iMX8MMINI DEMO

User Manual

Version: 1.6

2022-10-17

Revision History：

Version	Date	Description
1.0	2021-01-05	Initial Release
1.1	2021-02-23	Support HDMI
1.2	2021-05-24	Change eMMC Updating method
1.3	2022-05-28	Support IMX8MM-MIPI-ADAPTER
1.4	2022-06-21	Support MIPI-CSI OV5640
1.5	2022-08-18	Support CPT GT911
1.6	2022-10-17	Add Backlight Control

Table of Contents

1. Product Overview

1.1 Introduction

1.2 Resource Download

You can access to our remote server to download the hardware and softwar e resources through 2 ways below:

① Open web browser (Firefox is recommended), input the following url:

https://47.107.111.205/svn/NXP/IMX8MMINI_DEMO

If the browser reports a warning like "Potential Security Risk Ahead ", please choose the **Advanced...** option and "**Accept the Risk and Continue**".

Access Authorization:

User Name: guest

Password: guest

② Run svn command under Linux operating system, such as Debian and U buntu:

```
svn co svn://47.107.111.205/NXP/IMX8MMINI_DEMO
```

Note：

🕮 If the command not found, please install svn tools : apt-get update && apt-get install -y subversion

The svn program will ask for access authorization:

User Name: guest

Password: guest

1.3 Hardware features

1.4 Mechanical Demension

2. Linux Operation System

This chapter will give you a general map of the Linux software resources contained in the DVD-ROM provided along with the product, as well as detailed introduction to the process of Linux system development, drivers development, system update, functionality tests and application development examples.

Note：

📖 It is recommended to learn Ubuntu Linux installation and embedded Linux development technology before you get started.

2.1 Software Resources

The DVR-ROM provided along with the board contains demos, application examples, Linux source code and tools, helping you develop Linux applications and systems easily and quickly.

2.1.1 Locations of Resources

You can find software resources such as programs and codes contained in the DVD-ROM according to the information showed in the table below;

Categories	Location
Applications	
Source Code	CD\Source\u-boot-2020.04.git.tar.xz
	CD\Source\linux-5.4.47.git.tar.xz
Tools	CD\Tools\
Precompiled Images	CD\Image

2.1.2 BSP

The following table lists types and formats of the files contained in BSP;

Names		Note	Formats
BOOTLOADER	U-BOOT	MMC/SD	Source Code
		FAT	Source Code
		NET	Source Code
KERNEL	LINUX-5.4	Support ROM/CRAM/EXT4/FAT /NFS various of file system	Source Code
DEVICE DRIVER	SERIAL	Serials driver	Source Code
	RTC	Hardware RTC driver	Source Code
	NET	10/100M/1000M Ethernet driver	Source Code
	CAN	CAN bus driver	Source Code
	SPI	SPI driver	Source Code
	SPI FLASH	SPI Flash driver	Source Code
	I2C	I2C driver	Source Code
	LCD	MIPI-DSI driver	Source Code
	TOUCH SCREEN	I2C Resistive touch panel driver	Source Code
	MMC/SD	MMC/SD controller driver	Source Code
	USB OTG	USB OTG driver	Source Code
	AUDIO	Audio driver (supports audio r ecording and playback)	Source Code
	AUDIO SPDIF	SPDIF interface, playback	Source Code
	BUTTON	GPIO button driver	Source Code
	LED	LED driver	Source Code
	BUZZER	Buzzer driver	Source Code
	WIFI	SDIO WIFI dongle driver	Source Code
ROOTFS	YOCTO	Weston with Qt & Opencv	Image

2.2 Structure of Embedded Linux System

IMX8MMINI DEMO board is shipped with Linux-5.4 system in eMMC by default. This system consists of bootloader, kernel and rootfs. The following table shows the structure of embedded Linux system.

eMMC/SD			
Partition	MBR	FAT	EXT4

Image	Bootloader	DTB, Kernel	Yocto Rootfs

1) Bootloader is a program generated by u-boot compiling; its file name is **flash.bin**.
2) The kernel used in this document is Linux 5.4 and has been customized based on the hardware design.
3) Rootfs stores open-source system Yocto with EXT4 format.

2.3 Building Development Environment

Before the software development, users have to establish a Linux cross development environment on PC. This section will take **Ubuntu18.04** operating system as an example to describe how to establish a cross development environment.

It is strongly recommended to install necessary software packages for a newly installed Ubuntu through the following commands.

- **sudo apt-get update; sudo apt-get install -y build-essential git xz-utils ncurses-dev autoconf libtool automake texinfo bison flex libssl-dev libc6:i386 libncurses5:i386 libstdc++6:i386**

Note:

🕮 Each instruction has been put a bullets "•" before it to prevent confusion caused by the long instructions that occupy more than one line in the context.

🕮 Please note the SPACES within each instruction; Missing of any SPACE will cause failure when executing instructions.

2.3.1 Installing Cross Compilation Tools

We provide 2 cross-compilers under **Tools** directory:

① **gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu.tar.xz**

② **fsl-imx-wayland-glibc-x86_64-imx-image-full-aarch64-imx8mmddr4evk-toolchain-5.4-zeus.sh**

The item ① is mainly used to compile u-boot and kernel.

```
mkdir $HOME/tools
cd <YOUR_PATH>/Tools
tar -xvf gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu.tar.xz -C $HOME/tools
```

The item ② is used to compile user program based on Qt5 and Opencv etc.

```
sudo ./fsl-imx-wayland-glibc-x86_64-imx-image-full-aarch64-imx8mmddr4evk-toolchain-5.4-zeus.sh
```

It will extract and install under **/opt** directory, keep the default settings.

Start to compile an opencv4 example code with it:

```
source /opt/fsl-imx-wayland/5.4-zeus/environment-setup-aarch64-poky-linux
${CXX} example.cpp -lopencv_core -lopencv_imgproc -lopencv_highgui -lopencv_imgcodecs
```

2.3.2 Adding Environment Variables

Run the following commands to add them in the temporary environment variables:

```
export PATH=$HOME/tools/gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu/bin:$HOME/tools:$PATH
export ARCH=arm64
export CROSS_COMPILE=arm-linux-
```

Note：

- The instructions can be added in the .bashrc file located at the user directory, so that the addition of environment variables will be loaded automatically when the system is booting up;
- If you want to check the path, please use the instruction **printenv PATH**

2.4 Preparing the Source Code

Please refer to chapter <**1.2 Resource Download**> to get the development materials, You can get source code under **Source** directory.

- **tar -xvf u-boot-2020.04.git.tar.xz**
- **tar -xvf linux-5.4.47.git.tar.xz**

Then we can get the source code directory **u-boot-2020.04** and **linux-5.4.47**.

2.5 Compilation

1） Compiling Bootloader

Run the following instructions to compile bootloader:

- **cd u-boot-2020.04**
- **git checkout .**
- **vi make.sh**

```
export PATH=$HOME/tools/gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu/bin:
$HOME/tools:$PATH
export ARCH=arm64
export CROSS_COMPILE=arm-linux-

DESTDIR="/dev/shm/"
```

PATH: point to your cross-compiler installation directory.

DESTDIR: point to a directory to store the target image.

Please modify **PATH** and **DESTDIR** according to your local environment.

- **./make.sh**

After all the instructions are executed, you can find booting image named **flash.bin** under **DESTDIR** directory.

2） Compiling Kernel

Execute the following instructions to compile kernel:

- **cd linux-5.4.47**
- **git checkout .**

- **vi make.sh**

```
export PATH=$HOME/tools/gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu/bin:
$HOME/tools:$PATH
export ARCH=arm64
export CROSS_COMPILE=arm-linux-

DESTDIR="/dev/shm/"
```

PATH: point to your cross-compiler installation directory.

DESTDIR: point to a directory to store the target image.

Please modify **PATH** and **DESTDIR** according to your local environment.

- **make distclean**
- **./make.sh**

If it's successfully built, you can find kernel images named **Image** and **fsl-imx8mm-demo.dtb** under **DESTDIR** directory.

2.6 Linux System Customization

In order to satisfy different requirements of customers, designers commonly need to make some custom modification based on the default configuration of Linux kernel. This chapter will introduce the process of system customization with some examples.

2.6.1 Replace Kernel LOGO

- Prepare a picture suitable for your display screen size, named **my_logo.png** for example.
- Install some necessary programs under Ubuntu.
  - **sudo apt-get install netpbm gimp**
- Run command under Ubuntu desktop terminal:
  - **pngtopnm my_logo.png > linuxlogo.pnm**
  - **pnmquant 224 linuxlogo.pnm > linuxlogo224.pnm**
  - **pnmtoplainpnm linuxlogo224.pnm > logo_linux_clut224.ppm**

- Update Linux source code.

```
cp -f logo_linux_clut224.ppm <YOUR_PATH>/linux-5.4.47/drivers/video/logo/logo_linux_clut224.ppm
```

- Re-build the kernel.

```
make ARCH=arm64 distclean
./make.sh
```

Update the target file **Image** to the board, reboot and check the boot logo on the display screen.

