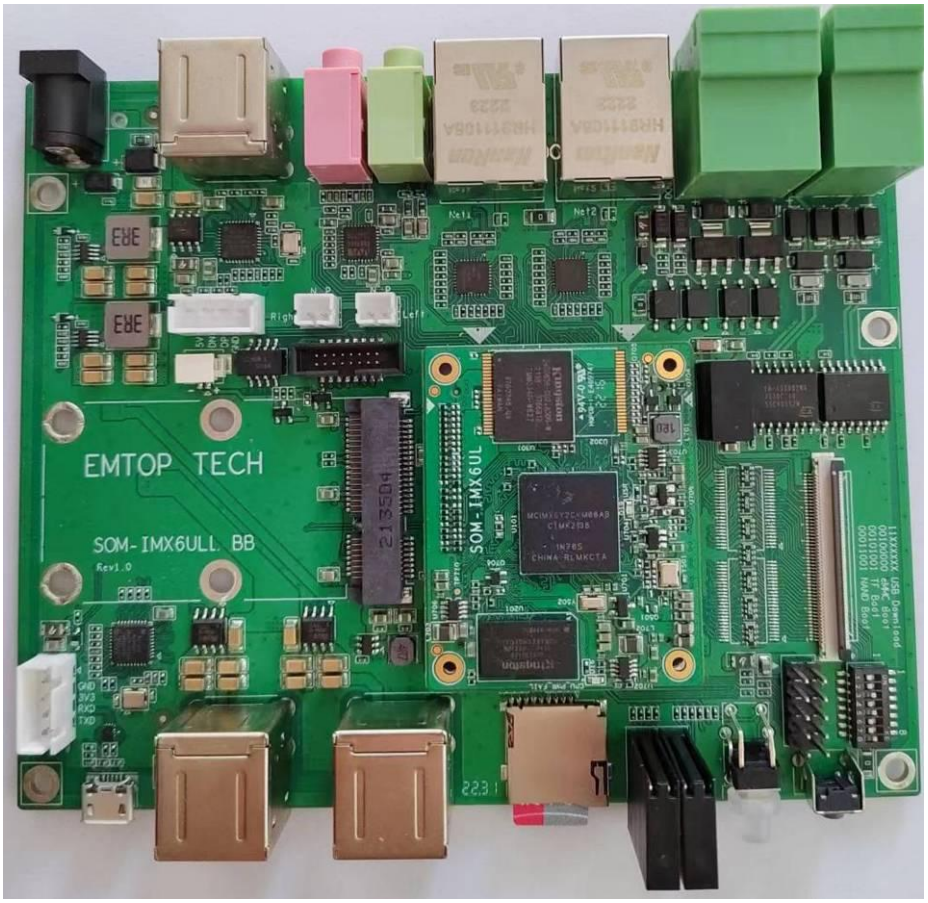
SOM-ET6ULL



SOM-ET6ULL + SOM-ET6ULL-BB

User Manual

Version: 0.1
2022-10-28

Revision History：

Version	Date	Description
0.1	2022-10-28	Initial Release

Table of Contents

1. Product Overview

1.1 Introduction

1.2 Resource Download

1.3 Hardware Features

1.4 Mechanical Dimension

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 in advance.

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 Location 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-imx-2016.03.git.tar.xz
	CD\Source\linux-imx-4.1.15.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-4.1.15	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 Ethernet driver	Source Code
	CAN	CAN bus driver	Source Code
	SPI	SPI driver	Source Code
	QSPI FLASH	QSPI Flash driver	Source Code
	I2C	I2C driver	Source Code
	LCD	TFT LCD driver	Source Code
	TOUCH SCREEN	I2C and TSC Resistive touch panel driver	Source Code
	MMC/SD	MMC/SD controller driver	Source Code
	USB OTG	USB OTG driver	Source Code
	USB HOST	USB HOST driver	Source Code
	AUDIO	WM8960 Audio driver(supports recording & playback)	Source Code
	BUTTON	GPIO button driver	Source Code
	LED	LED driver	Source Code
	BUZZER	Buzzer driver	Source Code
	WIFI	USB WIFI dongle driver	Source Code
ROOTFS	YOCTO	X11 with Qt5.6	Image

2.2 Structure of Embedded Linux System

SOM-ET6ULL board is shipped with Linux-4.1.15 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 **u-boot_sd.imx** or **u-boot_emmc.imx**.
2) The kernel used in this document is Linux 4.1.15 and has been customized according to the hardware design.
3) Rootfs stores open-source system Yocto with EXT4 format.

2.3 Building Development Environment

Before developing software, user has to establish a Linux cross development environment on PC. This section will take **Ubuntu20.04** 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 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 **fsl-imx-x11-gcc-5.3.0.tar.xz**

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

- **sudo tar -xvf fsl-imx-x11-gcc-5.3.0.tar.xz -C /opt**

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

2.3.2 Set Cross Compile Environment

Run the following commands to set the source code building environment:

- **source /opt/fsl-imx-x11/4.1.15-2.0.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi**
- **export ARCH=arm**
- **export CROSS_COMPILE=arm-poky-linux-gnueabi-**

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-imx-2016.03.git.tar.xz**
- **tar -xvf linux-imx-4.1.15.git.tar.xz**

Then we can get the source code directory **u-boot-imx-2016.03** and **linux-imx-4.1.15**.

2.5 Compilation

1） Compiling Bootloader

Run the following commands to compile bootloader:

- **cd u-boot-imx-2016.03**
- **git checkout .**
- **vi make.sh**

```
source /opt/fsl-imx-x11/4.1.15-2.0.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
export ARCH=arm
export CROSS_COMPILE=arm-poky-linux-gnueabi-

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

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

Change **DESTDIR** value to make it point to your target directory according to your local environment.

- **./make.sh**

After all the instructions are executed, you can find the booting images named **u-boot_emmc.imx** and **u-boot_sd.imx** under **DESTDIR** directory.

2） Compiling Kernel

Execute the following instructions to compile kernel:

- **cd linux-imx-4.1.15**
- **git checkout .**
- **vi make.sh**

```
source /opt/fsl-imx-x11/4.1.15-2.0.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
export ARCH=arm
export CROSS_COMPILE=arm-poky-linux-gnueabi-

