1 usb

2 sensors

3 camera

4 ddr

5 display

6 power

7 pwm

8 uart

9 Audio

主要描述 Audio 相关的概念、代码结构，还有实践演示

9.1 认知

9.1.1 概念

① CPU DAI：主控端的 Audio Data Interface,比如 i2s, spdif, pdm, tdm

② CODEC DAI： codec

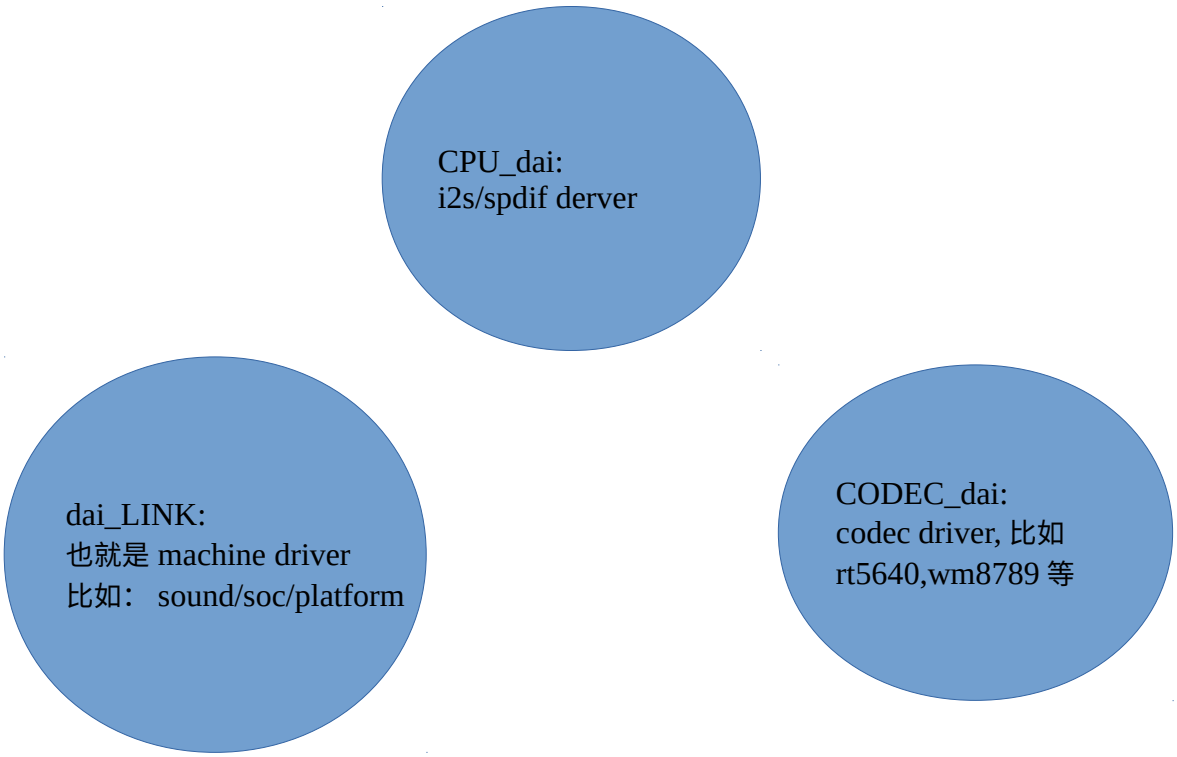
③ DAI_LINK：绑定 CPU_DAI 和 codec 为一个声卡，等同于 Machine Driver

④ DMAENGINE：用于 cpu 和 i2s spdif 等 DAI 的 DMA 传输引擎，实际是通过 DMA 来进行数据的搬运。

⑤ DAMP：动态音频电源管理，同于动态管理 codec 等电源，根据通路的开启配置开关，达到保证功能的前提下的功耗管理。

⑥ JACK：耳机接口，大部分用 codec 自身的检测机制，小部分使用 I/O 模拟。

9.1.2 数据链路



可以看到，一个声卡，实际上包括三部分组成。

9.1.2 代码结构

这里以 RK 为例子说明，其他平台也基本一致的，主要区别在 platform：

项目	功能	路径
Sound soc	主要包含公共部分代码，包括 dapm 控制， jack，dmaengine，core 等等	sound/soc/
rockchip platform	Rockchip 平台的 cpu dai 的驱动，比如 I²S, spdif 等以及自定义声卡 machine driver	sound/soc/rockchip
generic platform	simple card framework	sound/soc/generic
codec driver	所有的 codec driver 存放位置	sound/soc/codecs

9.2 codec 开发

codec 芯片的驱动一般在 2-3K 行，如果有兴趣可以自己跟着芯片手册写，在 sound/soc/codecs 下，可以找到非常多的 demo，甚至找到你当下芯片的几乎能用的驱动。