2.6.2 Setting Configuration Menu

A default configuration file is provided under kernel source codes:

linux-5.4.47/arch/arm64/configs/imx8mm_demo_defconfig

Please execute the following instructions to enter the configuration menu:

```
cd linux-5.4.47
make ARCH=arm64 imx8mm_demo_defconfig
make ARCH=arm64 menuconfig
```

Note：

🕮 If an error occurs when command 'make ARCH=arm64 menuconfig' is executed, you might need to install 'ncurse' in the Ubuntu system, ‘ncurses’ is a character graphic library required to generate configuration menu. Please enter the following instruction to install the library:

```
sudo apt-get install libncurses5-dev
```

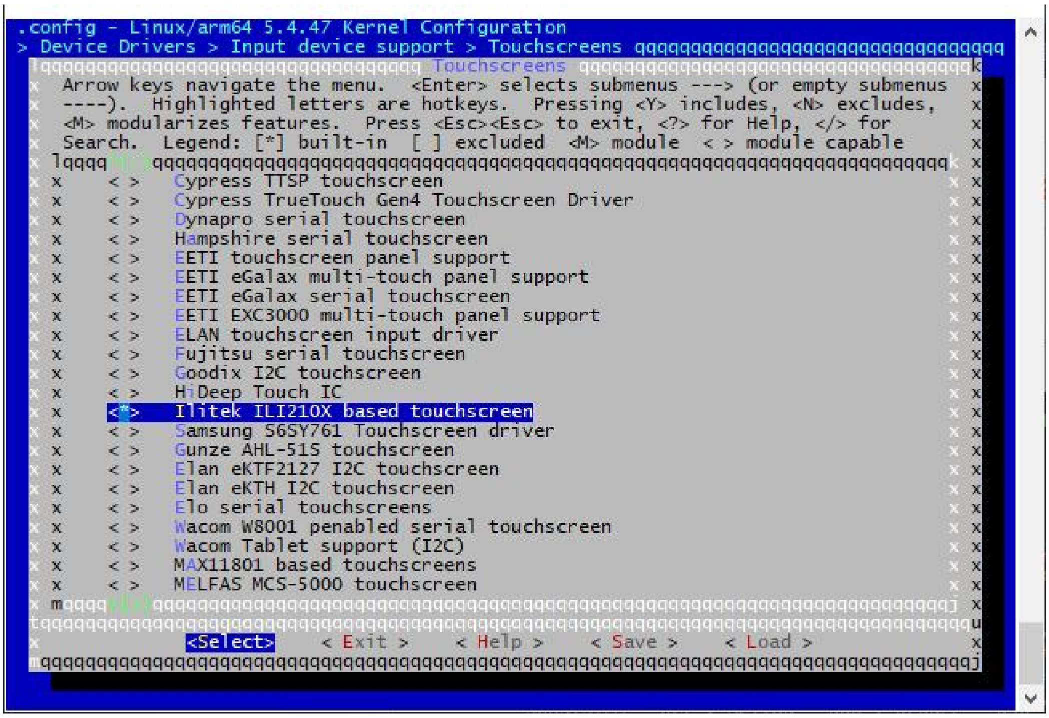
2.6.3 Menu Options

Configure options according to customization requirements after entering configuration menu, for example, access **Device Drivers** > **Input device support** > **Touchscreens** > **Ilitek ILI210X based touchscreen** as shown below:

```
-> Device Drivers
    -> Input device support
```

-> Touchscreens

-> Ilitek ILI210X based touchscreen



```
.config - Linux/arm64 5.4.47 Kernel Configuration
> Device Drivers > Input device support > Touchscreens qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
  lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq Touchscreens qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk
  x  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus  x
  x  ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,   x
  x  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for        x
  x  Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable       x
  x lqqqq(-)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk x
  x x     < >   Cypress TTSP touchscreen                                             x x
  x x     < >   Cypress TrueTouch Gen4 Touchscreen Driver                            x x
  x x     < >   Dynapro serial touchscreen                                           x x
  x x     < >   Hampshire serial touchscreen                                         x x
  x x     < >   EETI touchscreen panel support                                       x x
  x x     < >   EETI eGalax multi-touch panel support                                x x
  x x     < >   EETI eGalax serial touchscreen                                       x x
  x x     < >   EETI EXC3000 multi-touch panel support                               x x
  x x     < >   ELAN touchscreen input driver                                        x x
  x x     < >   Fujitsu serial touchscreen                                           x x
  x x     < >   Goodix I2C touchscreen                                               x x
  x x     < >   HiDeep Touch IC                                                      x x
  x x     <*>   Ilitek ILI210X based touchscreen                                     x x
  x x     < >   Samsung S6SY761 Touchscreen driver                                   x x
  x x     < >   Gunze AHL-51S touchscreen                                            x x
  x x     < >   Elan eKTF2127 I2C touchscreen                                        x x
  x x     < >   Elan eKTH I2C touchscreen                                            x x
  x x     < >   Elo serial touchscreens                                              x x
  x x     < >   Wacom W8001 penabled serial touchscreen                              x x
  x x     < >   Wacom Tablet support (I2C)                                           x x
  x x     < >   MAX11801 based touchscreens                                          x x
  x x     < >   MELFAS MCS-5000 touchscreen                                          x x
  x mqqqq(+)qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj x
  tqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqu
  x             <Select>    < Exit >    < Help >    < Save >    < Load >               x
  mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj
```

Set Ilitek ILI210X based touchscreen to **<*>**, exit and save changes.

2.6.4 Compile Kernel

Please execute the following instructions to recompile kernel:

- **./make.sh**

The script will **NOT** overwrite the configuration modifed by menuconfig. It mean s that your current setting is effective in your taget kernel image.

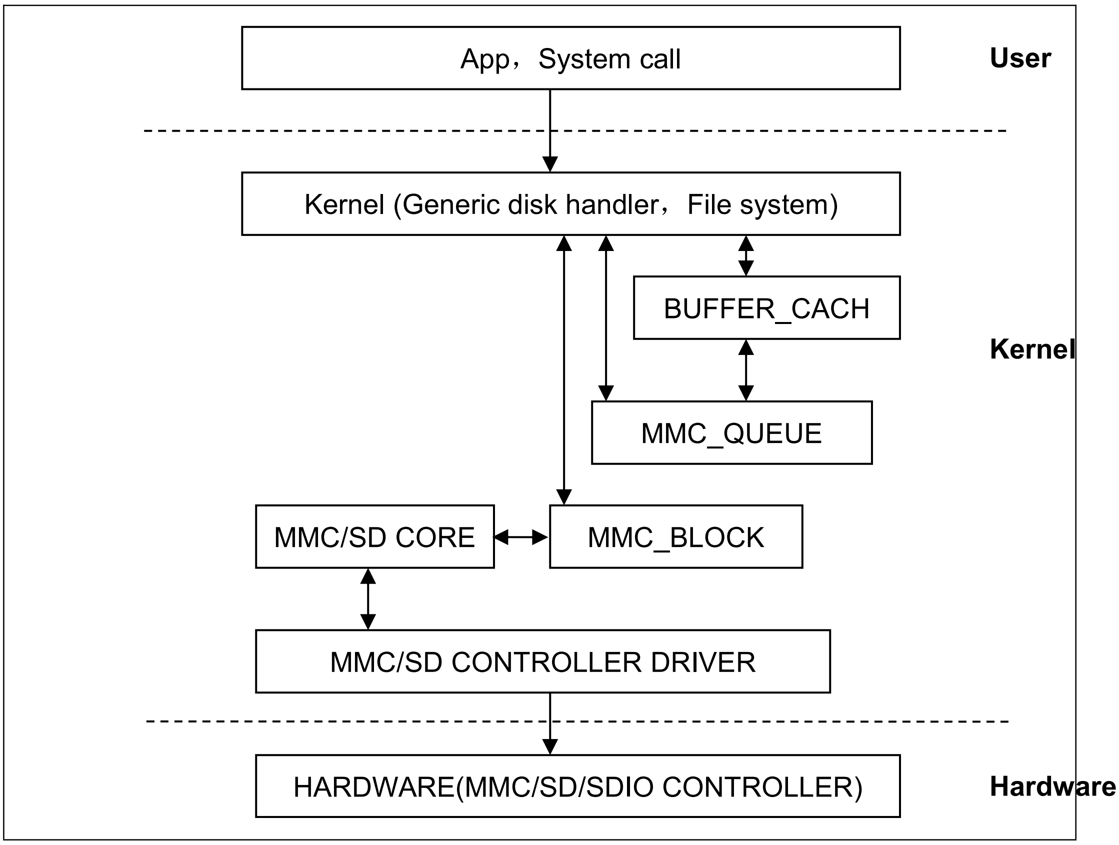
If you want to restore to the default configuration, please delete the file **.config** and run **make.sh**.

