys I2S Device Driver for Starfive JH7110 SOC platform x x
x x          SoC Audio for Freescale CPUs  --->                            x x
x x    < >   Hisilicon I2S controller                                      x x
x x    [ ]   Audio support for Imagination Technologies designs            x x
x x    < >   ALSA BT SCO CVSD/MSBC Driver                                  x x
x x    [ ]   Sound Open Firmware Support                                   x x
x x          STMicroelectronics STM32 SOC audio support  ----              x x
x x    < >   Audio support for the Xilinx I2S                              x x
x x    < >   Audio support for the Xilinx audio formatter                  x x
x x    < >   Audio support for the Xilinx SPDIF                            x x
x x    < >   XTFPGA I2S master                                             x x
x x    <*>   Audio support for Starfive                                    x x
x x    <*>     Starfive PWMDAC module                                      x x
x x    [ ]       PCM PIO extension for PWMDAC                              x x
x x    -*-     Starfive I2S_Rx Master module                               x x
x x    <*>     Starfive PDM module                                         x x
x x    <*>     Starfive TDM module                                         x x
x x    <*>     Starfive SPDIF module                                       x x
x x    [*]       PCM PIO extension for SPDIF                               x x
x x          CODEC drivers  --->                                           x x
x x    <*>   ASoC Simple sound card support                                x x
x x    < >   ASoC Audio Graph sound card support                           x x
x mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
x         <Select>    < Exit >    < Help >    < Save >    < Load >          x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