默认，直接向原厂获取驱动 codec.h && codec.c，理由如下：

① 我们写的不一定好，因为原厂的人，一个 ic 可能用了 n 个项目，已经积累了大量的经验，代码比较完善

② 时间，业务这么急，先产出，再琢磨

9.3 dts 编写

关于 dts 编写，还是那句老话，两条思路：

① 参考别人在现在的平台，已经编写好的 dts，修改为自己的节点信息

② linux/Documentation/devicetree/bindings/sound，这个路径，有你想知道的几乎任何信息

这里写个例子参考：

```
am82884f: am82884f@30 {
        compatible = "ESMT, am82884f";
        #sound-dai-cells = <0>;
        reg = <0x30>;
        status = "okay";
        reset_pin = <&portb 13 GPIO_ACTIVE_HIGH>;
};

av6210: av6210@40{
        compatible = "avnera,av6210";
        #sound-dai-cells = <0>;
        reg = <0x40>;
        status = "okay";
};
```

注意，这截取了 synaptics as371 的一份样板做演示，实际情况，你要根据 driver 来编写适当的 dts。

9.4 实践一：simple-card,rt5640

① 参考文档：kernel/Documentation/devicetree/bindings/sound/simple-card.txt

② 添加 codec driver：kernel/sound/soc/codec/rt5640.c && .h

③ 修改 Kbuild 信息：

A 在 sound/soc/codec/kconfig 中新增：

config SND_SOC_RT5640

tristate “Realtek ALC5640 CODEC”

depends on I2C

B 在同级目录中修改 Makefile：

snd-soc-rt5640-objs := rt5640.o

obj-$(CONFIG_SND_SOC_RT5640) += snd-soc-rt5640.o

④ enable simple card && codec

make menuconfig

Device Drivers --->

[*] Sound card support --->

[*] Advanced Linux Sound Architecture --->

[*] ALSA for SoC audio support --->

[*] ASoC support for Rockchip

[*] Rockchip I2S Device Driver

CODEC drivers --->

[*] Realtek ALC5640 CODEC

[*] ASoC Simple sound card support

⑤ 在 dts 中添加 simple card node:

```
rt5640-sound {
     compatible = "simple-audio-card";
     simple-audio-card,format = "i2s";
     simple-audio-card,name = "rockchip,rt5640-codec";
     simple-audio-card,mclk-fs = <256>;
     simple-audio-card,widgets =
          "Microphone", "Mic Jack",
          "Headphone", "Headphone Jack";
     simple-audio-card,routing =
          "Mic Jack", "MICBIAS1",
          "IN1P", "Mic Jack",
          "Headphone Jack", "HPOL",
```

```
                "Headphone Jack", "HPOR";
            simple-audio-card,cpu {
            sound-dai = <&i2s_8ch>;
            };
            simple-audio-card,codec {
                sound-dai = <&rt5640>;
            };
        };
&i2c1 {
        status = "okay";
        rt5640: rt5640@1c {
            #sound-dai-cells = <0>;
            compatible = "realtek,rt5640";
            reg = <0x1c>;
            clocks = <&cru SCLK_I2S_8CH_OUT>;
            /*
            注意：我们这里使用的时钟源 mclk
            upstream 代码，遵循谁用谁申请的原则，
            如果后续自己添加 codec driver，
            如果使用外部时钟作为 mclk，这里需要做适配!
            */
            clock-names = "mclk";
            realtek,in1-differential;
        };
};
```

