

Porting Notes for PianoPI Based 3.10.53

i.MX FAE Oct. 2015



External Use

Freescale, the Freescale logo, AltVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, Qorivva, SafeAssure, the SafeAssure logo, StarCore, Symphony and VortiQa are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Airfast, BeeKit, BeeStack, CoreNet, Flexis, Layerscape, MagnIV, MXC, Platform in a Package, QorIQ Qonverge, QUICC Engine, Ready Play, SMARTMOS, Tower, TurboLink, UMEMS, Vybrid and Xtrinsic are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2014 Freescale Semiconductor, Inc.



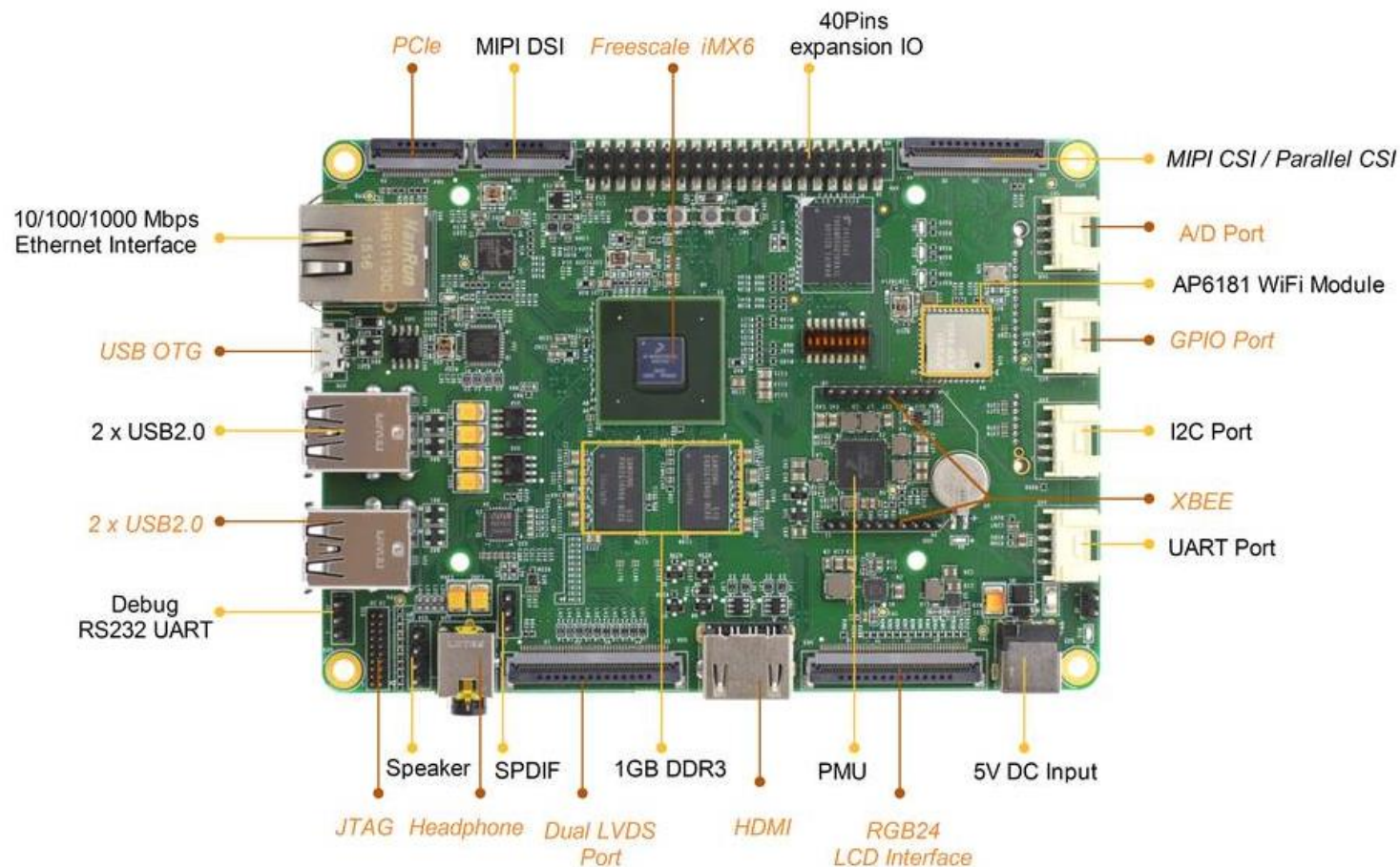
Agenda

- PianoPI board brief
- U-boot porting
- Kernel & device tree porting
 - eMMC
 - LVDS
 - touch screen
 - WM8960
 - camera
- Yocto project porting



PianoPI board brief

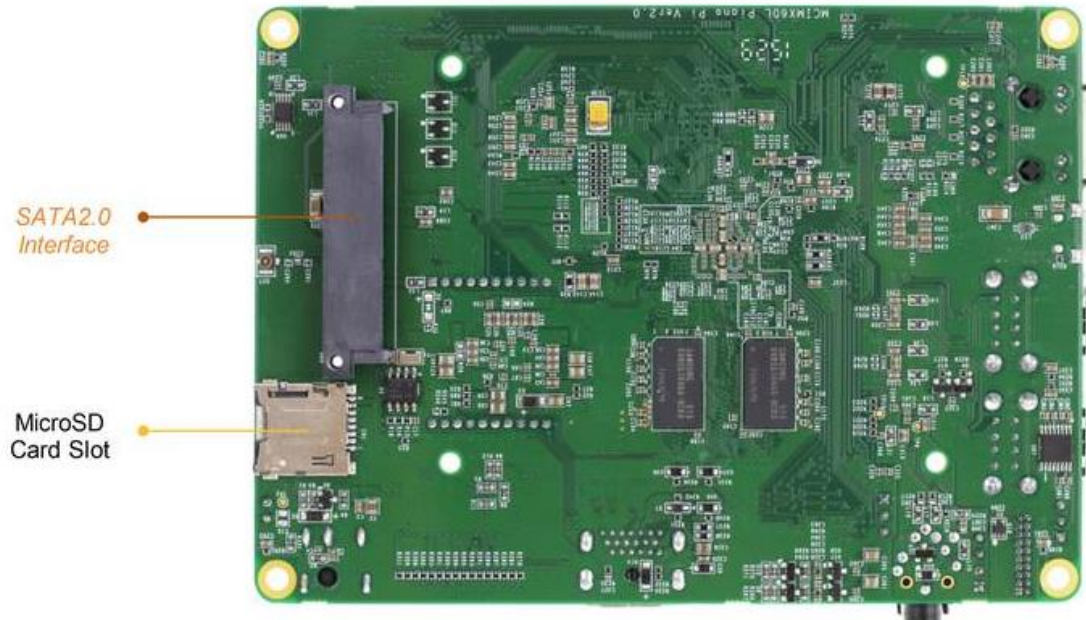
PianoPI board brief



Front side

PianoPI board is designed for IOT truck demo by freescale TIC team.
The HW is designed by Lemaker. SW developed by TIC base on BSP version L3.0.101.

PianoPI board brief



Back side

More details on PianoPI board please refer to:
<http://www.lemaker.org/article-48-1.html>



U-boot porting

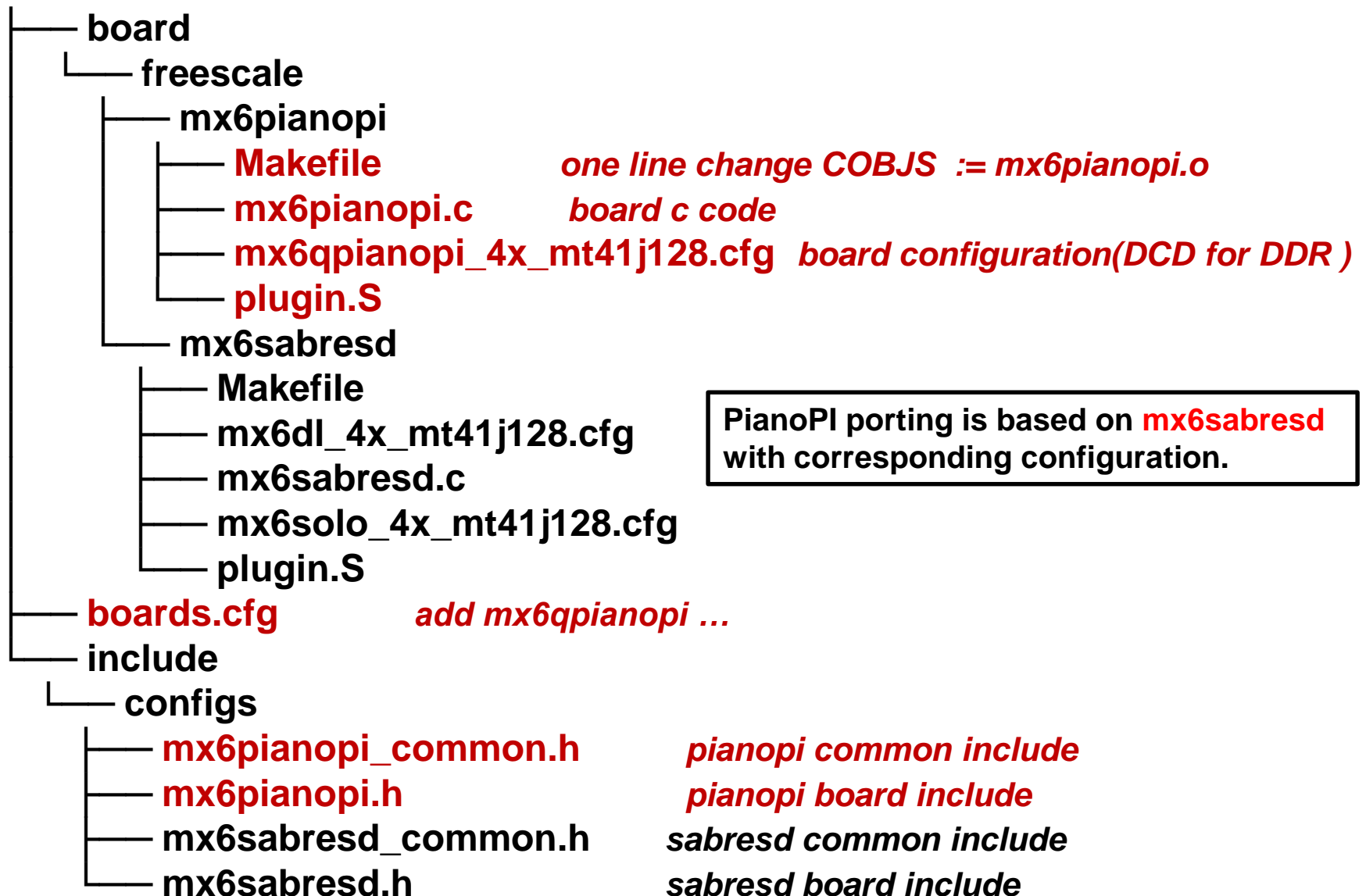
U-boot porting – boards.cfg

In L3.10.53_110, u-boot version is u-boot-2014.04.

To support pianoPI board, we add below line to boards.cfg:

```
Active arm      armv7      mx6      freescale    mx6pianopi    mx6qpianopi  
mx6pianopi:IMX_CONFIG=board/freescale/mx6pianopi/mx6q_4x_mt41j128.cfg,MX  
6Q,DEFAULT_FDT_FILE="imx6q-pianopi.dtb",DDR_MB=1024      -
```

U-boot porting – add files for PianoPI



PianoPI porting is based on **mx6sabresd** with corresponding configuration.

U-boot porting – mx6pianopi.c

```
...
iomux_v3_cfg_t const uart1_pads[] = {
//      MX6_PAD_CSI0_DAT10__UART1_TX_DATA | MUX_PAD_CTRL(UART_PAD_CTRL),
//      MX6_PAD_CSI0_DAT11__UART1_RX_DATA | MUX_PAD_CTRL(UART_PAD_CTRL),
//      MX6_PAD_SD3_DAT7__UART1_TX_DATA | MUX_PAD_CTRL(UART_PAD_CTRL),
//      MX6_PAD_SD3_DAT6__UART1_RX_DATA | MUX_PAD_CTRL(UART_PAD_CTRL),
};

iomux_v3_cfg_t const usdhc3_pads[] = {
    MX6_PAD_SD3_CLK__SD3_CLK | MUX_PAD_CTRL(USDHC_PAD_CTRL),
    MX6_PAD_SD3_CMD__SD3_CMD | MUX_PAD_CTRL(USDHC_PAD_CTRL),
    MX6_PAD_SD3_DAT0__SD3_DATA0 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
    MX6_PAD_SD3_DAT1__SD3_DATA1 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
    MX6_PAD_SD3_DAT2__SD3_DATA2 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
    MX6_PAD_SD3_DAT3__SD3_DATA3 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
//      MX6_PAD_SD3_DAT4__SD3_DATA4 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
//      MX6_PAD_SD3_DAT5__SD3_DATA5 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
//      MX6_PAD_SD3_DAT6__SD3_DATA6 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
//      MX6_PAD_SD3_DAT7__SD3_DATA7 | MUX_PAD_CTRL(USDHC_PAD_CTRL),
    MX6_PAD_NANDF_D0__GPIO2_IO00 | MUX_PAD_CTRL(NO_PAD_CTRL), /* CD */
};

int checkboard(void)
{
//      puts("Board: MX6-SabreSD\n");
//      puts("Board: MX6-PianoPI\n");
    return 0;
}
```

Base on **mx6sabre.c**, we made some modifications according to PianoPI board.





Kernel & device tree porting

Kernel & device tree porting

For kernel and device tree, we made source code modifications to drive **eMMC**, **LVDS**, **touch screen**, **audio codec** and **camera**, which will be detailed described in coming slides.