DESTDIR="/dev/shm"
```

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

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

- **make ARCH=arm distclean**
- **./make.sh modules**

If it's successfully built, you can find kernel images named **.dtb** files, **zImage** and **lib/modules/4.1.15** under **DESTDIR** directory.

Note：

🕮 The command **./make.sh**, without parameter, only build dtbs and zImage; but **./make.sh modules** will build dtbs, zImage and driver modules.

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 U-BOOT LOGO

- Prepare a picture; any format which can be recognized by Photoshop is OK.
- Run Photoshop program, open and load the picture.
- Resize to make it suitable for your display screen size.
- Save as RGB888 BMP named **logo.bmp**.
- Copy **logo.bmp** into the FAT partition of the bootable SDCard.
- Install the bootable SDCard on arm board, power up and check the LOGO displayed on screen.

Note：

🕮 Modify **uEnv.txt** to choose the correct panel type. If the setting is wrong, the LCD will not display normally.

2.6.2 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-imx-4.1.15/drivers/video/logo/logo_linux_clut224.ppm**
- Re-build the kernel.
  - **make ARCH=arm distclean**
  - **./make.sh**

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

2.6.3 Setting Configuration Menu

A default configuration file is provided under kernel source codes:

linux-imx-4.1.15/arch/arm/configs/emtop-som-et6ull_defconfig

Please execute the following commands to enter the configuration menu:

- **cd linux-imx-4.1.15**
- **make ARCH=arm emtop-som-et6ull_defconfig**
- **make ARCH=arm menuconfig**

Note：

If an error occurs when command 'make ARCH=arm 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

2.6.4 Menu Options

Configure options according to customization requirements after entering configu

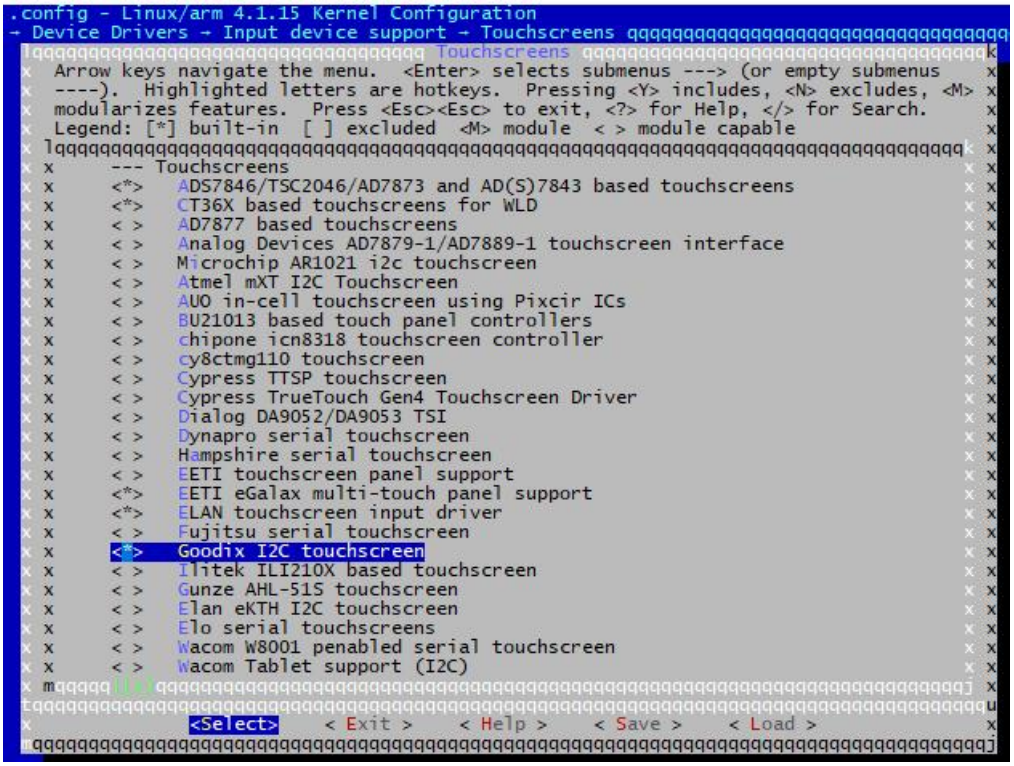
ration menu, for example, access **Device Drivers** > **Input device support** > **Touchscreens** > **Goodix I2C touchscreen** as shown below:

-> Device Drivers

-> Input device support

-> Touchscreens

-> Goodix I2C touchscreen



Set Goodix I2C touchscreen to **<*>**, exit and save changes.

2.6.5 Compile Kernel

Please execute the following instructions to recompile kernel:

- **./make.sh**

The script will **NOT** overwrite the configuration modified by menuconfig. It means that the current setting you modified is effective in your target kernel image.

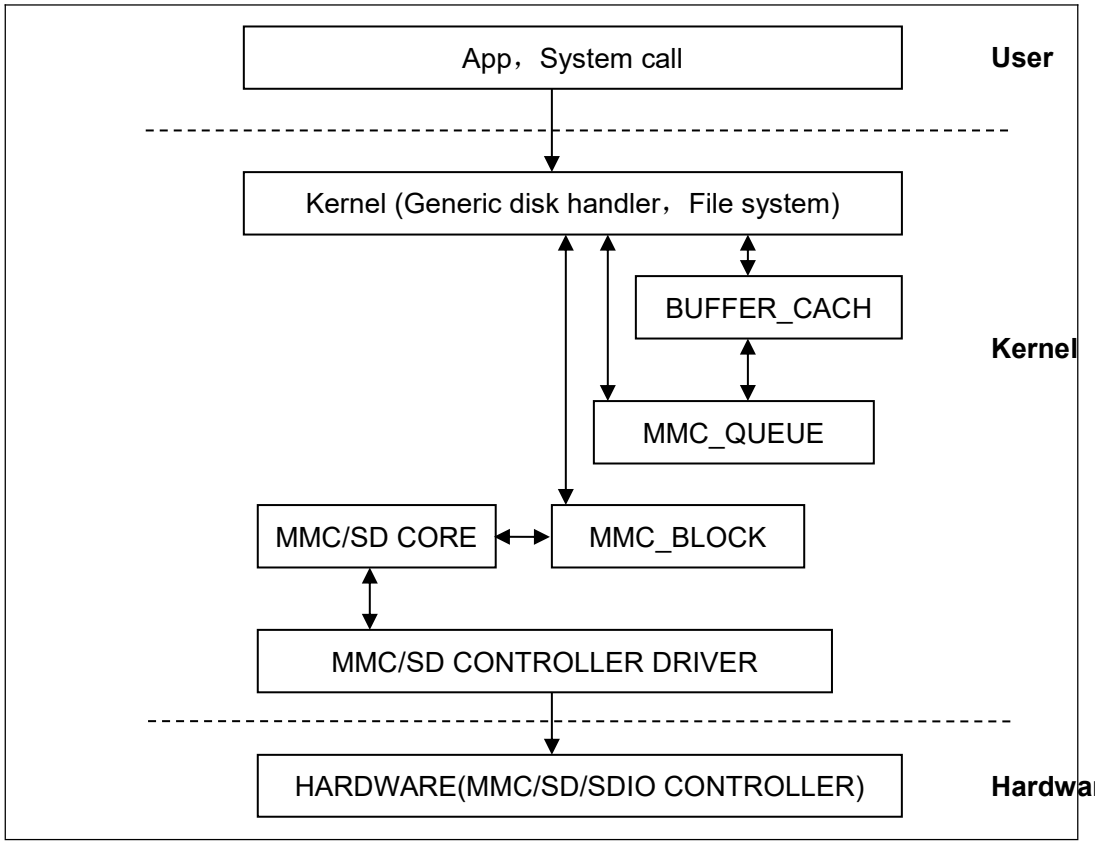
If you want to restore to the default configuration, please delete the file **.config** and run **./make.sh**.

2.7 Introduction to Drivers

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-4.1	Support ROM/CRAM/EXT4 /FAT/NFS	fs/
Devices	SERIAL	Serial driver	drivers/tty/serial/imx.c
	RTC	Hardware RTC driver	drivers/rtc/rtc-rx8010.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/video/fbdev/mxsfb.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/host/ehci-mxc.c
	AUDIO	Audio driver (supporting recording/playback)	sound/soc/fsl/imx-wm8960.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	USB WiFi dongle driver	drivers/net/wireless/realtek
	BLUETOOTH	USB Bluetooth driver	drivers/bluetooth/btusb.c
	PCIE	4G module driver	drivers/usb/serial/usb_wwan.c

2.7.1 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;

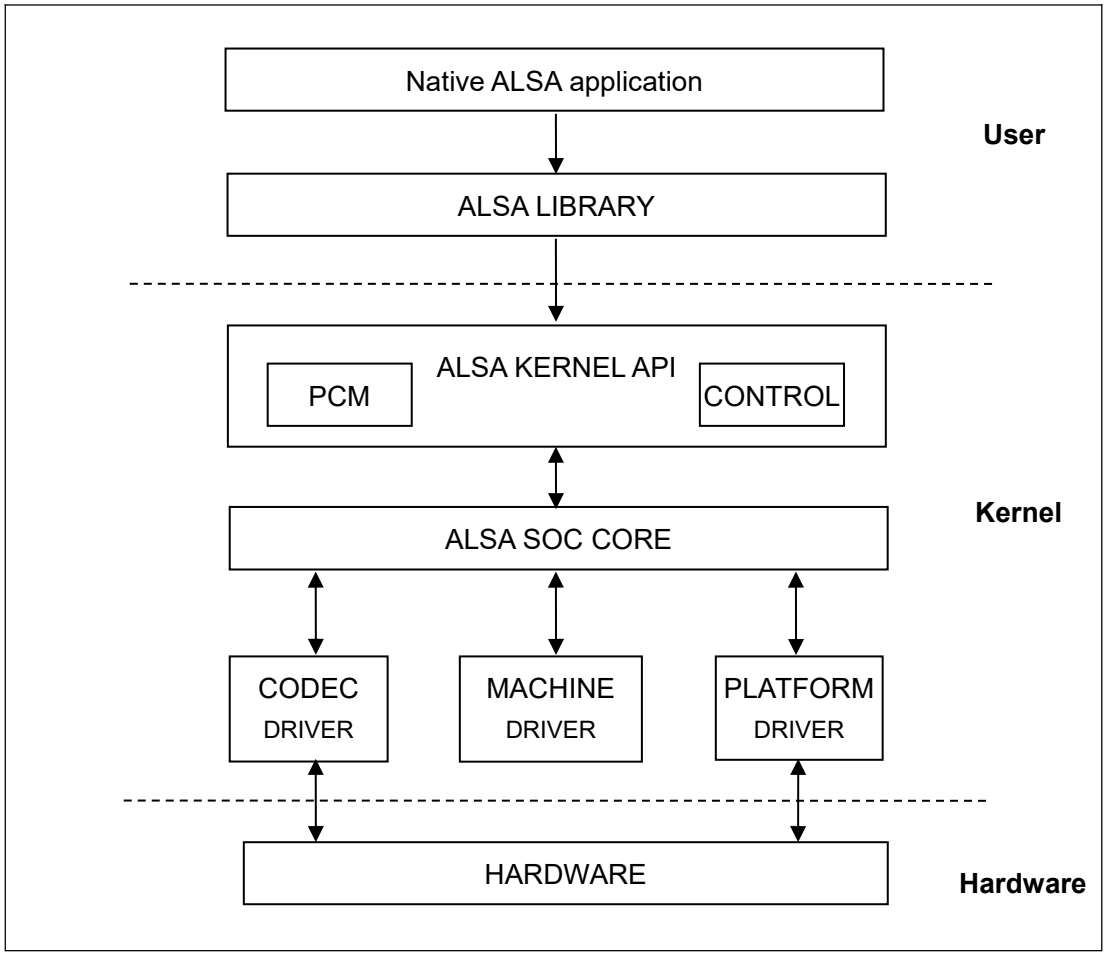
4） SD/MMC driver realizes specific controller driver.

Drivers and relevant documents:

linux-imx-4.1.15/drivers/mmc/

linux-imx-4.1.15/drivers/mmc/host/sdhci-esdhc-imx.c

2.7.2 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-imx-4.1.15/sound/soc/fsl

linux-imx-4.1.15/sound/soc/fsl/imx-wm8960.c

2.8 Driver development

2.8.1 GPIO_LEDs Driver

1）Device Definition

linux-imx-4.1.15/arch/arm/boot/dts/emtop-som-et6ull-emmc.dts

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

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

    sys {
        label = "sys";
        gpios = <&gpio5 9 GPIO_ACTIVE_LOW>;
        linux,default-trigger = "heartbeat";
    };
```