8. Select the **AC108** option on top of the list.

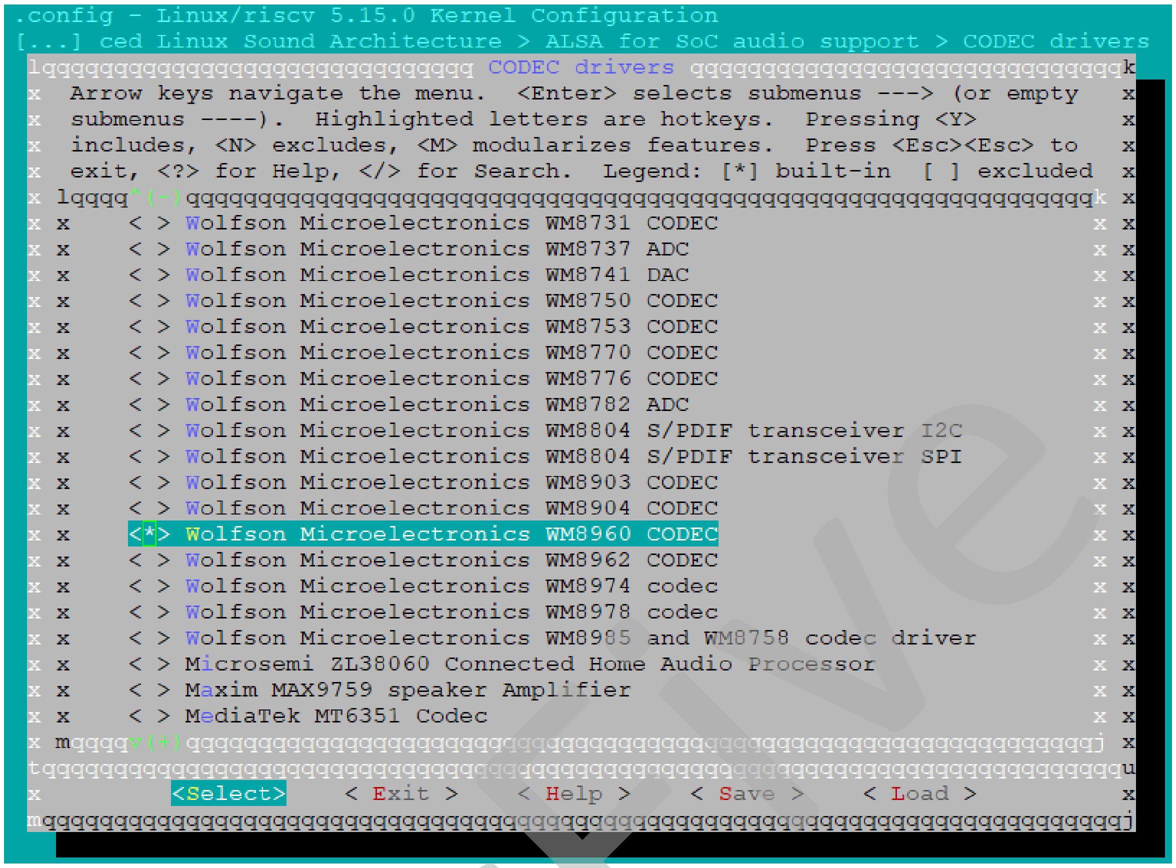
Figure 2-7 AC108



```
.config - Linux/riscv 5.15.0 Kernel Configuration
[...] ced Linux Sound Architecture > ALSA for SoC audio support > CODEC drivers
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq CODEC drivers qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty   x
x  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>          x
x  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to   x
x  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  x
x lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x     <*> AC108                                                        x x
x x     < > Build generic ASoC AC97 CODEC driver                         x x
x x     < > Analog Devices ADAU1372 CODEC (I2C)                          x x
x x     < > Analog Devices ADAU1372 CODEC (SPI)                          x x
x x     < > Analog Devices ADAU1701 CODEC                                x x
x x     < > Analog Devices AU1761 CODEC - I2C                            x x
x x     < > Analog Devices AU1761 CODEC - SPI                            x x
x x     < > Analog Devices ADAU7002 Stereo PDM-to-I2S/TDM Converter      x x
x x     < > Analog Devices ADAU7118 8 Channel PDM-to-I2S/TDM Converter - HWx x
x x     < > Analog Devices ADAU7118 8 Channel PDM-to-I2S/TDM Converter - I2x x
x x     < > AKM AK4104 CODEC                                             x x
x x     < > AKM AK4118 CODEC                                             x x
x x     < > AKM AK4458 CODEC                                             x x
x x     < > AKM AK4554 CODEC                                             x x
x x     < > AKM AK4613 CODEC                                             x x
x x     < > AKM AK4642 CODEC                                             x x
x x     < > AKM AK5638 CODEC                                             x x
x x     < > AKM AK5558 CODEC                                             x x
x x     < > Realtek ALC5623 CODEC                                        x x
x x     < > ROHM BD28623 CODEC                                           x x
x mqqqqv(+)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
x         <Select>    < Exit >    < Help >    < Save >    < Load >          x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