Beside these relatively significant modifications, we also have made several changes to the **device tree** due to the HW configuration difference between PianoPI board and mx6qsabresd board, such as **IOMUX**, see below an example.

```
&uart1 {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_uart1_2>;
    status = "okay";
};
```

imx6qdl-pianopi.dtsi

This is newly defined according to schematics and mx6q reference manual.

```
uart1 {
    pinctrl_uart1_1: uart1grp-1 {
        fsl,pins = <
            MX6QDL_PAD_CSI0_DAT10__UART1_TX_DATA 0x1b0b1
            MX6QDL_PAD_CSI0_DAT11__UART1_RX_DATA 0x1b0b1
        >;
    };
    pinctrl_uart1_2: uart1grp-2 {
        fsl,pins = <
            MX6QDL_PAD_SD3_DAT7__UART1_TX_DATA 0x1b0b1
            MX6QDL_PAD_SD3_DAT6__UART1_RX_DATA 0x1b0b1
        >;
    };
};
```

imx6qdl.dtsi



Kernel & device tree porting

Similar modifications include:

pinctrl_ipu1_5 , pinctrl_i2c3_3, pinctrl_pwm1_2...

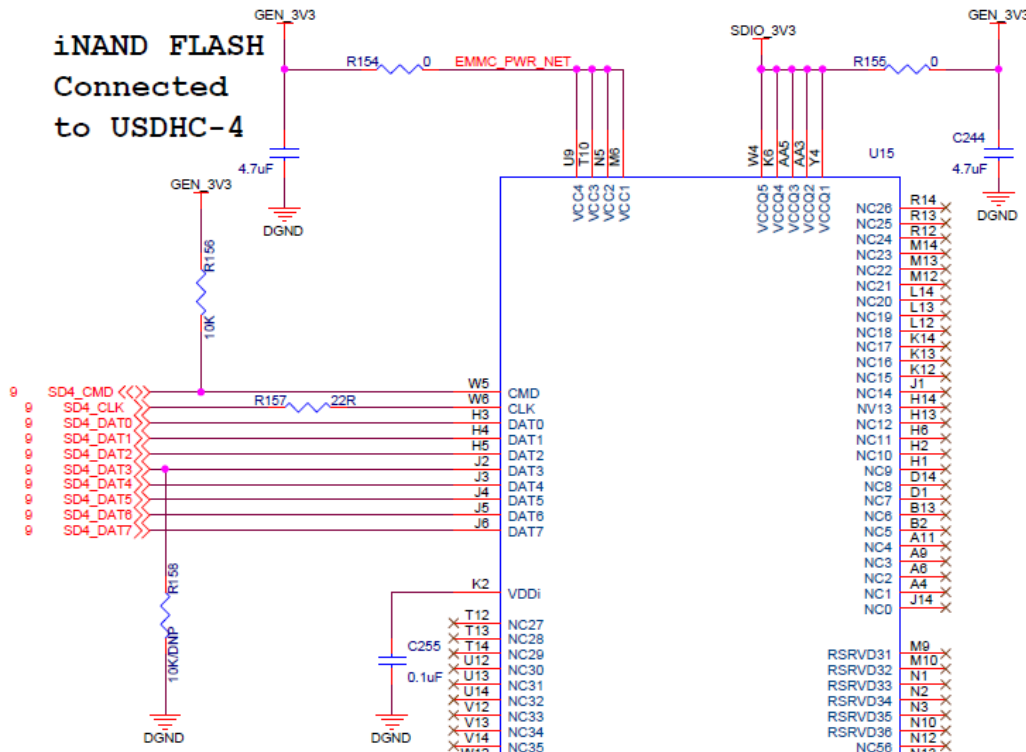
Please note some IOMUX is not existed in **imx6qdl.dtsi** and cannot be cataloged to certain component. We could place these to **hog** as below:

```
&iomuxc {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_hog_1>;

    hog {
        pinctrl_hog_1: hoggrp-1 {
            fsl,pins = <
                MX6QDL_PAD_NANDF_D4__GPIO2_IO04 0x80000000 /* TP_RST */
                MX6QDL_PAD_NANDF_D5__GPIO2_IO05 0x80000000 /* TP_INT */
                MX6QDL_PAD_NANDF_D6__GPIO2_IO06 0x80000000 /* TP_SHDN */
                MX6QDL_PAD_NANDF_D7__GPIO2_IO07 0x80000000 /* SPK_PWR_EN */
                MX6QDL_PAD_SD1_DAT0__GPIO1_IO16 0x80000000 /* CS10_PWDN */
                MX6QDL_PAD_SD1_DAT1__GPIO1_IO17 0x80000000 /*MIPI_PWDN */
                MX6QDL_PAD_SD1_DAT2__GPIO1_IO19 0x80000000 /* MIPI_RST */
                MX6QDL_PAD_SD1_CLK__GPIO1_IO20 0x80000000 /* CS10_RST */
            >
        }
    }
    ...
}
```

imx6qdl-pianopi.dtsi

Kernel & device tree porting – eMMC (1)



```

imx6qdl-piano.dtsi
&usdhc4 {
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_usdhc4_1>;
    bus-width = <8>;
    non-removable;
    no-1-8-v;
    keep-power-in-suspend;
    status = "okay";
};

```

eMMC on PianoPI board HW connection is the same as on mx6q sabresd. But we met a problem during L3.10.53 PianoPI porting. The mfgtool kernel cannot get `/dev/mmcblk3`, which is the emmc (SD4). See next slide the fail log.

Kernel & device tree porting – eMMC (2)

```
...
UTP: received command '$ sh mksdcard.sh /dev/mmcblk3'
UTP: executing "sh mksdcard.sh /dev/mmcblk3"
1+0 records in
1+0 records out
1024 bytes (1.0 kB) copied, 0.00686533 s, 149 kB/s
sfdisk: Warning: /dev/mmcblk3 is not a block device
sfdisk: Disk /dev/mmcblk3: cannot get geometry

Disk /dev/mmcblk3: 0 cylinders, 255 heads, 63 sectors/track
sfdisk: /dev/mmcblk3: unrecognized partition table type
Old situation:
sfdisk: No partitions found
sfdisk: Warning: given size (503) exceeds max allowable size (0)
New situation:
Units: 1MiB = 1024*1024 bytes, blocks of 1024 bytes, counting from 0

   Device Boot Start    End    MiB    #blocks    Id  System
/dev/mmcblk3p1          7+   509-   503-   514080    c  W95 FAT32 (LBA)
/dev/mmcblk3p2       596+   596-     0         0    83  Linux
/dev/mmcblk3p3         0         -     0         0     0  Empty
/dev/mmcblk3p4         0         -     0         0     0  Empty
sfdisk: Warning: partition 2 has size 0 but is not marked Empty
sfdisk: Warning: partition 1 extends past end of disk
Successfully wrote the new partition table

sfdisk: If you created or changed a DOS partition, /dev/foo7, say, then use dd(1)
to zero the first 512 bytes: dd if=/dev/zero of=/dev/foo7 bs=512 count=1
(See fdisk(8).)
UTP: sending Success to kernel for command $ sh mksdcard.sh /dev/mmcblk3.
utp_poll: pass returned.
UTP: received command '$ dd if=/dev/zero of=/dev/mmcblk3 bs=1k seek=384 conv=fsync count=129'
UTP: executing "dd if=/dev/zero of=/dev/mmcblk3 bs=1k seek=384 conv=fsync count=129"
129+0 records in
129+0 records out
132096 bytes (132 kB) copied, 0.00147833 s, 89.4 MB/s
UTP: sending Success to kernel for command $ dd if=/dev/zero of=/dev/mmcblk3 bs=1k seek=384 conv=fsync
count=129.
utp_poll: pass returned.
UTP: received command '$ echo 0 > /sys/block/mmcblk3boot0/force_ro'
UTP: executing "echo 0 > /sys/block/mmcblk3boot0/force_ro"
sh: /sys/block/mmcblk3boot0/force_ro: No such file or directory
UTP: sending Non-success to kernel for command $ echo 0 > /sys/block/mmcblk3boot0/force_ro.
utp_poll: exit with status 256
```

MFGTool programming fail log



Kernel & device tree porting – eMMC (3)

```
mxc_vpu 2040000.vpu: VPUinitialized
caam 2100000.caam: InstantiatedRNG4 SH0
mmc3: unrecognizedEXT_CSD revision 7
mmc3: error -22 whilstinitialising MMC card
caam 2100000.caam: InstantiatedRNG4 SH1
caam 2100000.caam: device ID =0x0a160100 (Era 4)
caam 2100000.caam: job rings =2, qi = 0
mmc3: unrecognizedEXT_CSD revision 7
mmc3: error -22 whilstinitialising MMC card
caam algorithms registered in/proc/crypto
```

We notice during the kernel boot up, the log shows `unrecognizedEXT_CSD revision 7`

See below snap of JEDEC Standard No. 84-B51.

So we modified the source code as below right.

`drivers/mmc/core/mmc.c`

7.4.61 EXT_CSD_REV [192]

Defines the fixed parameters related to the EXT_CSD, according to its revision

Table 139 — Extended CSD revisions

EXT_CSD_REV	Extended CSD Revision
255–9	Reserved
8	Revision 1.8 (for MMC v5.1)
7	Revision 1.7 (for MMC v5.0, v5.01)
6	Revision 1.6 (for MMC v4.5, v4.51)
5	Revision 1.5 (for MMC v4.41)
4	Revision 1.4 (Obsolete)
3	Revision 1.3 (for MMC v4.3)
2	Revision 1.2 (for MMC v4.2)
1	Revision 1.1 (for MMC v4.1)
0	Revision 1.0 (for MMC v4.0)

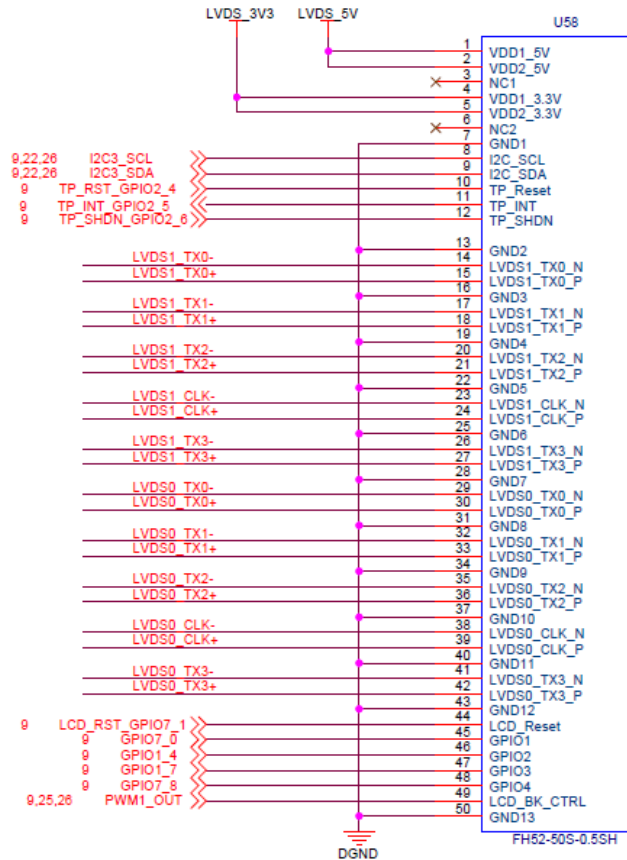
```
/* Version is coded in the CSD_STRUCTURE byte in the EXT_CSD register */
card->ext_csd.raw_ext_csd_structure = ext_csd[EXT_CSD_STRUCTURE];
if (card->csd.structure == 3) {
    if (card->ext_csd.raw_ext_csd_structure > 2) {
        pr_err("%s: unrecognized EXT_CSD structure "
            "version %d\n", mmc_hostname(card->host),
            card->ext_csd.raw_ext_csd_structure);
        err = -EINVAL;
        goto out;
    }
}

card->ext_csd.rev = ext_csd[EXT_CSD_REV];
// if (card->ext_csd.rev > 6) {
if (card->ext_csd.rev > 7) { //for eMMC 5.0 this revision will be 7
    pr_err("%s: unrecognized EXT_CSD revision %d\n",
        mmc_hostname(card->host), card->ext_csd.rev);
    err = -EINVAL;
    goto out;
}
```

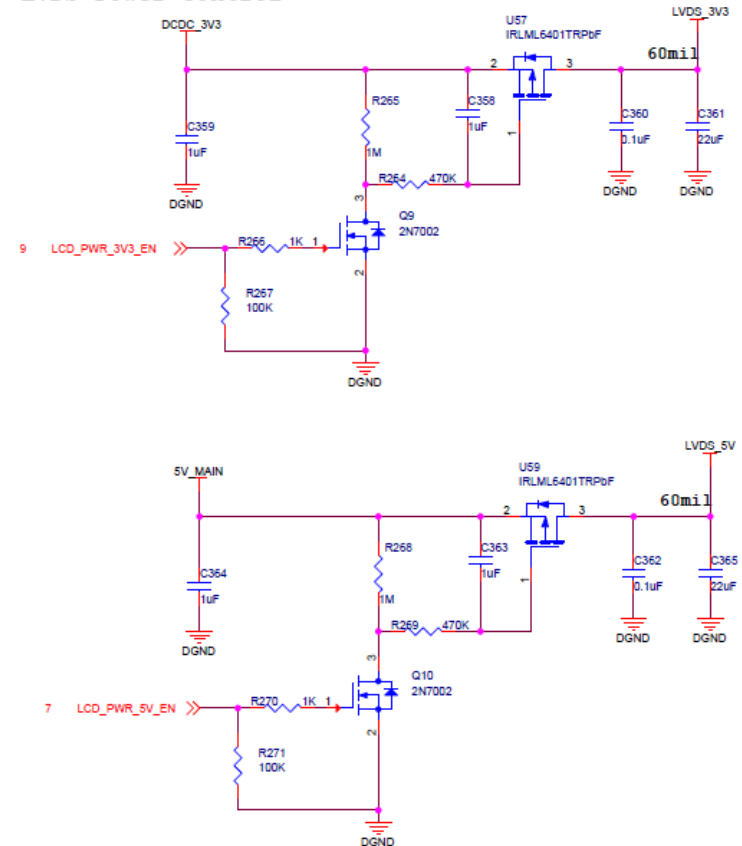


Kernel & device tree porting – LVDS (1)

Dual LVDS Port



LVDS Power Control



LVDS schematics of PianoPI board is similar with mx6qsabresd. We just need to make some IOMUX adaptations.