2.7 Introduction to Drivers

2.7.1 The table below shows the access path to find all the drivers :

Category	Name	Description	Location
Bootloader	u-boot	MMC/SD	drivers/mmc/fsl_esdhc_imx.c
		FAT	fs/
		NET	drivers/net/ fec_mxc.c
Kernel	Linux-5.4	Support ROM/CRAM/EXT4 /FAT/NFS	fs/
Devices	SERIAL	Serial driver	drivers/tty/serial/imx.c
	RTC	Hardware RTC driver	drivers/rtc/rtc-ds1307.c
	NET	10/100M/1000M Ethernet driver	drivers/net/ethernet/freescale/fec_main.c
	CAN	CAN bus driver	drivers/net/can/spi/mcp251x.c
	SPI	SPI driver	drivers/spi/spi-imx.c
	SPI FLASH	SPI Flash driver	drivers/mtd/spi-nor/spi-nor.c
	LCD	MIPI-DSI driver	drivers/gpu/drm/panel/panel-ronbo-rb070d30.c
	TOUCH SCREEN	I2C Resistive touch panel driver	drivers/input/touchscreen/ili210x.c
	MMC/SD	MMC/SD controller dirver	drivers/mmc/host/sdhci-esdhc-imx.c
	USB	USB controller dirver	drivers/usb/dwc3
	AUDIO	Audio driver (supporting recording/playback)	sound/soc/fsl/imx-wm8904.c
	AUDIO SPDIF	SPDIF interface, playback	sound/soc/fsl/imx-spdif.c
	BUTTON	GPIO button driver	drivers/input/keyboard/snvs_pwrkey.c
	LED	LED driver	drivers/leds/leds-gpio.c
	BUZZER	Buzzer driver	drivers/leds/leds-gpio.c
	WIFI	SDIO wifi dongle driver	drivers/net/wireless/broadcom/brcm80211/brcmfmac

2.7.2 SD/MMC



SD/MMC drivers in Linux are mainly consisted of SD/MMC core, mmc_block, mmc_queue and SD/MMC driver：

1） SD/MMC core realizes the codes unrelated to structure in the SD/MMC card operation;

2） mmc_block realizes driver structure when SD/MMC card is used as a block device;

3） mmc_queue realizes management of request queue;

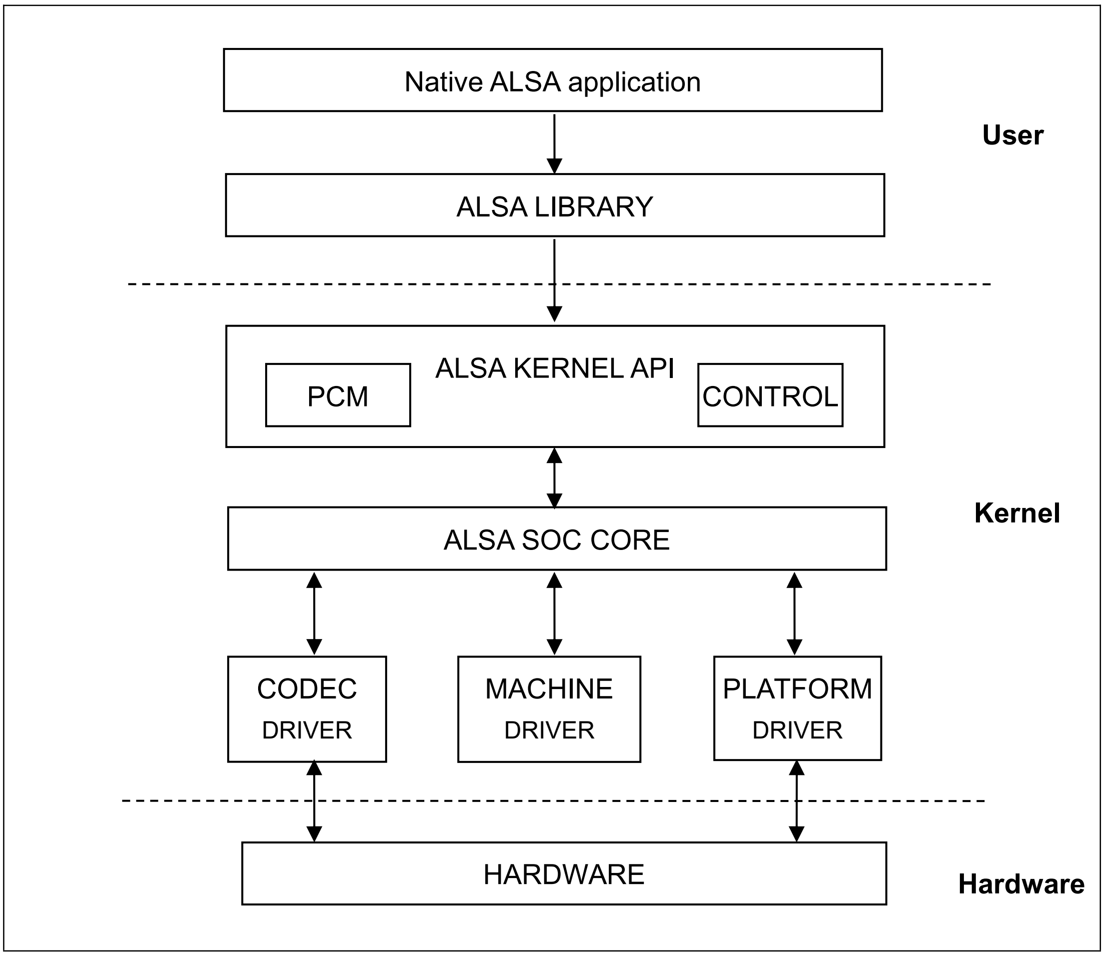
4） SD/MMC driver realizes specific controller driver.

Drivers and relevant documents:

linux-5.4.47/drivers/mmc/

linux-5.4.47/drivers/mmc/host/sdhci-esdhc-imx.c

2.7.3 Audio In/Out



ASoC embedded audio system basically consists of three components:

1) Codec driver: The codec driver is platform independent and contains audio controls, audio interface capabilities, codec dapm definition and codec IO functions.

2) Platform driver: It contains the audio dma engine and audio interface drivers (e.g. I2S, AC97, PCM) of that platform.

3) Machine driver: The machine driver handles any machine specific controls and audio events i.e. turning on an amp at start of playback.

Drivers and relevant documents:

linux-5.4.47/sound/soc/fsl

linux-5.4.47/sound/soc/fsl/imx-wm8904.c

linux-5.4.47/sound/soc/fsl/imx-spdif.c

2.8 Driver development

2.8.1 GPIO_LEDs Driver

1）Device Definition

linux-5.4.47/arch/arm64/boot/dts/freescale/fsl-imx8mm-demo.dts

Configure GPIO3.16 as system runing status indicator, blinking as heartbeat.

```
leds {
      compatible = "gpio-leds";
      pinctrl-names = "default";
      pinctrl-0 = <&pinctrl_gpio_led>;

      sys {
            label = "sys";
            gpios = <&gpio3 16 GPIO_ACTIVE_HIGH>;
            linux,default-trigger = "heartbeat";
      };
```

2）GPIO pinmux Configuration

linux-am335x/arch/arm/boot/dts/am335x-som860e.dts

Config GPIO2.5as MODE7(gpiomode),AM33XX_PIN_OUTPUT（output mode）

```
pinctrl_gpio_led: gpioledgrp {
      fsl,pins = <
            MX8MM_IOMUXC_NAND_READY_B_GPIO3_IO16            0x19
            ……
      >;
};
```

3）Driver Design

linux-5.4.47/drivers/leds/leds-gpio.c

a) Call platform_driver_register to register gpio_leds driver

```
static struct platform_driver gpio_led_driver = {
      .probe          = gpio_led_probe,
      .shutdown        = gpio_led_shutdown,
      .driver        = {
            .name     = "leds-gpio",
            .of_match_table = of_gpio_leds_match,
       },
};
```

```
module_platform_driver(gpio_led_driver);


MODULE_AUTHOR("Raphael Assenat <raph@8d.com>, Trent Piepho <tpiepho@freesc
ale.com>");
MODULE_DESCRIPTION("GPIO LED driver");
MODULE_LICENSE("GPL");
MODULE_ALIAS("platform:leds-gpio");
```

b) Apply for gpio and call led_classdev_register to led_classdev drivr.

```
static int gpio_led_probe(struct platform_device *pdev)
{
...
    priv->num_leds = pdata->num_leds;
        for (i = 0; i < priv->num_leds; i++) {
            const struct gpio_led *template = &pdata->leds[i];
            struct gpio_led_data *led_dat = &priv->leds[i];

            if (template->gpiod)
                led_dat->gpiod = template->gpiod;
            else
                led_dat->gpiod =
                    gpio_led_get_gpiod(&pdev->dev,
                               i, template);
            if (IS_ERR(led_dat->gpiod)) {
                dev_info(&pdev->dev, "Skipping unavailable LED gpio %d (%s)\n",
                     template->gpio, template->name);
                continue;
            }

            ret = create_gpio_led(template, led_dat,
                          &pdev->dev, NULL,
                          pdata->gpio_blink_set);
            if (ret < 0)
                return ret;
        }
    } else {
        priv = gpio_leds_create(pdev);
        if (IS_ERR(priv))
            return PTR_ERR(priv);
    }

    platform_set_drvdata(pdev, priv);
```

```
	return 0;
}

static int create_gpio_led(const struct gpio_led *template,
	struct gpio_led_data *led_dat, struct device *parent,
	struct fwnode_handle *fwnode, gpio_blink_set_t blink_set)
{
	struct led_init_data init_data = {};
	int ret, state;

	led_dat->cdev.default_trigger = template->default_trigger;
	led_dat->can_sleep = gpiod_cansleep(led_dat->gpiod);
	if (!led_dat->can_sleep)
		led_dat->cdev.brightness_set = gpio_led_set;
	else
		led_dat->cdev.brightness_set_blocking = gpio_led_set_blocking;
	led_dat->blinking = 0;
	if (blink_set) {
		led_dat->platform_gpio_blink_set = blink_set;
		led_dat->cdev.blink_set = gpio_blink_set;
	}
	if (template->default_state == LEDS_GPIO_DEFSTATE_KEEP) {
		state = gpiod_get_value_cansleep(led_dat->gpiod);
		if (state < 0)
			return state;
	} else {
		state = (template->default_state == LEDS_GPIO_DEFSTATE_ON);
	}
	led_dat->cdev.brightness = state ? LED_FULL : LED_OFF;
	if (!template->retain_state_suspended)
		led_dat->cdev.flags |= LED_CORE_SUSPENDRESUME;
	if (template->panic_indicator)
		led_dat->cdev.flags |= LED_PANIC_INDICATOR;
	if (template->retain_state_shutdown)
		led_dat->cdev.flags |= LED_RETAIN_AT_SHUTDOWN;

	ret = gpiod_direction_output(led_dat->gpiod, state);
	if (ret < 0)
		return ret;

	if (template->name) {
		led_dat->cdev.name = template->name;
```

```
            ret = devm_led_classdev_register(parent, &led_dat->cdev);
      } else {
            init_data.fwnode = fwnode;
            ret = devm_led_classdev_register_ext(parent, &led_dat->cdev,
                                          &init_data);
      }
      return ret;
}
```

c) Users may access the file named brightness under **/sys/class/leds/xxx/brightness**, and call gpio_led_set to configure LED status

```
static void gpio_led_set(struct led_classdev *led_cdev,
      enum led_brightness value)
{
...
                  gpiod_set_value(led_dat->gpiod, level);
}
```