9. Go down and select the **WM8960 CODEC** option.

Figure 2-8 WM8960 Codec



```
.config - Linux/riscv 5.15.0 Kernel Configuration
[...] ced Linux Sound Architecture > ALSA for SoC audio support > CODEC drivers
lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq CODEC drivers qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
x  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty   x
x  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>          x
x  includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to   x
x  exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  x
x lqqqq^(-)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
x x     < > Wolfson Microelectronics WM8731 CODEC                        x x
x x     < > Wolfson Microelectronics WM8737 ADC                          x x
x x     < > Wolfson Microelectronics WM8741 DAC                          x x
x x     < > Wolfson Microelectronics WM8750 CODEC                        x x
x x     < > Wolfson Microelectronics WM8753 CODEC                        x x
x x     < > Wolfson Microelectronics WM8770 CODEC                        x x
x x     < > Wolfson Microelectronics WM8776 CODEC                        x x
x x     < > Wolfson Microelectronics WM8782 ADC                          x x
x x     < > Wolfson Microelectronics WM8804 S/PDIF transceiver I2C       x x
x x     < > Wolfson Microelectronics WM8804 S/PDIF transceiver SPI       x x
x x     < > Wolfson Microelectronics WM8903 CODEC                        x x
x x     < > Wolfson Microelectronics WM8904 CODEC                        x x
x x     <*> Wolfson Microelectronics WM8960 CODEC                        x x
x x     < > Wolfson Microelectronics WM8962 CODEC                        x x
x x     < > Wolfson Microelectronics WM8974 codec                        x x
x x     < > Wolfson Microelectronics WM8978 codec                        x x
x x     < > Wolfson Microelectronics WM8985 and WM8758 codec driver      x x
x x     < > Microsemi ZL38060 Connected Home Audio Processor             x x
x x     < > Maxim MAX9759 speaker Amplifier                              x x
x x     < > MediaTek MT6351 Codec                                        x x
x mqqqqv(+)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
x          <Select>    < Exit >    < Help >    < Save >    < Load >          x
mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