The **LVDS panel** used is different. We need to configure it in the **device tree** based on the panel data sheet.



Kernel & device tree porting – LVDS (2)

```
mxcfb1: fb@0 {
    compatible = "fsl,mxc_sdc_fb";
    disp_dev = "ldb";
    interface_pix_fmt = "RGB666";
    default_bpp = <16>;
    int_clk = <0>;
    late_init = <0>;
    status = "disabled";
};

...
&ldb {
    status = "okay";

    lvds-channel@0 {
        fsl,data-mapping = "spwg";
        fsl,data-width = <18>;
        status = "okay";

        display-timings {
            native-mode = <&timing0>;
            timing0: hsd100pxn1 {
                clock-frequency = <71100000>;
                hactive = <1280>;
                vactive = <800>;
                hback-porch = <40>;
                hfront-porch = <40>;
                vback-porch = <10>;
                vfront-porch = <3>;
                hsync-len = <80>;
                vsync-len = <10>;
            };
        };
    };
};
```

imx6qdl-pianopi.dtsi

Kernel & device tree porting – LVDS (3)

display-timings Binding Guide

17 required properties:

18 - hactive, vactive: display resolution

19 - hfront-porch, hback-porch, hsync-len: horizontal display timing parameters

20 in pixels

21 vfront-porch, vback-porch, vsync-len: vertical display timing parameters in

22 lines

23 - clock-frequency: display clock in Hz

24

25 optional properties:

26 - hsync-active: hsync pulse is active low/high/ignored

27 - vsync-active: vsync pulse is active low/high/ignored

28 - de-active: data-enable pulse is active low/high/ignored

29 - pixelclk-active: with

30 - active high = drive pixel data on rising edge/

31 sample data on falling edge

32 - active low = drive pixel data on falling edge/

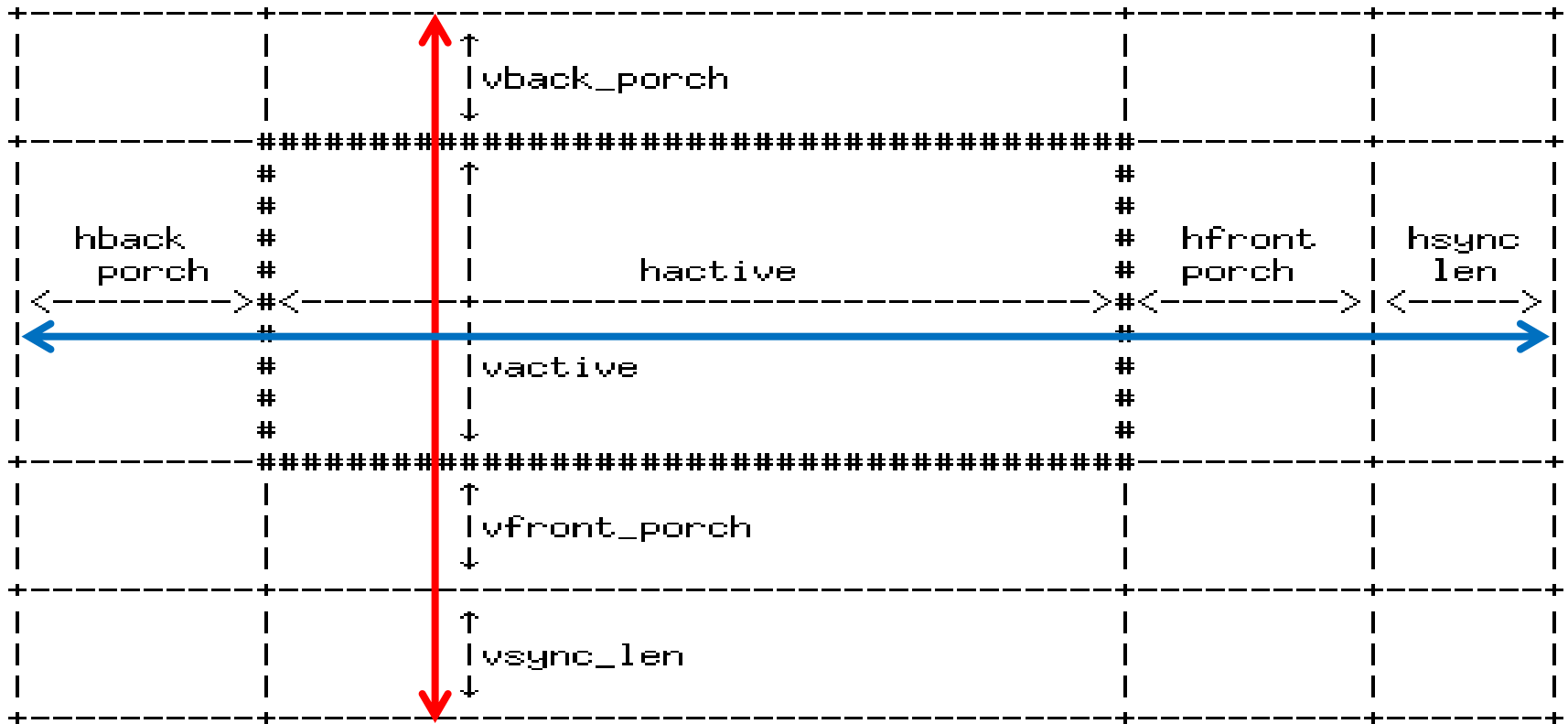
33 sample data on rising edge

34 - ignored = ignored

35 - interlaced (bool): boolean to enable interlaced mode

display-timing.txt

Kernel & device tree porting – LVDS (4)



In the DE mode

hback-porch + hfront-porch + hsync-len = Horizontal Black Time

vback-porch + vfront-porch + vsync-len = Vertical Black Time

hback-porch, hfront-porch, hsync-len are no-zero value

vback-porch, vfront-porch, vsync-len are no-zero value

Kernel & device tree porting – LVDS (5)

4.5 DISPLAY TIMING SPECIFICATIONS

The input signal timing specifications are shown as the following table and timing diagram.

Signal	Item	Symbol	Min.	Typ.	Max.	Unit	Note
DCLK	Frequency	1/Tc	(64)	(71.1)	(74.7)	MHz	-
DE	Vertical Total Time	TV	(810)	(823)	(829)	TH	-
	Vertical Active Display Period	TVD	800	800	800	TH	-
	Vertical Active Blanking Period	TVB	TV-TVD	(23)	TV-TVD	TH	-
	Horizontal Total Time	TH	(1362)	(1440)	(1480)	Tc	-
	Horizontal Active Display Period	THD	1280	1280	1280	Tc	-
	Horizontal Active Blanking Period	THB	TH-THD	(160)	TH-THD	Tc	-

Note (1) Because this module is operated by DE only mode, Hsync and Vsync are ignored.

LVDS has two modes: DE mode or SYNC mode.

According to the LVDS panel datasheet, it is DE mode only. So we could set the values accordingly as below.

```

&ldb {
    status = "okay";

    lvds-channel@0 {
        fsl,data-mapping = "spwg";
        fsl,data-width = <18>;
        status = "okay";

        display-timings {
            native-mode = <&timing0>;
            timing0: hsd100pxn1 {
                clock-frequency = <71100000>;
                hactive = <1280>;
                vactive = <800>;
                hback-porch = <40>;
                hfront-porch = <40>;
                vback-porch = <10>;
                vfront-porch = <3>;
                hsync-len = <80>;
                vsync-len = <10>;
            }
        }
    }
}
    
```

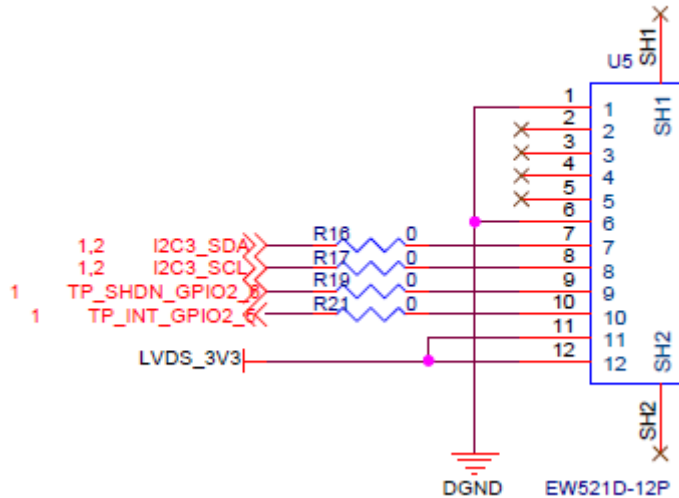
Vertical Black Time =
 $vback-porch + vfront-porch + vsync-len$
 $23 = 10 + 3 + 10$

Horizontal Black Time =
 $hback-porch + hfront-porch + hsync-len$
 $160 = 40 + 40 + 80$



Kernel & device tree porting – touch screen (1)

Touch Interface



The touch screen IC on PianoPI board is GOODix GT9xx. In our BSP release, it is not supported. We have get the source code supporting device tree from GOODix FAE and added it into the kernel source code directory `drivers/input/touchscreen/`. We also made modifications to the Makefile and Kconfig file. Device tree configuration see below.

Work ▶ PianoPI ▶ linux_3.10.53_11.0-pianopi-upload ▶ drivers ▶ input ▶ touchscreen

with New folder

Name	Date modified	Type	Size
gt9xx.c	2015-09-24 1:33 PM	C File	81 KB
gt9xx.h	2015-09-10 4:38 PM	H File	12 KB
Makefile	2015-08-27 5:11 PM	File	4 KB
Kconfig	2015-08-27 5:11 PM	File	25 KB
gt9xx_firmware.h	2014-11-28 11:06 ...	H File	213 KB
gt9xx_update.c	2014-11-28 10:56 ...	C File	104 KB
goodix_tool.c	2014-11-28 9:39 AM	C File	17 KB

```
&i2c3 {
    clock-frequency = <100000>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_i2c3_3>;
    status = "okay";

    goodix_ts@5d {
        compatible = "goodix,gt9xx";
        reg = <0x5d>;
        interrupt-parent = <&gpio2>;
        interrupts = <5 2>;
        goodix,rst-gpio = <&gpio2 6 0x00>;
        goodix,irq-gpio = <&gpio2 5 0x00>;

        ...
    };
};
```

Kernel & device tree porting – touch screen (2)

We have met the problem that, from debug log it seems touch screen module has get the touch coordinates. But touch on the screen UI has no response.

```
root@imx6qsabresd:~# evtest /dev/input/event0
...

Testing ... (interrupt to exit)
Event: time 1440752924.596007, type 1 (Key), code 330 (Touch), value 1
Event: time 1440752924.596007, type 3 (Absolute), code 53 (Position X), value 240
Event: time 1440752924.596007, type 3 (Absolute), code 54 (Position Y), value 807
Event: time 1440752924.596007, type 3 (Absolute), code 48 (Touch Major), value 28
Event: time 1440752924.596007, type 3 (Absolute), code 50 (Width Major), value 28
Event: time 1440752924.596007, type 3 (Absolute), code 57 (Tracking ID), value 0
Event: time 1440752924.596007, ----- Config Sync -----
Event: time 1440752924.596007, ----- Report Sync -----
Event: time 1440752924.610400, type 3 (Absolute), code 53 (Position X), value 240
Event: time 1440752924.610400, type 3 (Absolute), code 54 (Position Y), value 807
Event: time 1440752924.610400, type 3 (Absolute), code 48 (Touch Major), value 28
Event: time 1440752924.610400, type 3 (Absolute), code 50 (Width Major), value 28
Event: time 1440752924.610400, type 3 (Absolute), code 57 (Tracking ID), value 0
Event: time 1440752924.610400, ----- Config Sync -----
Event: time 1440752924.610400, ----- Report Sync -----
```

Kernel & device tree porting – touch screen (3)

We compared the evtest log with mx6q sabresd board and find the difference as below.

```
root@imx6qsabresd:~# evtest /dev/input/event0
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0xdead product
0xbeef version 0x28bb
Input device name: "goodix-ts"
Supported events:
  Event type 0 (Sync)
  Event type 1 (Key)
    Event code 330 (Touch)
  Event type 3 (Absolute)
    Event code 48 (Touch Major)
      Value      0
      Min        0
      Max       255
    Event code 50 (Width Major)
      Value      0
      Min        0
      Max       255
    Event code 53 (Position X)
      Value      0
      Min        0
      Max       800
    Event code 54 (Position Y)
      Value      0
      Min        0
      Max     1280
    Event code 57 (Tracking ID)
      Value      0
      Min        0
      Max       255
Testing ... (interrupt to exit)
```

PianoPI
board

```
root@imx6qdlsolo:~# evtest /dev/input/event0
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0x0 product 0x0
version 0x0
Input device name: "EETI eGalax Touch Screen"
Supported events:
  Event type 0 (Sync)
  Event type 1 (Key)
    Event code 330 (Touch)
  Event type 3 (Absolute)
    Event code 0 (X)
      Value     1472
      Min        0
      Max     32760
    Event code 1 (Y)
      Value     1824
      Min        0
      Max     32760
    Event code 47 (?)
      Value      0
      Min        0
      Max         4
    Event code 53 (Position X)
      Value      0
      Min        0
      Max     32760
    Event code 54 (Position Y)
      Value      0
      Min        0
      Max     32760
    Event code 57 (Tracking ID)
      Value      0
      Min        0
      Max    65535
Testing ... (interrupt to exit)
```

sabresd
board



Kernel & device tree porting – touch screen (4)

Below are the event definitions for touch screen.


```
#define ABS_X                0x00
#define ABS_Y                0x01
#define ABS_Z                0x02
#define ABS_RX               0x03
#define ABS_RY               0x04
#define ABS_RZ               0x05
...
#define ABS_MT_SLOT          0x2f /* MT slot being modified */
#define ABS_MT_TOUCH_MAJOR   0x30 /* Major axis of touching ellipse */
#define ABS_MT_TOUCH_MINOR   0x31 /* Minor axis (omit if circular) */
#define ABS_MT_WIDTH_MAJOR   0x32 /* Major axis of approaching ellipse */
#define ABS_MT_WIDTH_MINOR   0x33 /* Minor axis (omit if circular) */
#define ABS_MT_ORIENTATION   0x34 /* Ellipse orientation */
#define ABS_MT_POSITION_X    0x35 /* Center X touch position */
#define ABS_MT_POSITION_Y    0x36 /* Center Y touch position */
#define ABS_MT_TOOL_TYPE     0x37 /* Type of touching device */
#define ABS_MT_BLOB_ID       0x38 /* Group a set of packets as a blob */
#define ABS_MT_TRACKING_ID   0x39 /* Unique ID of initiated contact */
#define ABS_MT_PRESSURE     0x3a /* Pressure on contact area */
#define ABS_MT_DISTANCE     0x3b /* Contact hover distance */
```

include/uapi/linux/input.h

The ABS_X, ABS_Y are for **single touch** only while the definitions with prefix ABS_MT_ are for **multi-touch**.

