everyday genius

AIV8183 MIPI DSI LCM bringup SOP

Version: 1.0

Release date: 2019-03-21

Specifications are subject to change without notice.

Document Revision History

Revision	Date	Description
1.0	2019-03-18	Initial Draft

Table of Contents

1. Introduction

This file is for AIV8183 MIPI DSI panel bring up.

1.1 Purpose

You can refer to this file if you want to add a new MIPI panel to AIV8183 project.

1.2 Definitions, Acronyms and Abbreviations

LCM: LCD modules.

Panel ID: Panel identification

1.3 References

NA.

1.4 Overview

Sections overview. Such as:

Section 1 is the introduction and includes a description of the project, applicable and reference documents.

Section 2 provides file path will be used.

Section 3 describes how to add a new panel of lk.

Section 4 describes how to add a new panel of kernel.

Section 5 describes how to add multiple panels.

2. File path

2.1 File path of LK

Modify or add these LCM setting related files if you want to add a new panel under lk.

Files need added: lcm makefile, lcm_driver.c.

Files need modified: projectconfig.mk, mt65xx_lcm_list.c, codegen.dws.

2.1.1 Path of LCM driver

lcm makefile: To configure which lcm file to be build.

lcm_driver.c: This file is about how to power a panel. Such as: lcm power on, lcm initial, lcm suspend, lcm resume.

Path: vendor\mediatek\proprietary\bootable\bootloader\lk\dev\lcm\

2.1.2 Path of mt65xx_lcm_list.c

mt65xx_lcm_list.c: To add LCM main structure.

Path: Vendor\mediatek\proprietary\bootable\bootloader\lk\dev\lcm\

2.1.3 Path of codegen.dws

codegen.dws: To configure LCM GPIO.

Path: vendor\mediatek\propority\bootable\bootloader\lk\target\<project>\dct\dct\

2.1.4 Path of Dct tool

DCT tool is used for GPIO configuration. You can configure LCM GPIO by modify codegen.dws directly, or by DCT tool. It is a convenient way to configure LCM GPIO by DCT tool. You just need select the property by DCT tool and save it, then you can see your selection is in codegen.dws. For more detail, please refer to Chapter 3.

Path: vendor\mediatek\propority\scripts\dct\DrvGen.exe

2.1.5 Path of projectconfig.mk

projectconfig.mk: To configure panel setting, such as: lcm configuration, lcm height, lcm width, logo.

Path: Vendor\mediatek\proprietary\bootable\bootloader\lk\project\

2.2 File path of Kernel

Modify or add these LCM setting related files if you want to add a new panel under kernel.

Files need added: lcm makefile, lcm_driver.c

Files need modified: mt65xx_lcm_list.c, mt65xx_lcm_list.h, dts, defconfig

2.2.1 Path of LCM driver

lcm makefile: To configure which lcm file to be build.

lcm_driver.c: This file is about how to power a panel. Such as: lcm power on, lcm initial, lcm suspend, lcm resume.

Path: Kernel-4.4\drivers\misc\mediatek\lcm\

2.2.2 Path of mt65xx_lcm_list.c and mt65xx_lcm_list.h

mt65xx_lcm_list.c: To add LCM main structure.

mt65xx_lcm_list.h: To add LCM driver define.

2.2.3 Path of dts file

dts: To add panel node

Path: Kernel-4.4\arch\arm64\boot\dts\mediatek\

2.2.4 Path of defconfig file

defconfig: To configure panel setting, such as: lcm configuration, lcm height, lcm width, logo

Path: Kernel-4.4\arch\arm64\configs\

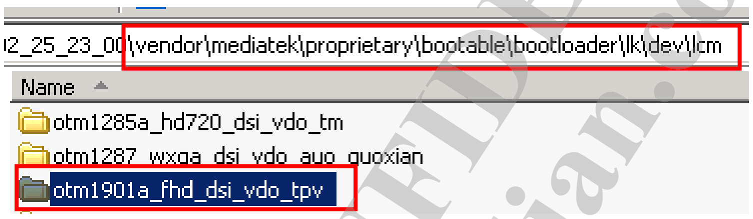
3. LK LCM driver porting

This chapter is about how to add a new panel driver to lk. In lk, LCM have to power on and show logo. In order to facilitate the description of "how to add a new panel driver", we will take lcm "otm1901a_fhd_dsi_vdo_tpv", project "aiv8183m1_64_bsp" for example in this chapter.

3.1 Create <lcm_driver> folder

Path: \vendor\mediatek\proprietary\bootable\bootloader\lk\dev\lcm\

Before implement the new lcm driver code, please create a new <lcm_driver> folder at above path, which named otm1901a_fhd_dsi_vdo_tpv.



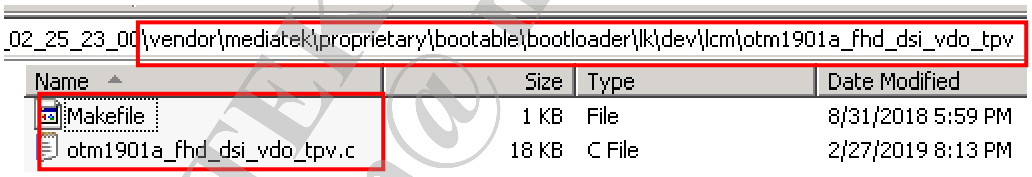
```
2_25_23_00\vendor\mediatek\proprietary\bootable\bootloader\lk\dev\lcm
Name
otm1285a_hd720_dsi_vdo_tm
otm1287_wxga_dsi_vdo_auo_guoxian
otm1901a_fhd_dsi_vdo_tpv
```

Figure 3-1. Create a new folder for panel will be add

3.2 Create <makefile> and <lcm_driver.c> in < lcm_driver > folder

Path: \vendor\mediatek\proprietary\bootable\bootloader\lk\dev\lcm\otm1901a_fhd_dsi_vdo_tpv\

Pleased be noted that, the name of lcm_driver.c is same as lcm_driver folder.

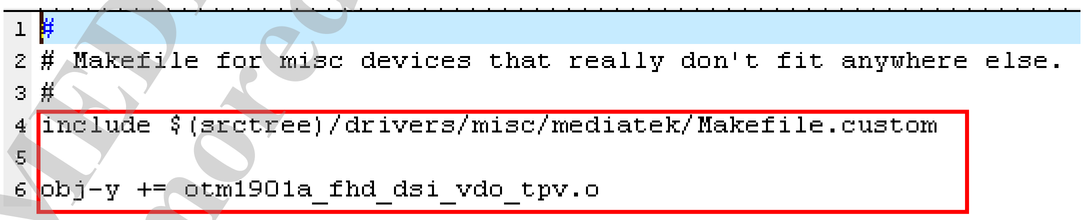


_02_25_23_00\vendor\mediatek\proprietary\bootable\bootloader\lk\dev\lcm\otm1901a_fhd_dsi_vdo_tpv

Name	Size	Type	Date Modified
Makefile	1 KB	File	8/31/2018 5:59 PM
otm1901a_fhd_dsi_vdo_tpv.c	18 KB	C File	2/27/2019 8:13 PM

Figure 3-2. Create makefile and lcm_driver.c

3.3 Implement <makefile> in < lcm_driver > folder



```
#
# Makefile for misc devices that really don't fit anywhere else.
#
include $(srctree)/drivers/misc/mediatek/Makefile.custom

obj-y += otm1901a_fhd_dsi_vdo_tpv.o
```

Figure 3-3. Implement makefile

3.4 Add <lcm main function> of <lcm_driver.c>

Path:\vendor\mediatek\proprietary\bootable\bootloader\lk\dev\lcm\<lcm_driver>\<lcm_driver.c>

lcm_set_util_funcs(): Set util functions.

lcm_get_params(): Setting panel parameters, such as: height, width, video timing, and so on.

Lcm_init(): Initialize panel. The initial code is provided by panel vendor.

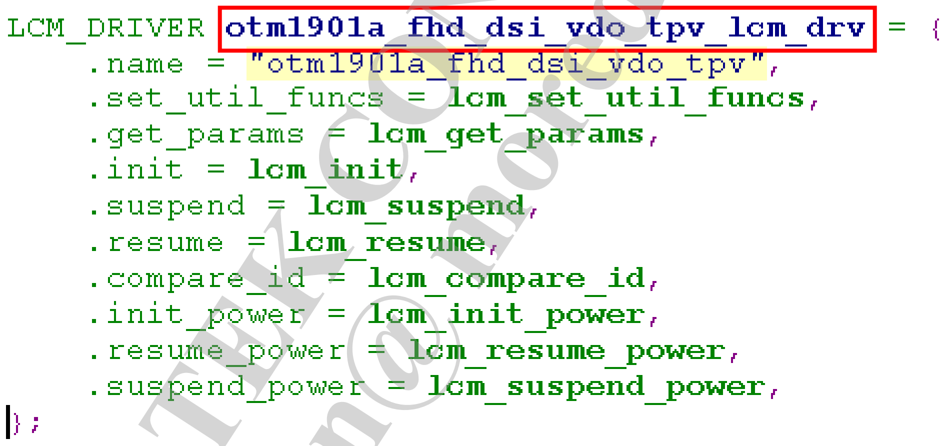
Lcm_init_power(): Initialize lcm power supply. Power sequence must suitable for the panel.

Lcm_compare_id(): Get panel id. If you want to distinguish different panels by panel ID, you need add this function in lk.

Lcm_suspend() & lcm_resume(): Suspend and resume panel. Suspend & resume is used in kernel, Lk no need do this.

Lcm_suspend_power() & Lcm_resume_power(): Suspend and resume power of panel. Suspend & resume is used in kernel, Lk no need do this.