2.9 System Update

IMX8MMINI DEMO can boot up from both TF card and eMMC, this section briefly introduce the process of system update on TF card and eMMC.

2.9.1 Update of TF Card System Image

1） Make A Bootable TF Card

a) Get the system image from **Image** directory, named as **imx-image-xxx.img.xz**, unxz it and create a raw image **imx-image-xxx.img**.

b) If you work under Windows system, please run **Tools/win32diskimager** to write the **imx-image-xxx.img** into TF Card. If you work under Linux system, please use **dd** command to write it into TF Card.

Image Name	Display Supported
imx-image-full-imx8mmddr4evk-xxx.img	MIPI-DSI
imx-image-full-imx8mmddr4evk-xxx-hdmi.img	HDMI

Note:

📖 The difference between **imx-image-full-imx8mmddr4evk-xxx.img** and **imx-image-full-imx8mmddr4evk-xxx-hdmi.img** is that the **fdt_file** in **uEnv.txt** chooses a different dtb.

2） Update U-Boot

If you've made some changes to the u-boot source code, and want to update it into TFCard, please run the below command:

```
dd if=<YOUR_PATH>/flash.bin of=/dev/sdx bs=1K seek=33 conv=notrunc
```

Note:

📖 /dev/sdx is the TFCard device node recognized under Ubuntu system.

3） Update Kernel

If you have modified the kernel source code, please update the dtb and Image under Partition 1 [FAT32] of the TF Card. That partition can be recognized by Windows or Linux.

4） Update Rootfs

Because EXT4 isn't accessible Under Windows, please mount the Partiton 2 of TF Card under Ubuntu, change the target file and umount the Card.

Note:

📖 Press down the button "**BOOT**" before power up, don't release it. Wait until booting message shows on the terminal [3 seconds at most], release the button.

2.9.2 Update eMMC with TFCard

- Make a bootable TFCard and boot up the system;
- Choose the target image [under directory **Image/**] and copy it into the USB disk. If it is **.xz** file, please unxz it to generate **.img** file.
- Install the USB disk on the ARM board, it will be automatically mounted under directory **/run/media/**, for example, the USB disk is recognized as **sda1**;
- Run command to start writing eMMC:

```
root@imx8mmddr4evk:~# umount /dev/mmcblk2*
```

- root@imx8mmddr4evk:~# **dd if=/run/media/sda1/imx-image-full-imx8mmddr4evk-xxx.img of=/dev/mmcblk2**

After it's done, power off the board, remove the TFCard, then reboot the board, it should boot from eMMC and enter into Linux prompt.

2.9.3 Update eMMC with u-boot Command

- Connect MicroUSB slot on ARM board with PC;
- Make it boot from TF Card and enter into u-boot command mode;
- Run command:
  - u-boot=> **mmc list**

```
FSL_SDHC: 1 (SD)
FSL_SDHC: 2
```

Then we know "mmc2" is the target device eMMC.

- u-boot=> **ums 0 mmc 2**

```
UMS: LUN 0, dev mmc 2, hwpart 0, sector 0x0, count 0x1d34000
\
```

Then the PC will discover a removable disk installed [USB mass storage disk].

- Run Win32DiskImager on PC side;
- Write the target system image **imx-image-full-imx8mmddr4evk-xxx-hdmi.img** into the USB mass storage disk;
- Wait until it completes, remove USB cable and TF Card, reboot the board and check the booting message.

2.10 Test and Demonstration

This section will run some tests on the peripheral devices.

POWER: **12V DC**

Debug Port: **UART2**, **115200 1N8**.

2.10.1 RTC

There is a RTC chip RX8025 on board, so the integrated RTC is disabled. So there is only one RTC accessible under system.

Let's set the current time to 2021-1-4 10:12,

- root@arm:~# **date -s "2021-1-4 10:12"; hwclock -w**

Reboot the board, and check the hardware RTC time with below command:

- root@arm:~# **hwclock**

2.10.2 Timezone Setting

Set Beijing Time for example,

- root@arm:~# **echo "Asia/Shanghai" > /etc/timezone**
- root@arm:~# **ln -sf /usr/share/zoneinfo/Asia/Shanghai /etc/localtime**
- root@arm:~# **sync**

Note:

📖 NXP Yocto image doesn't contain zoneinfo, copy **/usr/share/zoneinfo** under Ubuntu system to the board, and retry the above commands.

2.10.3 USB OTG

Connect UDisk to the OTG slot, it can be recognized as a removable disk.

Connect the OTG to the HOST PC[Ubuntu], it can be recognized as a RNDIS network device.

2.10.4 USB HUB

There are 4 USB host channels extended from USB0:

SIGNAL	USAGE
DN1 | DP1	
DN2 | DP2	4G module
DN3 | DP3	J24 USB slot
DN4 | DP4	J23 USB slot

Force the HUB to reset:

-
  ```
  root@arm:~# node=/sys/class/leds/usbhub_reset/brightness; echo 0 > $node;sleep 1;echo 1 > $node
  ```

```
[ 1967.294776] usb 1-1: USB disconnect, device number 3
[ 1967.299981] usb 1-1.2: USB disconnect, device number 4
( reseting ... )
[ 1030.068743] usb 1-1: new high-speed USB device number c
[ 1030.230896] hub 1-1:1.0: USB hub found
[ 1030.234947] hub 1-1:1.0: 4 ports detected
```

2.10.5 NETWORK

There is a 1Gbps network chip AR8035 on board.

-
  ```
  root@arm:~# ifconfig eth0
  ```

```
eth0: flags=-28669<UP,BROADCAST,MULTICAST,DYNAMIC>   mtu 1500
        ether 1c:ba:8c:98:8b:58   txqueuelen 1000   (Ethernet)
        RX packets 0   bytes 0 (0.0 B)
        RX errors 0   dropped 0   overruns 0   frame 0
        TX packets 0   bytes 0 (0.0 B)
        TX errors 0   dropped 0 overruns 0   carrier 0   collisions 0
        device interrupt 176
```

DHCP feature is enabled as default; the board can request a valid IP address from DHCP server in local network.

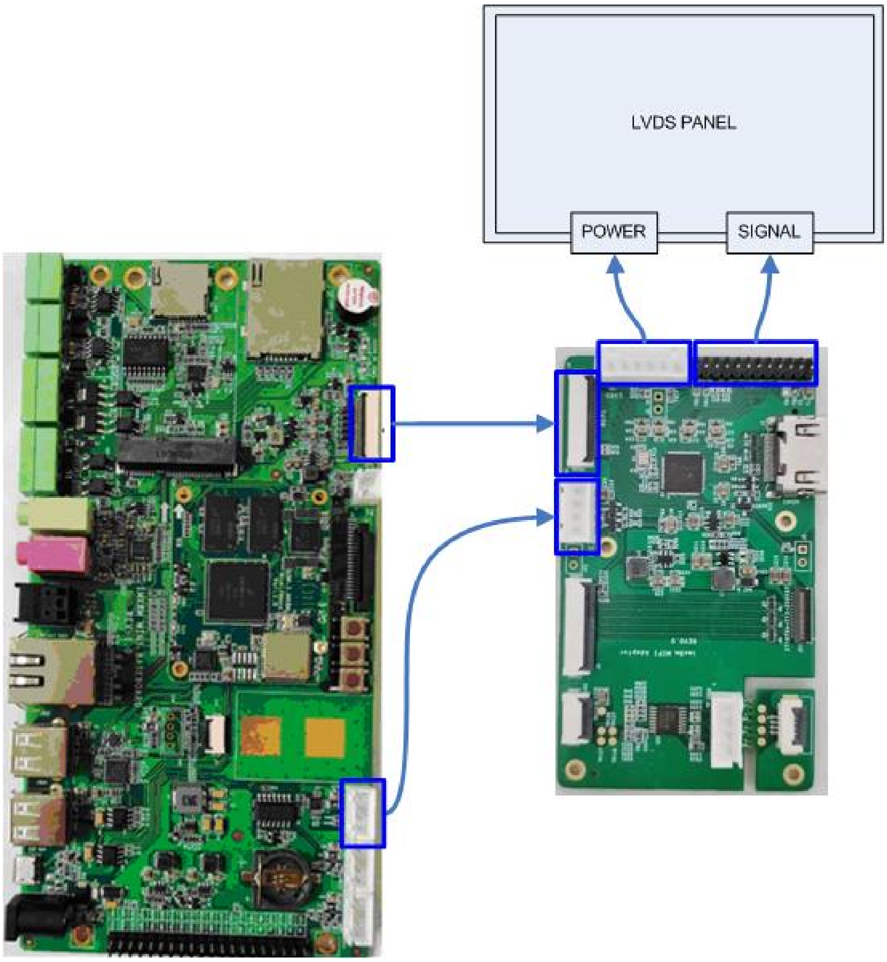
-
  ```
  root@arm:~# ping -I eth0 www.baidu.com
  ```

```
PING www.a.shifen.com (14.215.177.38) from 192.168.8.26 eth0: 56(84) bytes of d.
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=1 ttl=55 time=7.77 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 ttl=55 time=7.73 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 ttl=55 time=7.22 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=4 ttl=55 time=7.05 ms
^C
--- www.a.shifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
```