10 HDMI

10.1 HDMI IN

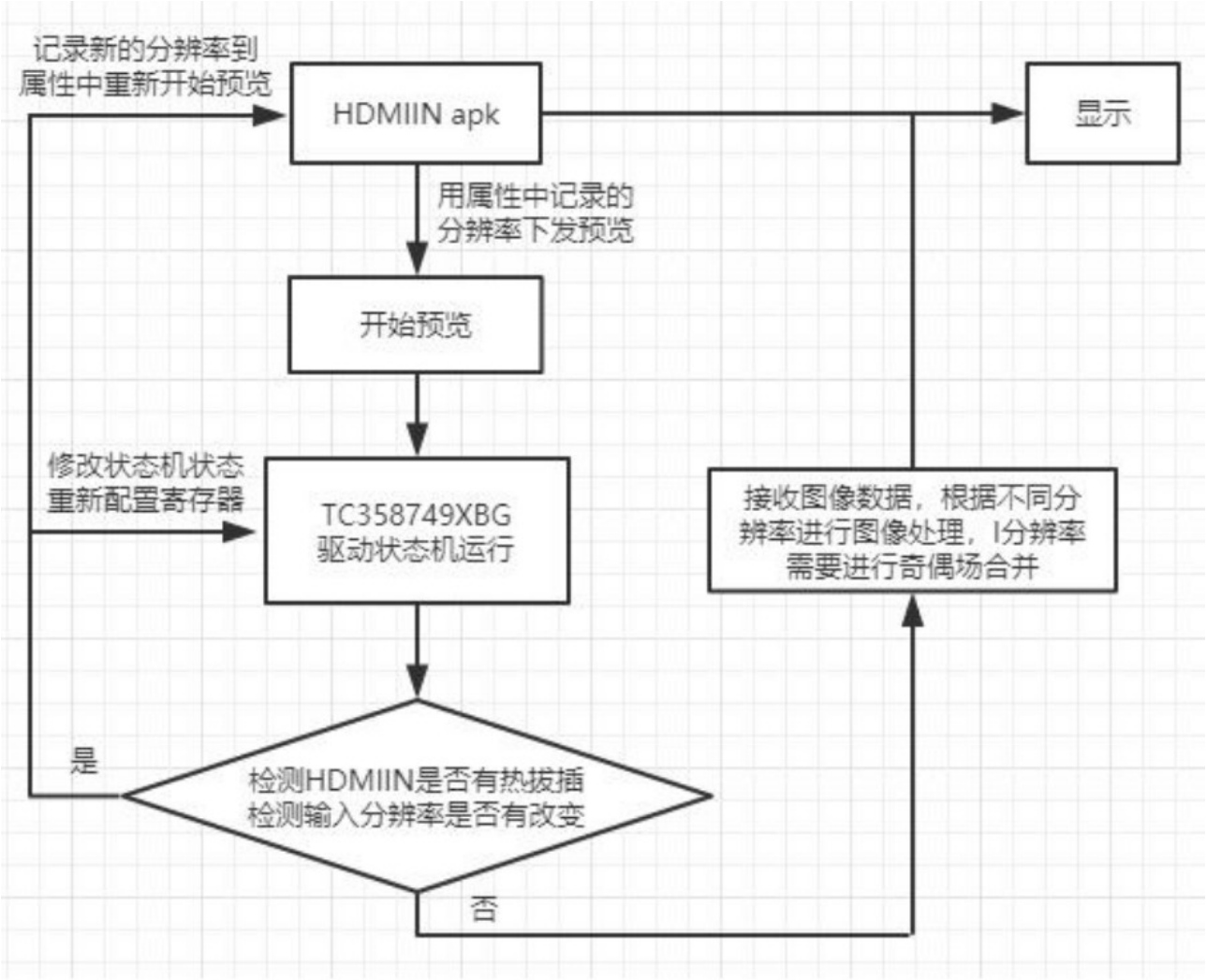
10.1.1 RockChip

10.1.1.1 概要

软硬件背景：RK3288/RK3399 基于 Linux 4.4 内核的 Android7.1/8.1 与东芝 TC358749XBG 芯片组合实现 HDMI IN 功能，支持 HDMI IN 热拔插，分辨率自适应：1080P/I、720P、480P/I、576P/I。

10.1.1.2 HDMI IN VIDEO 框架说明

HDMI IN VIDEO 的软件方案，是将 3587 这款芯片模拟成一个 MIPI 的 camera 设备，通过 camera 的框架接受 video 数据并在上层显示出来。



HDMI IN 分辨率自适应流程

HDMI IN 支持自适应分辨率:1080P/I、720P、480P/I、576P/I。camera 架构不支持动态切换预览分辨率,根据 HDMI IN 应用场景需要,在 TC358749XBG 驱动和 camera HAL 层增加 HDMI IN 分辨率自动识别、预览自动切换、不同分辨率图像区分处理的流程。在 TC358749XBG 驱动中创建线程,运行状态机,查询 HDMI SOURCE 的分辨率,并设置属性值“sys.hdmi.resolution”。HDMI IN APK 从该属性获取输入源分辨率,并根据分辨率变化,切换预览分辨率重新开始预览。在 camera HAL 中从该属性获取输入源分辨率,并对 P 和 I 分辨率的图像数据区分处理,对 I 分辨率的图像进行奇偶场合成后,再送显示。