Take “otm1901a_fhd_dsi_vdo_tpv” panel for example, please be notified that, “otm1901a_fhd_dsi_vdo_tpv_lcm_drv” should be same as that in mt65xx_lcm_list.c/lcm_driver_list[].



```
LCM_DRIVER otm1901a_fhd_dsi_vdo_tpv_lcm_drv = {
    .name = "otm1901a_fhd_dsi_vdo_tpv",
    .set_util_funcs = lcm_set_util_funcs,
    .get_params = lcm_get_params,
    .init = lcm_init,
    .suspend = lcm_suspend,
    .resume = lcm_resume,
    .compare_id = lcm_compare_id,
    .init_power = lcm_init_power,
    .resume_power = lcm_resume_power,
    .suspend_power = lcm_suspend_power,
};
```

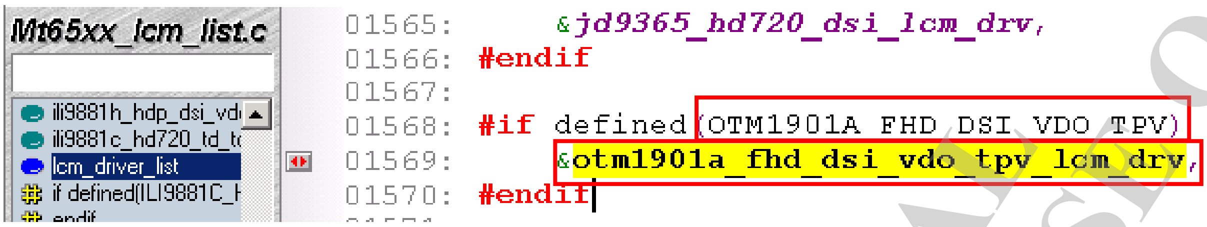
Figure 3-4. Add main function of lcm_driver.c

3.5 Add <lcm main structure> to lcm_driver_list[]

Path: \vendor\mediatek\proprietary\bootable\bootloader\lk\dev\lcm\mt65xx_lcm_list.c

Please be noted that, “OTM1901A_FHD_DSI_VDP_TPV” in lcm_driver_list[] should in capitals.

Please be noted that, “otm1901a_fhd_dsi_vdo_tpv_lcm_drv” should be same as that in otm1901a_fhd_dsi_vdo_tpv.c.



```
      &jd9365_hd720_dsi_lcm_drv,
#endif

#if defined(OTM1901A_FHD_DSI_VDO_TPV)
      &otm1901a_fhd_dsi_vdo_tpv_lcm_drv,
#endif
```

Figure 3-5. Add lcm main structure of lcm_driver.c

```
extern LCM_DRIVER jd9365_hd720_dsi_lcm_drv;
extern LCM_DRIVER otm1901a_fhd_dsi_vdo_tpv_lcm_drv;
extern LCM_DRIVER es6311_anx6585_zigzag_wxga_lcm_drv;
```

Figure 3-6. Add lcm main structure of lcm_driver.c

3.6 Implement <lcm main function> of <lcm_driver.c>

3.6.1 Implement lcm_set_util() function

```
#define SET_RESET_PIN(v)     (lcm_util.set_reset_pin((v)))
#define MDELAY(n)          (lcm_util.mdelay(n))
#define UDELAY(n)          (lcm_util.udelay(n))

#define dsi_set_cmdq_V2(cmd, count, ppara, force_update) \
        lcm_util.dsi_set_cmdq_V2(cmd, count, ppara, force_update)
#define dsi_set_cmdq(pdata, queue_size, force_update) \
        lcm_util.dsi_set_cmdq(pdata, queue_size, force_update)
#define wrtie_cmd(cmd) lcm_util.dsi_write_cmd(cmd)
#define write_regs(addr, pdata, byte_nums) \
        lcm_util.dsi_write_regs(addr, pdata, byte_nums)
#define read_reg(cmd) \
        lcm_util.dsi_dcs_read_lcm_reg(cmd)
#define read_reg_v2(cmd, buffer, buffer_size) \
        lcm_util.dsi_dcs_read_lcm_reg_v2(cmd, buffer, buffer_size)

static void lcm_set_util_funcs(const LCM_UTIL_FUNCS *util)
{
    memcpy(&lcm_util, util, sizeof(LCM_UTIL_FUNCS));
}
```

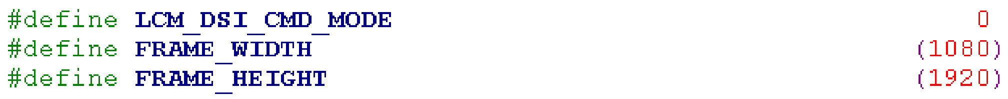
Figure 3-7. Implement lcm_set_util_funcs() function

3.6.2 Implement lcm_get_params() function

This function is used to set LCM parameters, such as: lcm interface mode, panel size, video timing, PLL_CLOCK, and so on.

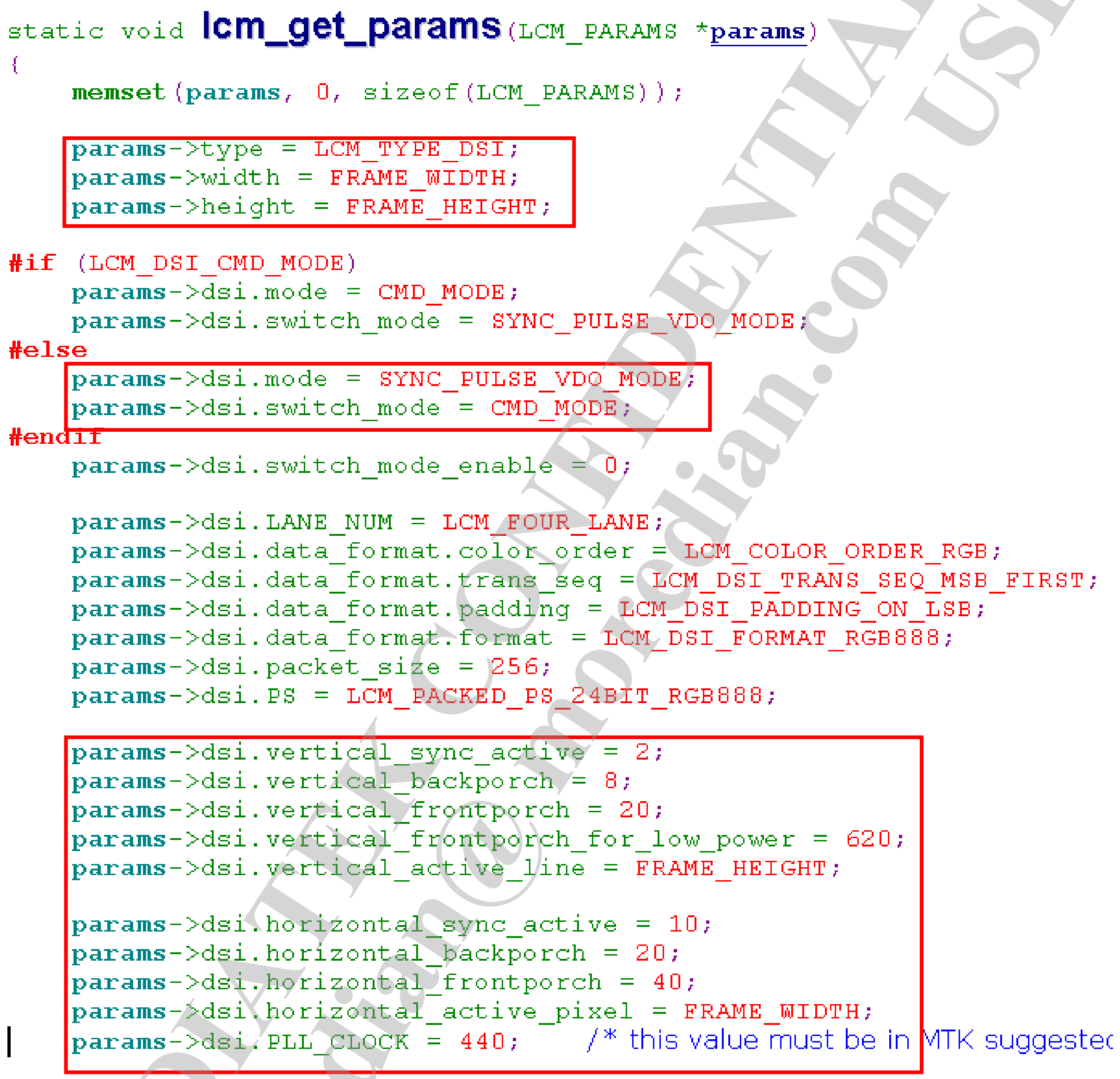
Interface mode: DSI video mode, or DSI command mode; DPI

Panel size: Panel height, panel width



```
#define LCM_DSI_CMD_MODE                                   0
#define FRAME_WIDTH                                   (1080)
#define FRAME_HEIGHT                                  (1920)
```

Figure 3-8. LCM mode and size



```
static void lcm_get_params(LCM_PARAMS *params)
{
    memset(params, 0, sizeof(LCM_PARAMS));

    params->type = LCM_TYPE_DSI;
    params->width = FRAME_WIDTH;
    params->height = FRAME_HEIGHT;

#if (LCM_DSI_CMD_MODE)
    params->dsi.mode = CMD_MODE;
    params->dsi.switch_mode = SYNC_PULSE_VDO_MODE;
#else
    params->dsi.mode = SYNC_PULSE_VDO_MODE;
    params->dsi.switch_mode = CMD_MODE;
#endif
    params->dsi.switch_mode_enable = 0;

    params->dsi.LANE_NUM = LCM_FOUR_LANE;
    params->dsi.data_format.color_order = LCM_COLOR_ORDER_RGB;
    params->dsi.data_format.trans_seq = LCM_DSI_TRANS_SEQ_MSB_FIRST;
    params->dsi.data_format.padding = LCM_DSI_PADDING_ON_LSB;
    params->dsi.data_format.format = LCM_DSI_FORMAT_RGB888;
    params->dsi.packet_size = 256;
    params->dsi.PS = LCM_PACKED_PS_24BIT_RGB888;

    params->dsi.vertical_sync_active = 2;
    params->dsi.vertical_backporch = 8;
    params->dsi.vertical_frontporch = 20;
    params->dsi.vertical_frontporch_for_low_power = 620;
    params->dsi.vertical_active_line = FRAME_HEIGHT;

    params->dsi.horizontal_sync_active = 10;
    params->dsi.horizontal_backporch = 20;
    params->dsi.horizontal_frontporch = 40;
    params->dsi.horizontal_active_pixel = FRAME_WIDTH;
    params->dsi.PLL_CLOCK = 440;     /* this value must be in MTK suggestec
```

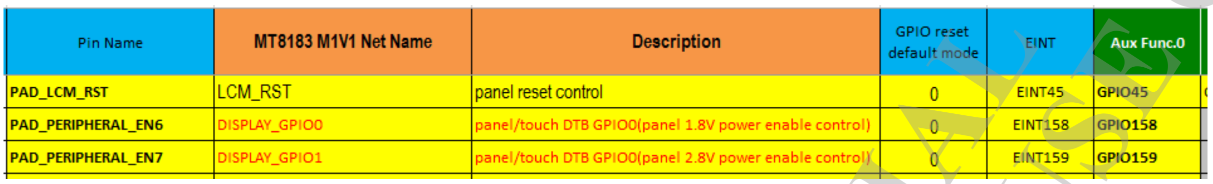
Figure 3-8. Implement lcm_get_params() function

3.6.3 Implement lcm_init_power() function

This function is used to set LCM power on sequence.