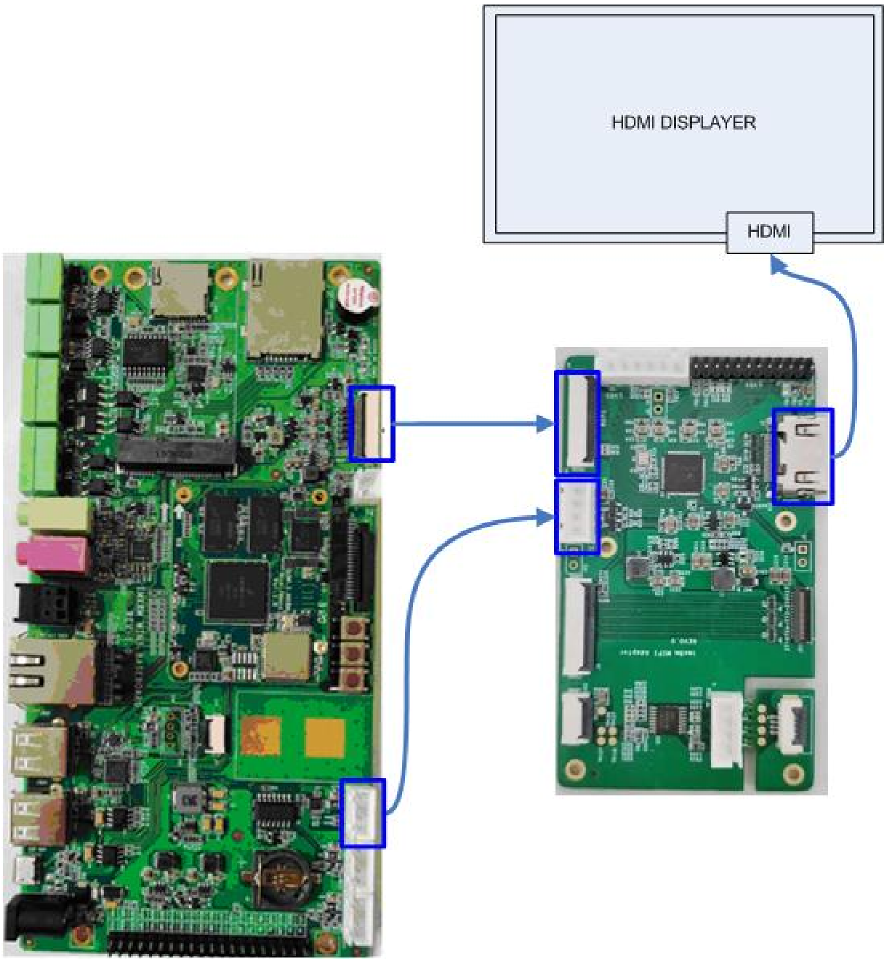
```
rtt min/avg/max/mdev = 7.058/7.447/7.771/0.319 ms
```

2.10.6 IMX8MM-MIPI-ADAPTER

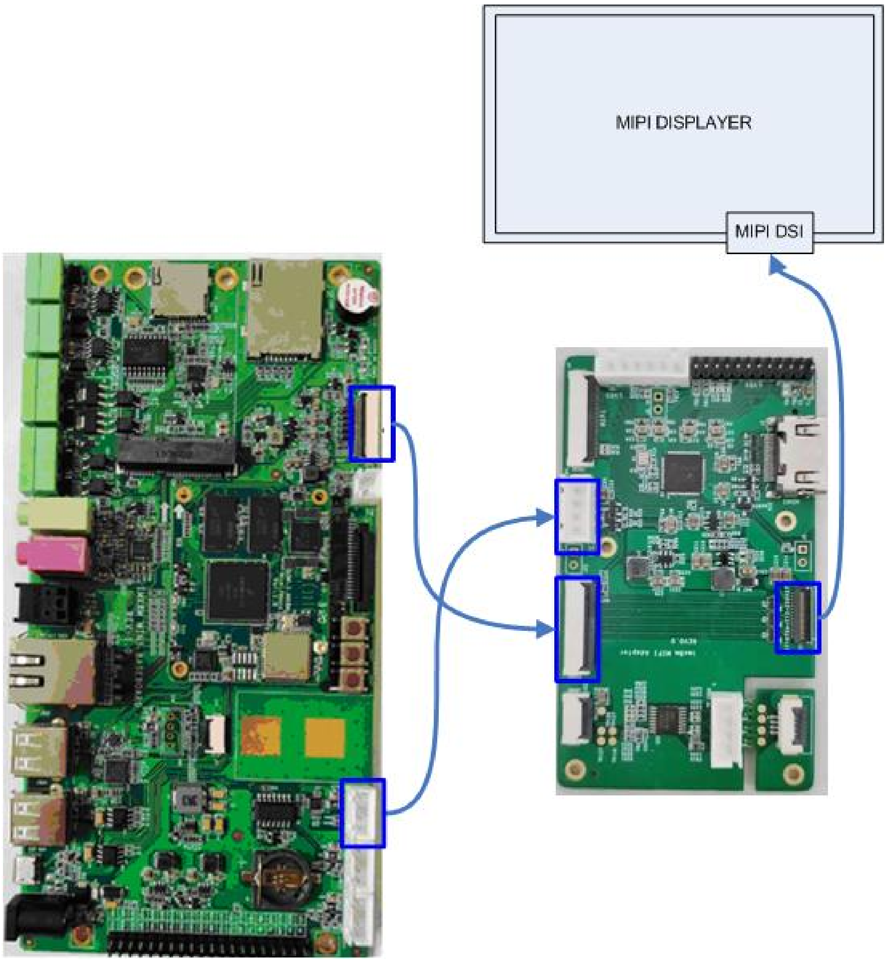
- LVDS



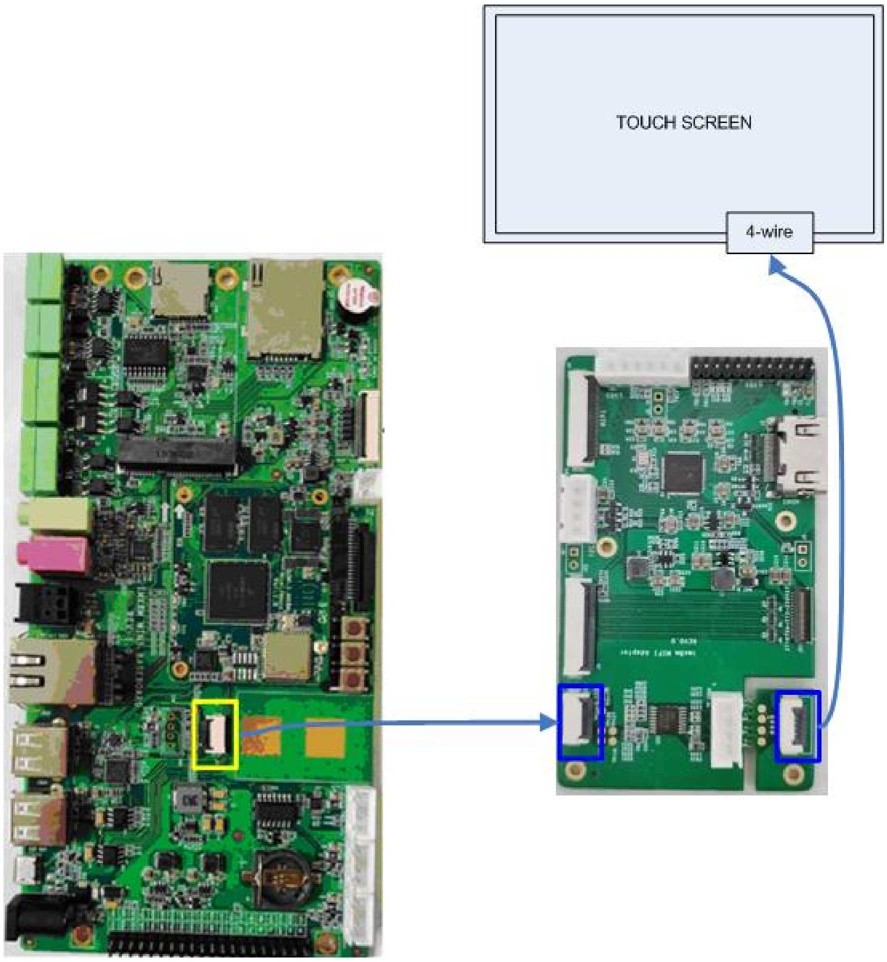
- HDMI



- MIPI



- Touch Screen



2.10.7 MIPI-DSI

The default configuration is tested with below devices:

TYPE	PANEL	RESOLUTION	DTB
MIPI-DSI	EK79007 VISLCD_070HYC530NCT61	1024 * 600	fsl-imx8mm-demo.dtb
MIPI-DSI	VISLCD-101HYSX31P	800 * 1280	fsl-imx8mm-demo-vislcd-mipi.dtb
LVDS	VISLCD-101BWX1	1280 * 800	fsl-imx8mm-demo-lt8912-lvds.dtb

User can modify **uEnv.txt** and choose the dtb file for the corresponding displayer.