We could learn from last slide that GOODix provided driver reports **multi-touch type B** touch events. The mx6q sabresd used e-galax driver reports **single touch** events together with **multi-touch type A** events.

More details on multi-touch protocol please refer to:

 <https://www.kernel.org/doc/Documentation/input/multi-touch-protocol.txt>



Kernel & device tree porting – touch screen (5)

On mx6q sabresd board, the evtest log looks like this:

```
root@imx6qsabresd:~# evtest /dev/input/event0
...
Testing ... (interrupt to exit)
Event: time 1420438798.864839, type 3 (Absolute), code 57 (Tracking ID), value 27
Event: time 1420438798.864839, type 3 (Absolute), code 53 (Position X), value 5584
Event: time 1420438798.864839, type 3 (Absolute), code 54 (Position Y), value 15536
Event: time 1420438798.864839, type 1 (Key), code 330 (Touch), value 1
Event: time 1420438798.864839, type 3 (Absolute), code 0 (X), value 5584
Event: time 1420438798.864839, type 3 (Absolute), code 1 (Y), value 15536
Event: time 1420438798.864839, ----- Report Sync -----
Event: time 1420438798.944020, type 3 (Absolute), code 53 (Position X), value 5616
Event: time 1420438798.944020, type 3 (Absolute), code 54 (Position Y), value 15568
Event: time 1420438798.944020, type 3 (Absolute), code 0 (X), value 5616
Event: time 1420438798.944020, type 3 (Absolute), code 1 (Y), value 15568
Event: time 1420438798.944020, ----- Report Sync -----
Event: time 1420438798.970491, type 3 (Absolute), code 53 (Position X), value 6144
Event: time 1420438798.970491, type 3 (Absolute), code 54 (Position Y), value 15520
Event: time 1420438798.970491, type 3 (Absolute), code 0 (X), value 6144
Event: time 1420438798.970491, type 3 (Absolute), code 1 (Y), value 15520
Event: time 1420438798.970491, ----- Report Sync -----
```

As we see, it reports both single touch and multi-touch coordinates and the Yocto built UI responds to the touch. While the Goodix GT9xx driver reports only the multi-touch coordinates, and the UI doesn't respond to the touch operations.

So **the Yocto built UI responds to the single touch coordinates report events only.** We need to add single touch events into the driver.

Kernel & device tree porting – touch screen (6)

We added single touch events to the driver source codes as below:

```
...
static s8 gtp_request_input_dev(struct goodix_ts_data *ts)
...
    input_set_abs_params(ts->input_dev, ABS_X, 0, ts->abs_x_max, 0, 0);
    input_set_abs_params(ts->input_dev, ABS_Y, 0, ts->abs_y_max, 0, 0);

    input_set_abs_params(ts->input_dev, ABS_MT_POSITION_X, 0, ts->abs_x_max, 0, 0);
    input_set_abs_params(ts->input_dev, ABS_MT_POSITION_Y, 0, ts->abs_y_max, 0, 0);
    input_set_abs_params(ts->input_dev, ABS_MT_WIDTH_MAJOR, 0, 255, 0, 0);
    input_set_abs_params(ts->input_dev, ABS_MT_TOUCH_MAJOR, 0, 255, 0, 0);
    input_set_abs_params(ts->input_dev, ABS_MT_TRACKING_ID, 0, 255, 0, 0);
...
static void gtp_touch_down(struct goodix_ts_data* ts,s32 id,s32 x,s32 y,s32 w)
...
    input_report_abs(ts->input_dev, ABS_X, x);
    input_report_abs(ts->input_dev, ABS_Y, y);
    input_sync(ts->input_dev);
...

```

gt9xx.c

We registered two events ABS_X and ABS_Y, and reported them when triggered. After the modification, touch on the panel UI get correct responses.

Kernel & device tree porting – WM8960 (1)

driver structure

Platform:

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

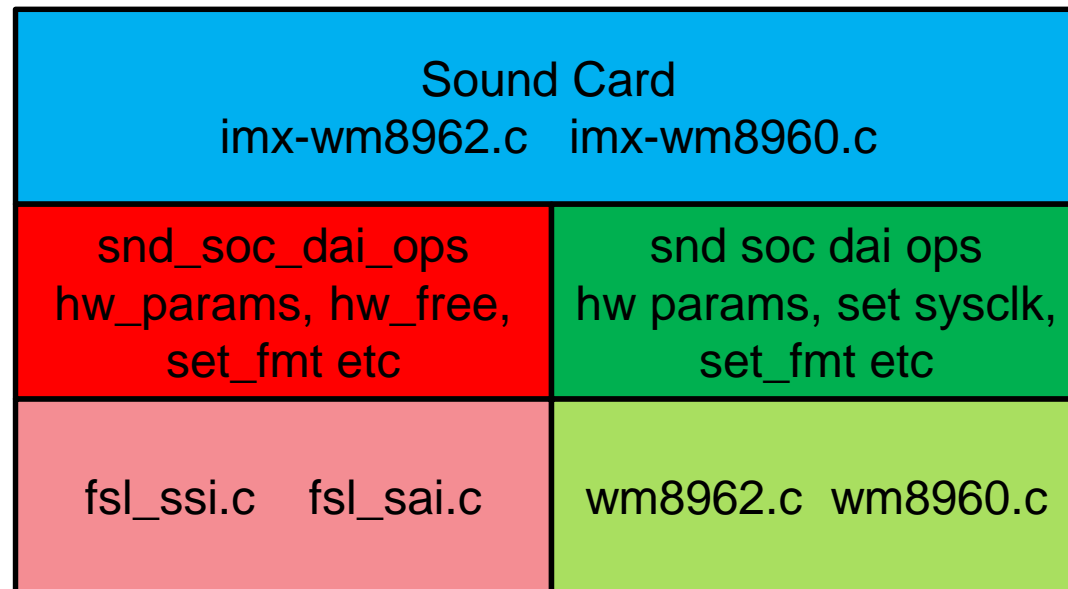
Machine:

imx-wm8960.c

config cpu_dai, codec_dai, control
and audio routing map

Codec:

wm8960.c



Kernel & device tree porting – WM8960 (2)

```
&i2c2 {  
    clock_frequency = <100000>;  
    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;  
    };
```

Codec

```
reg_audio: wm8960_supply {  
    compatible = "regulator-fixed";  
    regulator-name = "wm8960-supply";  
    gpio = <&gpio2 7 0>;  
    enable-active-high;  
};
```

**Power
Regulator**

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

Machine



Kernel & device tree porting – WM8960 (3)

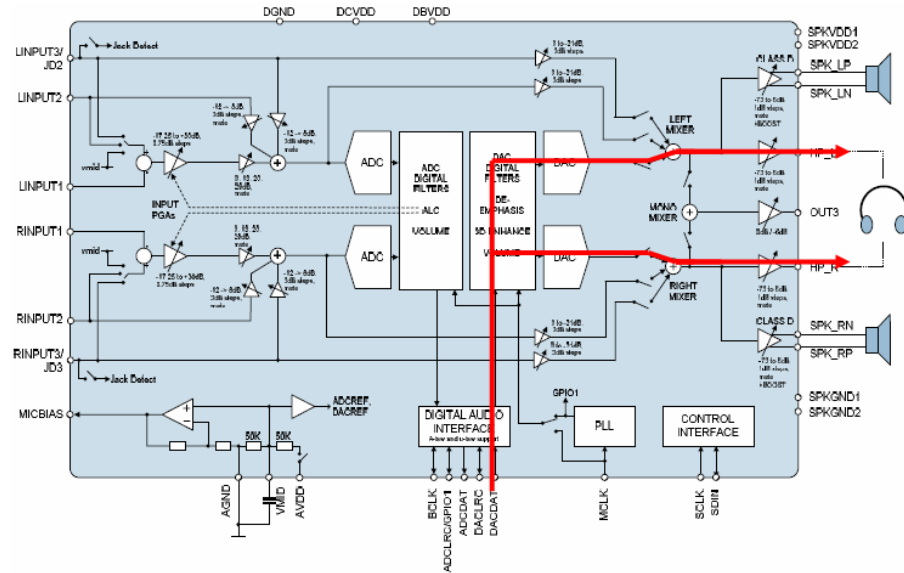
Kcontrol

```
static const struct snd_kcontrol_new wm8960_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),
```

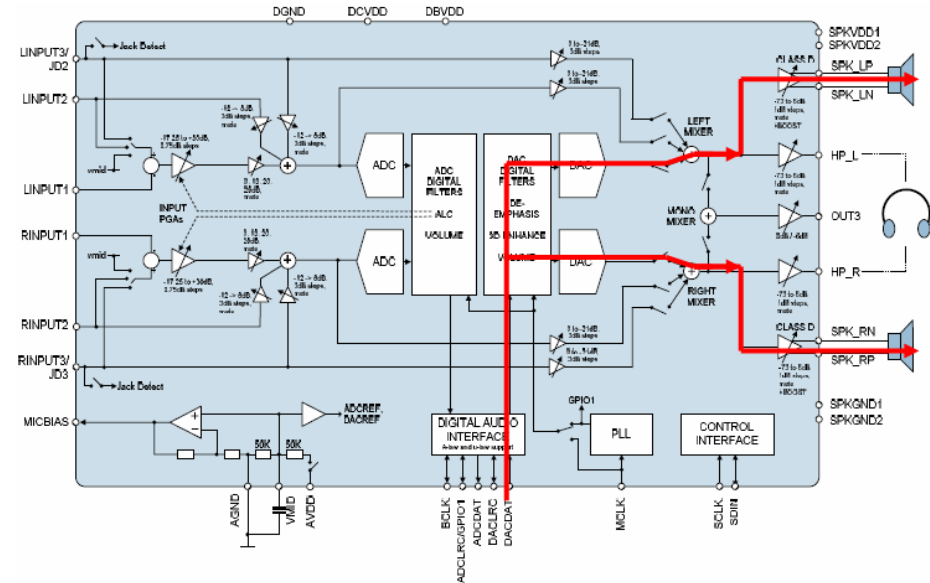
Kcontrol set the codec wm8960 registers via i2c interface

Kernel & device tree porting – WM8960 (4)

audio routing



headphone



speakers

Audio routing for headphone and speakers

Kernel & device tree porting – WM8960 (5)

audio routing

```
static const struct snd_soc_dapm_route audio_paths[] = { wm8960.c
    { "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

Kernel & device tree porting – WM8960 (7)

control widget

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

imx-wm8960.c

machine driver imx-wm8960:

define widgets which will be linked to **codec driver** power management widget in last slide

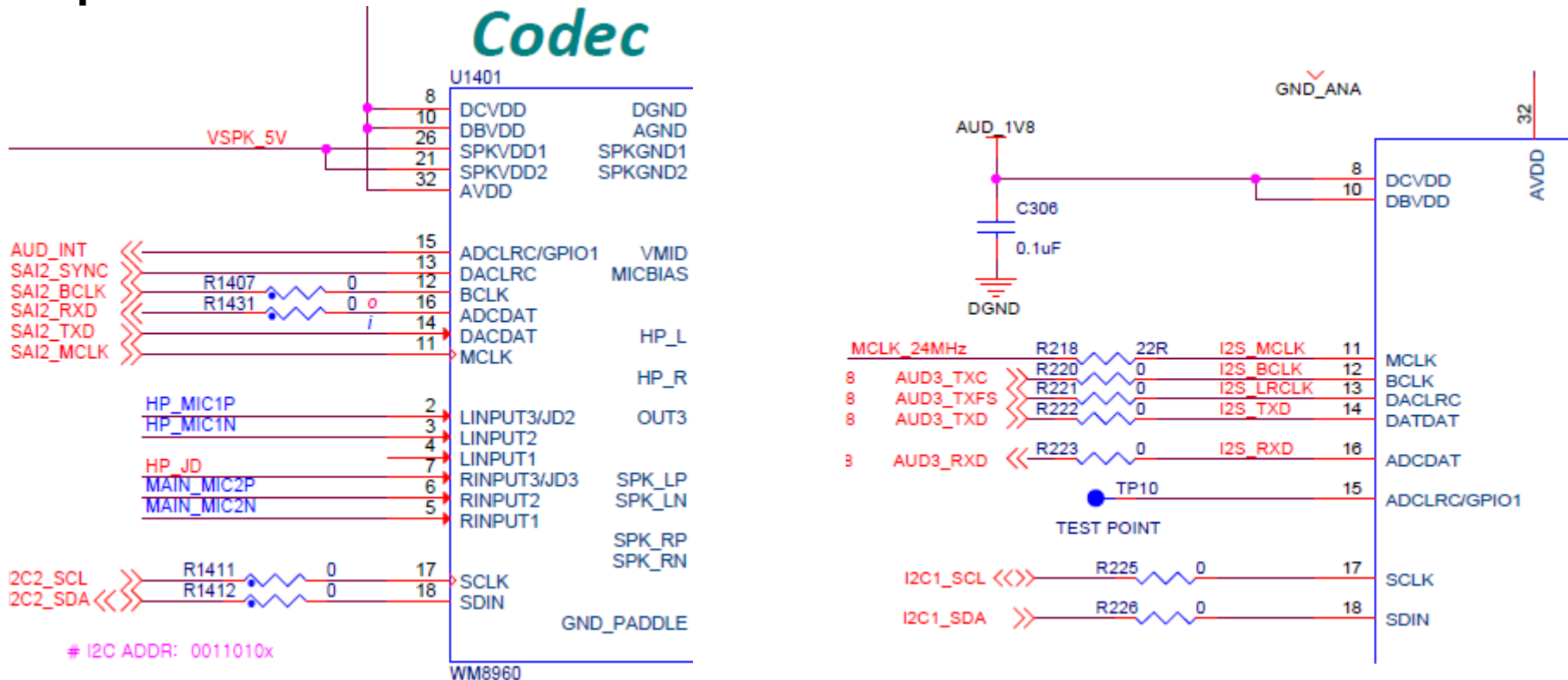
see device tree **audio-routing** on the right

imx6qdl-pianopi.dtsi