First, you have to check GPIO table to figure out how many GPIOs are used for this LCM.
Take "otm1901a_fhd_dsi_vdo_tpv" for example, this panel uses 3 GPIOs to control its power on sequence: 2.8V panel power enable pin, 1.8V panel power enable pin, panel reset pin, see figure 3-9.

We have already configure these 3 GPIOs and named VarName by DCT tool (please refer to Chapter 3.7 for this part) , so we can use VarName to define GPIOs directly.



Pin Name	MT8183 M1V1 Net Name	Description	GPIO reset default mode	EINT	Aux Func.0
PAD_LCM_RST	LCM_RST	panel reset control	0	EINT45	GPIO45
PAD_PERIPHERAL_EN6	DISPLAY_GPIO0	panel/touch DTB GPIO0(panel 1.8V power enable control)	0	EINT158	GPIO158
PAD_PERIPHERAL_EN7	DISPLAY_GPIO1	panel/touch DTB GPIO0(panel 2.8V power enable control)	0	EINT159	GPIO159

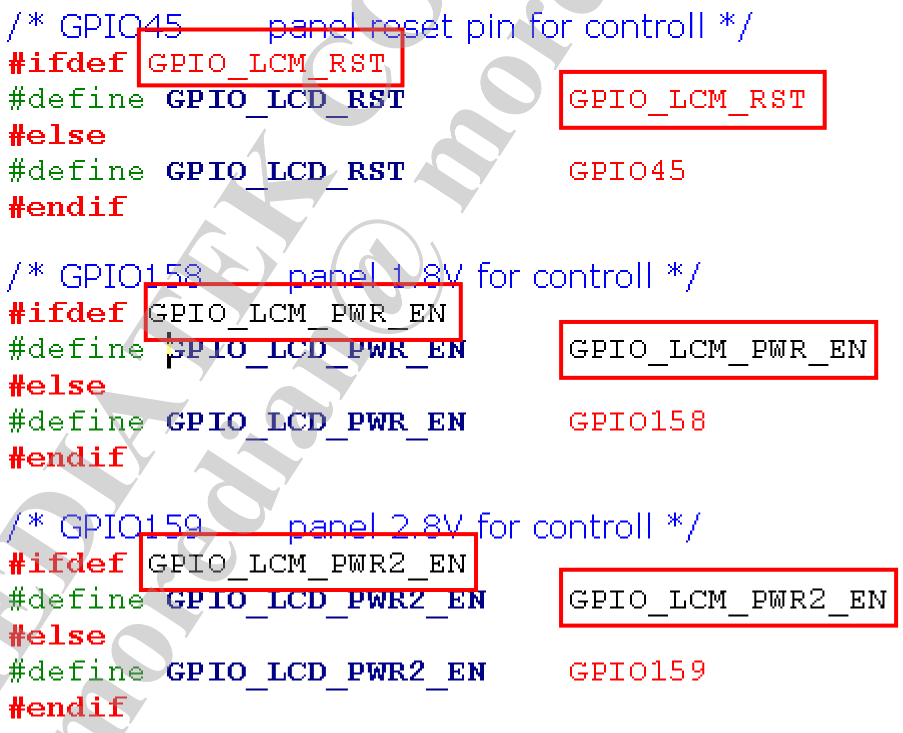
Figure 3-9. GPIO of otm1901a_fhd_dsi_vdo_tpv panel

Second, you have to check LCM spec to figure out the power sequence of this panel.

Third, configure these GPIOs by using DCT tool, please refer to Chapter 3.7 for "how to use DCT tool to configure LCM GPIOs".

Forth, define these GPIOs in lcm_driver.c.
Take "otm1901a_fhd_dsi_vdo_tpv" for example, VarName "GPIO_LCM_RST", "GPIO_LCM_PWR_EN", "GPIO_LCM_PWR2_EN" are already defined in codegen.dws by DCT tool (please refer to Chapter 3.7 for this part). So they can be used to define local definition (for example: GPIO_LCD_RST, GPIO_LCD_PWR_EN, GPIO_LCD_PWR2_EN) directly. After definition, you can use in your driver.

Pleased to be notified that, #ifdef **xxx** should be same as VarName in codegen.dws. And #define ??? **xxx** should be same as VarName in codegen.dws.

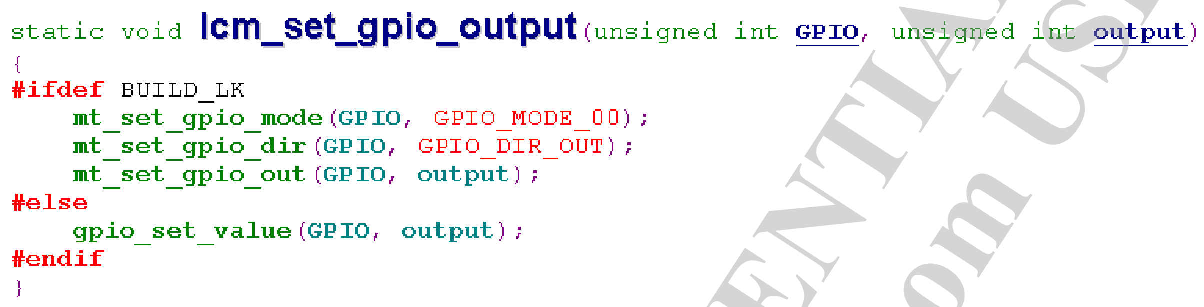


```
/* GPIO45       panel reset pin for controll */
#ifdef GPIO_LCM_RST
#define GPIO_LCD_RST            GPIO_LCM_RST
#else
#define GPIO_LCD_RST            GPIO45
#endif

/* GPIO158       panel 1.8V for controll */
#ifdef GPIO_LCM_PWR_EN
#define GPIO_LCD_PWR_EN         GPIO_LCM_PWR_EN
#else
#define GPIO_LCD_PWR_EN         GPIO158
#endif

/* GPIO159       panel 2.8V for controll */
#ifdef GPIO_LCM_PWR2_EN
#define GPIO_LCD_PWR2_EN        GPIO_LCM_PWR2_EN
#else
#define GPIO_LCD_PWR2_EN        GPIO159
#endif
```

Figure 3-10. Example GPIOs definition of otm1901a_fhd_dsi_vdo_tpv

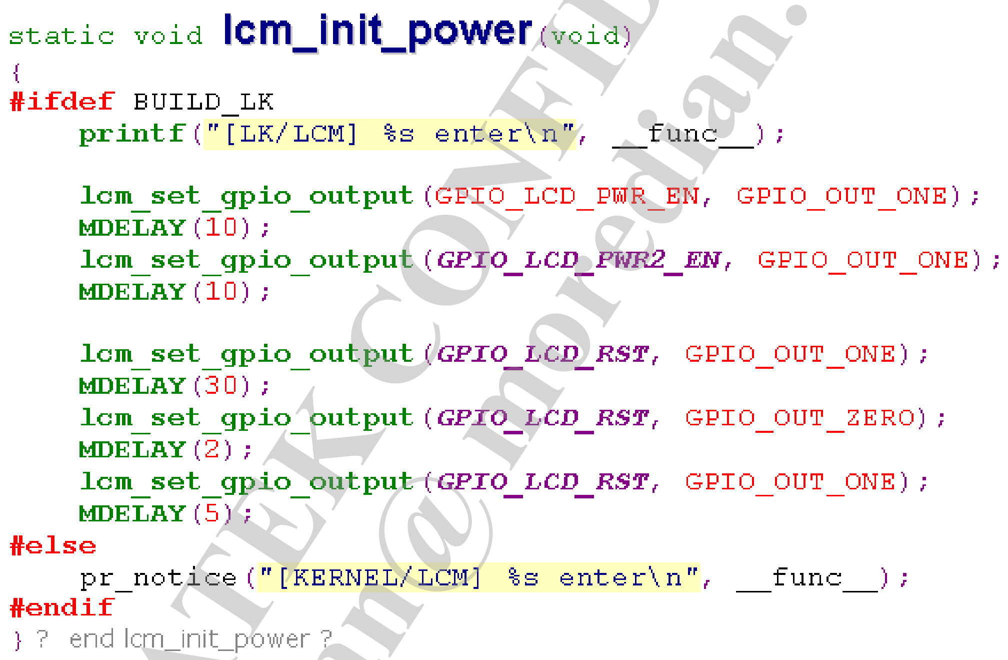
.

Pleased to be notified that, appropriate delay is needed when power on and power off. You can get this information from panel spec too. If you are not sure about power on or power off sequence, you can get help from panel vendor.



```
static void lcm_set_gpio_output(unsigned int GPIO, unsigned int output)
{
#ifdef BUILD_LK
    mt_set_gpio_mode(GPIO, GPIO_MODE_00);
    mt_set_gpio_dir(GPIO, GPIO_DIR_OUT);
    mt_set_gpio_out(GPIO, output);
#else
    gpio_set_value(GPIO, output);
#endif
}
```

Figure 3-11. GPIO setting function



```
static void lcm_init_power(void)
{
#ifdef BUILD_LK
    printf("[LK/LCM] %s enter\n", __func__);

    lcm_set_gpio_output(GPIO_LCD_PWR_EN, GPIO_OUT_ONE);
    MDELAY(10);
    lcm_set_gpio_output(GPIO_LCD_PWR2_EN, GPIO_OUT_ONE);
    MDELAY(10);

    lcm_set_gpio_output(GPIO_LCD_RST, GPIO_OUT_ONE);
    MDELAY(30);
    lcm_set_gpio_output(GPIO_LCD_RST, GPIO_OUT_ZERO);
    MDELAY(2);
    lcm_set_gpio_output(GPIO_LCD_RST, GPIO_OUT_ONE);
    MDELAY(5);
#else
    pr_notice("[KERNEL/LCM] %s enter\n", __func__);
#endif
} ? end lcm_init_power ?
```

Figure 3-12. Example of otm1901a_fhd_dsi_vdo_tpv power on sequence

3.6.4 Implement lcm_init_lcm() function

This function is used to initialize LCM. You can get LCM initial code from vendor.