Check the kernel source code:

- **vi arch/arm64/boot/dts/freescale/fsl-imx8mm-demo.dts**

```
&mipi_dsi {
        status = "okay";


        panel_ek79007: panel@0 {
```

```
        compatible = "ronbo,rb070d30";
        reg = <0>;
        vcc-lcd-supply = <&dummy_reg>;
        backlight = <&backlight_pwm>;
        status = "okay";
    };

    panel_vislcd_101hysx31p: panel@1 {
        compatible = "bananapi,lhr050h41", "ilitek,ili9881c";
        reg = <1>;
        power-supply = <&dummy_reg>;
        backlight = <&backlight_pwm>;
        status = "disabled";
    };

    panel_hdmi: port@1 {
        status = "disabled";

        mipi_dsi_hdmi_out: endpoint {
            remote-endpoint = <&lt8912_1_in>;
            attach-bridge;
        };
    };
};
```

```
    hdmi_bridge: lt8912b@48 {
        #address-cells = <1>;
        #size-cells = <0>;
        compatible = "lontium,lt8912";
        reg = <0x48>;
        lontium,dsi-lanes = <4>;
        vdd1-supply = <&dummy_reg>;
        no-hpd;
        no-edid;
        status = "disabled";

        port {
            lt8912_1_in: endpoint {
                remote-endpoint = <&mipi_dsi_hdmi_out>;
            };
        };
```

Rebuild the kernel, generate dtb and replace the image on board.

2.10.8 BACKLIGHT

- **vi arch/arm64/boot/dts/freescale/fsl-imx8mm-demo.dts**

```
    backlight_pwm: backlight {
        compatible = "pwm-backlight";
        pwms = <&pwm1 0 1250 0>;
        brightness-levels = <0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
23 ......   234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
253 254 255>;
        default-brightness-level = <200>;
    };
```

pwms = <&pwm1 0 **1250** 0>;

Sets the PWM signal period(ns):

$$T = 1/f$$

The target frequency we set is 800KHz, so:

$$T = 1/800000 = 1.25\text{e} - 06(\text{s}) = 1250(\text{ns})$$

brightness-levels =<...>

Assign the duty cycle value.

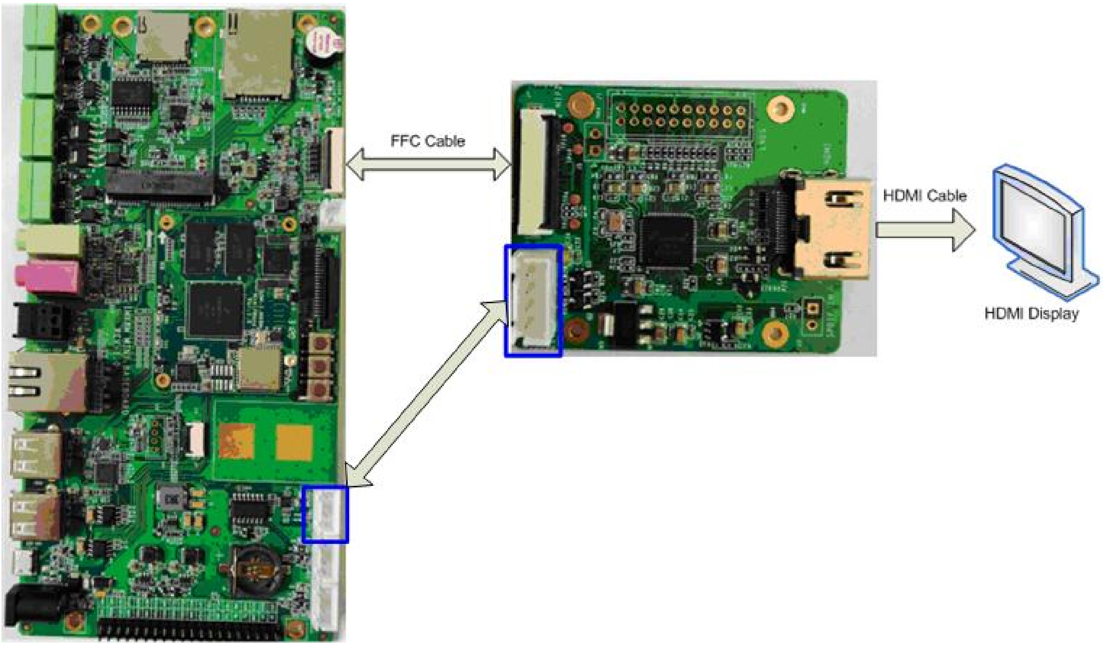
default-brightness-level

Set the default duty cycle value after booting up.

Set the backlight:

- root@arm:~# **echo 100 > /sys/class/backlight/backlight/brightness**

2.10.9 HDMI



The MIPI to HDMI chip LT8912B is already tested.

Check the kernel source code:

- **vi arch/arm64/boot/dts/freescale/fsl-imx8mm-demo-hdmi.dts**

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

&display_timings {
    native-mode = <&timing0>;          /* 1080p */
};

&panel_ek79007 {
    status = "disabled";
};

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

Rebuild the kernel, generate dtb and replace the image on board.

The HDMI EDID feature is not supported on our board. Please choose the target resolution according to your displayer manually:

- **vi arch/arm64/boot/dts/freescale/fsl-imx8mm-demo.dts**

```
        display-timings {
            native-mode = <&timing0>;

            /* 1920 * 1080 MIPI */
            timing0: timing0 {
                clock-frequency = <148500000>;
                hactive = <1920>;
                vactive = <1080>;
                hfront-porch = <88>;
                hsync-len = <44>;
                hback-porch = <148>;
                vfront-porch = <36>;
                vsync-len = <5>;
                vback-porch = <4>;
                hsync-active = <0>;
                vsync-active = <0>;
                de-active = <0>;
                pixelclk-active = <0>;
            };

            /* 1280 * 720 MIPI */
            timing1: timing1 {
                clock-frequency = <74250000>;
                hactive = <1280>;
                vactive = <720>;
                hfront-porch = <110>;
                hsync-len = <40>;
                hback-porch = <220>;
                vfront-porch = <5>;
                vsync-len = <5>;
                vback-porch = <20>;
                hsync-active = <0>;
                vsync-active = <0>;
                de-active = <0>;
                pixelclk-active = <0>;
            };

            /* 1280 * 800 MIPI */
```

```
                timing2: timing2 {
                        clock-frequency = <71000000>;
                        hactive = <1280>;
                        vactive = <800>;
                        hfront-porch = <48>;
                        hsync-len = <32>;
                        hback-porch = <80>;
                        vfront-porch = <3>;
                        vsync-len = <6>;
                        vback-porch = <14>;
                        hsync-active = <0>;
                        vsync-active = <0>;
                        de-active = <0>;
                        pixelclk-active = <0>;
                };
        };
```

Modify the **native-mode** with value: timing0, timing1 and timing2, to set the corresponding resolution.

2.10.10 CAPACITIVE TOUCH PANEL GT911

Device already tested: VISLCD_070HYC530NCT61 with GT911 CTP chip.

- **vi arch/arm64/boot/dts/freescale/fsl-imx8mm-demo.dts**

```
&i2c2 {
        ......
        touch@5d {
                compatible = "goodix,gt911";
                reg = <0x5d>;
                pinctrl-names = "default";
                pinctrl-0 = <&pinctrl_touch>;
                interrupt-parent = <&gpio3>;
                interrupts = <25 IRQ_TYPE_EDGE_FALLING>;
                irq-gpios = <&gpio3 25 GPIO_ACTIVE_HIGH>;
                reset-gpios = <&gpio5 9 GPIO_ACTIVE_LOW>;
                touchscreen-inverted-x;
                touchscreen-inverted-y;
                panel_is_vislcd_070hyc530nct61;
        };
};
```

- **make ARCH=arm64 menuconfig**

```
Device Drivers   --->
   Input device support   --->
      [*]   Touchscreens   --->
         <*>   Goodix I2C touchscreen
```

- Update driver source code: **drivers/input/touchscreen/goodix.c**, please refer to the latest kernel code released by Emtop Tech.
- Rebuild the kernel and get dtb files and '**Image**', update them on ARM board.
- Modify uEnv.txt:

```
fdt_file=fsl-imx8mm-demo.dtb
```

- Boot the board, login and run below commands to calibrator and test touch panel:

  - root@arm:~# **weston-touch-calibrator**

```
could not load cursor 'dnd-move'
could not load cursor 'dnd-copy'
could not load cursor 'dnd-none'
device "/sys/devices/platform/soc@0/soc@0:bus@30800000/30a30000.i2c/i2c-1/1-005
d/input/input3/event1" - head "DSI-1"
```

  - root@arm:~# **weston-touch-calibrator /sys/devices/platform/soc@0/soc@0:bus@30800000/30a30000.i2c/i2c-1/1-005d/input/input3/event1**

The LCD will display symbol +, click it four times to calibrate touch panel.