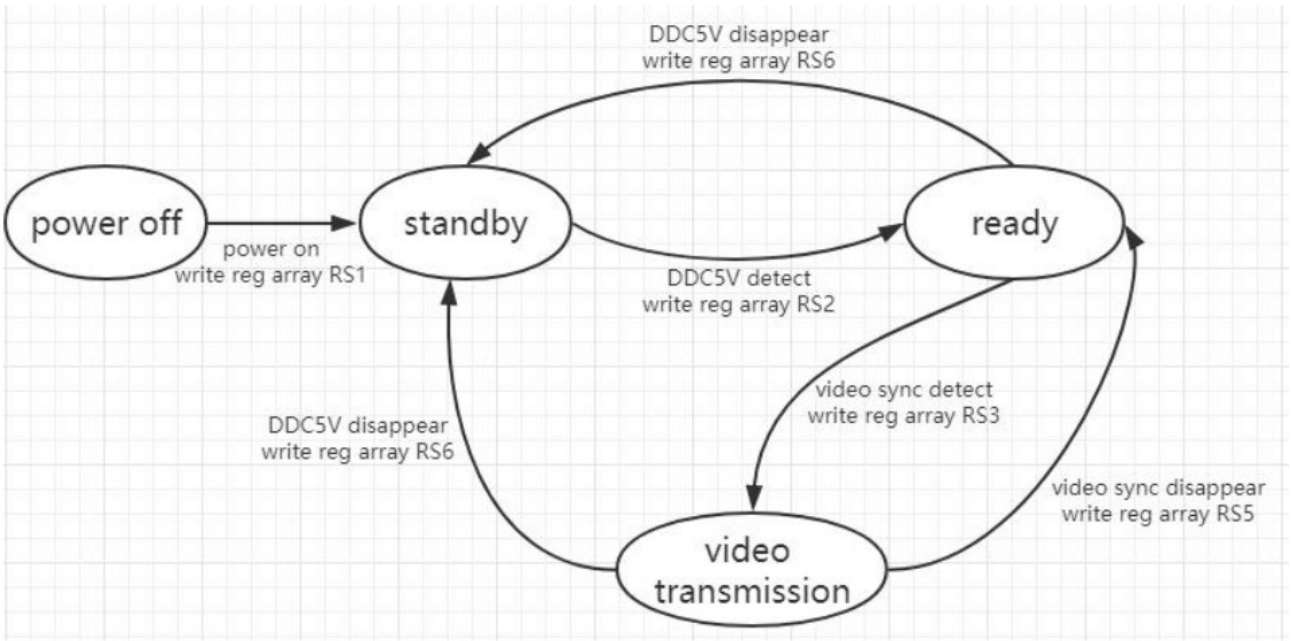
HDMI IN 支持哪些分辨率是由 TC358749XBG 配置的 EDID 和 camera 框架支持的分辨率决定的。 TC358749XBG 芯片的 EDID 使用模式可配,目前使用 Internal EDID-RAM & DDC2B mode,具体可参考 TC358749XBG 的 DATASHEET。当前 EDID-RAM 的数据是通过配置寄存

器写入,该配置通过东芝原厂提供的 excel 表格生成。不建议自行修改 EDID 的配置信息,若需要支持其他分辨率,请联系东芝原厂 FAE 及 RK FAE 咨询。

TC358749XBG 驱动状态机

在 TC358749XBG 驱动加载后,会创建一个线程,运行驱动状态机,并通过寄存器 0x8520 查询 HDMI 的 S_DDC5V 插入状态,视频同步信号 S_SYNC,通过寄存器 0x8521 查询视频分辨率 S_V_FORMAT。查询到 HDMI IN 状态变化后,切换状态机状态,并配置相关寄存器序列。

TC358749XBG 驱动状态机如下图所示:



10.1.1.3 HDMI IN VIDEO 实践

一、 首先确定原厂 SDK 的版本是否拥有 HDMI IN 的支持:

kernel/ 代码需要包含以下提交:

```
commit 05f148f30f4c95c988099a76977a56f546707215
Author: Zhong Yichong <zyc@rock-chips.com>
Date:   Fri Feb 2 09:25:38 2018 +0800

    camera: rockchip: camsys_drv: v0.0x26.0

    fix iommu resource not been released when process mediaserver
    crashes unexpectly.

    Change-Id: Ia8209f7d0a60f6a86d273e313260b87d5facecc3
    Signed-off-by: Zhong Yichong <zyc@rock-chips.com>
```

hardware/rockchip/camera/ 代码需要包含以下提交:

```
commit 0fc19969755395eac1a3daae967656f001ab4e47
Author: Dingxian Wen <shawn.wen@rock-chips.com>
Date:   Thu Feb 8 16:12:31 2018 +0800

    TC358749XBG:
    fix bug: set the property "sys.hdmiin.resolution" to
    false when TC358749XBG is released.

    Change-Id: I9b7c31eb13f23e1b923142caef9b9d0787bbd155
    Signed-off-by: Dingxian Wen <shawn.wen@rock-chips.com>
```

二、编写 DTS：

```
&i2c1 {
        status = "okay";
        clock-frequency = <400000>;

        tc358749x: tc358749x@0f {
                compatible = "toshiba,tc358749x";
                reg = <0x0f>;
                power-gpios = <&gpio7 21 GPIO_ACTIVE_HIGH>;
                stanby-gpios = <&gpio7 5 GPIO_ACTIVE_HIGH>;
                reset-gpios = <&gpio8 8 GPIO_ACTIVE_HIGH>;
                int-gpios = <&gpio8 9 GPIO_ACTIVE_HIGH>;
                pinctrl-names = "default";
                pinctrl-0 = <&hdmiin_gpios>;
                status = "okay";
        };
};
```

插注：编写 dts 有非常多的技巧，当然我们首先应该具备完善的 dts 知识，如果后续有时间，我会编写一篇 dts 的文章，在我们的实践中，我可以提供两个小技巧，当然熟悉之后，根本谈不上技巧。第一，利用好内核自带的 Documentation，里面也许能找到非常完善的 example 说明；第二，利用 git 或者 tig 工具，