```
audio-routing =  
    "Headset Jack", "HP_L",  
    "Headset Jack", "HP_R",  
    "Ext Spk", "SPK_LP",  
    "Ext Spk", "SPK_LN",
```

Kernel & device tree porting – WM8960 (8)

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.



Kernel & device tree porting – WM8960 (9)

cpu dai: ssi

```
#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;
    }
    */
```

imx-wm8960.c

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

imx6qdl-pianopi.dtsi

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.



Kernel & device tree porting – WM8960 (10)

cpu dai: ssi



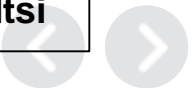
AUD3_TXC	GPIO5_22	N1	CSIO_DAT4
AUD3_TXD	GPIO5_23	P2	CSIO_DAT5
AUD3_TXFS	GPIO5_24	N4	CSIO_DAT6
AUD3_RXD	GPIO5_25	N3	CSIO_DAT7
I2C1_SDA	I2C1_SDA	N6	CSIO_DAT8
I2C1_SCL	I2C1_SCL	N5	CSIO_DAT9
		M1	

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;
    ...
};
    
```



Kernel & device tree porting – WM8960 (11)

cpu dai: ssi

```
...
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

imx6qdl-pianopi.dtsi

```
sound {
    compatible = "fsl,imx6-pianopi-wm8960",
        "fsl,imx-audio-wm8960";
    model = "wm8960-audio";
    cpu-dai = <&ssi2>;
    ...
    mux-int-port = <2>;
    mux-ext-port = <3>;
    hp-det = <3 0>;
    hp-det-gpios = <&gpio3 19 1>;
    spk-en-gpio = <&gpio2 7 0>;
};
```

imx-wm8960.c should add audmux support.

The device tree ports configurations see the above.

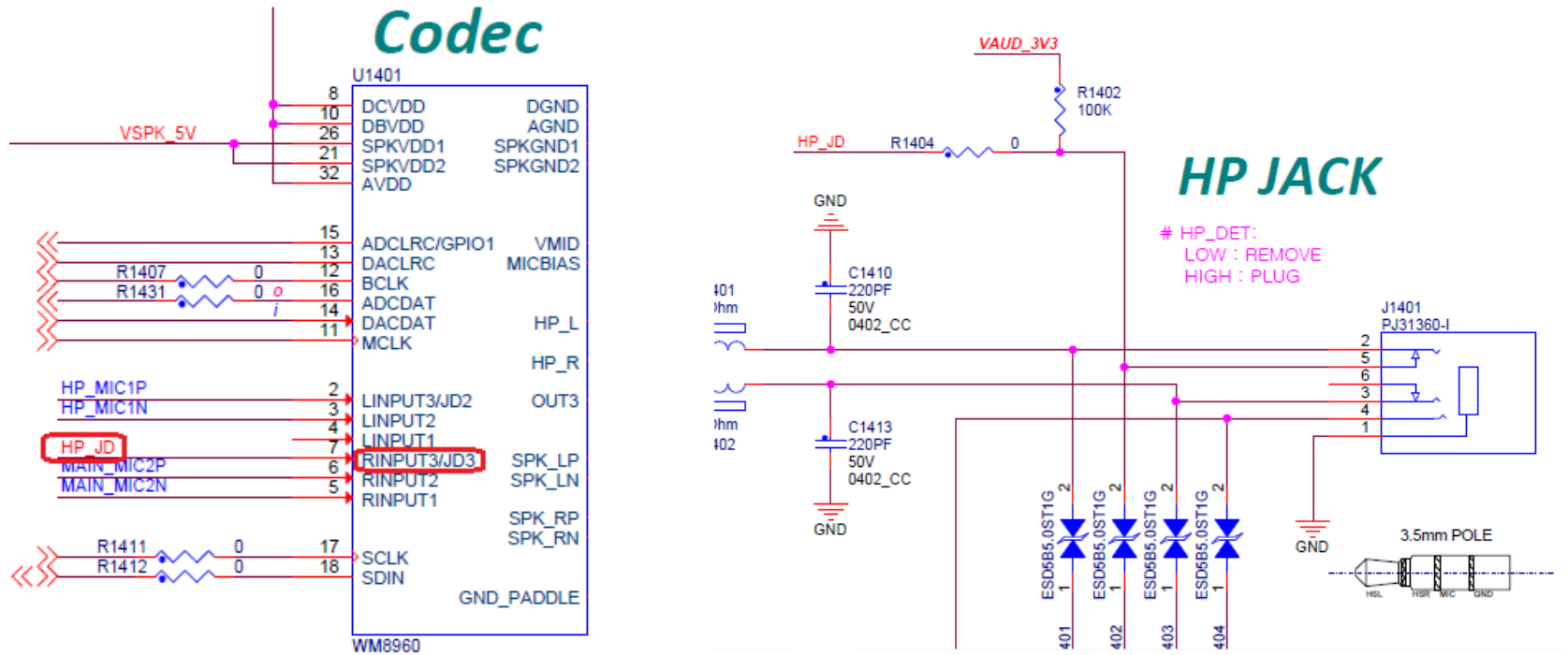
Please note ssi should be corresponding to mux-int-port as listed in last slide:

ssi1<->1, ssi2<->2, ssi3<->7.



Kernel & device tree porting – WM8960 (12)

hp detect: i.MX6UL EVK



i.MX6UL EVK Audio Codec Headphone Detect:

HP_JD signal is connected to wm8960 JD3 pin.

Kernel & device tree porting – WM8960 (13)

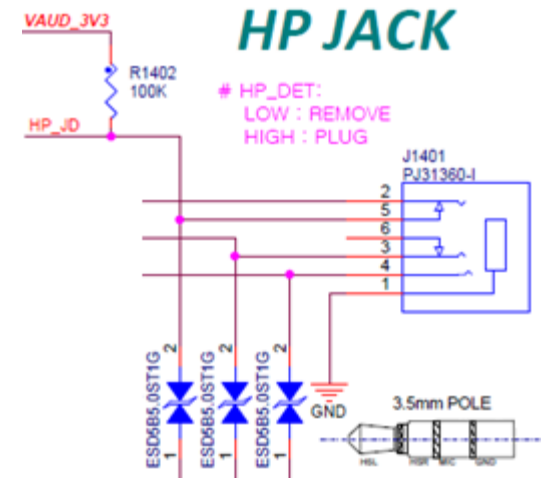
hp detect: i.MX6UL EVK

R48 (30h) Additional Control (4)	3:2	HPSEL[1:0]	00	Headphone Switch Input Select 0X = GPIO1 used for jack detect input (Requires ADCLRC pin to be configured as a GPIO) 10 = JD2 used for jack detect input 11 = JD3 used for jack detect input	Headphone Jack Detect
	1	TSENSEN	1	Temperature Sensor Enable 0 = Temperature sensor disabled 1 = Temperature sensor enabled	Thermal Shutdown

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

REGISTER ADDRESS	BIT	LABEL	DEFAULT	DESCRIPTION	REFER TO
Additional Control (2)	6	HPSWEN	0	Headphone Switch Enable 0 = Headphone switch disabled 1 = Headphone switch enabled	Headphone Jack Detect
	5	HPSWPOL	0	Headphone Switch Polarity 0 = HPDETECT high = headphone 1 = HPDETECT high = speaker	Headphone Jack Detect
	4			Reserved	

update R24(18h) with hp-det = <3 0>;



Kernel & device tree porting – WM8960 (14)

hp detect: i.MX6UL EVK

```
                                imx6ul-14x14-evk.dts
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>;
};
```

```
                                imx-wm8960.c
wm8960_init():

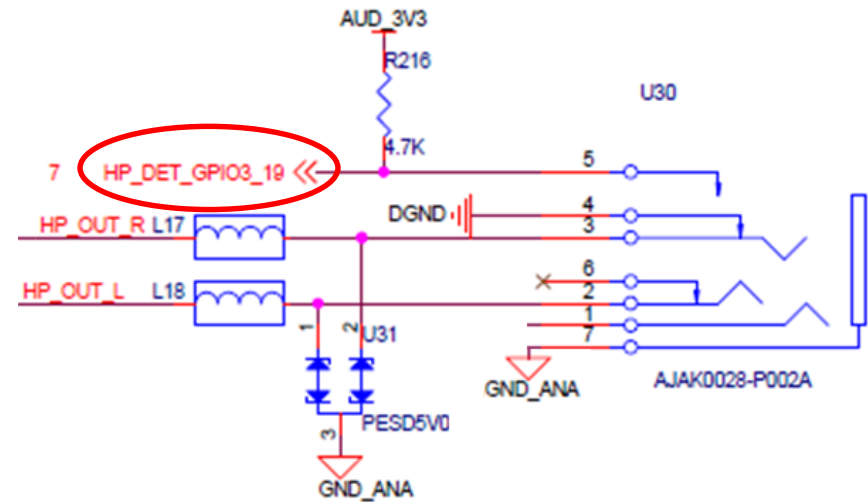
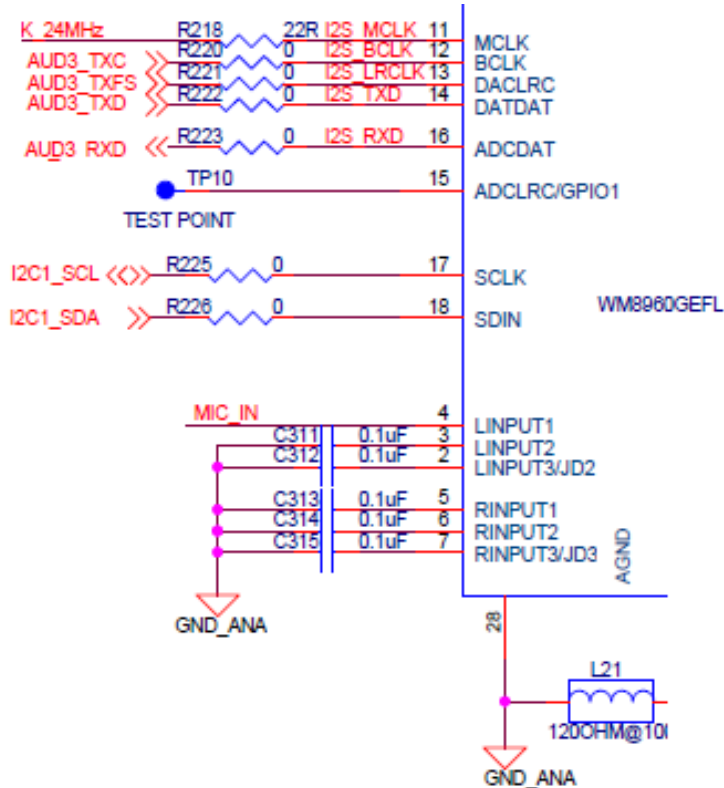
/* Enable headphone jack detect */
snd_soc_update_bits(codec, WM8960_ADDCTL2, 1<<6, 1<<6);
snd_soc_update_bits(codec, WM8960_ADDCTL2, 1<<5, data->hp_det[1]<<5);
snd_soc_update_bits(codec, WM8960_ADDCTL4, 3<<2, data->hp_det[0]<<2);
snd_soc_update_bits(codec, WM8960_ADDCTL1, 3, 3);
```

update R24(18h) with
hp-det = <3 0>;

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

Kernel & device tree porting – WM8960 (15)

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.

Kernel & device tree porting – WM8960 (16)

hp detect: PianoPI

```
imx-wm8960.c
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, 3<<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");
...
}
```