2）GPIO pinmux Configuration

linux-imx-4.1.15/arch/arm/boot/dts/emtop-som-et6ull-emmc.dts

Configure SNVS_TAMPER as GPIO5_IO09 function:

```
&iomuxc_snvs {
    ...
    imx6ul-evk {
        pinctrl_snvs_leds: snvs_leds {
            fsl,pins = <
                MX6UL_PAD_SNVS_TAMPER9__GPIO5_IO09          0xb0
            >;
        };
    };
};

/ {
    ...
    leds {
        compatible = "gpio-leds";
        pinctrl-names =  "default";
        pinctrl-0 =  <&pinctrl_snvs_leds >;
        status = "okay";
```

```
            sys    {
                label =       "sys";
                gpios =       <&gpio5     9 GPIO_ACTIVE_LOW>;
                linux,default-trigger =    "heartbeat";
            };
        };
        ...
};
```

3）Driver Design

linux-imx-4.1.15/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/sys/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);
}
```

2.8.2 Pinmux Configuration Guide

Let's take the pad GPIO1_IO00 as an example to explain the pinmux setting steps.

- **vi arch/arm/boot/dts/emtop-som-et6ull-emmc.dts**

```
&iomuxc {
        pinctrl_dummy: dummygrp {
                fsl,pins = <
                        MX6UL_PAD_GPIO1_IO00__GPIO1_IO00             0xb0
                >;
        };
};
```

The macro MX6UL_PAD_SNVS_TAMPER9__GPIO5_IO09 is defined in

arch/arm/boot/dts/imx6ul-pinfunc.h:

```
#define  MX6UL_PAD_GPIO1_IO00__GPIO1_IO00                                0x005C
0x02E8  0x0000  0x5  0x0
```

The value means:

mux_reg	conf_reg	input_reg	mux_mode	input_val
0x005C	0x02E8	0x0000	0x5	0x0

Usually we don't need to care about the value it defines, the only thing we need to do is to select the target function from the head file.