针对性的查找代码中出现的关键字眼，看看别人是怎么做的某件事情的，举个例子，git log –grep=HDMI，我们也许就可以看到别人关于 HDMI 所做的事情，用法很多，灵活使用。

三、修改 hardware/rockchip/camera/Config/cam_board_rk3288.xml

修改方法有两种:

1) 修改 hardware/rockchip/camera/Config/cam_board_rk3288.xml 文件，重新编译 android,生成固件,烧写固件。

2) 修改 xml 文件后,用 adb push 到 /etc/cam_board.xml(一般在调试时使用这种方法)

根据实际硬件连接,在 cam_board.xml 文件配置对应的 I2C 和 MIPI PHY 通道:

下图中 I2C 通道配置为 1，若 TC358749XBG 连接在 I2C3，则 SensorI2cBusNum busnum ="3"。

TC358749XBG 的 MIPI 接口连接至 RK3288 的 MIPI_RX,则配置 phyIndex =“0”,若连接至 MIPI_TX/RX,则配置为 phyIndex =“1”。

```
<HardWareInfo>
    <Sensor>
        <SensorName name="TC358749XBG" ></SensorName>
        <SensorDevID IDname="CAMSYS_DEVID_SENSOR_1B"></SensorDevID>
        <SensorHostDevID busnum="CAMSYS_DEVID_MARVIN" ></SensorHostDevID>
        <SensorI2cBusNum busnum="1"></SensorI2cBusNum>          I2C通道配置
        <SensorI2cAddrByte byte="2"></SensorI2cAddrByte>
        <SensorI2cRate rate="100000"></SensorI2cRate>
        <SensorAvdd name="NC" min="0" max="0" delay="0"></SensorAvdd>
        <SensorDovdd name="NC" min="0" max="0" delay="0"></SensorDovdd>
        <SensorDvdd name="NC" min="0" max="0" delay="0"></SensorDvdd>
        <SensorMclk mclk="27000000" delay="0"></SensorMclk>
        <SensorGpioPwen ioname="NC" active="1" delay="1000"></SensorGpioPwen>
        <SensorGpioRst ioname="NC" active="0" delay="1000"></SensorGpioRst>
        <SensorGpioPwdn ioname="NC" active="0" delay="1000"></SensorGpioPwdn>
        <SensorFacing facing="back"></SensorFacing>
        <SensorInterface interface="MIPI"></SensorInterface>
        <SensorMirrorFlip mirror="0"></SensorMirrorFlip>
        <SensorOrientation orientation="0"></SensorOrientation>
        <SensorPowerupSequence seq="1234"></SensorPowerupSequence>
        <SensorFovParemeter h="60.0" v="60.0"></SensorFovParemeter>
        <SensorAWB_Frame_Skip fps="0"></SensorAWB_Frame_Skip>
        <SensorPhy phyMode="CamSys_Phy_Mipi" lane="4"  phyIndex="0" sensorFmt="CamSys_Fmt_Yuv422_8b"></SensorPhy>
    </Sensor>
```

MIPI PHY通道配置

10.1.1.4 HDMI IN AUDIO 框架说明

HDMI IN audio 硬件设计有多种方案如下:

TC358749X 通过 Codec 间接连接到主控

HDMIIn 通路 1：

HDMIIn 声音直接通过 codec 输出到喇叭、耳机,不需要送到主控进行处理。

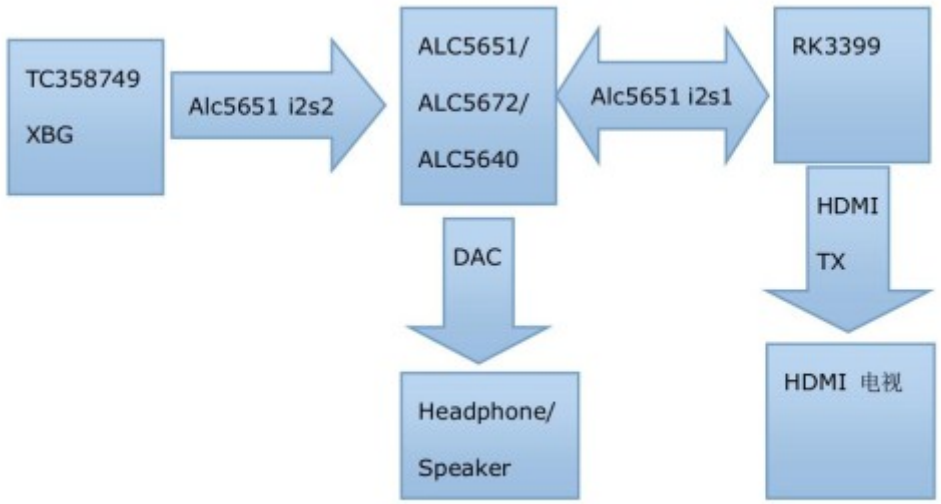
TC358749XBG-> alc5651 i2s2 -> alc5651 dac -> hp/lineout