```
imx6qdl-pianopi.dtsi
sound {
    compatible = "fsl,imx6-pianopi-wm8960",
        "fsl,imx-audio-wm8960";
    model = "wm8960-audio";
    cpu-dai = <&ssi2>;
    audio-codec = <&codec>;
    asrc-controller = <&asrc>;
    codec-master;
    gpr = <&gpr>;
    /* hp-det = <3 1>; */
    hp-det-gpios = <&gpio5 4 0>;
};
```

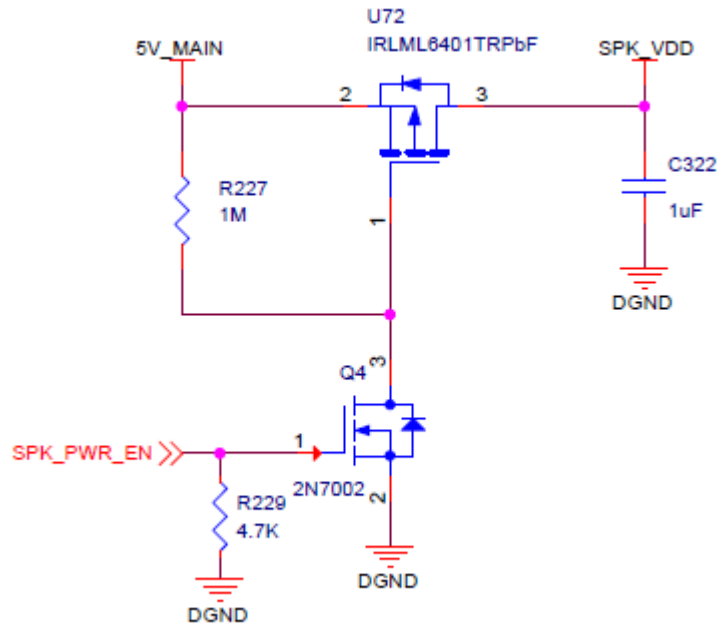
We should disable wm8960 headphone detection function and change dapm as left.

The device tree **hp-det** configuration is not supported anymore. We delete it as above.



Kernel & device tree porting – WM8960 (17)

speaker power enable



imx6qdl-pianopi.dtsi

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

PianoPI speaker power enable:

In Documentation/devicetree/bindings/regulator/regulated-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.

Kernel & device tree porting – WM8960 (18)

speaker power enable

Why modify GPIO polarity gpio = <&gpio2 7 1> has no effect

drivers/regulator/fixed.c

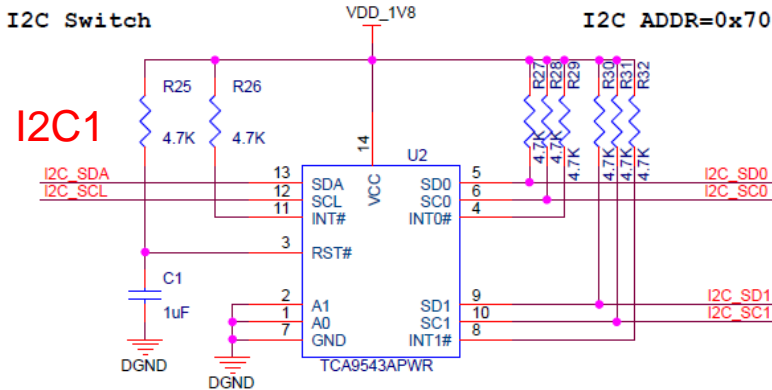
```
config->gpio = of_get_named_gpio(np, "gpio", 0);
if (of_find_property(np, "enable-active-high", NULL))
    config->enable_high = true;

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

Set GPIO init voltage here, the configuration of GPIO polarity is overwritten.

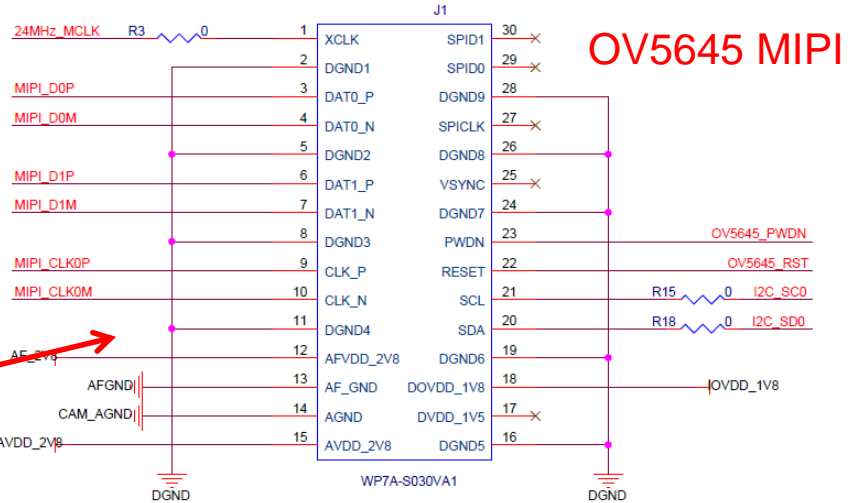
Kernel & device tree porting – camera (1)

drive structure

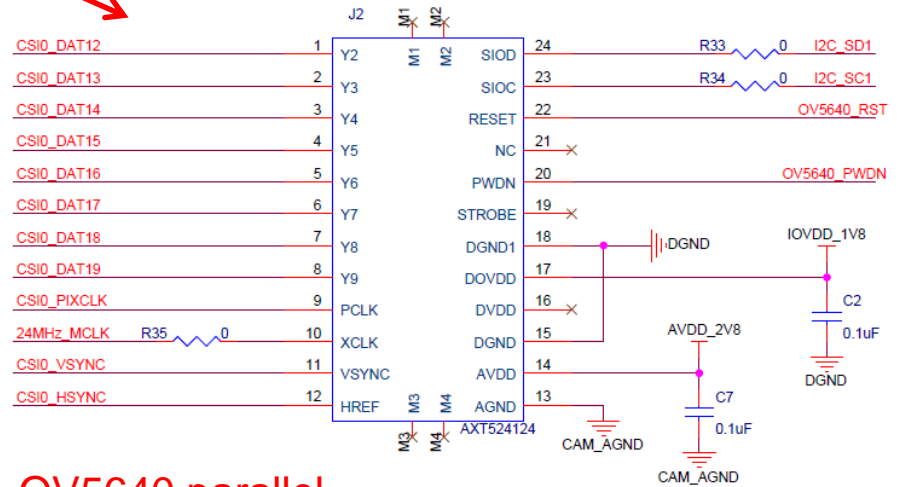


I2C Switch: PCA9543

On PianoPI board there are two cameras. One is OV5645 MIPI, the other is OV5640 parallel. The two OV564x cameras with same I2C address 0x3C are both connected to I2C1 via an I2C switch PCA9543.



7bit I2C address : 0x3C



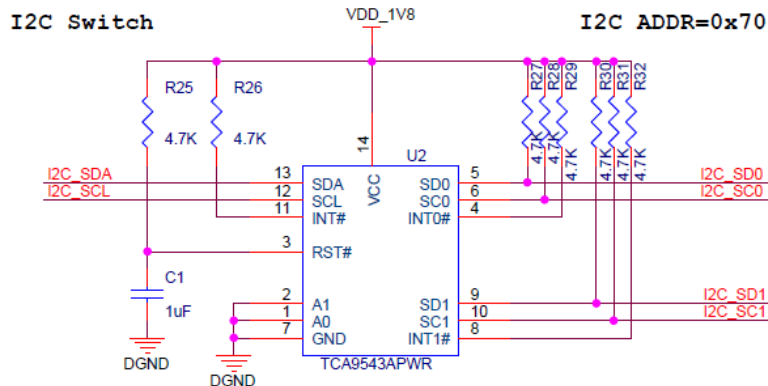
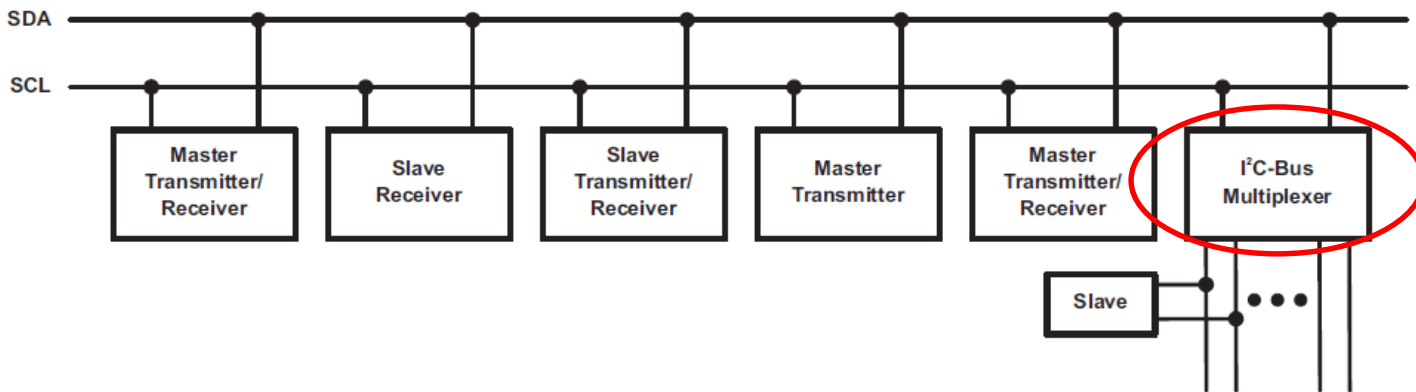
OV5640 parallel

7bit I2C address : 0x3C



Kernel & device tree porting – camera (2)

I2C switch principle










I2C Switch: PCA9543

I2C switch drive principle please see above. It usually connects several devices with same I2C address.

PCA9543 is a NXP I2C switch which supports two sub-devices. On PianoPI board, it connects to OV564x cameras.

Kernel & device tree porting – camera (3)

The PCA9543 driver source code is copied from L3.14.38-6dqp-ga, including several files under linux directory `./drivers/i2c/muxes/`.

Name	Date modified	Type	Size
 i2c-mux-pca954x.c	2015-10-10 4:45 PM	C File	9 KB
 i2c-arb-gpio-challenge.c	2015-09-23 5:27 PM	C File	8 KB
 i2c-mux-gpio.c	2015-09-23 5:27 PM	C File	7 KB
 i2c-mux-pca9541.c	2015-09-23 5:27 PM	C File	11 KB
 i2c-mux-pinctrl.c	2015-09-23 5:27 PM	C File	8 KB
 Kconfig	2015-09-23 5:27 PM	File	2 KB
 Makefile	2015-09-23 5:27 PM	File	1 KB

It is supposed to support **device tree** as it has device tree binding guide under Documentation directory. In fact it doesn't. If go through the code of `i2c-mux-pca954x.c`, we could find it supports old style **platform matching** only.

We need to pass the I2C switch mux number and the adapter ID into the driver. So we add some codes in the driver to parse the settings from device tree, and configure them in the device tree file as below:

```
i2c-mux-num = <2>; /* simulate two I2C adapters */  
i2c-mux-0-adap_id = <3>;  
i2c-mux-1-adap_id = <4>;
```

Please refer to the next slides.

Kernel & device tree porting – camera (4)

i2c-mux-pca954x.c

```
...
static int pca954x_probe(struct i2c_client *client,
                        const struct i2c_device_id *id)
{
    struct i2c_adapter *adap = to_i2c_adapter(client->dev.parent);
    struct pca954x_platform_data *pdata = dev_get_platdata(&client->dev);
    struct pca954x_platform_data *pdata_of;
    ...
    pdata_of = devm_kzalloc(&client->dev, sizeof(struct pca954x_platform_data), GFP_KERNEL);
    if (!pdata_of)
        return -ENOMEM;

    ret = of_property_read_u32(np, "i2c-mux-num", &(pdata_of->num_modes));
    if (ret) {
        dev_err(&client->dev, "i2c-mux-num missing or invalid\n");
        return ret;
    }

    if(pdata_of->num_modes > chips[data->type].nchans) {
        pdata_of->num_modes = chips[data->type].nchans;
    }

    pdata_of->modes = devm_kzalloc(&client->dev, sizeof(struct pca954x_platform_mode)*pdata_of->num_modes,
    GFP_KERNEL);
}
```

mux number will be defined in device tree, which could be 2-8 for specific pca954x. For pca9543 it should be no more than 2.

Kernel & device tree porting – camera (5)

i2c-mux-pca954x.c

```
....  
pdata_of->modes = devm_kzalloc(&client->dev, sizeof(struct pca954x_platform_mode)*pdata_of->num_modes,  
GFP_KERNEL);
```

```
if (!pdata_of)
```

```
    return -ENOMEM;
```

```
for (num = 0; num < pdata_of->num_modes; num++) {
```

```
    pdata_of->modes[num].deselect_on_exit = true;
```

```
}
```

```
ret = of_property_read_u32(np, "i2c-mux-0-adap_id", &(pdata_of->modes[0].adap_id));
```

```
if (ret) {
```

```
    dev_err(&client->dev, "i2c-mux-0-adap_id missing or invalid\n");
```

```
return ret;
```

```
}
```

```
ret = of_property_read_u32(np, "i2c-mux-1-adap_id", &(pdata_of->modes[1].adap_id));
```

```
if (ret) {
```

```
    dev_err(&client->dev, "i2c-mux-1-adap_id missing or invalid\n");
```

```
return ret;
```

```
}
```

```
pdata = pdata_of;
```

In device tree we will define two simulated I2C adapters id. Should not conflict with existed ones. That is, in i.MX6Q board, they could not be 0-2.

Kernel & device tree porting – camera (6)

imx6qdl-pianopi.dtsi

```
...
&i2c1 {
    clock-frequency = <100000>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_i2c1_2>;
    status = "okay";

    codec: wm8960@1a { ... };

    i2c-switch@70 {
        compatible = "nxp,pca9543";
        reg = <0x70>;
        i2c-mux-num = <2>;
        i2c-mux-0-adap_id = <3>;
        i2c-mux-1-adap_id = <4>;
    }

    i2c@0 {
        reg = <0>;

        ov5645_mipi: ov5645_mipi@3c { /* i2c2 driver */
            compatible = "ovti,ov5645_mipi";
            reg = <0x3c>;
            ...
        }
    }
}

i2c@1 {
    reg = <1>;

    ov5640: ov5640@3c {
        compatible = "ovti,ov5640";
        reg = <0x3c>;
        ...
    }
}
...

```

Configure for I2C switch in device tree

Kernel & device tree porting – camera (7)

```
root@imx6qsabresd:~# i2cdetect -l
i2c-0    i2c          21a0000.i2c
i2c-1    i2c          21a4000.i2c
i2c-2    i2c          21a8000.i2c
i2c-3    i2c          i2c-0-mux (chan_id 0)
i2c-4    i2c          i2c-0-mux (chan_id 1)
```

The last two I2C adapter is generated by I2C switch pca9543.