First, fill initial code in struct LCM_setting_table xxx[] ={}, the format of struct LCM_setting_table is {address, count, {data}}.
Address: The register address of LCM.
Count: Setting how many initial data to this address.
Data: Initial data

```
struct LCM_setting_table {
    unsigned int cmd;
    unsigned char count;
    unsigned char para_list[64];
};
```

Figure 3-13. Format of LCM_setting_table

```
static struct LCM_setting_table init_setting[] = {
    {0x00,  1,  {0x00}},
    {0xFF,  4,  {0x19,0x01,0x01,0x00}},
    {0x00,  1,  {0x80}},
    {0xFF,  2,  {0x19,0x01}},
    {0x00,  1,  {0x00}},
    {0x1C,  1,  {0x33}},
    {0x00,  1,  {0xA0}},
    {0xC1,  1,  {0xE8}},
    {0x00,  1,  {0xA7}},
    {0xC1,  1,  {0x00}},
    {0x00,  1,  {0x90}},
    {0xC0,  6,  {0x00,0x2F,0x00,0x00,0x00,0x01}},
    {0x00,  1,  {0xC0}},
    {0xC0,  6,  {0x00,0x2F,0x00,0x00,0x00,0x01}},
```

Figure 3-14. Example of otm1901a_fhd_dsi_vdo_tpv initial code

Second, implement lcm_init_lcm() function

```
static void lcm_init_lcm(void)
{
    push_table(init_setting,
        sizeof(init_setting) / sizeof(struct LCM_setting_table), 1);
}
```

Figure 3-15. Example of otm1901a_fhd_dsi_vdo_tpv Initial function

3.6.5 Implement lcm_suspend() and lcm_resume() function

Lcm_suspend() and lcm_resume() function will be discussed in chapter 4.

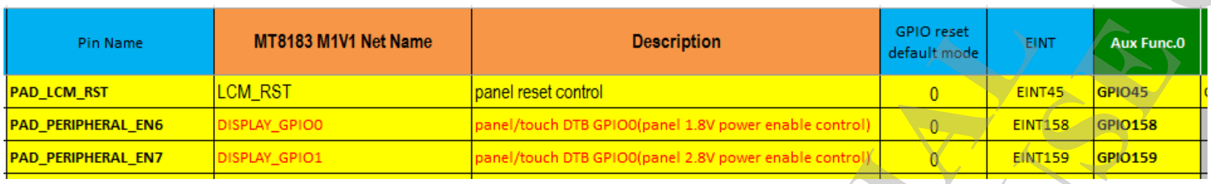
3.7 LCM GPIO configuration

DCT tool Path: vendor\mediatek\proprity\scripts\dct\DrvGen.exe

Codegen.dws Path: vendor\mediatek\proprity\bootable\bootloader\lk\target\<project>\dct\dct\

You can find which GPIO is used for panel power, panel reset, and backlight by looking up GPIO table. Then configure them in codegen.dws by DCT tool.

Generally, the GPIOs have to be controlled of a panel as below. You can check the GPIOs in GPIO table. Now take otm1901a_fhd_dsi_vdo_tpv panel for example.

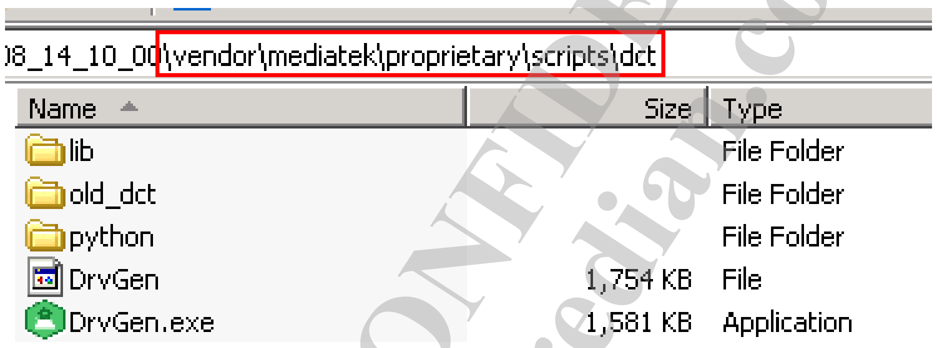


Pin Name	MT8183 M1V1 Net Name	Description	GPIO reset default mode	EINT	Aux Func.0
PAD_LCM_RST	LCM_RST	panel reset control	0	EINT45	GPIO45
PAD_PERIPHERAL_EN6	DISPLAY_GPIO0	panel/touch DTB GPIO0(panel 1.8V power enable control)	0	EINT158	GPIO158
PAD_PERIPHERAL_EN7	DISPLAY_GPIO1	panel/touch DTB GPIO0(panel 2.8V power enable control)	0	EINT159	GPIO159

Figure 3-16. GPIO of otm1901a_fhd_dsi_vdo_tpv panel

3.7.1 Open DCT tool

Find DCT tool (DrvGen.exe) in below path, then double click to open it.



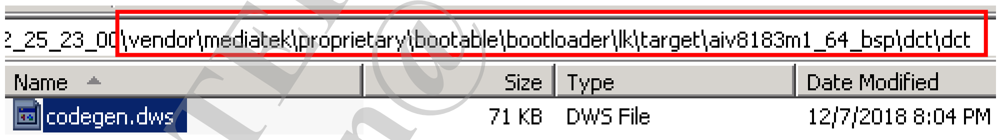
)8_14_10_00\vendor\mediatek\proprietary\scripts\dct

Name	Size	Type
lib		File Folder
old_dct		File Folder
python		File Folder
DrvGen	1,754 KB	File
DrvGen.exe	1,581 KB	Application

Figure 3-17. DCT tool

3.7.2 Open codegen.dws file

Path: Drvgen.exe—>Pro—>Open—>lk\target\<project>\dct\dct\codegen.dws



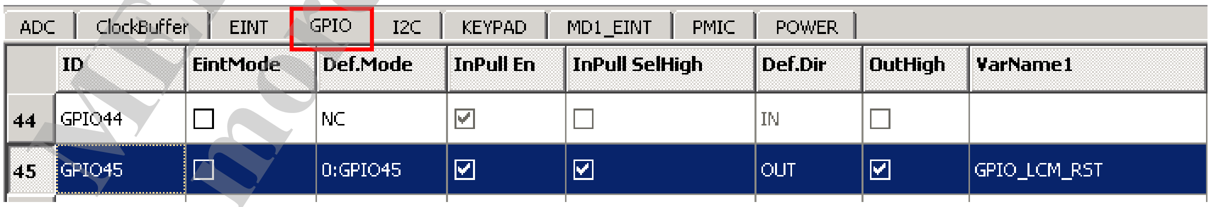
2_25_23_00\vendor\mediatek\proprietary\bootable\bootloader\lk\target\aiv8183m1_64_bsp\dct\dct

Name	Size	Type	Date Modified
codegen.dws	71 KB	DWS File	12/7/2018 8:04 PM

Figure 3-18. Open codegen.dws

3.7.3 Select GPIO

After open codegen.dws, click “GPIO” to select GPIO sheet.



	ID	EintMode	Def.Mode	InPull En	InPull SelHigh	Def.Dir	OutHigh	VarName1
44	GPIO44	☐	NC	✓	☐	IN	☐	
45	GPIO45	☐	0:GPIO45	✓	✓	OUT	✓	GPIO_LCM_RST

Figure 3-19. Select GPIO

3.7.4 Configure GPIO

Configuration LCM related GPIOs.

Def.Mode: Set this to GPIOxx, and select VarName1 for it. Then you can use VarName1 to define GPIO in lcm_driver.c.

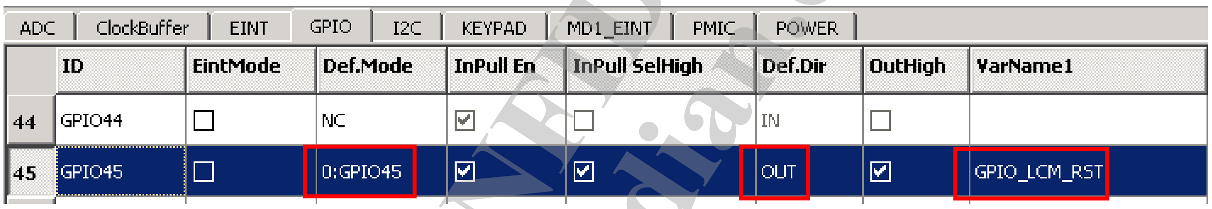
Def.Dir: The direction of pin. LCM power, power enable, panel reset are all output pin.

InPull En: If you select this, means enable pull.

InPull SelHigh: If you select this, means pull up.

OutHigh: If you select this, means output high voltage.

VarName1: Define a name for this GPIO, you can take it as an configure. Then you can use this configure to define GPIO in lcm_driver.c.