HDMIIn 通路 2：

HDMIIn 声音通过 codec 送到主控,由主控来选择输出设备,

比如 HDMI 电视机,或是喇叭。

TC358749XBG-> alc5651 i2s2-> alc5651 i2s1-> RK3399 i2s1->RK3399 HDMI TX-> 电视



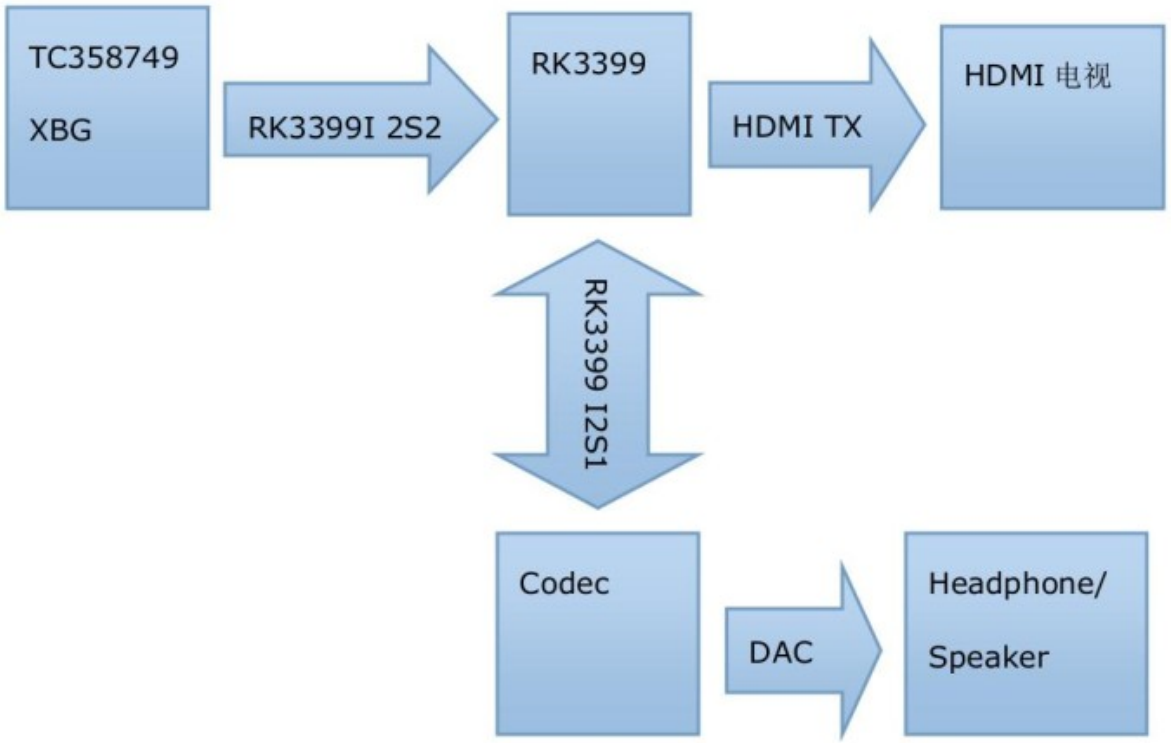
TC358749X 直接连接到主控

HDMIIn 通路 3：

HDMIIn –> RK3399 I2S2- > RK3399 HDMI TX -> HDMI 电视机

HDMIIn 通路 4：

HDMIIn –> RK3399 I2S2- > RK3399 I2S1 -> CODEC →hp/Speaker



10.1.1.5 HDMI IN AUDIO 实践

一、首先确定原厂 SDK 支持包：

kernel 代码至少应该更新到以下的提交之后:

```
commit ff5c8869774df73e7e49c982b918e335473910d9
Author: Meiyou Chen <cmy@rock-chips.com>
Date:   Fri Oct 26 15:05:15 2018 +0800

    ASoC: codecs: rockchip_rt5651_tc358749x: add HDMIIN widget for complete audio path

    Change-Id: I9750a05ffe242c5946389b2e90902f22cfdf18e8
    Signed-off-by: Meiyou Chen <cmy@rock-chips.com>
```

hardware/rockchip/audio 代码至少应该更新到以下的提交之后:

```
commit afba637425ab9a2a3695b2bf11bed013799b7181
Author: Meiyou Chen <cmy@rock-chips.com>
Date:   Tue Dec 11 15:36:15 2018 +0800

    support HDMIIn capture mode

    Change-Id: I4b421781d9e37f29017fb78e1a1833317e81f960
    Signed-off-by: Meiyou Chen <cmy@rock-chips.com>
```