```
    #define  MX6UL_PAD_GPIO1_IO00__I2C2_SCL                              0x00
5C  0x02E8  0x05AC  0x0  0x1
    #define  MX6UL_PAD_GPIO1_IO00__GPT1_CAPTURE1                         0x
005C  0x02E8  0x058C  0x1  0x0
    #define  MX6UL_PAD_GPIO1_IO00__ANATOP_OTG1_ID                        0x
005C  0x02E8  0x04B8  0x2  0x0
    #define  MX6UL_PAD_GPIO1_IO00__ENET1_REF_CLK1                        0x
005C  0x02E8  0x0574  0x3  0x0
    #define  MX6UL_PAD_GPIO1_IO00__MQS_RIGHT                             0x0
05C  0x02E8  0x0000  0x4  0x0
    #define  MX6UL_PAD_GPIO1_IO00__GPIO1_IO00                            0x00
5C  0x02E8  0x0000  0x5  0x0
    #define  MX6UL_PAD_GPIO1_IO00__ENET1_1588_EVENT0_IN                  0x
005C  0x02E8  0x0000  0x6  0x0
    #define  MX6UL_PAD_GPIO1_IO00__SRC_SYSTEM_RESET                      0
x005C  0x02E8  0x0000  0x7  0x0
    #define  MX6UL_PAD_GPIO1_IO00__WDOG3_WDOG_B
0x005C  0x02E8  0x0000  0x8  0x0
```

You can refer to the below description in <**iMX6ULLRM.pdf**>

IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO00 field descriptions (continued)

Field	Description
	1 **ENABLED** — Force input path of pad GPIO1_IO00 0 **DISABLED** — Input Path is determined by functionality
MUX_MODE	MUX Mode Select Field. Select 1 of 9 iomux modes to be used for pad: GPIO1_IO00. 0000 **ALT0** — Select mux mode: ALT0 mux port: I2C2_SCL of instance: i2c2 0001 **ALT1** — Select mux mode: ALT1 mux port: GPT1_CAPTURE1 of instance: gpt1 0010 **ALT2** — Select mux mode: ALT2 mux port: ANATOP_OTG1_ID of instance: anatop 0011 **ALT3** — Select mux mode: ALT3 mux port: ENET1_REF_CLK1 of instance: enet1 0100 **ALT4** — Select mux mode: ALT4 mux port: MQS_RIGHT of instance: mqs 0101 **ALT5** — Select mux mode: ALT5 mux port: GPIO1_IO00 of instance: gpio1 0110 **ALT6** — Select mux mode: ALT6 mux port: ENET1_1588_EVENT0_IN of instance: enet1 0111 **ALT7** — Select mux mode: ALT7 mux port: SRC_SYSTEM_RESET of instance: src 1000 **ALT8** — Select mux mode: ALT8 mux port: WDOG3_WDOG_B of instance: wdog3

```
&iomuxc {
        pinctrl_dummy: dummygrp {
                fsl,pins = <
                        MX6UL_PAD_GPIO1_IO00__GPIO1_IO00          <PADCtrlValue>
                >;
        };
};
```

The **PADCtrlValue** is described in <**iMX6ULLRM.pdf**>

32.6.153 SW_PAD_CTL_PAD_GPIO1_IO00 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO00)

SW_PAD_CTL Register

Address: 20E_0000h base + 2E8h offset = 20E_02E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															HYS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PUS		PUE	PKE	ODE	Reserved			SPEED		DSE			Reserved		SRE
Reset	0	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO00 field descriptions

Field	Description
31–17 -	This field is reserved. Reserved
16 HYS	Hyst. Enable Field Select one out of next values for pad: GPIO1_IO00 0 **HYS_0_Hysteresis_Disabled** — Hysteresis Disabled 1 **HYS_1_Hysteresis_Enabled** — Hysteresis Enabled
15–14 PUS	Pull Up / Down Config. Field Select one out of next values for pad: GPIO1_IO00 00 **PUS_0_100K_Ohm_Pull_Down** — 100K Ohm Pull Down 01 **PUS_1_47K_Ohm_Pull_Up** — 47K Ohm Pull Up 10 **PUS_2_100K_Ohm_Pull_Up** — 100K Ohm Pull Up 11 **PUS_3_22K_Ohm_Pull_Up** — 22K Ohm Pull Up
13 PUE	Pull / Keep Select Field Select one out of next values for pad: GPIO1_IO00 0 **PUE_0_Keeper** — Keeper 1 **PUE_1_Pull** — Pull
12 PKE	Pull / Keep Enable Field Select one out of next values for pad: GPIO1_IO00 0 **PKE_0_Pull_Keeper_Disabled** — Pull/Keeper Disabled 1 **PKE_1_Pull_Keeper_Enabled** — Pull/Keeper Enabled
11 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO00

Field	Description
	0 **ODE_0_Open_Drain_Disabled** — Open Drain Disabled 1 **ODE_1_Open_Drain_Enabled** — Open Drain Enabled
10–8 -	This field is reserved. Reserved
7–6 SPEED	Speed Field Select one out of next values for pad: GPIO1_IO00 00 **SPEED_0_low_50MHz_** — low(50MHz) 01 **SPEED_1_medium_100MHz_** — medium(100MHz) 10 **SPEED_2_medium_100MHz_** — medium(100MHz) 11 **SPEED_3_max_200MHz_** — max(200MHz)
5–3 DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO00 000 **DSE_0_output_driver_disabled_** — output driver disabled; 001 **DSE_1_R0_260_Ohm___3_3V__150_Ohm_1_8V__240_Ohm_for_DDR_** — R0(260 Ohm @ 3.3V, 150 Ohm@1.8V, 240 Ohm for DDR) 010 **DSE_2_R0_2** — R0/2 011 **DSE_3_R0_3** — R0/3 100 **DSE_4_R0_4** — R0/4 101 **DSE_5_R0_5** — R0/5 110 **DSE_6_R0_6** — R0/6 111 **DSE_7_R0_7** — R0/7
2–1 -	This field is reserved. Reserved
0 SRE	Slew Rate Field Select one out of next values for pad: GPIO1_IO00 0 **SRE_0_Slow_Slew_Rate** — Slow Slew Rate 1 **SRE_1_Fast_Slew_Rate** — Fast Slew Rate

Sometimes, the PADCtrlValue can be set like 0x800xxxxx or 0x400xxxxx,

0x800xxxxx means no need to set its value, keep it as it was;

0x400xxxxx means to set SION bit, force input path of the pad.

2.9 System Update

SOM-ET6ULL board can boot up from both TF card and eMMC, this section br iefly 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.

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 partition 2 of TF Card under Ubuntu, change the target file and umount the Card.

Note:

Set the Boot Select Switch[SW2 on the base board] to "TF Boot", which is marked on the silk layer of the base board.

2.9.2 Update eMMC

Make a bootable TFCard and boot up the system, run below command to update eMMC:

- root@arm:~# **system-update.sh**

```
&iomuxc {
        pinctrl_dummy: dummygrp {
                fsl,pins = <
                        MX6UL_PAD_GPIO1_IO00__GPIO1_IO00          <PADCtrlValue>
                >;
```