	ID	EintMode	Def.Mode	InPull En	InPull SelHigh	Def.Dir	OutHigh	VarName1
44	GPIO44	☐	NC	☑	☐	IN	☐	
45	GPIO45	☐	0:GPIO45	☑	☑	OUT	☑	GPIO_LCM_RST

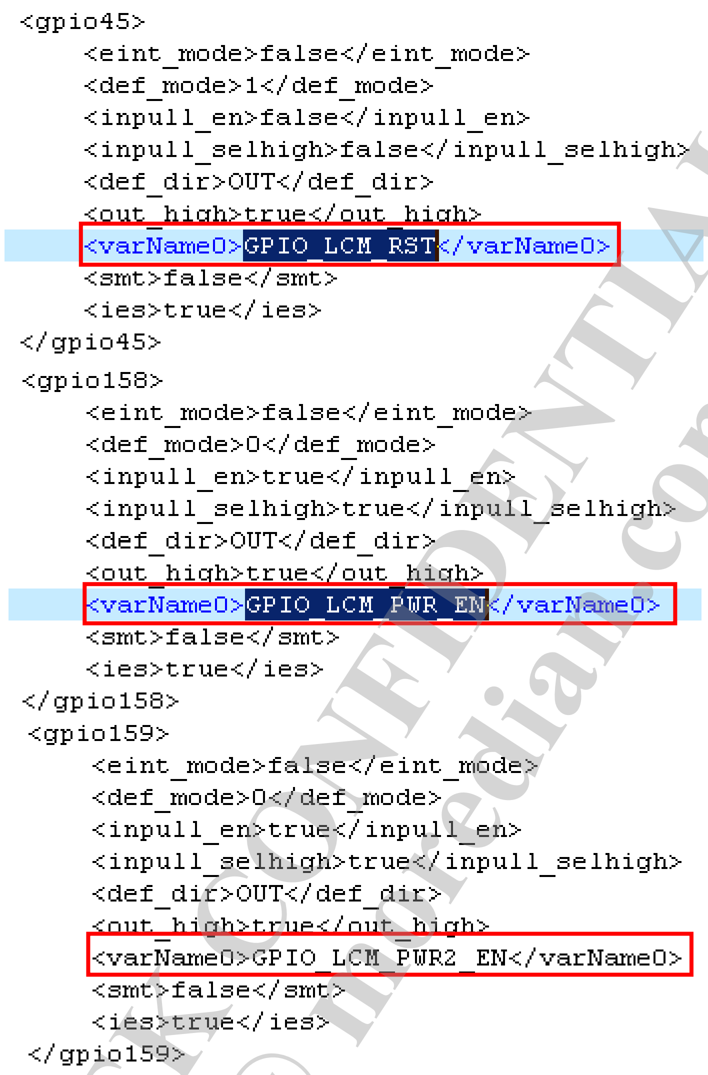
Figure 3-20. Configure GPIO

3.7.5 Save codegen.dws

3.7.6 Check result in codegen.dws

Path: vendor\mediatek\proprity\bootable\bootloader\lk\target\<project>\dct\dct\codegen.dws

Take "otm1901a_fhd_dsi_vdo_tpv" for example, GPIO_LCM_RST, GPIO_LCM_PWR_EN, GPIO_LCM_PWR2_EN is defined as GPIO45, GPIO158, GPIO159. You can use them in lcm_driver.c to control power on sequence.



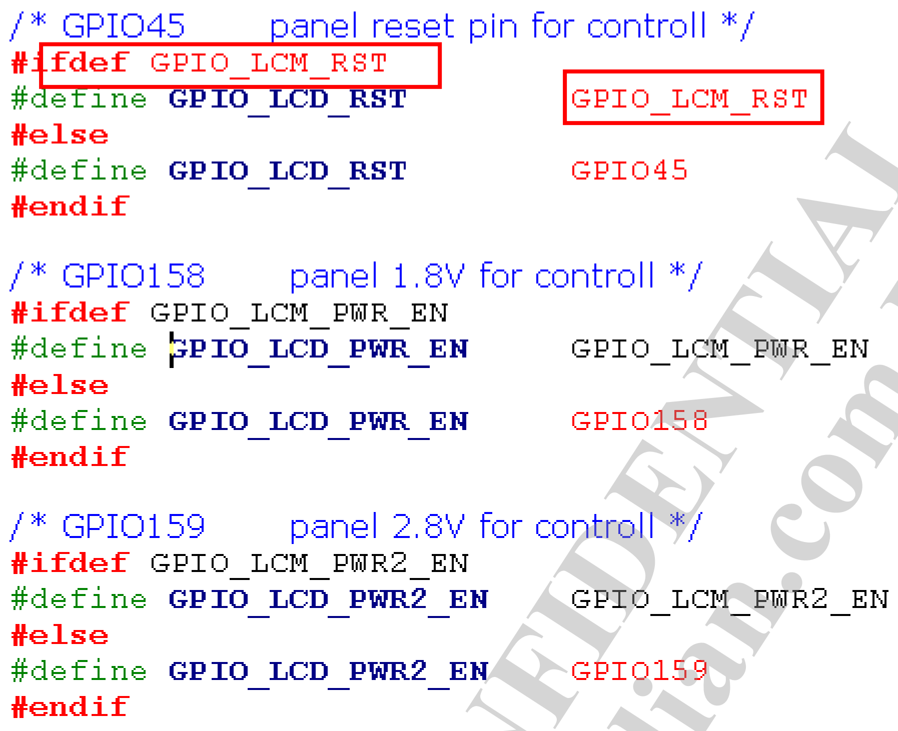
```
<gpio45>
    <eint_mode>false</eint_mode>
    <def_mode>1</def_mode>
    <inpull_en>false</inpull_en>
    <inpull_selhigh>false</inpull_selhigh>
    <def_dir>OUT</def_dir>
    <out_high>true</out_high>
    <varName0>GPIO_LCM_RST</varName0>
    <smt>false</smt>
    <ies>true</ies>
</gpio45>
<gpio158>
    <eint_mode>false</eint_mode>
    <def_mode>0</def_mode>
    <inpull_en>true</inpull_en>
    <inpull_selhigh>true</inpull_selhigh>
    <def_dir>OUT</def_dir>
    <out_high>true</out_high>
    <varName0>GPIO_LCM_PWR_EN</varName0>
    <smt>false</smt>
    <ies>true</ies>
</gpio158>
<gpio159>
    <eint_mode>false</eint_mode>
    <def_mode>0</def_mode>
    <inpull_en>true</inpull_en>
    <inpull_selhigh>true</inpull_selhigh>
    <def_dir>OUT</def_dir>
    <out_high>true</out_high>
    <varName0>GPIO_LCM_PWR2_EN</varName0>
    <smt>false</smt>
    <ies>true</ies>
</gpio159>
```

Figure 3-21. Example of otm1901a_fhd_dsi_vdo_tpv panel GPIO configuration

3.7.7 Configure GPIO in <lcm_driver.c>

Path: vendor\mediatek\proprity\bootable\bootloader\lk\dev\lcm\<lcm_driver>\<lcm_driver.c>

Use VarName1 to define GPIOs which will be used to control panel power on sequence. Take "otm1901a_fhd_dsi_vdo_tpv" for example, GPIO_LCM_RST, GPIO_LCM_PWR_EN, GPIO_LCM_PWR2_EN are defined in codegen.dws. So GPIO_LCD_RST, GPIO_LCD_PWR_EN, GPIO_LCD_PWR2_EN can be used directly.



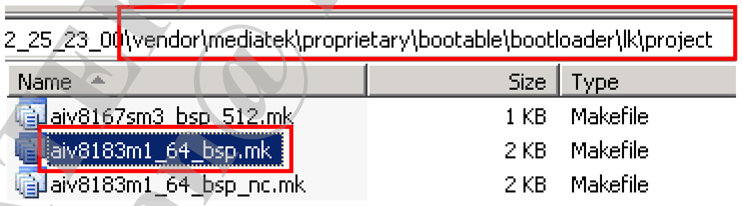
```
/* GPIO45      panel reset pin for controll */
#ifdef GPIO_LCM_RST
#define GPIO_LCD_RST          GPIO_LCM_RST
#else
#define GPIO_LCD_RST          GPIO45
#endif

/* GPIO158      panel 1.8V for controll */
#ifdef GPIO_LCM_PWR_EN
#define GPIO_LCD_PWR_EN       GPIO_LCM_PWR_EN
#else
#define GPIO_LCD_PWR_EN       GPIO158
#endif

/* GPIO159      panel 2.8V for controll */
#ifdef GPIO_LCM_PWR2_EN
#define GPIO_LCD_PWR2_EN      GPIO_LCM_PWR2_EN
#else
#define GPIO_LCD_PWR2_EN      GPIO159
#endif
```

Figure 3-22. Example of otm1901a_fhd_dsi_vdo_tpv panel GPIO definition

3.8 Add <LCM config> to <project.mk>

Path: vendor\mediatek\propreitary\bootable\bootloader\lk\project

Take < project=aiv8183m1_64_bsp> for example, the project.mk file is aiv8183m1_64_bsp.mk



2_25_23_00\vendor\mediatek\proprietary\bootable\bootloader\lk\project

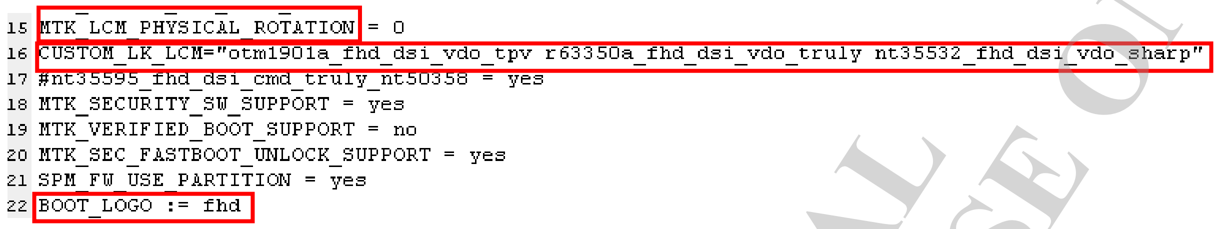
Name	Size	Type
aiv8167sm3_bsp_512.mk	1 KB	Makefile
aiv8183m1_64_bsp.mk	2 KB	Makefile
aiv8183m1_64_bsp_nc.mk	2 KB	Makefile

Figure 3-23. project.mk of otm1901a_fhd_dsi_vdo_tpv panel

MTK_LCM_PHYSICAL_ROTATION: Configuration lcm rotation, it can be: 0/90/180/270

CUSTOM_LK_LCM: Enable lcm configuration. If the case is single LCM, mask previous <lcm configuration> and add yours here. If is multiple LCMs, add other <lcm configuration> after previous one with " " between them. In figure 3-24, there configure 3 LCMs. This means 3 LCMs are build into binary, you can choose one of them by which panel is used.
Pleased be noted that, there is a " " between lcm configuration.