I2C adapter
I2C adapter
I2C adapter
I2C adapter
I2C adapter

```
root@imx6qsabresd:~# i2cdetect -y 3
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- UU -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- UU -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: UU -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

0x1a is audio codec wm8960.
0x3c is ov5645_mipi.
0x70 is I2C switch pca9543.

```
root@imx6qsabresd:~# i2cdetect -y 4
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:                -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- UU -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- UU -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: UU -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

Here 0x3c is ov5640 parallel.

root@imx6qsabresd:~#



Kernel & device tree porting – camera (8)

OV5640 parallel

There are two cameras on the board. For OV5640 parallel, we could use the driver in L3.10.53_1.1.0_GA directly. Once I2C switch is debugged OK, device tree correctly configured parallel camera OV5640 will work.

```
...                               imx6qdl-pianopi.dtsi
                                ov564x: ov564x@3c {
                                    compatible = "ovti,ov564x";
                                    reg = <0x3c>;
...
                                };
```

```
...                               imx6qdl-pianopi.dtsi
                                ov5640: ov5640@3c {
                                    compatible = "ovti,ov5640";
                                    reg = <0x3c>;
...
                                };
```

Please note two cameras on i.MX6Q sabresd board, one is OV5640/OV5642 parallel camera, the other is OV5640 MIPI.

If in device tree we set camera as above left, the ov5640_camera and ov5642_camera module driver will both be probed as below left. If we set as above right, only ov5640_camera will be probed as below right.

```
root@imx6qsabresd:~# lsmod
Module                Size  Used by
ov5642_camera         75119  0
mxc_v4l2_capture      22322  2
ipu_fg_overlay_sdc   4877   1 mxc_v4l2_capture
ipu_csi_enc           2841   1 mxc_v4l2_capture
ipu_prp_enc           4645   1 mxc_v4l2_capture
ipu_still             1663   1 mxc_v4l2_capture
ipu_bg_overlay_sdc   4001   1 mxc_v4l2_capture
ov5640_camera         17959  0
ov5645_camera_mipi   19193  0
mxc_dcic              5334   0
```

```
root@imx6qsabresd:~# lsmod
Module                Size  Used by
mxc_v4l2_capture      22322  2
ipu_fg_overlay_sdc   4877   1 mxc_v4l2_capture
ipu_csi_enc           2841   1 mxc_v4l2_capture
ipu_prp_enc           4645   1 mxc_v4l2_capture
ipu_still             1663   1 mxc_v4l2_capture
ipu_bg_overlay_sdc   4001   1 mxc_v4l2_capture
ov5640_camera         17959  0
ov5645_camera_mipi   19193  0
mxc_dcic              5334   0
```

Kernel & device tree porting – camera (9)

OV5640 parallel

```
...  
    ov564x: ov564x@3c {  
        compatible = "ovti,ov564x";  
        reg = <0x3c>;  
    };  
...
```

Why set compatible string as ov564x probes both ov5640 and ov5642 module?

See below drivers source codes.

```
...  
static const struct i2c_device_id ov5640_id[] = {  
    {"ov5640", 0},  
    {"ov564x", 0},  
    {},  
};  
MODULE_DEVICE_TABLE(i2c, ov5640_id);  
...
```

```
...  
static const struct i2c_device_id ov5642_id[] = {  
    {"ov5642", 0},  
    {"ov564x", 0},  
    {},  
};  
MODULE_DEVICE_TABLE(i2c, ov5642_id);  
...
```



Kernel & device tree porting – camera (10)

OV5645 MIPI

OV5645 MIPI driver is based on freescale TIC SUN Weidong developed driver which is for L3.0.101 with old style platform matching.

Have added device tree support into it. During the porting, I have referred to the ov5640_mipi.c file changes between L3.0.101 and L3.10.53.

Below is the main changes made in **ov5645_mipi.c** to support device tree.

```
...
static int ov5645_probe(struct i2c_client *client,
                        const struct i2c_device_id *id)
{
...
    /* request power down pin */
    pwn_gpio = of_get_named_gpio(dev->of_node, "pwn-gpios", 0);
...
    retval = devm_gpio_request_one(dev, pwn_gpio, GPIOF_OUT_INIT_HIGH, "ov5645_mipi_pwdn");
...
    rst_gpio = of_get_named_gpio(dev->of_node, "rst-gpios", 0);
...
    retval = devm_gpio_request_one(dev, rst_gpio, GPIOF_OUT_INIT_HIGH, "ov5645_mipi_reset");
...
}
```

ov5645_mipi.c

Kernel & device tree porting – camera (11)

OV5645 MIPI

```
...
ov5645_data.sensor_clk = devm_clk_get(dev, "csi_mclk");
...
retval = of_property_read_u32(dev->of_node, "mclk", &(ov5645_data.mclk));
...
retval = of_property_read_u32(dev->of_node, "mclk_source", (u32 *) &(ov5645_data.mclk_source));
...
retval = of_property_read_u32(dev->of_node, "csi_id", &(ov5645_data.csi));
...
io_regulator = devm_regulator_get(dev, "DOVDD");
...
core_regulator = devm_regulator_get(dev, "DVDD");
...
analog_regulator = devm_regulator_get(dev, "AVDD");
...
```

ov5645_mipi.c

There are still some other tiny modifications like adding below functions, which is not listed in detail here, please refer to the source code:

ov5645_mipi_powerdown

ov5645_mipi_sensor_io_init

ioctl_enum_frameintervals

Kernel & device tree porting – camera (12)

OV5645 MIPI

imx6qdl-pianopi.dtsi

```
...
&i2c1 {
    ...
    i2c-switch@70 {
        compatible = "nxp,pca9543";
        reg = <0x70>;
        ...
        i2c@0 {
            reg = <0>;

            ov5645_mipi: ov5645_mipi@3c { /* i2c2 driver */
                compatible = "ovti,ov5645_mipi";
                reg = <0x3c>;
                clocks = <&clks 201>;
                clock-names = "csi_mclk";
                DOVDD-supply = <&vgen4_reg>; /* 1.8v */
                AVDD-supply = <&vgen3_reg>; /* 2.8v, rev C board is VGEN3
                    rev B board is VGEN5 */
                DVDD-supply = <&vgen2_reg>; /* 1.5v */
                pwn-gpios = <&gpio1 17 1>; /* active low */
                rst-gpios = <&gpio1 19 0>; /* active high */
                csi_id = <1>;
                mclk = <24000000>;
                mclk_source = <0>;
            };
        };
    };
    ...
}
```


Kernel & device tree porting – camera (13)

debugging cameras commands

We debug cameras use utilities under unit_tests directory.

```
/unit_tests/mxc_v4l2_overlay.out
```

Default output /dev/video0, csi0, which is ov5640.

```
/unit_tests/mxc_v4l2_overlay.out -di /dev/video1
```

output /dev/video1, csi1, which is ov5645_mipi.

```
/unit_tests/mxc_v4l2_overlay.out -ot 100 -ol 100 -ow 720 -oh 572 &
```

set output top, left, width and height

```
/unit_tests/mxc_v4l2_overlay.out -help
```

For more parameters



Yocto project porting

Yocto Modification

add a layer (mylayer) for:

1. u-boot and kernel patching
2. support new machines (...)
3. appoint new defconfig files:
 - imx_v7_pianopi_defconfig
 - imx_v7_pianopi_mfg_defconfig

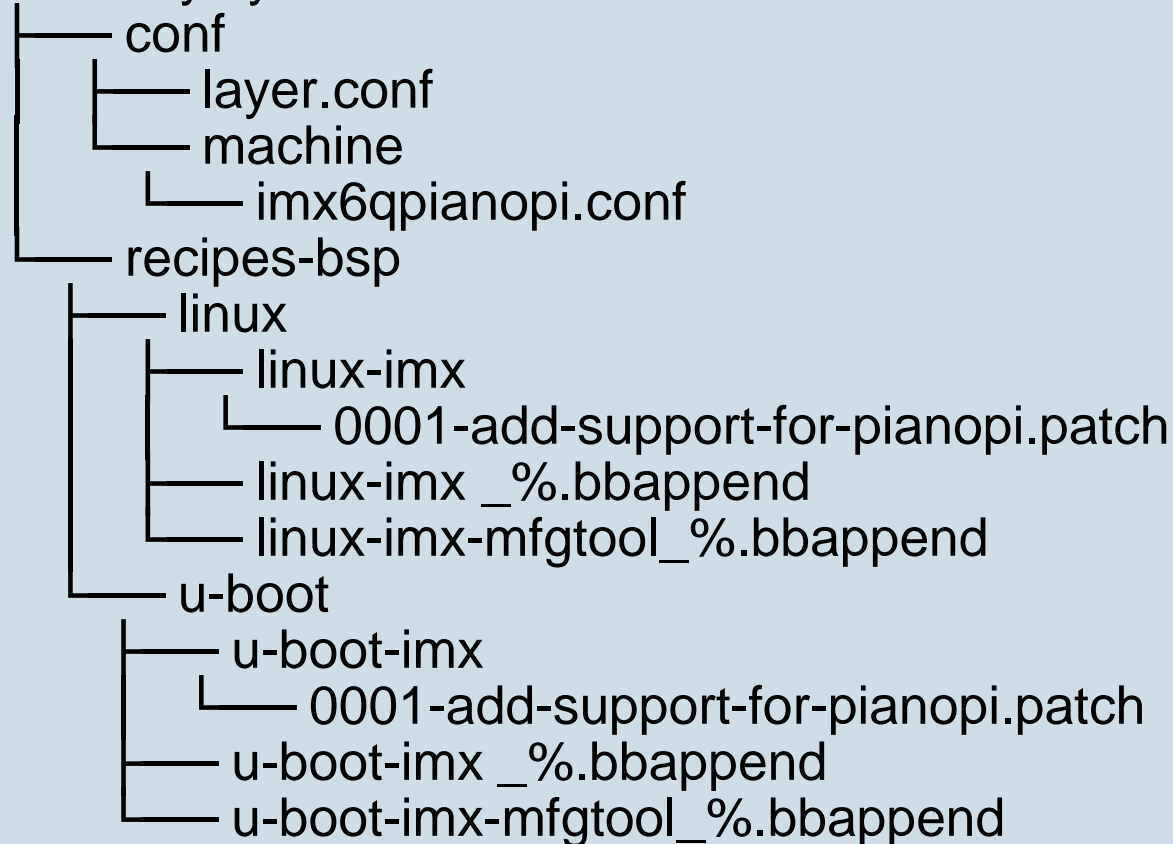
```
yocto-layer create mylayer
```

We could run this command to create our new-add layer under directory sources/

mylayer directory structure

- glance at the added layer (mylayer) under directory sources/

meta-mylayer/



yocto-layer create mylayer

yocto-layer command is activated after we run:

```
source fsl-setup-release.sh
```

```
yocto-layer create mylayer
```

if all setting use default value, there will no example recipe and no bbappend, and the created layer looks like below:



yocto-layer create mylayer (cont.)

layer.conf

```
# We have a conf and classes directory, add to BBPATH
BBPATH .= ":{LAYERDIR}"
```

```
# We have recipes-* directories, add to BBFILES
BBFILES += "${LAYERDIR}/recipes-*/*/*.bb \
           ${LAYERDIR}/recipes-*/*/*.bbappend"
```

```
BBFILE_COLLECTIONS += "meta-mylayer"
BBFILE_PATTERN_ meta-mylayer = "^${LAYERDIR}/"
BBFILE_PRIORITY_ meta-mylayer = "6"
```

yocto-layer create mylayer (cont.)

```
yocto-layer create mylayer
```

if we create layer with sample recipe and sample bbappend, the created layer looks like this:

meta-mylayer/

```
├── conf
│   └── layer.conf
├── COPYING.MIT
├── README
├── recipes-example
│   └── example
│       ├── example-0.1
│       │   ├── example.patch
│       │   └── helloworld.c
│       └── example_0.1.bb
└── recipes-example-bbappend
    └── example-bbappend
        ├── example-0.1
        │   └── example.patch
        └── example_0.1.bbappend
```

yocto-layer create mylayer (cont.)

example_0.1.bb

```
#
# This file was derived from the 'Hello World!' example recipe in the
# Yocto Project Development Manual.
#

DESCRIPTION = "Simple helloworld application"
SECTION = "examples"
LICENSE = "MIT"
LIC_FILES_CHKSUM =
"file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"
PR = "r0"

SRC_URI = "file://helloworld.c"

S = "${WORKDIR}"

do_compile() {
    ${CC} helloworld.c -o helloworld
}

do_install() {
    install -d ${D}${bindir}
    install -m 0755 helloworld ${D}${bindir}
}
```


yocto-layer create mylayer (cont.)