frameworks/av 代码需要包含以下提交:

```
commit 23db837fe959eea4b008c2f6bc3111457027196d
Author: Meiyou Chen <cmy@rock-chips.com>
Date:   Tue Dec 11 15:26:25 2018 +0800

    audio: support HDMIIn and more output audio switch

    Change-Id: Ia424214943ae0301c1601ca6b9ade45d2c309abe
    Signed-off-by: Meiyou Chen <cmy@rock-chips.com>
```

device/rockchip/common 代码需要包含以下提交:

```
commit 9df1bb1d77ba34f0ff20c68553a2a449e5cdea50
Author: Meiyou Chen <cmy@rock-chips.com>
Date:   Fri Oct 26 15:12:34 2018 +0800

    audio: support HDMI IN device

    Change-Id: Ic8df5ef27019be85858a18abb12c2b89ad05c14a
    Signed-off-by: Meiyou Chen <cmy@rock-chips.com>
```

二、编写 DTS

if (是 HDMI IN Audio 通路 1/通路 2) {

```
hdmiin-sound {
        compatible = "rockchip,rockchip-rt5651-tc358749x-sound";
        rockchip,cpu = <&i2s0>;
        rockchip,codec = <&rt5651 &rt5651 &tc358749x>;
        status = "okay";
};
```

} else if (HDMI IN Audio 通路 3/通路 4) {

在 DTS 中为 TC358749X 添加一个 audio card,以 TC358749X 连接到 3399 I2S1 为例,先关闭掉 hdmiin-sound,再添加 tc358749x-sound,如下所示:

（Documentation/devicetree/bindings/sound/simple-card.txt 找到说明，或者直接到内核社区：https://www.kernel.org/doc/Documentation/devicetree/bindings/sound/simple-card.txt 找到最新的说明。

）

```
/ {
        ....
        hdmiin-sound {
                compatible = "rockchip,rockchip-rt5651-tc358749x-sound";
                rockchip,cpu = <&i2s0>;
                rockchip,codec = <&rt5651 &rt5651 &tc358749x>;
                status = "disabled";
        };

        tc358749x_sound: tc358749x-sound {
            status = "okay";
                compatible = "simple-audio-card";
                simple-audio-card,format = "i2s";
                simple-audio-card,name = "rk,hdmiin-tc358749x-codec";
                simple-audio-card,bitclock-master = <&sound0_master>;
                simple-audio-card,frame-master = <&sound0_master>;

                simple-audio-card,cpu {
                        sound-dai = <&i2s1>;
                };
                sound0_master: simple-audio-card,codec {
                        sound-dai = <&tc358749x>;
                };
        };
};

&i2s1 {
        status = "okay";
};
```

}

三、 配置声卡

以 HDMIIn 通路 1/2 为例,hdmiin 是接到 codec 的,获取系统中的声卡列表

```
rk3399_all:/ # cat /proc/asound/cards
 0 [realtekrt5651co]: realtekrt5651co - realtekrt5651codec_hdmiin
                      realtekrt5651codec_hdmiin
 1 [rockchiphdmi   ]: rockchip_hdmi - rockchip,hdmi
                      rockchip,hdmi
```

依据 codec 列表,修改 hardware/rockchip/audio/tinyalsa_hal/audio_hw.h

```
int PCM_CARD = 0;
int PCM_CARD_HDMI = 1;
int PCM_CARD_HDMIIN = 0;
```