BOOT_LOGO: Configuration BOOT_LOGO of LCM, please refer to chapter 3.9.



```
MTK_LCM_PHYSICAL_ROTATION = 0
CUSTOM_LK_LCM="otm1901a_fhd_dsi_vdo_tpv r63350a_fhd_dsi_vdo_truly nt35532_fhd_dsi_vdo_sharp"
#nt35595_fhd_dsi_cmd_truly_nt50358 = yes
MTK_SECURITY_SW_SUPPORT = yes
MTK_VERIFIED_BOOT_SUPPORT = no
MTK_SEC_FASTBOOT_UNLOCK_SUPPORT = yes
SPM_FW_USE_PARTITION = yes
BOOT_LOGO := fhd
```

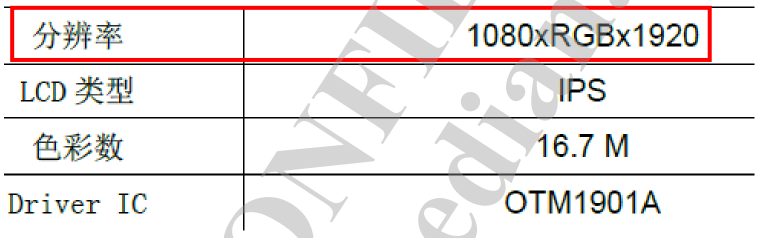
Figure 3-24. Example of LCM configuration

3.9 Configure BOOT_LOGO in <project.mk>

Logo path: vendor\mediatek\propreitary\bootable\bootloader\lk\dev\logo

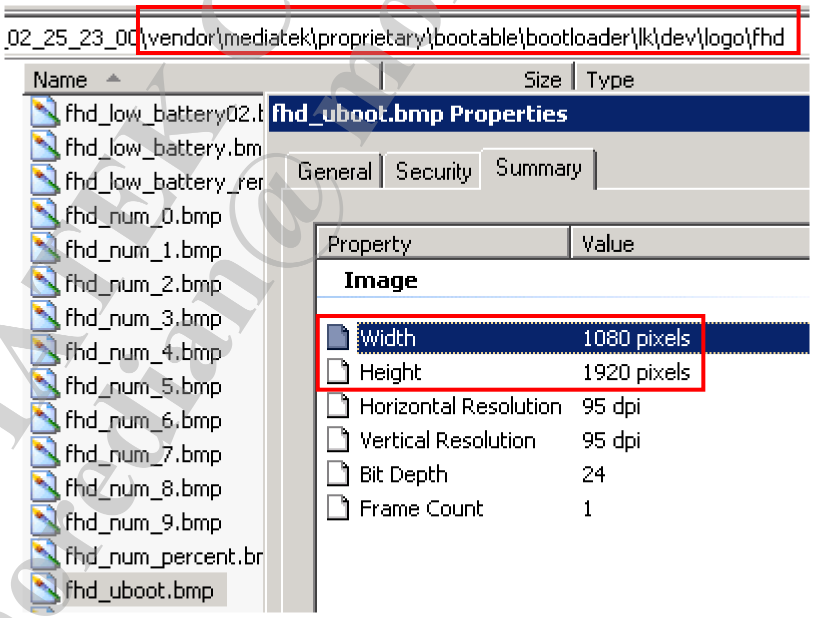
Project.mk Path: vendor\mediatek\propreitary\bootable\bootloader\lk\project

You can get panel size from panel spec, then select suitable LOGO for it. You can check the LOGO size by "right button—>properities—>summary".



分辨率	1080xRGBx1920
LCD 类型	IPS
色彩数	16.7 M
Driver IC	OTM1901A

Figure 3-25. Example of otm1901a_fhd_dsi_vdo_tpv size



02_25_23_00\vendor\mediatek\proprietary\bootable\bootloader\lk\dev\logo\fhd

Name	Size	Type
fhd_low_battery02.b		
fhd_low_battery.bm		
fhd_low_battery_rer		
fhd_num_0.bmp		
fhd_num_1.bmp		
fhd_num_2.bmp		
fhd_num_3.bmp		
fhd_num_4.bmp		
fhd_num_5.bmp		
fhd_num_6.bmp		
fhd_num_7.bmp		
fhd_num_8.bmp		
fhd_num_9.bmp		
fhd_num_percent.br		
fhd_uboot.bmp		

fhd_uboot.bmp Properties

General | Security | Summary

Property	Value
Image	
Width	1080 pixels
Height	1920 pixels
Horizontal Resolution	95 dpi
Vertical Resolution	95 dpi
Bit Depth	24
Frame Count	1

Figure 3-25. How to check LOGO size

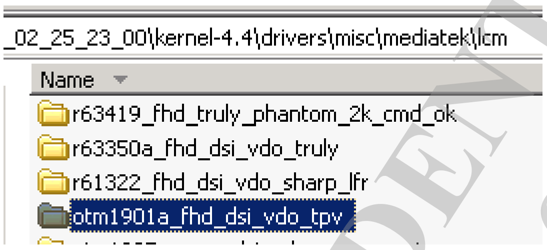
After above 9 steps, lk lcm driver is added ok. Chapter 4 will about "how to add lcm driver to kernel".

4. Kernel LCM driver porting

We will take lcm “otm1901a_fhd_dsi_vdo_tpv”, project “aiv8183m1_64_bsp” for example.

4.1 Create <lcm_driver> folder

Path: kernel-4.4\drivers\misc\mediatek\lcm

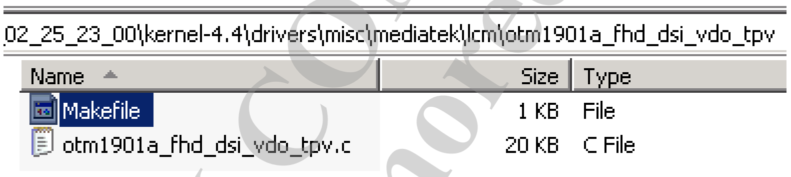


```
_02_25_23_00\kernel-4.4\drivers\misc\mediatek\lcm
Name
r63419_fhd_truly_phantom_2k_cmd_ok
r63350a_fhd_dsi_vdo_truly
r61322_fhd_dsi_vdo_sharp_lfr
otm1901a_fhd_dsi_vdo_tpv
```

Figure 4-1. Create lcm_driver folder

4.2 Create <makefile> and < lcm_driver .c> in < lcm_driver > folder

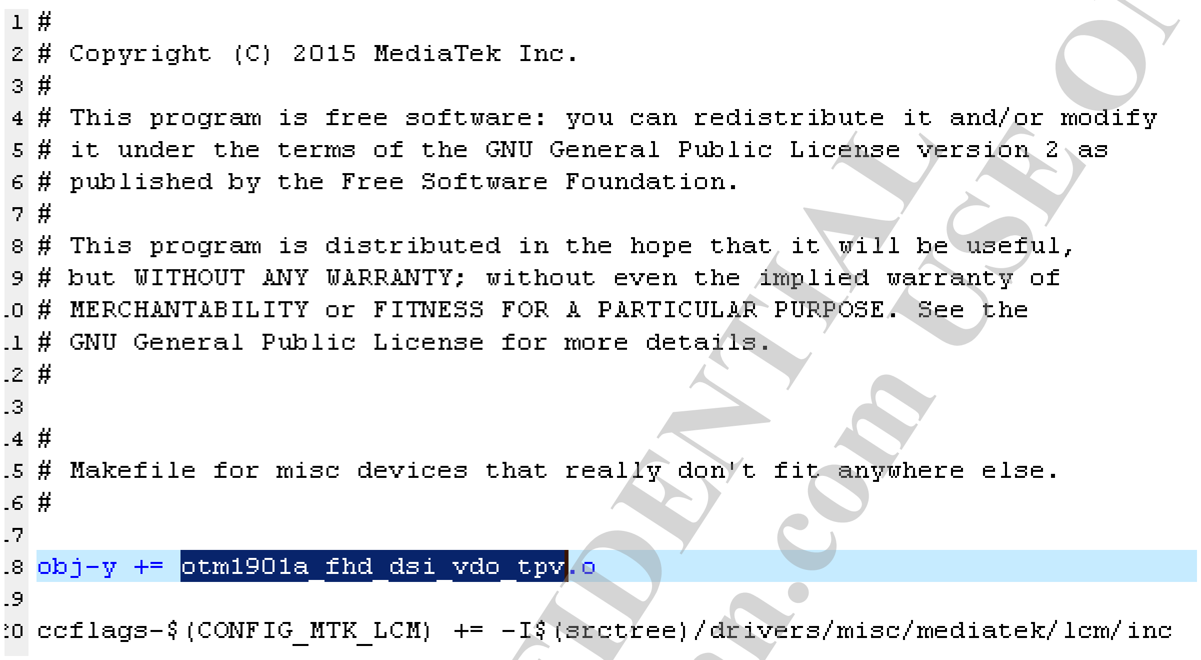
Path: kernel-4.4\drivers\misc\mediatek\lcm\<lcm_driver>



```
02_25_23_00\kernel-4.4\drivers\misc\mediatek\lcm\otm1901a_fhd_dsi_vdo_tpv
Name                         Size   Type
Makefile                     1 KB   File
otm1901a_fhd_dsi_vdo_tpv.c  20 KB   C File
```

Figure 4-2. Create <makefile> and < lcm_driver .c>

4.3 Implement <makefile> in < lcm_driver > folder



```
#
# Copyright (C) 2015 MediaTek Inc.
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License version 2 as
# published by the Free Software Foundation.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#

#
# Makefile for misc devices that really don't fit anywhere else.
#

obj-y += otm1901a_fhd_dsi_vdo_tpv.o

ccflags-$(CONFIG_MTK_LCM) += -I$(srctree)/drivers/misc/mediatek/lcm/inc
```

Figure 4-3. Implement <makefile>

4.4 Add <main function> to <lcm_driver.c>

Path: kernel-4.4\drivers\misc\mediatek\lcm\<lcm_driver>\<lcm_driver.c>

lcm_set_util_funcs(): Set util functions.

lcm_get_params(): Setting panel parameters, such as: height, width, video timing, and so on.

Lcm_init_lcm():Initialize panel. The initial code is provided by panel vendor.