example_0.1.bbappend

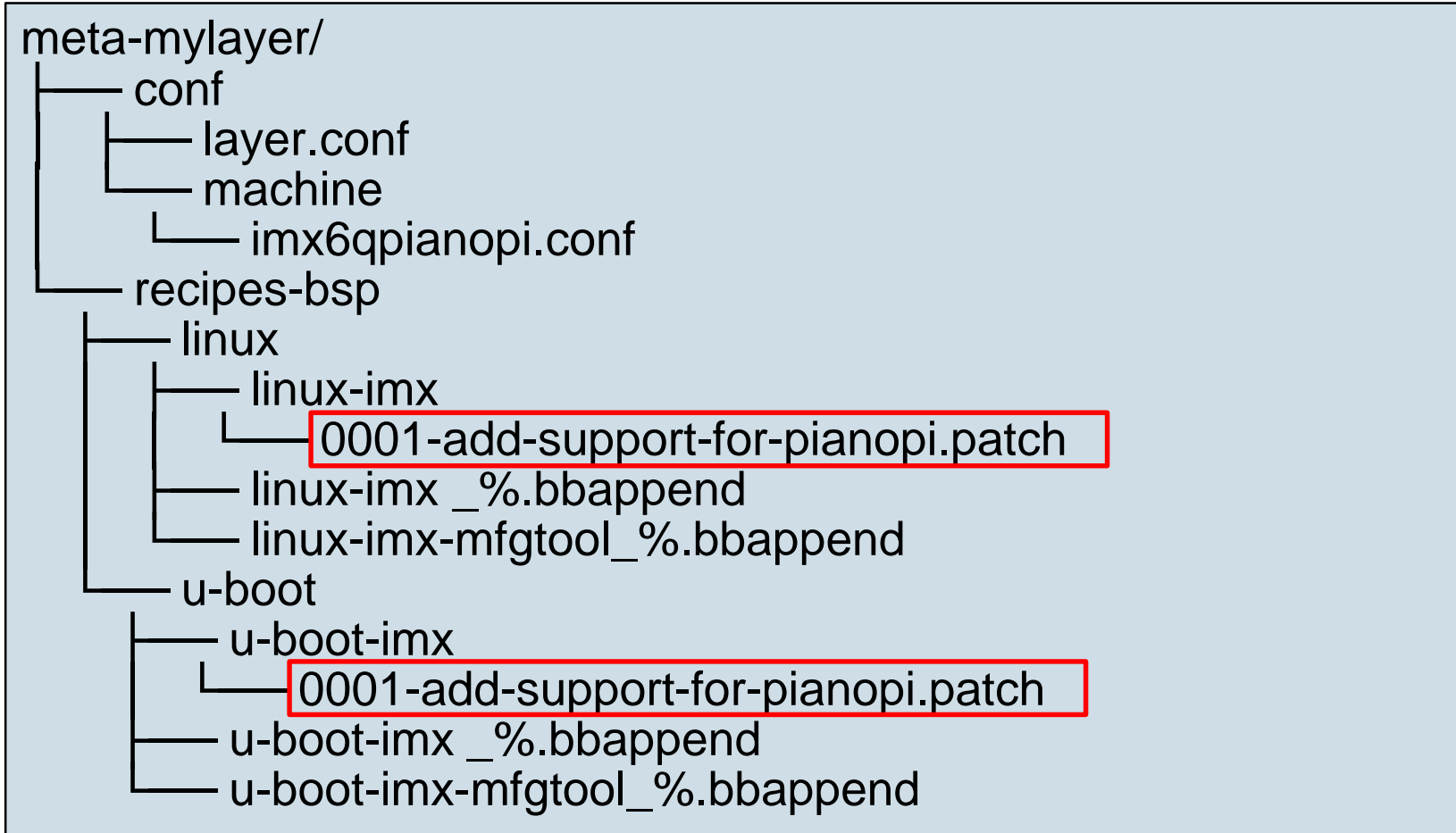
```
FILESEXTRAPATHS_prepend := "${THISDIR}/${PN}-${PV}:"
```

```
#  
# This .bbappend doesn't yet do anything - replace this text with  
# modifications to the example_0.1.bb recipe, or whatever recipe it is  
# that you want to modify with this .bbappend (make sure you change  
# the recipe name (PN) and version (PV) to match).  
#
```

```
└─ recipes-example-bbappend  
  └─ example-bbappend  
    └─ example-0.1  
      └─ example.patch  
    └─ example_0.1.bbappend
```

mylayer directory structure

- add a layer (mylayer) under directory sources/



1, u-boot and kernel patching

- layer.conf:

```
# We have a conf and classes directory, add to BBPATH
BBPATH .= ":{LAYERDIR}"

# We have recipes-* directories, add to BBFILES
BBFILES += "${LAYERDIR}/recipes-*/*/*.bb \
           ${LAYERDIR}/recipes-*/*/*.bbappend"

BBFILE_COLLECTIONS += "mylayer"
BBFILE_PATTERN_mylayer = "^${LAYERDIR}/"
BBFILE_PRIORITY_mylayer = "5"
```

This file is created manually.

1, u-boot and kernel patching (cont.)

- u-boot-imx_%.bbappend:

```
FILESEXTRAPATHS_prepend := "${THISDIR}/${PN}:"
```

```
SRC_URI += "file://0001-add-support-for-pianopi.patch"
```

- u-boot-imx-mfgtool_%.bbappend:

```
FILESEXTRAPATHS_prepend := "${THISDIR}/u-boot-imx:"
```

```
SRC_URI += "file://0001-add-support-for-pianopi.patch"
```

1, u-boot and kernel patching (cont.)

- linux-imx_%.bbappend:

```
FILESEXTRAPATHS_prepend := "${THISDIR}/${PN}:"
```

```
SRC_URI += "file://0001-add-support-for-pianopi.patch"
```

```
...
```

- if file name is *linux-imx_3.10.53.bbappend*, it also works as the preferred linux-imx version is 3.10.53. But suppose we bitbake *linux-imx_3.14.38.bb*, this patch will not be appended.
- we could add more than one patches here.

1, u-boot and kernel patching (cont.)

- linux-imx_%.bbappend:

```
FILESEXTRAPATHS_prepend := "${THISDIR}/${PN}:"
```

```
SRC_URI += "file://0001-add-support-for-pianopi.patch"
```

```
...
```

- linux-imx-mfgtool_%.bbappend:

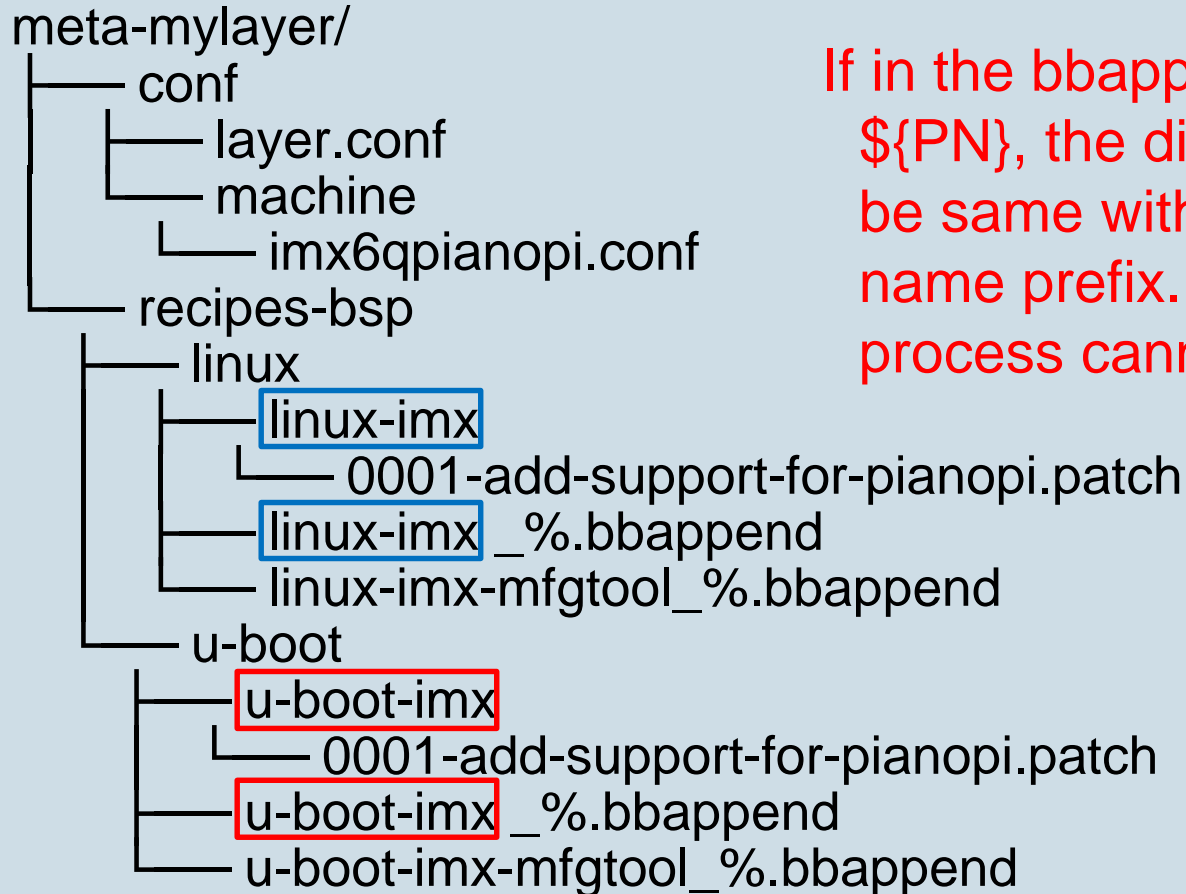
```
FILESEXTRAPATHS_prepend := "${THISDIR}/linux-imx:"
```

```
SRC_URI += "file://0001-add-support-for-pianopi.patch"
```

```
...
```

1, u-boot and kernel patching (cont.)

- add a layer (mylayer) under directory sources/



If in the bbappend file, the dir use $\${PN}$, the directory name should be same with the bbappend file name prefix. Or, the bitbake process cannot get the patch file.

2, support new machine

- add a layer (mylayer) under directory sources/

```
meta-mylayer/  
├── conf  
│   ├── layer.conf  
│   └── machine  
│       └── imx6qpianopi.conf  
├── recipes-bsp  
│   ├── linux  
│   │   ├── linux-imx  
│   │   │   └── 0001-add-support-for-pianopi.patch  
│   │   ├── linux-imx_%.bbappend  
│   │   └── linux-imx-mfgtool_%.bbappend  
│   └── u-boot  
│       ├── u-boot-imx  
│       │   └── 0001-add-support-for-pianopi.patch  
│       ├── u-boot-imx_%.bbappend  
│       └── u-boot-imx-mfgtool_%.bbappend
```


2, support new machines (cont.)

imx6qpianopi.conf:

```
...  
require conf/machine/include/imx6sabresd-common.inc  
  
SOC_FAMILY = "mx6:mx6q"  
KERNEL_IMAGETYPE = "zImage"  
KERNEL_DEVICETREE = "imx6q-pianopi.dtb"  
  
UBOOT_CONFIG ??= "emmc"  
UBOOT_CONFIG[emmc] = "mx6qpianopi"  
UBOOT_CONFIG[mfgtool] = "mx6qpianopi"
```

Kernel image type,
default ulmage

Device tree
to build

3, appoint new defconfig file

- *linux-imx_3.10.53.bb* has defined below function:

```
...
do_configure_prepend() {
# copy latest defconfig for imx_v7_defconfig to use
cp ${S}/arch/arm/configs/imx_v7_defconfig ${S}/.config
cp ${S}/arch/arm/configs/imx_v7_defconfig ${S}/../defconfig
fi
}
```

- however, PianoPI use *imx_v7_pianopi_defconfig* instead of *imx_v7_defconfig* due to the specific HW configuration (see next page).

3, appoint new defconfig file (cont.)

- Why we need new defconfig:

On mx6q sabresd board there is max11801 for touch screen. PianoPI board has no such component. So in the kernel the serial console will keep printing below:

FAIL max11801_client not initialize

or

max11801_ts 1-0048: FIFO_RD_AUX_MSB read fails

Besides, the release BSP original **imx_v7_defconfig** has incorporated almost all probably needed drivers, like SCSI, SPI-NOR, digital accelerometer, proximity/opto sensor, ambient light sensor... For PMIC, several vendor's PMIC drivers including PFUSE100 are built into the kernel. Actually on PianoPI board none of them are used.

So, we reconfigured base on the **imx_v7_defconfig** for PianoPI board, which is **imx_v7_pianopi_defconfig**.

3, appoint new defconfig file (cont.)

- in *linux-imx_%.bbappend* add below:

```
...
do_configure_append() {
if [ "${MACHINE}" = "imx6qpianopi" ] ; then
    # copy latest defconfig for imx_v7_pianopi_defconfig to
    use
    cp ${S}/arch/arm/configs/imx_v7_pianopi_defconfig
    ${S}/.config
    cp ${S}/arch/arm/configs/imx_v7_pianopi_defconfig
    ${S}/../defconfig
fi
}
```

3, appoint new defconfig file (cont.)

- Similarly, in *linux-imx-mfgtool_%.bbappend* add:

```
...
do_configure_append() {
if [ "${MACHINE}" = "imx6qpianopi" ] ; then
    # copy latest defconfig for imx_v7_pianopi_defconfig to
    use
    cp ${S}/arch/arm/configs/imx_v7_pianopi_mfg_defconfig
    ${S}/.config
    cp ${S}/arch/arm/configs/imx_v7_pianopi_mfg_defconfig
    ${S}/../defconfig
fi
}
```

the last step of adding a layer

After creating the build directory and before bitbaking any image, we need to modify build/conf/**bblayer.conf** to add below line:

```
BBLAYERS += "${BSPDIR}/sources/meta-mylayer"
```

Image Building Instructions

1. get Yocto L3.10.53_1.1.0_GA source code, following user's guide

```
mkdir ~/fsl-release-bsp
cd ~/fsl-release-bsp
git config --global user.name "vmuser"
git config --global user.email "vmuser@freescale.com"
git config --list
repo init -u git://git.freescale.com/imx/fsl-arm-yocto-bsp.git -b imx-
  3.10.53-1.1.0_ga
repo sync
```

2. apply the addon tarball in directory ~/fsl-release-bsp

```
tar xvjf fsl-release-bsp_L3.10.53_1.1.0_GA_addon.tar.bz2
```

Image Building Instructions (cont.)

3. configure for pianopi board:

```
MACHINE=imx6qpianopi source fsl-setup-release.sh -b  
build-x11 -e x11
```

4. modify build/conf/bblayers.conf, add below line

```
BBLAYERS += "${BSPDIR}/sources/meta-mylayer"
```

5. bitbake images

Commands	Generated Image
bitbake fsl-image-gui	Gui Image
bitbake fsl-image-qt5	QT5 Image
bitbake u-boot-imx	U-boot image only
bitbake linux-imx	Kernel and device tree only
bitbake fsl-image-mfgtool-initramfs	MFGTool firmware images: u-boot, kernel, device tree and init ramfs.



www.Freescale.com