```
        };
};
```

There are some updating messages show on terminal, and sys LED will blink.

After it's done, remove the TFCard, set the **Boot Select Switch[SW2 on the base board]** to "eMMC Boot" and reboot the board, it should boot from eMMC and enter into Linux prompt.

2.10 Test and Demonstration

This section will run some tests on the peripheral devices.

POWER: **12V DC**

Debug Port: **UART1**, **115200 1N8**.

2.10.1 RTC

There is a RTC chip RX8010 on board, so the integrated RTC is disabled by default. So there is only one RTC accessible under system.

Let's set the current time to 2022-10-10 10:12,

- root@arm:~# **date -s "2022-10-10 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**

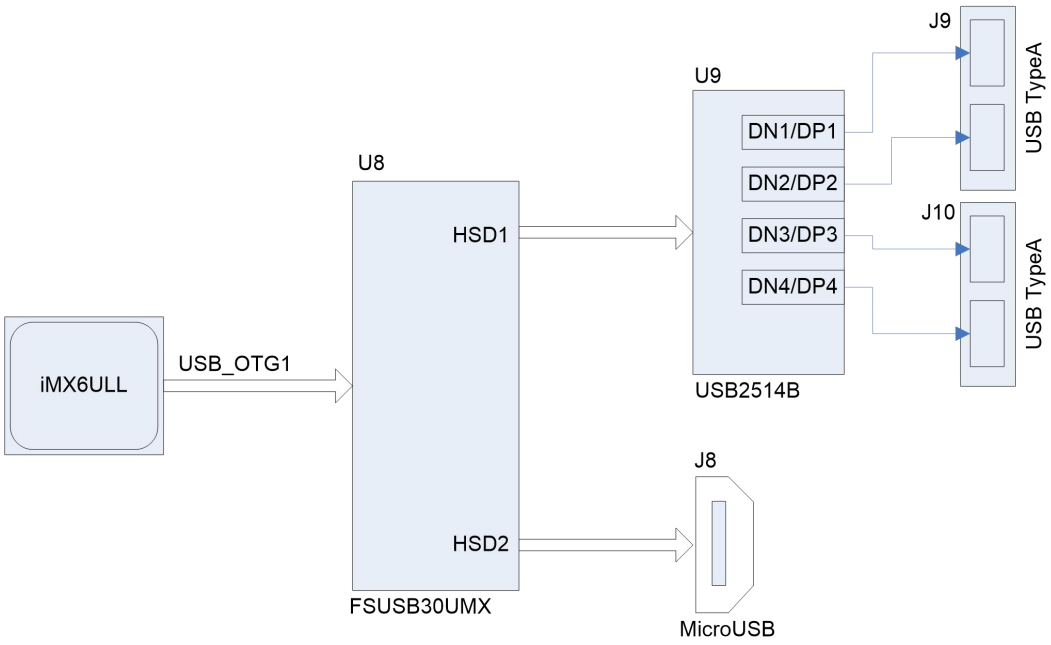
Note:

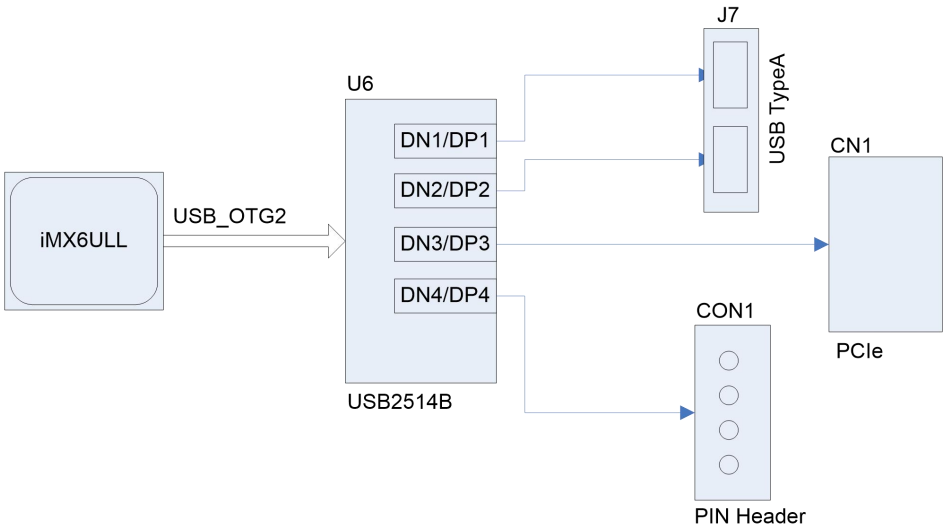
🕮 If NXP Yocto image doesn't contain zoneinfo, copy **/usr/share/zoneinfo** under Ubuntu system to the board, and retry the above commands.

2.10.3 USB OTG

(**To be continued**)

2.10.4 USB HOST





There are 6 USB host channels [USB typeA slot] extended on the base board. Install an USB disk on these slots, check message below:

```
[   272.082860] usb-storage 2-1.1:1.0: USB Mass Storage device detected
[   272.098248] scsi host0: usb-storage 2-1.1:1.0
[   273.104255] scsi 0:0:0:0: Direct-Access     SanDisk  Flash Memory     0.1  PQ: 0
ANSI: 2
[   273.130158] sd 0:0:0:0: [sda] 2001888 512-byte logical blocks: (1.02 GB/977 MiB)
[   273.143825] sd 0:0:0:0: [sda] Write Protect is off
[   273.147410] sd 0:0:0:0: [sda] Mode Sense: 03 00 00 00
[   273.148611] sd 0:0:0:0: [sda] No Caching mode page found
[   273.155755] sd 0:0:0:0: [sda] Assuming drive cache: write through
[   273.176207]  sda: sda1
[   273.199625] sd 0:0:0:0: [sda] Attached SCSI removable disk
[    273.783449] FAT-fs (sda1): Volume was not properly unmounted. Some data may be
corrupt. Please run fsck.
```

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

```
......
/dev/sda1 on /run/media/sda1 type vfat (rw,relatime,gid=6,fmask=0007,dmask=0007,all
ow_utime=0020,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro)
```

The USB disk is automatically mounted under **/run/media/sda1** by udev.

Reset USB Device

USB_OTG1	/sys/devices/platform/soc/2100000.aips-bus/2184000.usb
USB_OTG2	/sys/devices/platform/soc/2100000.aips-bus/2184200.usb

If you install the USB disk in slot **J7** which is extended through USB_OTG2, search

the device name **'sda'** under the system device path of USB_OTG2:

- root@arm:~# **egrep -nir sda /sys/devices/platform/soc/2100000.aips-bus/2184200.usb 2>/dev/null**

```
/sys/devices/platform/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/usb2/2-1/2-1.1/2-1.1:1.0/host3/target3:0:0/3:0:0:0/block/sda/sda1/uevent:3:DEVNAME=sda1
/sys/devices/platform/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/usb2/2-1/2-1.1/2-1.1:1.0/host3/target3:0:0/3:0:0:0/block/sda/uevent:3:DEVNAME=sda
```

- root@arm:~# **node=/sys/devices/platform/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/usb2/2-1/authorized; echo 0 > $node;sleep 1;echo 1 > $node**

```
usb 2-1.1: USB disconnect, device number 7
FAT-fs (sda1): FAT read failed (blocknr 32)
hub 2-1:1.0: USB hub found
hub 2-1:1.0: 4 ports detected
usb 2-1: authorized to connect
usb 2-1.1: new high-speed USB device number 8 using ci_hdrc
usb-storage 2-1.1:1.0: USB Mass Storage device detected
scsi host5: usb-storage 2-1.1:1.0
scsi 5:0:0:0: Direct-Access     SanDisk  Flash Memory     0.1  PQ: 0 ANSI: 2
sd 5:0:0:0: [sda] 2001888 512-byte logical blocks: (1.02 GB/977 MiB)
sd 5:0:0:0: [sda] Write Protect is off
sd 5:0:0:0: [sda] No Caching mode page found
sd 5:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1
sd 5:0:0:0: [sda] Attached SCSI removable disk
FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
```

It resets successfully.

Of course, you can reset the USB bus, then all devices connected via the bus will reset at the same time.

- root@arm:~# **node=/sys/devices/platform/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/usb2/authorized; echo 0 > $node;sleep 1;echo 1 > $node**