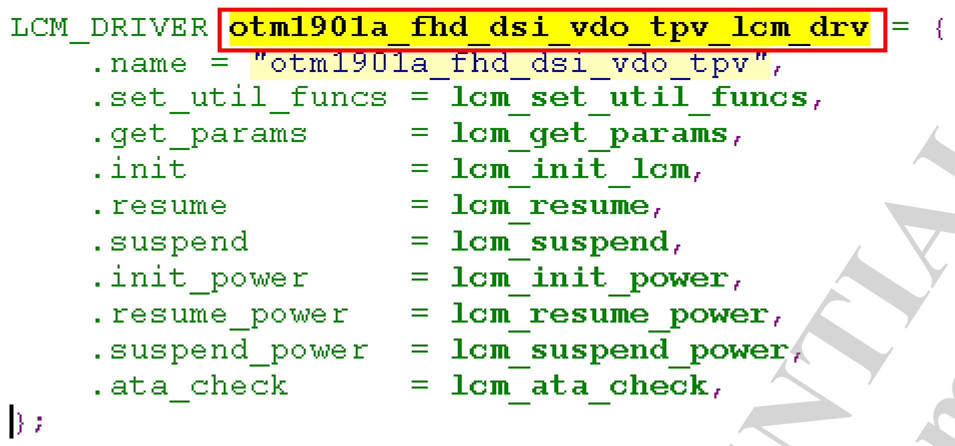
Lcm_init_power():Initialize lcm power supply. Power sequence must suitable for the panel.

Lcm_compare_id(): Get panel id. If you want to distinguish different panels by ID, you need add this function in lk.

Lcm_suspend() & lcm_resume(): Suspend and resume panel. Suspend & resume is done in kernel, Lk no need do this.

Lcm_suspend_power() & Lcm_resume_power(): Suspend and resume power of panel. Suspend & resume is done in kernel, Lk no need do this.

Take “otm1901a_fhd_dsi_vdo_tpv” panel for example, please be notified that, “otm1901a_fhd_dsi_vdo_tpv_lcm_drv” should be same as that in mt65xx_lcm_list.c and lcm_driver_list[].



```
LCM_DRIVER otm1901a_fhd_dsi_vdo_tpv_lcm_drv = {
    .name = "otm1901a_fhd_dsi_vdo_tpv",
    .set_util_funcs = lcm_set_util_funcs,
    .get_params     = lcm_get_params,
    .init           = lcm_init_lcm,
    .resume         = lcm_resume,
    .suspend        = lcm_suspend,
    .init_power     = lcm_init_power,
    .resume_power   = lcm_resume_power,
    .suspend_power  = lcm_suspend_power,
    .ata_check      = lcm_ata_check,
};
```

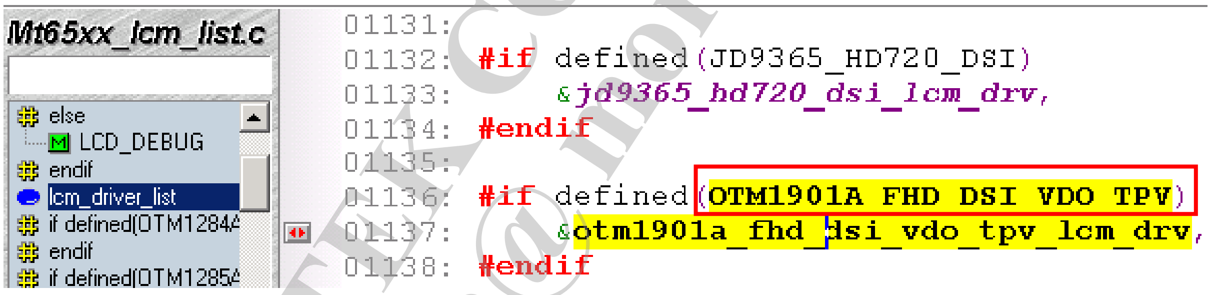
Figure 4-4. Add main function of lcm_driver.c

4.5 Add <LCM main structure> to lcm_driver_list[]

Path: kernel-4.4\drivers\misc\mediatek\lcm\mt65xx_lcm_list.c
kernel-4.4\drivers\misc\mediatek\lcm \mt65xx_lcm_list.h

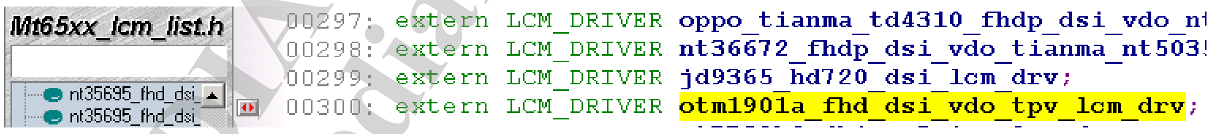
Please be noted that, "OTM1901A_FHD_DSI_VDP_TPV" in lcm_driver_list[] should in capitals.

Please be noted that, "otm1901a_fhd_dsi_vdo_tpv_lcm_drv" should be same as that in otm1901a_fhd_dsi_vdo_tpv.c.



```

#if defined(JD9365_HD720_DSI)
    &jd9365_hd720_dsi_lcm_drv,
#endif

#if defined(OTM1901A_FHD_DSI_VDO_TPV)
    &otm1901a_fhd_dsi_vdo_tpv_lcm_drv,
#endif
```

Figure 4-5. Add lcm main structure of lcm_driver.c



```
extern LCM_DRIVER oppo_tianma_td4310_fhdp_dsi_vdo_n
extern LCM_DRIVER nt36672_fhdp_dsi_vdo_tianma_nt503
extern LCM_DRIVER jd9365_hd720_dsi_lcm_drv;
extern LCM_DRIVER otm1901a_fhd_dsi_vdo_tpv_lcm_drv;
```

Figure 4-6. Add lcm main structure of lcm_driver.c

4.6 Implement <main function> of <lcm_driver.c>

4.6.1 Implement lcm_get_params() function

This function is used to set LCM parameters, such as: lcm interface mode, panel size, video timing, PLL_CLOCK, and so on. Not discuss again.

4.6.2 Implement lcm_init_power() function

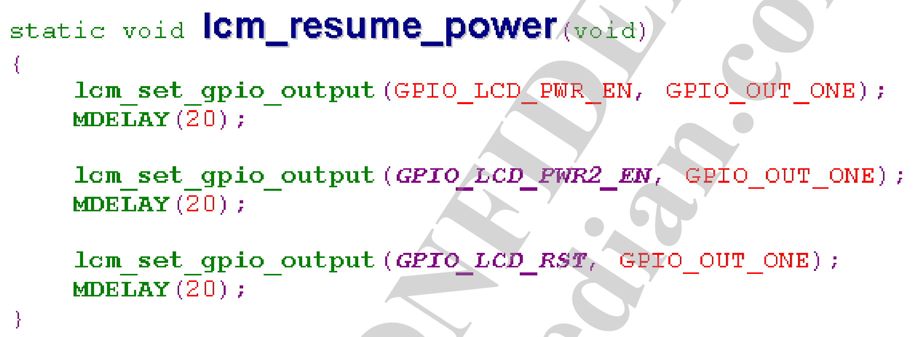
This function is for setting LCM power control. The power supply sequence is same as that in lk. If If lcm already power on in lk, here needn't power on again.

4.6.3 Implement lcm_init_lcm() function

This function if for setting initial code of LCM. The initial code is same as lk. If lcm initialized in lk already, here needn't initialize again.

4.6.4 Implement lcm_resume_power() function

This function is for LCM resume power control. Should set lcm initial power sequence again.

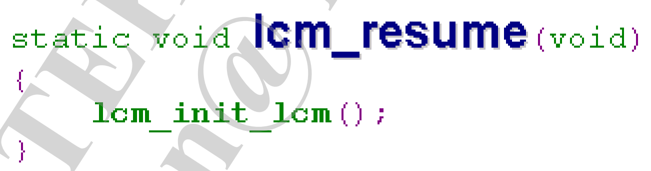


```
static void lcm_resume_power(void)
{
    lcm_set_gpio_output(GPIO_LCD_PWR_EN, GPIO_OUT_ONE);
    MDELAY(20);

    lcm_set_gpio_output(GPIO_LCD_PWR2_EN, GPIO_OUT_ONE);
    MDELAY(20);

    lcm_set_gpio_output(GPIO_LCD_RST, GPIO_OUT_ONE);
    MDELAY(20);
}
```

Figure 4-7. Example of otm1901a_fhd_dsi_vdo_tpv resume power sequence

4.6.5 Implement lcm_resume() function

This function is for LCM resume control. Should set lcm initial code again in lcm resume.

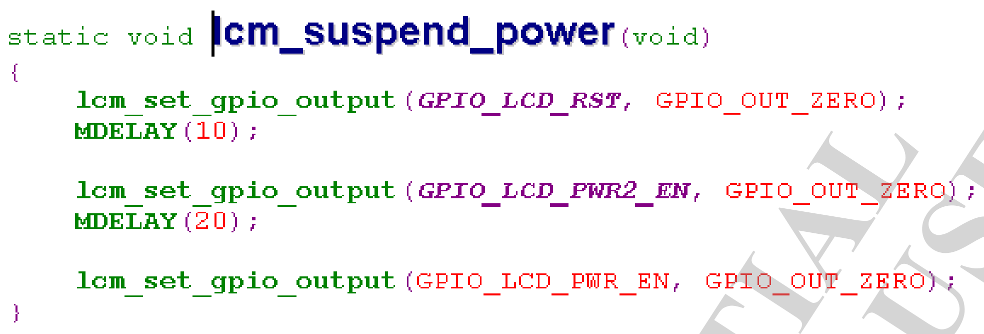


```
static void lcm_resume(void)
{
    lcm_init_lcm();
}
```

Figure 4-8. Example of otm1901a_fhd_dsi_vdo_tpv resume function

4.6.6 Implement lcm_suspend_power() function

This function is for LCM suspend power control. Should set lcm suspend power sequence in lcm_suspend_power().