10. Save your change before you exit the kernel configuration dialog.

2.5. U-Boot Parameter Configuration

Follow the steps below to configure the U-boot parameters for Linux boot.

1. Use the following command to switch IO voltage to 3.3 V.

```
mw.l 0x1303000c 0x0 0x1
```

2. Set IP address and gateway address according to actual application environments.

   The following code block provides an example based on the test environment.

```
setenv bootfile vmlinuz;
setenv fdt_addr_r 0x48000000;
setenv fdt_high 0xffffffffffffffff;
setenv fdtcontroladdr 0xffffffffffffffff;
setenv initrd_high 0xffffffffffffffff;
setenv kernel_addr_r 0x44000000;
setenv fileaddr a0000000;
setenv ipaddr 192.168.125.211;
setenv gatewayip 192.168.125.1;
setenv serverip 192.168.125.139;
setenv kernel_comp_addr_r 0xb0000000;
setenv kernel_comp_size 0x10000000
```

3. Use the following command to boot Linux.

```
tftpboot ${fdt_addr_r} jh7110-visionfive-v2-pcie-i2s-sd.dtb;
tftpboot ${kernel_addr_r} Im-age.gz;
tftpboot ${ramdisk_addr_r} initramfs.cpio.gz;
booti ${kernel_addr_r} ${ramdisk_addr_r}:${filesize} ${fdt_addr_r};
```

3. Functional Description

3.1. View Sound Card and Device Status

Take the external codec (wm8960) as an example, the virtual sound card is described as follows.

- For virtual sound card registered in I2S interfaces:

```
# cat /proc/asound/cards
0 [StarfiveMultiSo]: simple-card - Starfive-Multi-Sound-Card
        Starfive-Multi-Sound-Card
```

- To view the device for recording:

```
# arecord -l
**** List of CAPTURE Hardware Devices ****
card 0: StarfiveMultiSo [Starfive-Multi-Sound-Card], device 0: 100e0000.i2srx_3ch-wm8960-hifi
 wm8960-hifi-0 [100e0000.i2srx_3ch-wm8960-hifi wm8960-hifi-0]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

- To view the device for playback:

```
# aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: StarfiveMultiSo [Starfive-Multi-Sound-Card], device 1: 120c0000.i2stx_4ch1-wm8960-hifi
 wm8960-hifi-1 [120c0000.i2stx_4ch1-wm8960-hifi wm8960-hifi-1]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
```

Use the following commands to check sound flow settings.

- To view playback parameters (Check during play. The example is based on checking a file of 32-bit, 48 KHz sampling rate.)

```
# aplay -Dhw:0,1 -f S32_LE -r48000 -t wav tdm_32bit_48k.wav &
# cat /proc/asound/card0/pcm1p/sub0/hw_params
access: RW_INTERLEAVED
format: S32_LE // Sample rate: 32-bit
subformat: STD
channels: 2 // Number of channels: 2 channels
rate: 48000 (48000/1) // Sample rate: 48 KHz
period_size: 512
buffer_size: 24064
```

- To view playback parameters (Check during play.)