```
usb 2-1: USB disconnect, device number 2
usb 2-1.1: USB disconnect, device number 10
usb 2-1.2: USB disconnect, device number 9
FAT-fs (sda1): FAT read failed (blocknr 32)
```

```
hub 2-0:1.0: USB hub found
hub 2-0:1.0: 1 port detected
usb usb2: authorized to connect
usb 2-1: new high-speed USB device number 11 using ci_hdrc
hub 2-1:1.0: USB hub found
hub 2-1:1.0: 4 ports detected
usb 2-1.1: new high-speed USB device number 12 using ci_hdrc
usb-storage 2-1.1:1.0: USB Mass Storage device detected
scsi host9: usb-storage 2-1.1:1.0
usb 2-1.2: new high-speed USB device number 13 using ci_hdrc
Bluetooth: hci0: rtl: examining hci_ver=06 hci_rev=000b lmp_ver=06 lmp_subver=8723
Bluetooth: hci0: rtl: loading rtl_bt/rtl8723b_fw.bin
Bluetooth: hci0: rom_version status=0 version=1
scsi 9:0:0:0: Direct-Access     SanDisk  Flash Memory     0.1  PQ: 0 ANSI: 2
sd 9:0:0:0: [sda] 2001888 512-byte logical blocks: (1.02 GB/977 MiB)
sd 9:0:0:0: [sda] Write Protect is off
sd 9:0:0:0: [sda] No Caching mode page found
sd 9:0:0:0: [sda] Assuming drive cache: write through
 sda: sda1
sd 9:0:0:0: [sda] Attached SCSI removable disk
FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. Please run
fsck.
IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
```

You can see that the USB disk and USB WiFi/BT module reset.

2.10.5 NETWORK

There are two 100Mbps network chip KSZ8081 on board.

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

```
eth0      Link encap:Ethernet  HWaddr 3a:f7:82:bc:fa:0a
          inet addr:192.168.1.81  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::38f7:82ff:febc:fa0a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:4 overruns:0 frame:0
          TX packets:42 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1555 (1.5 KiB)  TX bytes:7192 (7.0 KiB)
```

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 192.168.1.81 eth0: 56(84) bytes of data.
64 bytes from www.baidu.com (183.232.231.174): icmp_seq=1 ttl=56 time=12.1 ms
64 bytes from www.baidu.com (183.232.231.174): icmp_seq=2 ttl=56 time=12.2 ms
64 bytes from www.baidu.com (183.232.231.174): icmp_seq=3 ttl=56 time=12.1 ms
64 bytes from www.baidu.com (183.232.231.174): icmp_seq=4 ttl=56 time=12.5 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
```

Do the same operations to eth1.

2.10.6 LCD

Devices already tested:

MODEL	RESOLUTION	TOUCH	LINUX DTB
EMTOP LCD8000-43T	480x272	TSADC	emtop-som-et6ull-480x272.dtb
EMTOP LCD8000-70T	800x480	TSADC	emtop-som-et6ull-emmc.dtb
LCD70-CTP	1024x600	FT5X06	emtop-som-et6ull-1024x600.dtb

Modify **uEnv.txt** in SD/eMMC FAT32 partition:

```
panel=480x272
......
```

Change the value of 'panel' to the corresponding resolution of your target device, save and reboot the arm board.

```
U-Boot 2016.03-00286-ga57b13b942-dirty (Oct 26 2022 - 16:26:11 +0800)

CPU:   Freescale i.MX6ULL rev1.1 69 MHz (running at 396 MHz)
CPU:   Industrial temperature grade (-40C to 105C) at 35C
Reset cause: POR
Board: EMTOP SOM-ET6ULL
I2C:   ready
DRAM:  512 MiB
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
*** Warning - bad CRC, using default environment
reading uEnv.txt
263 bytes read in 9 ms (28.3 KiB/s)
```

```
PANEL: 1024x600 in uEnv.txt
Display: 1024x600 (1024x600)
Video: 1024x600x24
reading logo.bmp
In:       serial
Out:      serial
Err:      serial
switch to partitions #0, OK
```

2.10.7 TOUCH PANEL

The X11 system will run input calibrating program during its first booting. If you want to recalibrate the touch panel, please try command below:

- root@arm:~# **rm -f /etc/pointercal.xinput; reboot**

Next boot up, the system will display the calibration dialog.

2.10.8 BACKLIGHT

Get the max brightness level:

- root@arm:~# **cat /sys/class/backlight/backlight/max_brightness**

```
255
```

So the valid brightness level range is 0 ~ 255:

- root@arm:~# **for i in `seq 50 250`; do echo $i > /sys/class/backlight/backlight/brightness; echo $i;sleep 0.1;done**

2.10.9 WM8960 AUDIO

Playback:

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

Record:

- root@arm:~# **amixer sset Headphone 127,127**
- root@arm:~# **amixer cset name='Playback Volume' 255,255**

- root@arm:~# **amixer cset name='Capture Volume' 0,31**
- root@arm:~# **amixer sset 'Left Output Mixer PCM' on**
- root@arm:~# **amixer sset 'Right Output Mixer PCM' on**
- root@arm:~# **amixer cset name='ADC PCM Capture Volume' 220,220**
- root@arm:~# **amixer cset name='Left Boost Mixer LINPUT1 Switch' off**
- root@arm:~# **amixer cset name='Left Boost Mixer LINPUT2 Switch' off**
- root@arm:~# **amixer cset name='Left Input Boost Mixer LINPUT2 Volume' 0**
- root@arm:~# **amixer cset name='Left Input Boost Mixer LINPUT3 Volume' 0**
- root@arm:~# **amixer cset name='Right Input Boost Mixer RINPUT1 Volume' 1**
- root@arm:~# **amixer cset name='ADC Data Output Select' 'Left Data = Right ADC; Right Data = Right ADC'**

- root@arm:~# **arecord -r 44100 -f S16_LE -c 2 -d 10 record.wav**

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

- root@arm:~# **aplay record.wav**

2.10.10 UART

DEVICE NODE	HARDWARE	USAGE
/dev/ttymxc0	UART1	DEBUG PORT
/dev/ttymxc1	UART2	RS485
/dev/ttymxc2	UART3	3-WIRE @ J22
/dev/ttymxc4	UART5	5-WIRE @ J23

Connect TXD and RXD of UART3 to run external loopback test:

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

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

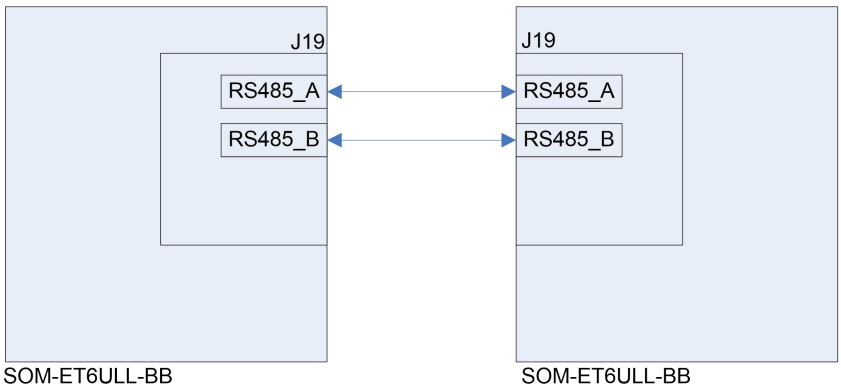
Connect TXD and RXD, RTS and CTS of UART5 to run external loopback test with flow control:

- root@arm:~# **/test/app/com -f -d /dev/ttymxc4**

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

2.10.11 RS485

Connect a RS485 device, or connect 2 boards directly:



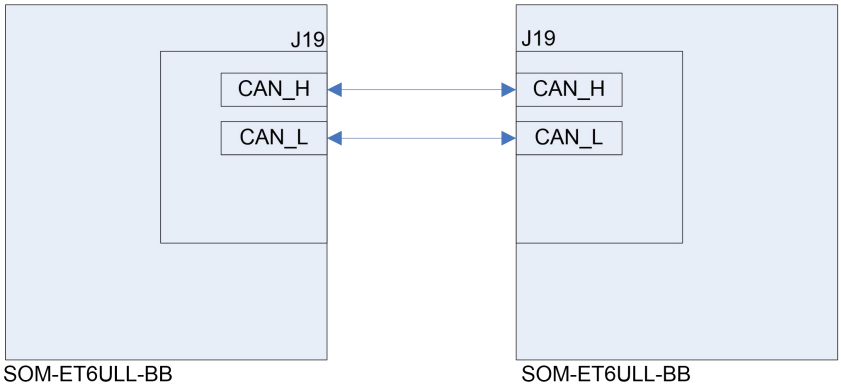
Run below command on both of the boards:

- root@arm:~# **/test/app/com -d /dev/ttymxc1 -m rs485**

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

2.10.12 CAN BUS

Connect 2 boards directly:



- root@arm:~# **ifconfig can0**

```
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          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 [both side]:

- 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 on one board:

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

Send package on the other board:

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

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

2.10.13 POWER BUTTON

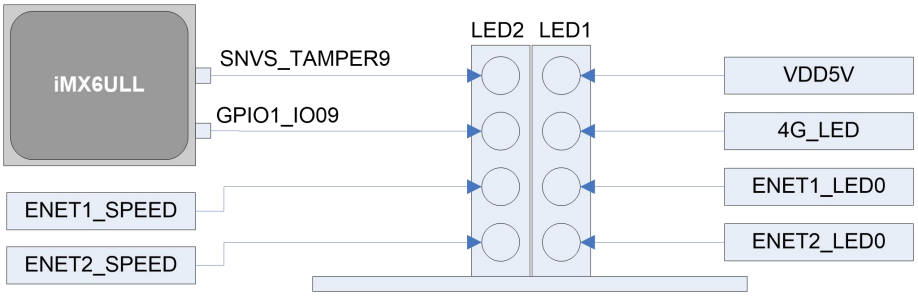
- root@arm:~# **evtest /dev/input/event0**

```
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
```

2.10.14 LED

There are 2 LED array on the base board:



But there are only 2 leds we can control by software:

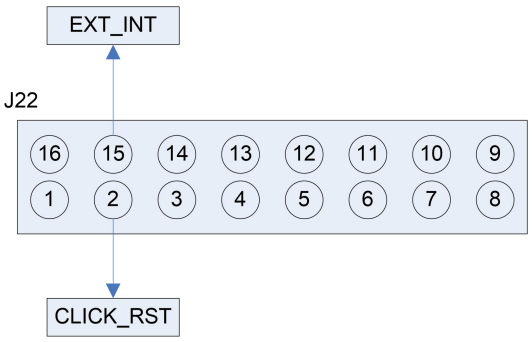
HARDWARE SIGNAL	DEVICE NODE	USAGE
SNVS_TAMPER9	/sys/class/leds/sys	System Status Heartbeat
GPIO1_IO09	/sys/class/leds/led2	User Control

Let's test GPIO1_IO09:

- root@arm:~# **echo none > /sys/class/leds/led2/trigger**
- root@arm:~# **while test 1; do echo 1 > /sys/class/leds/led2/brightness;sleep 1; echo 0 > /sys/class/leds/led2/brightness;sleep 1;done**

Your can see the corresponding LED blinking with 2Hz frequency.

2.10.15 GPIO



SIGNAL	GPIO	GPIO ID [GPIOn_IOm: 32*(n-1)+m]
EXT_INT	GPIO1_IO10	10
CLICK_RST	GPIO5_IO03	131

Initialization:

- root@arm:~# **echo 10 > /sys/class/gpio/export**

Set output:

- root@arm:~# **echo out > /sys/class/gpio/gpio10/direction**

Output high:

- root@arm:~# **echo 1 > /sys/class/gpio/gpio10/value**

Output Low:

- root@arm:~# **echo 0 > /sys/class/gpio/gpio10/value**

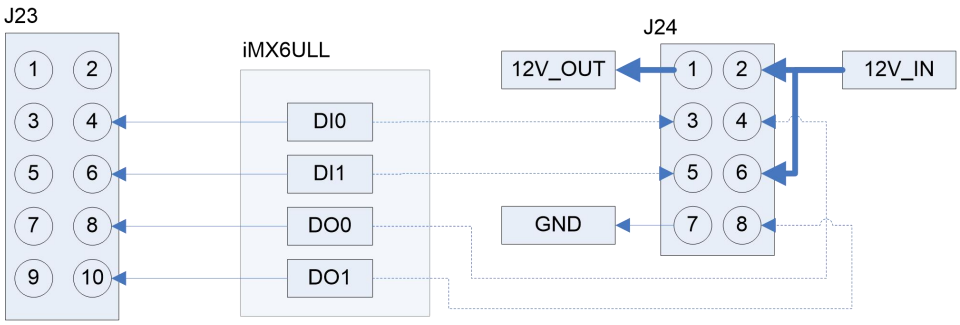
Or set input:

- root@arm:~# **echo in > /sys/class/gpio/gpio10/direction**

Read input:

- root@arm:~# **cat /sys/class/gpio/gpio10/value**

2.10.16 DI/DO



SIGNAL	GPIO	GPIO ID [GPIOn_IOm: 32*(n-1)+m]
DO0	GPIO4_IO17	113
DO1	GPIO4_IO19	115
DI0	GPIO4_IO18	114
DI1	GPIO4_IO20	116

Initialization:

- root@arm:~# **for id in 113 114 115 116; do echo $id > /sys/class/gpio/export 2>/dev/null; done**
- root@arm:~# **for id in 113 115; do echo out > /sys/class/gpio/gpio$id/direction; done**
- root@arm:~# **for id in 114 116; do echo in > /sys/class/gpio/gpio$id/direction; done**

Connect DO0 and DI0 in **J23**

Let DO0 output a signal sequence and read value from DI0:

- root@arm:~# **while test 1; do for i in 0 1; do echo $i > /sys/class/gpio/gpio113/value; sleep 1; cat /sys/class/gpio/gpio114/value; done; done**

```
1
0
1
0
1
0
1
0
1
0
```

2.10.17 PCIe

Devices already tested:

MODEL	DESCRIPTION
QUECTEL EC20	4G module

About QUECTEL EC20 test, please refer to chapter <4G>.

2.10.18 SPI FLASH

The SPIFlash chip is NOT placed on the core board.

2.10.19 TFCard

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

2.10.20 eMMC

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

2.10.21 Unique ID

- root@arm:~# **cat /sys/fsl_otp/HW_OCOTP_CFG[0-1]**