10.1.2 基于 synaptics 平台实践

实例背景： GVA 专案中，根据业务需求，需要新增一个 HDMI AUDIO IN 的功能。

板级环境：

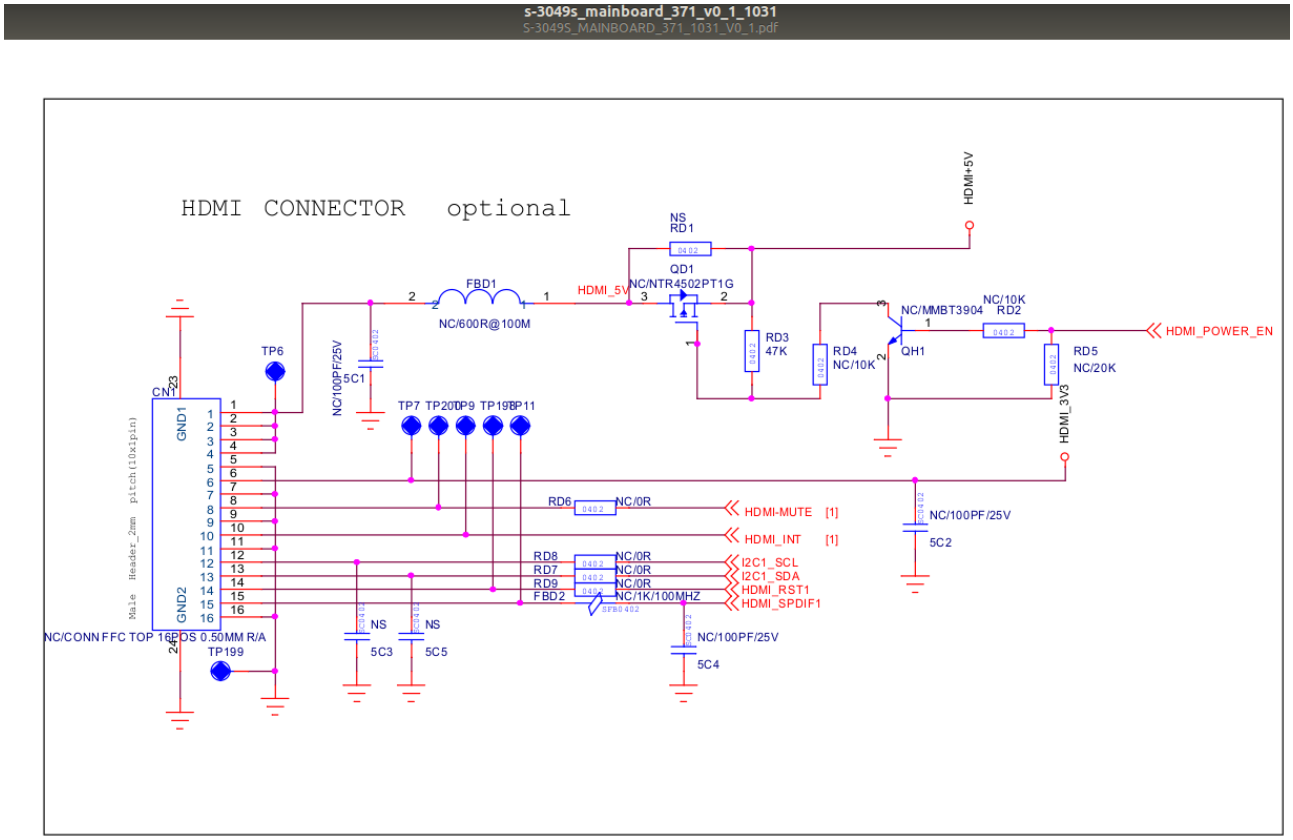
主控：as371

HDMI IN IC: ep950e

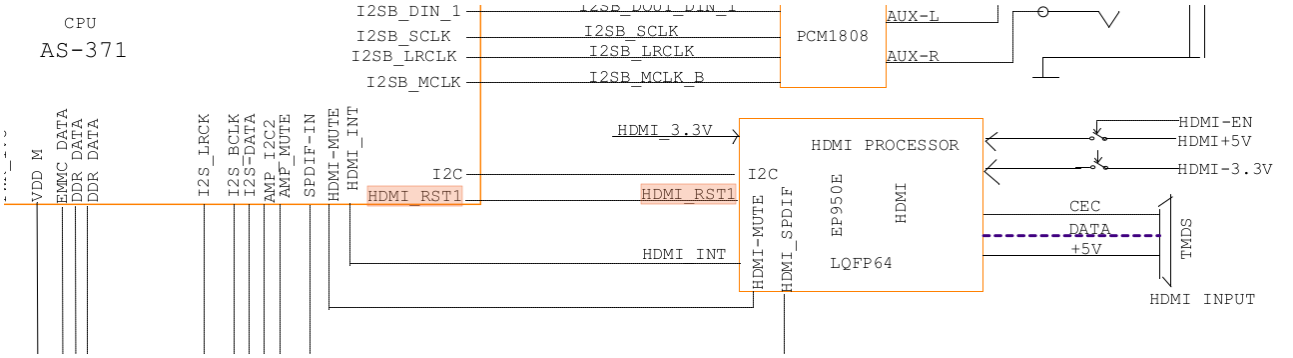
Kernel: Linux4.14

裁剪后的 Android Java1.8 环境

10.1.2.1 硬件信息



这是一个外接 FPC（柔性电路板）的插座



与 CPU 371 的框架连线

详细的管教脚线，还需要到电路图中查找具体 GPIO 数值。

10.1.2.2 通道链路

根据电路原理图接线的信息，我们可以知道这个 HDMI AUDIO IN 的链路如下：

明显，符合这章内容前面描述的第四种通道链路：

HDMI IN ——> CPU ——>codec——>功放

11 emmc

12 spi

13 i2c

14 pmic

15 pcie

16 uboot

17 security&trust

18 mobile-net

19 I/O domain-Pinctrl

20 CRU-clock

21 DVFS-CPUFreq-Info

22 Ethernet

23 ARM-context-M

24 thermal

25 Performance optimization

26 Android

26.1 Android8.1_BT

26.2 Android8.1_WIFI

26.3 预置 OEM

26.3.1 预置 oem 内容

26.3.2 预安装应用

26.4 定制开机动画

26.5 性能模式

26.6 恢复出厂设置保护

26.7 verity-boot

26.8 增加一个分区

26.9 显示框架

26.10 媒体中心

26.11 Recovery