```
# arecord -Dhw:0,0 -f S32_LE -r16000 -c2 -d 10 -t wav music_32b_16k.wav &
# cat /proc/asound/card0/pcm0c/sub0/hw_params
access: RW_INTERLEAVED
format: S32_LE // Sample rate: 32-bit
subformat: STD
channels: 2 // Number of channels: 2 channels
rate: 16000 (16000/1) // Sample rate: 16 KHz
period_size: 512
buffer_size: 8192
```

3.2. Audio Controller Configuration

The following code block provides an example of showing the sound card controller list and audio router of the audio module:

```
# cat /proc/asound/cards
0 [StarfiveMultiSo]: simple-card - Starfive-Multi-Sound-Card
      Starfive-Multi-Sound-Card
# amixer -controls
numid=12,iface=MIXER,name='Headphone Playback ZC Switch'
```

```
numid=11,iface=MIXER,name='Headphone Playback Volume'
numid=17,iface=MIXER,name='PCM Playback -6dB Switch'
numid=57,iface=MIXER,name='Mono Output Mixer Left Switch'
numid=58,iface=MIXER,name='Mono Output Mixer Right Switch'
numid=41,iface=MIXER,name='ADC Data Output Select'
numid=19,iface=MIXER,name='ADC High Pass Filter Switch'
numid=36,iface=MIXER,name='ADC PCM Capture Volume'
numid=18,iface=MIXER,name='ADC Polarity'
numid=2,iface=MIXER,name='Capture Volume ZC Switch'
numid=3,iface=MIXER,name='Capture Switch'
numid=1,iface=MIXER,name='Capture Volume'
numid=10,iface=MIXER,name='Playback Volume'
numid=23,iface=MIXER,name='3D Filter Lower Cut-Off'
numid=22,iface=MIXER,name='3D Filter Upper Cut-Off'
numid=25,iface=MIXER,name='3D Switch'
numid=24,iface=MIXER,name='3D Volume'
numid=33,iface=MIXER,name='ALC Attack'
numid=32,iface=MIXER,name='ALC Decay'
numid=26,iface=MIXER,name='ALC Function'
numid=30,iface=MIXER,name='ALC Hold Time'
numid=27,iface=MIXER,name='ALC Max Gain'
numid=29,iface=MIXER,name='ALC Min Gain'
numid=31,iface=MIXER,name='ALC Mode'
numid=28,iface=MIXER,name='ALC Target'
numid=21,iface=MIXER,name='DAC Deemphasis Switch'
numid=42,iface=MIXER,name='DAC Mono Mix'
numid=20,iface=MIXER,name='DAC Polarity'
numid=45,iface=MIXER,name='Left Boost Mixer LINPUT1 Switch'
numid=43,iface=MIXER,name='Left Boost Mixer LINPUT2 Switch'
numid=44,iface=MIXER,name='Left Boost Mixer LINPUT3 Switch'
numid=9,iface=MIXER,name='Left Input Boost Mixer LINPUT1 Volume'
numid=5,iface=MIXER,name='Left Input Boost Mixer LINPUT2 Volume'
numid=4,iface=MIXER,name='Left Input Boost Mixer LINPUT3 Volume'
numid=49,iface=MIXER,name='Left Input Mixer Boost Switch'
numid=53,iface=MIXER,name='Left Output Mixer Boost Bypass Switch'
numid=37,iface=MIXER,name='Left Output Mixer Boost Bypass Volume'
numid=52,iface=MIXER,name='Left Output Mixer LINPUT3 Switch'
numid=38,iface=MIXER,name='Left Output Mixer LINPUT3 Volume'
numid=51,iface=MIXER,name='Left Output Mixer PCM Playback Switch'
numid=35,iface=MIXER,name='Noise Gate Switch'
numid=34,iface=MIXER,name='Noise Gate Threshold'
numid=48,iface=MIXER,name='Right Boost Mixer RINPUT1 Switch'
numid=46,iface=MIXER,name='Right Boost Mixer RINPUT2 Switch'
numid=47,iface=MIXER,name='Right Boost Mixer RINPUT3 Switch'
numid=8,iface=MIXER,name='Right Input Boost Mixer RINPUT1 Volume'
numid=7,iface=MIXER,name='Right Input Boost Mixer RINPUT2 Volume'
numid=6,iface=MIXER,name='Right Input Boost Mixer RINPUT3 Volume'
numid=50,iface=MIXER,name='Right Input Mixer Boost Switch'
numid=56,iface=MIXER,name='Right Output Mixer Boost Bypass Switch'
numid=39,iface=MIXER,name='Right Output Mixer Boost Bypass Volume'
numid=54,iface=MIXER,name='Right Output Mixer PCM Playback Switch'
numid=55,iface=MIXER,name='Right Output Mixer RINPUT3 Switch'
numid=40,iface=MIXER,name='Right Output Mixer RINPUT3 Volume'
numid=16,iface=MIXER,name='Speaker AC Volume'
numid=15,iface=MIXER,name='Speaker DC Volume'
numid=13,iface=MIXER,name='Speaker Playback Volume'
numid=14,iface=MIXER,name='Speaker Playback ZC Switch'
```

3.2.1. Audio Controller Description

The following table shows the commonly used audio controller.

Table 3-1 Audio Controller Description

Index	Controller	Values	Description
10	Playback Volume	• **min**=0 • **max**=255	To set the volume value to 224: `amixer cset numid=10 224`

Table 3-1 Audio Controller Description (continued)

Index	Controller	Values	Description
			To get the volume value: `asmixer cget numid=10`
36	ADC PCM Capture Volume	• **min**=0 • **max**=255	To set the volume value to 224: `amixer cset numid=36 224` To get the volume value: `asmixer cget numid=36`

3.2.2. Audio Channel Configuration

Before a physical codec has been associated with the I2S audio card, all the registered audio cards are virtual, thus configuration is not needed. Follow the instructions of the external codec when you have the physical codec connected.

3.3. Function Validation

3.3.1. Validate Recording

Use the commands below to validate the recording function.

- To record a 10-second sound track in the sample rate of 16 KHz:

```
# arecord -Dhw:0,0 -f S32_LE -r16000 -c2 -d 10 -t wav sound_32b_16k.wav
```

- To record a 10-second sound track in the sample rate of 44.1 KHz:

```
# arecord -Dhw:0,0 -f S32_LE -r44100 -c2 -d 10 -t wav sound_32b_44100.wav
```

3.3.2. Validate Playback

Use the commands below to validate the playback function.

- To play a 10-second sound track in the sample rate of 16 KHz:

```
# aplay -Dhw:0,1 -f S32_LE -r16000 -t wav sound_32b_16k.wav
```

- To play a 10-second sound track in the sample rate of 44.1 KHz:

```
# arecord -Dhw:0,1 -f S32_LE -r44100 -t wav sound_32b_44100.wav
```

3.3.3. Validate Loopback

Loopback is essentially the test channel between TXFIFO and RXFIFO in IC. No external pin connections are required. Once loopback has been enabled, if you can write data in I2S TX, you can read data from I2S RX. The function is usually used in audio data recording scenarios.

Use the commands below to validate the loopback function.

```
# arecord -D hw:0,0 -r 16000 -c 2 -f S32_LE -t raw | aplay -D hw:0,1 -t raw -r 16000 -c 2 -f S32_LE
```

4. External Codecs

4.1. Supported External Codecs

The JH7110 I2S module supports up to 8 channels. All the 8 channels can connect to the external codecs of **wm8960** or **ac108**.

4.2. Connection Procedure

4.2.1. Prepare Hardware

Follow the steps below to prepare hardware for connecting to external codecs.

1. Make sure hardware connections are correct.
    a. Make sure all the hardware connections are correct, including I2C_SDA, I2C_SCK, I2S_MCLK, I2S_BCLK, I2S_LRCK, I2S_DIN, and I2S_DOUT, all the clock and data pins are confirmed as working properly.
    b. Make sure the I2S module of the SoC and the external codec modules are fed with the proper power supply.
2. Make sure you have checked the hardware schematics. And by checking the schematics, all the following items are correct.
    a. Confirm that the target I2S and the corresponding pins and pin multiplexing are working properly.
    b. Confirm that the I2C used for communication between the external codec and SoC is working properly.
3. Make sure you have checked the datasheet of the external codec. And by checking the datasheet, confirm all the following items are correct.
    a. Confirm the required work mode is master mode or slave mode.
    b. Confirm that the mclk, sysclk, bclk and lrclk are derived from sysclk.
    c. Confirm the supported sample rate range.
    d. Confirm the clock polarity.
    e. Confirm the data format. (For sample bit width, Left justified, Right justified, I2S, and DSP mode).
    f. Confirm the I2C address of the external codec.

4.2.2. Prepare Software

Follow the steps below to prepare software for connecting to external codecs.

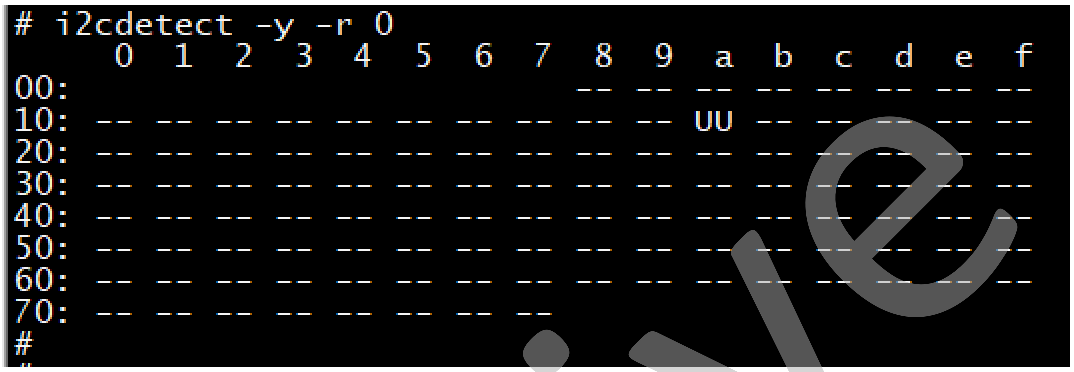
1. Make sure you have the drivers for I2S and the external codec.
2. Make sure you have enabled the driver modules in kernel.
    a. SoC I2S driver modules, including `linux-menuconfig` and `jh7110.dtsi`, `jh7110-common.dtsi`.
    b. External codec driver modules, including `linux-menuconfig` and `sf_wm8960.dtsi`, `sf_ac108.dtsi`.
3. Build and pack your program correctly.

4.2.3. Board Level Validation

Follow the steps below to validate board-level settings for connecting to external codecs.

1. Make sure OS can boot and the audio card has been registered successfully.
    a. Use the command `cat /proc/asound/cards` to check the existing audio cards.
    b. Use `aplay -l` to check the audio playback device.
    c. Use `arecord -l` to check the audio record device.
2. Make sure I2C is working properly.
    - Use the command i2cdetect to search codec device (**wm8960**'s address is 0x1a)

Figure 4-1 I2C Detect



```
# i2cdetect -y -r 0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                         -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- UU -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
#
```

Note:
UU in the above screen indicates **wm8960** is present.

    - Use the debug nodes of the external codec to read and write the registers for I2C communication debug.

```
mount -t debugfs none /mnt
```

```
cat /mnt/wm8960_reg
```

3. Verify the playback and record functions.

Usage:

Playback:

```
aplay -Dhw:0,1 -f S32_LE -r16000 -t wav sound_32b_16k.wav
```

Record:

```
arecord -Dhw:0,0 -f S32_LE -r16000 -c2 -d 10 -t wav sound_32b_16k.wav
```