```
0x5d5b3595
0x73a11d7
```

2.10.22 WIFI

Devices already tested:

MODEL	DESCRIPTION

REALTEK RTL8723B	USB WiFi/Bluetooth module

- 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:~# **killall wpa_supplicant udhcpc**
- root@arm:~# **wpa_passphrase WIFI_AP 12345678 >> /etc/wpa_supplicant.conf**

File: **/etc/wpa_supplicant.conf**

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1

network={
        key_mgmt=NONE
}
network={
        ssid="WIFI_AP"
        #psk="12345678"
        psk=6d561555aad99b23640971509aa7594494e3e9cf4b2eece2cb5781d21f351d
b1
}
```

- root@arm:~# **wpa_supplicant -B -iwlan0 -c/etc/wpa_supplicant.conf**

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:192.168.1.101  Bcast:192.168.1.255  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; ifup wlan0; udhcpc -i wlan0**

Wait a while for getting IP:

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

```
wlan0     Link encap:Ethernet  HWaddr d0:c5:d3:d0:9c:33
          inet addr:192.168.1.101  Bcast:192.168.1.255  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)
```

2.10.23 BLUETOOTH

Devices already tested:

MODEL	DESCRIPTION
REALTEK RTL8723B	USB WiFi/Bluetooth module
CSR8510 A10	USB Bluetooth 4.0 module

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

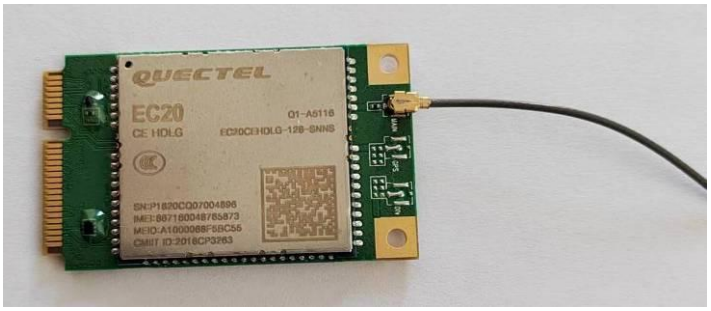
```
hci0:   Type: BR/EDR  Bus: USB
        BD Address: 30:7B:C9:44:2F:F7  ACL MTU: 820:8  SCO MTU: 255:16
        DOWN
        RX bytes:1225 acl:0 sco:0 events:122 errors:0
        TX bytes:23230 acl:0 sco:0 commands:122 errors:0
        Features: 0xff 0xff 0xff 0xfe 0xdb 0xff 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF PARK
        Link mode: SLAVE ACCEPT
```

- root@arm:~# **rfkill unblock all**
- root@arm:~# **/etc/init.d/bluetooth restart**
- 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.

2.10.24 4G



Install QUECTEL EC20, SIM card and antenna.

```
[      1.520949] usbserial: USB Serial support registered for GSM modem (1-port)
[     12.025951] option 2-1.3:1.0: GSM modem (1-port) converter detected
[     12.053488] usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB0
[     12.083780] option 2-1.3:1.1: GSM modem (1-port) converter detected
[     12.109653] usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB1
[     12.124588] option 2-1.3:1.2: GSM modem (1-port) converter detected
[     12.152472] usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB2
[     12.173640] option 2-1.3:1.3: GSM modem (1-port) converter detected
[     12.192775] usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB3
[     12.213280] option 2-1.3:1.4: GSM modem (1-port) converter detected
[     12.241140] usb 2-1.3: GSM modem (1-port) converter now attached to ttyUSB4
```

- root@arm:~# **pppd call quectel-ppp &**

```
... ...
CHAP authentication succeeded
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfReq id=0x0]
sent [IPCP ConfNak id=0x0 <addr 0.0.0.0>]
rcvd [IPCP ConfNak id=0x1 <addr 10.32.232.200> <ms-dns1 202.96.128.86> <ms-dns2
202.96.134.133>]
sent [IPCP ConfReq id=0x2 <addr 10.32.232.200> <ms-dns1 202.96.128.86> <ms-dns2
202.96.134.133>]
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfAck id=0x1]
rcvd [IPCP ConfAck id=0x2 <addr 10.32.232.200> <ms-dns1 202.96.128.86> <ms-dns2
202.96.134.133>]
Could not determine remote IP address: defaulting to 10.64.64.64
local  IP address 10.32.232.200
```

```
remote IP address 10.64.64.64
primary     DNS address 202.96.128.86
secondary DNS address 202.96.134.133
Script /etc/ppp/ip-up started (pid 981)
Script /etc/ppp/ip-up finished (pid 981), status = 0x0
```

Connection test:

- root@arm:~# **ping -I ppp0 www.baidu.com**

```
PING www.a.shifen.com (14.215.177.38) from 10.32.232.200 ppp0: 56(84) bytes of data.
64 bytes from 14.215.177.38: icmp_seq=1 ttl=54 time=37.0 ms
64 bytes from 14.215.177.38: icmp_seq=2 ttl=54 time=43.5 ms
64 bytes from 14.215.177.38: icmp_seq=3 ttl=54 time=51.8 ms
64 bytes from 14.215.177.38: icmp_seq=4 ttl=54 time=41.4 ms
^C64 bytes from 14.215.177.38: icmp_seq=5 ttl=54 time=33.4 ms

--- www.a.shifen.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 20329ms
rtt min/avg/max/mdev = 33.408/41.456/51.856/6.272 ms
```

4G Disable

It's usually called **'airplane mode'**, disable wireless transmission.

- root@arm:~# **echo 1 > /sys/class/leds/4g_disable/brightness**

4G Enable

- root@arm:~# **echo 0 > /sys/class/leds/4g_disable/brightness**

4G Reset:

- root@arm:~# **echo 0 > /sys/class/leds/4g_reset/brightness; sleep 1; echo 1 > /sys/class/leds/4g_reset/brightness**

2.10.25 SUSPEND and RESUME

Suspend to ram:

- root@arm:~# **echo mem > /sys/power/state**

```
PM: Syncing filesystems ... done.
Freezing user space processes ... (elapsed 0.006 seconds) done.
Freezing remaining freezable tasks ... (elapsed 0.001 seconds) done.
Suspending console(s) (use no_console_suspend to debug)
[Click the POWER KEY on the base board...]
PM: suspend of devices complete after 137.489 msecs
PM: suspend devices took 0.140 seconds
PM: late suspend of devices complete after 2.748 msecs
PM: noirq suspend of devices complete after 2.736 msecs
Disabling non-boot CPUs ...
PM: noirq resume of devices complete after 1.455 msecs
PM: early resume of devices complete after 1.690 msecs
mmc1: mmc_select_hs200 failed, error -74
: switch to mmc1 failed
usb 2-1.1: reset high-speed USB device number 5 using ci_hdrc
PM: resume of devices complete after 1690.457 msecs
PM: resume devices took 1.690 seconds
Bluetooth: hci0: rtl: examining hci_ver=06 hci_rev=000b lmp_ver=06 lmp_subver=8723
Bluetooth: hci0: rtl: loading rtl_bt/rtl8723b_fw.bin
Bluetooth: hci0: rom_version status=0 version=1
Restarting tasks ...
usb 2-1.3: USB disconnect, device number 4
done.
```