  - root@arm:~# **weston-flower**

The LCD will display a flower figure, press it and drag within the window to check the touch point.

2.10.11 USB TOUCH

(To be continued)

2.10.12 SPDIF AUDIO

- root@arm:~# **aplay -L**

```
null
    Discard all samples (playback) or generate zero samples (capture)
pulse
    PulseAudio Sound Server
sysdefault:CARD=imxspdif
    imx-spdif, S/PDIF PCM snd-soc-dummy-dai-0
    Default Audio Device
sysdefault:CARD=wm8904audio
    wm8904-audio,
    Default Audio Device
```

Transform the digital audio to anolog voice signal with a SPDIF converter.

- root@arm:~# **aplay /usr/share/sounds/alsa/*.wav**

```
Playing WAVE '/usr/share/sounds/alsa/Front_Center.wav' : Signed 16 bit Little Eo
Playing WAVE '/usr/share/sounds/alsa/Front_Left.wav' : Signed 16 bit Little Endo
Playing WAVE '/usr/share/sounds/alsa/Front_Right.wav' : Signed 16 bit Little Eno
......
```

2.10.13 WM8904 AUDIO

Playback:

- root@arm:~# **aplay -D plughw:1,0 /usr/share/sounds/alsa/*.wav**

Record:

- root@arm:~# **amixer -c 1 set 'Left Capture Mux' IN1L**
- root@arm:~# **arecord -D plughw:1,0 -t wav -f cd test.wav**

Wait several seconds, Ctrl+C to terminate arecord program. Now, let's play it to check:

- root@arm:~# **aplay -D plughw:1,0 test.wav**

2.10.14 UART

Device Node	Hardware	Usage
/dev/ttymxc0	UART1	BLUETOOTH
/dev/ttymxc1	UART2	DEBUG PORT
/dev/ttymxc2	UART3	RS485
/dev/ttymxc3	UART4	

There is only UART4 can be tested here. Connect TXD and RXD to run loopback test:

- root@arm:~# **/test/com -d /dev/ttymxc3**

```
SEND: 1234567890
RECV: 1234567890
SEND: 1234567890
RECV: 1234567890
```

2.10.15 RS485

Connect a RS485 device, or connect 2 boards directly.

- root@arm:~# **/test/com -d /dev/ttymxc2 -m rs485**

```
SEND: 1234567890
RECV: 1234567890
SEND: 1234567890
RECV: 1234567890
```

2.10.16 POWER BUTTON

- root@arm:~# **evtest /dev/input/event0**

```
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x0 product 0x0 version 0x0
Input device name: "30370000.snvs:snvs-powerkey"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 116 (KEY_POWER)
Properties:
Testing ... (interrupt to exit)
Event: time 1591238021.080788, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 1591238021.080788, -------------- SYN_REPORT ------------
Event: time 1591238021.144791, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 1591238021.144791, -------------- SYN_REPORT ------------
```

```
Event: time 1591238021.544772, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 1591238021.544772, -------------- SYN_REPORT ------------
Event: time 1591238021.608776, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
```

2.10.17 LED

There is LED named D21, used as system running indicator under hearbeat blinking mode. But we can change its mode manually.

- root@arm:~# **echo none > /sys/class/leds/sys/trigger**
- root@arm:~# **while test 1; do echo 1 > /sys/class/leds/sys/brightness;sleep 1;echo 0 > /sys/class/leds/sys/brightness;sleep 1;done**

2.10.18 BUZZER

- root@arm:~# **while test 1; do echo 1 > /sys/class/leds/beep/brightness;sleep 1;echo 0 > /sys/class/leds/beep/brightness;sleep 1;done**

2.10.19 PCIe

Aready test a PCIe to USB3.0 module uPD72020x.

2.10.20 SPI FLASH

- root@arm:~# **cat /proc/mtd**

```
dev:    size   erasesize  name
mtd0: 00800000 00010000 "30bb0000.spi"
```

Erase:

- root@arm:~# **flash_erase /dev/mtd0 0 0**

Format:

- root@arm:~# **mkfs.ext4 /dev/mtdblock0**

```
mke2fs 1.45.3 (14-Jul-2019)
Creating filesystem with 8192 1k blocks and 2048 inodes
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done
```

Mount:

- `root@arm:~# mount /dev/mtdblock0 /mnt`

```
[ 2107.531052] EXT4-fs (mtdblock0): mounted filesystem with ordered data mode. )
[ 2107.539223] ext4 filesystem being mounted at /mnt supports timestamps until )
```

Create and write your data under directory **/mnt**, it will be stored in SPI FLASH.

2.10.21 TFCard

When booting from eMMC, the TFCard will be recognized as a removable disk device.

2.10.22 eMMC

eMMC is mainly used for storing system image, needless to test it manually.

2.10.23 Unique ID

- `root@arm:~# cat /sys/devices/soc0/soc_uid`

```
0C259209DAB3DAF3
```

2.10.24 CAN Bus

- `root@arm:~# ifconfig can0`

```
can0      Link encap:UNSPEC   HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-0
          NOARP   MTU:16   Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)   TX bytes:0 (0.0 B)
```

Configure parameters:

- root@arm:~# **ifconfig can0 down**
- root@arm:~# **ip link set can0 type can bitrate 125000**
- root@arm:~# **ip link set can0 type can restart-ms 100**
- root@arm:~# **ifconfig can0 up**

Start to listen:

- root@arm:~# **candump can0 &**

Send package:

- root@arm:~# **cansend can0 "5A1#1122334455667788"**

For more information, please refer to project can-utils.

2.10.25 WIFI

- root@arm:~# **ifconfig wlan0 up**

If it reports message: ***SIOCSIFFLAGS: Operation not possible due to RF-kill***, please try below command:

- root@arm:~# **rfkill unblock all**

Now, we can control it manually.

- root@arm:~# **ifconfig wlan0 up; iw wlan0 scan**

```
BSS f0:b0:52:70:e2:58(on wlan0)
        last seen: 214.948s [boottime]
        TSF: 0 usec (0d, 00:00:00)
        freq: 2447
        beacon interval: 100 TUs
        capability: ESS Privacy ShortPreamble ShortSlotTime (0x0431)
        signal: -70.00 dBm
        last seen: 15156 ms ago
        SSID: Embest_Guest
        Supported rates: 1.0* 2.0* 5.5* 11.0*
        DS Parameter set: channel 8
        Country: US     Environment: Indoor/Outdoor
                Channels [1 - 11] @ 36 dBm
        ERP: <no flags>
        Extended supported rates: 6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0
        HT capabilities:
```

```
                    Capabilities: 0x1ad
                            RX LDPC
                            HT20
                            SM Power Save disabled
                            RX HT20 SGI
                            TX STBC
                            RX STBC 1-stream
                            Max AMSDU length: 3839 bytes
                            No DSSS/CCK HT40
... ...
```

- root@arm:~# **wpa_passphrase WIFI_AP 12345678 >> /etc/wpa_supplicant.conf**
- root@arm:~# **systemctl restart wpa_supplicant.service**

If everything works fine, it will get IP after several seconds.

- root@arm:~# **ifconfig wlan0**

```
wlan0       Link encap:Ethernet   HWaddr d0:c5:d3:d0:9c:33
            inet addr:192.168.52.101   Bcast:192.168.52.255   Mask:255.255.255.0
            inet6 addr: fe80::d2c5:d3ff:fed0:9c33/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST DYNAMIC   MTU:1500   Metric:1
            RX packets:60 errors:0 dropped:0 overruns:0 frame:0
            TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:7380 (7.2 KiB)   TX bytes:12849 (12.5 KiB)
```

Now you can do some connection test.

- root@arm:~# **sync; reboot**

Next boot, turn it on:

- root@arm:~# **rfkill unblock all; ifconfig wlan0 up**

Wait a while for getting IP:

- root@arm:~# **ifconfig wlan0**

```
wlan0       Link encap:Ethernet   HWaddr d0:c5:d3:d0:9c:33
            inet addr:192.168.52.101   Bcast:192.168.52.255   Mask:255.255.255.0
            inet6 addr: fe80::d2c5:d3ff:fed0:9c33/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST DYNAMIC   MTU:1500   Metric:1
            RX packets:60 errors:0 dropped:0 overruns:0 frame:0
            TX packets:94 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
```

```
          RX bytes:7380 (7.2 KiB)   TX bytes:12849 (12.5 KiB)
```

2.10.26 BLUETOOTH

- root@arm:~# **hciattach /dev/ttymxc0 bcm43xx 921600**

```
bcm43xx_init
Set Controller UART speed to 921600 bit/s
Flash firmware /etc/firmware/BCM4345C0.1MW.hcd
Set Controller UART speed to 921600 bit/s
Setting TTY to N_HCI line discipline
Device setup complete
```

- root@arm:~# **hciconfig -a**

```
hci0:    Type: Primary   Bus: UART
         BD Address: D0:C5:D3:F9:60:06   ACL MTU: 1021:8   SCO MTU: 64:1
         DOWN
         RX bytes:708 acl:0 sco:0 events:38 errors:0
         TX bytes:443 acl:0 sco:0 commands:38 errors:0
         Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
         Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
         Link policy: RSWITCH SNIFF
         Link mode: SLAVE ACCEPT
```

- root@arm:~# **rfkill unblock all**
- root@arm:~# **bluetoothctl**

```
Agent registered
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# scan on
Discovery started
[CHG] Controller D0:C5:D3:F9:60:06 Discovering: yes
[NEW] Device 63:EB:0D:5C:3D:F6 63-EB-0D-5C-3D-F6
[NEW] Device 51:02:9F:66:76:EC 51-02-9F-66-76-EC
[NEW] Device 78:C5:28:67:88:03 78-C5-28-67-88-03
[NEW] Device 7B:A2:1E:1D:15:60 7B-A2-1E-1D-15-60
.. ..
[bluetooth]# scan off
```

Please search bluetoothctl usage on web for more information.

2.10.27 4G

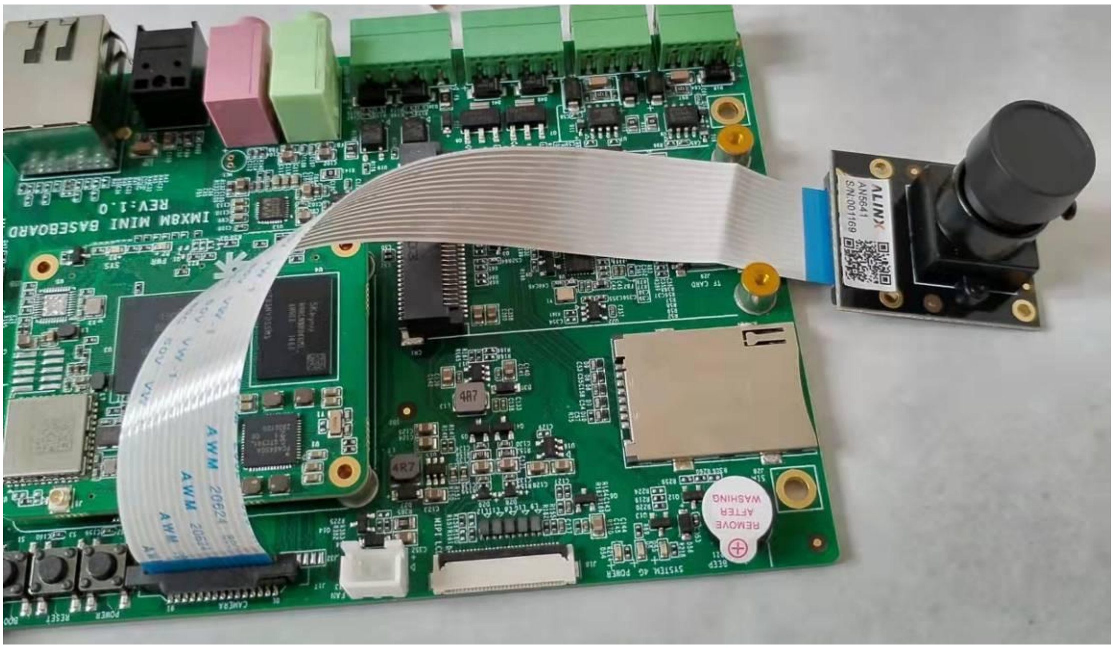
(To be continued)

2.10.28 MIPI-CSI Camera

Tested Devices: ALINX **AN5641** [**OV5640** inside]

Kernel Version: **linux-imx-5.4.47**

- Hardware Connection:



- Software Configuration:

  - **cd <YOUR_PATH>/linux-imx-5.4.47**
  - **vi arch/arm64/boot/dts/freescale/fsl-imx8mm-demo.dts**

```
&iomuxc {
    pinctrl-names = "default";
      pinctrl-0 = <&pinctrl_default>;
      ......
      pinctrl_csi_pwn: csi_pwn_grp {
              fsl,pins = <
                      MX8MM_IOMUXC_GPIO1_IO05_GPIO1_IO5               0x19
              >;
```

```
        };
};
```

```
&i2c3 {
    clock-frequency = <100000>;
    pinctrl-names = "default";
    pinctrl-0 = <&pinctrl_i2c3>;
    status = "okay";

    ......

    ov5640_mipi: ov5640_mipi@3c {
        compatible = "ovti,ov5640_mipi";
        reg = <0x3c>;
        status = "okay";
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_csi_pwn>;
        clocks = <&clk IMX8MM_CLK_CLKO1>;
        clock-names = "csi_mclk";
        assigned-clocks = <&clk IMX8MM_CLK_CLKO1>;
        assigned-clock-parents = <&clk IMX8MM_CLK_24M>;
        assigned-clock-rates = <24000000>;
        csi_id = <0>;
        pwn-gpios = <&gpio1 5 GPIO_ACTIVE_LOW>;
        mclk = <24000000>;
        mclk_source = <0>;
        port {
            ov5640_mipi1_ep: endpoint {
                remote-endpoint = <&mipi1_sensor_ep>;
            };
        };
    };
};
```

```
&csi1_bridge {
    fsl,mipi-mode;
    status = "okay";

    port {
        csi1_ep: endpoint {
            remote-endpoint = <&csi1_mipi_ep>;
        };
    };
};

&mipi_csi_1 {
    #address-cells = <1>;
    #size-cells = <0>;
    status = "okay";

    port {
        mipi1_sensor_ep: endpoint@1 {
            remote-endpoint = <&ov5640_mipi1_ep>;
            data-lanes = <2>;
            csis-hs-settle = <13>;
            csis-clk-settle = <2>;
            csis-wclk;
        };

        csi1_mipi_ep: endpoint@2 {
            remote-endpoint = <&csi1_ep>;
        };
    };
};
```

- **vi drivers/media/platform/mxc/capture/ov5640_mipi_v2.c [refer to the below patch to update it]**

```
--- a/drivers/media/platform/mxc/capture/ov5640_mipi_v2.c
+++ b/drivers/media/platform/mxc/capture/ov5640_mipi_v2.c
@@ -123,6 +123,7 @@ struct ov5640 {

        void (*io_init)(struct ov5640 *);
        int pwn_gpio, rst_gpio;
+       enum of_gpio_flags pwn_gpio_flags;
 };

 struct ov5640_res {
@@ -499,9 +500,9 @@ static inline void ov5640_power_down(struct ov5640 *sensor,
 int enable)
                return;
        if (!enable)
-               gpio_set_value_cansleep(sensor->pwn_gpio, 0);
+               gpio_set_value_cansleep(sensor->pwn_gpio, !!(sensor->pwn_gpio_flag
s & OF_GPIO_ACTIVE_LOW));
        else
-               gpio_set_value_cansleep(sensor->pwn_gpio, 1);
+               gpio_set_value_cansleep(sensor->pwn_gpio, !(sensor->pwn_gpio_flags
 & OF_GPIO_ACTIVE_LOW));

        msleep(2);
 }
@@ -551,13 +552,13 @@ static void ov5640_reset(struct ov5640 *sensor)
        gpio_set_value(sensor->rst_gpio, 1);

        /* camera power dowmn */
-       gpio_set_value(sensor->pwn_gpio, 1);
+       gpio_set_value(sensor->pwn_gpio, !(sensor->pwn_gpio_flags & OF_GPIO_ACT
IVE_LOW));
        msleep(5);

        gpio_set_value(sensor->rst_gpio, 0);
        msleep(1);

-       gpio_set_value(sensor->pwn_gpio, 0);
+       gpio_set_value(sensor->pwn_gpio, !!(sensor->pwn_gpio_flags & OF_GPIO_AC
TIVE_LOW));
        msleep(5);

        gpio_set_value(sensor->rst_gpio, 1);
@@ -1712,7 +1713,7 @@ static int ov5640_probe(struct i2c_client *client,
                dev_warn(dev, "No pin available\n");

        /* request power down pin */
-       sensor->pwn_gpio = of_get_named_gpio(dev->of_node, "pwn-gpios", 0);
+       sensor->pwn_gpio = of_get_named_gpio_flags(dev->of_node, "pwn-gpios", 0,
&(sensor->pwn_gpio_flags));
        if (!gpio_is_valid(sensor->pwn_gpio))
                dev_warn(dev, "No sensor pwdn pin available");
        else {
```

- **make ARCH=arm64 imx8mm_demo_defconfig**
- **make ARCH=arm64 menuconfig**

```
Device Drivers  --->
  <*> Multimedia support  --->
    [*]   V4L platform devices  --->
      <*>    mxc mipi csi driver
      MXC Camera/V4L2 PRP Features support  --->
        <*> OmniVision ov5640 camera support using mipi
```

- ➢ Rebuild the kernel to get dtb and Image, update them on ARM board.
- ➢ Boot and test:
  - root@arm:~# **gst-launch-1.0 v4l2src device=/dev/video0 ! video/x-raw,width=1920,height=1080 ! waylandsink window-width=1280 window-height=720**

```
Setting pipeline to PAUSED ...
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
[  154.230969] ov5640_mipi 2-003c: s_stream: 1
```

Now you can see the live video captured by camera on wayland desktop.