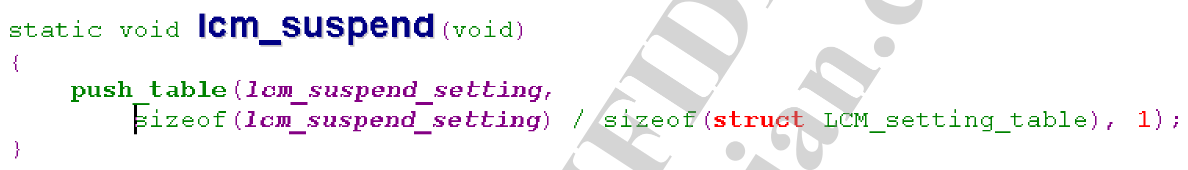


```
static void lcm_suspend_power(void)
{
    lcm_set_gpio_output(GPIO_LCD_RST, GPIO_OUT_ZERO);
    MDELAY(10);

    lcm_set_gpio_output(GPIO_LCD_PWR2_EN, GPIO_OUT_ZERO);
    MDELAY(20);

    lcm_set_gpio_output(GPIO_LCD_PWR_EN, GPIO_OUT_ZERO);
}
```

Figure 4-9. Example of otm1901a_fhd_dsi_vdo_tpv suspend power sequence

4.6.7 Implement lcm_suspend() function

This function is for LCM suspend control. Should set lcm suspend code in lcm_suspend().



```
static void lcm_suspend(void)
{
    push_table(lcm_suspend_setting,
        sizeof(lcm_suspend_setting) / sizeof(struct LCM_setting_table), 1);
}
```

Figure 4-10. Example of otm1901a_fhd_dsi_vdo_tpv suspend function

4.7 Add <panel node> to <project.dts>

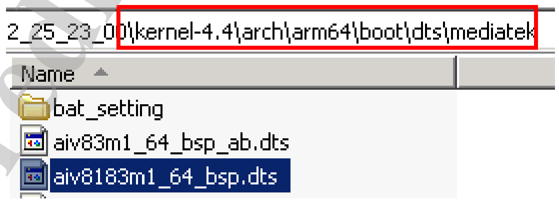
Path: kernel-4.4\arch\arm64\boot\dts\mediatek

Add panel node to project dts file, and configure LCM GPIO in this node. Then use gpio_request() in lcm_driver.c to get GPIO from dts.

Compatible: Should same as that in lcm_driver.c
<&pio 45 0>: &pio: means it is included in gpio-controller. (there is pio).

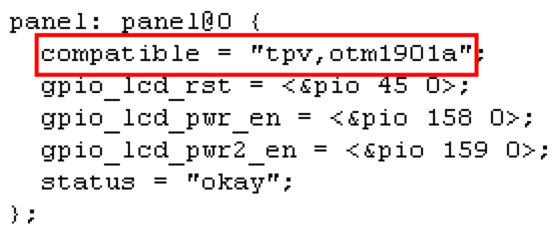
45 & 158 & 159: gpio num that you want to control.
0: flag, do not need care.

Status: okay. Compatible match ok, then run lcm driver probe function. Should set “okay”.



```
2_25_23_00\kernel-4.4\arch\arm64\boot\dts\mediatek
Name
bat_setting
aiv83m1_64_bsp_ab.dts
aiv8183m1_64_bsp.dts
```

Figure 4-11. dts path

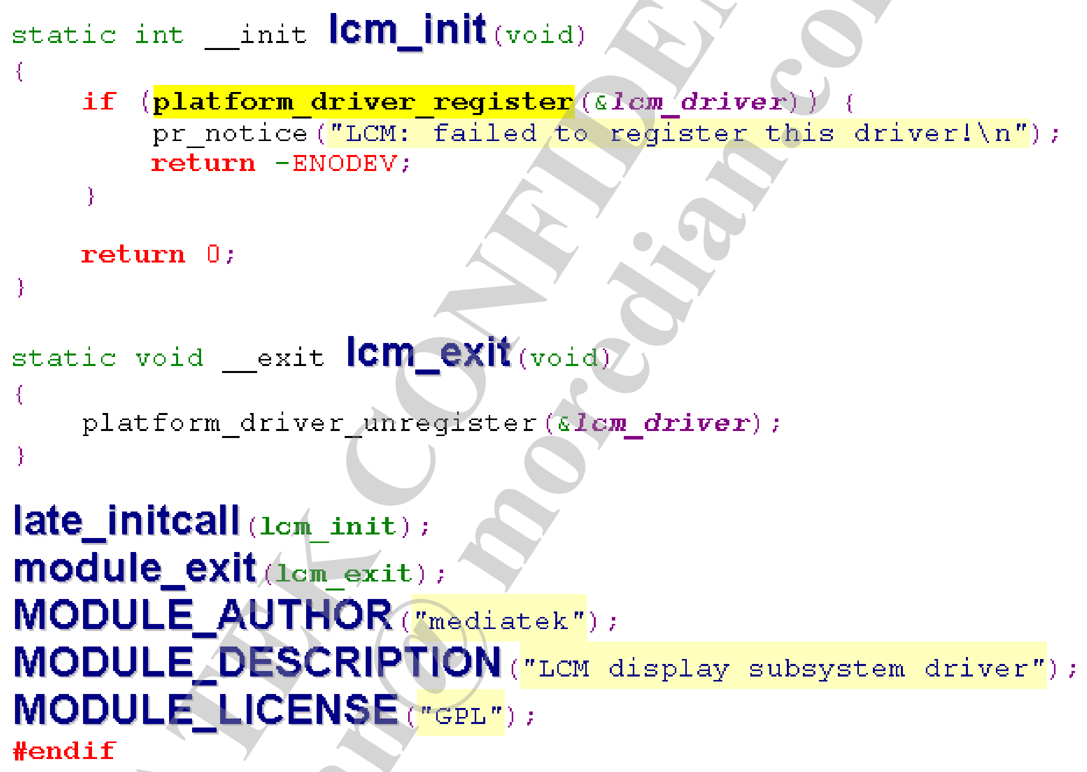


```
panel: panel@0 {
  compatible = "tpv,otm1901a";
  gpio_lcd_rst = <&pio 45 0>;
  gpio_lcd_pwr_en = <&pio 158 0>;
  gpio_lcd_pwr2_en = <&pio 159 0>;
  status = "okay";
};
```

Figure 4-12. How to add panel node

4.8 Register LCM platform in <lcm_driver.c> and request gpio

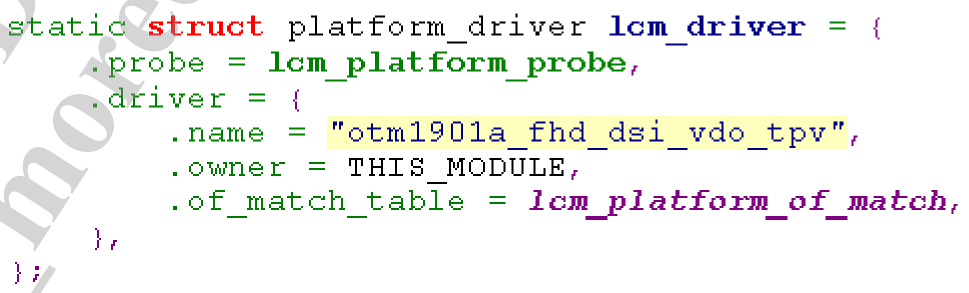
First, register LCM platform driver in lcm_driver.c.



```
static int __init lcm_init(void)
{
    if (platform_driver_register(&lcm_driver)) {
        pr_notice("LCM: failed to register this driver!\n");
        return -ENODEV;
    }

    return 0;
}

static void __exit lcm_exit(void)
{
    platform_driver_unregister(&lcm_driver);
}

late_initcall(lcm_init);
module_exit(lcm_exit);
MODULE_AUTHOR("mediatek");
MODULE_DESCRIPTION("LCM display subsystem driver");
MODULE_LICENSE("GPL");
#endif
```

Figure 4-13. Register lcm platform driver

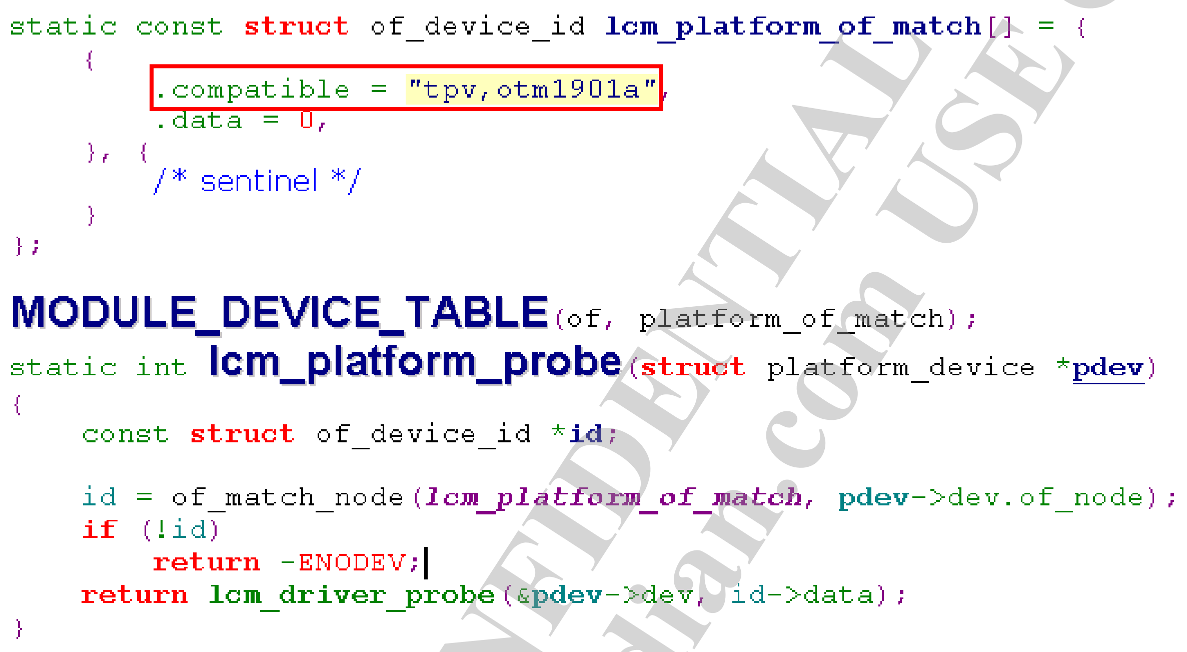
Second, Implement platform_driver lcm_driver = {}



```
static struct platform_driver lcm_driver = {
    .probe = lcm_platform_probe,
    .driver = {
        .name = "otm1901a_fhd_dsi_vdo_tpv",
        .owner = THIS_MODULE,
        .of_match_table = lcm_platform_of_match,
    },
};
```

Figure 4-14. Implement lcm platform_driver

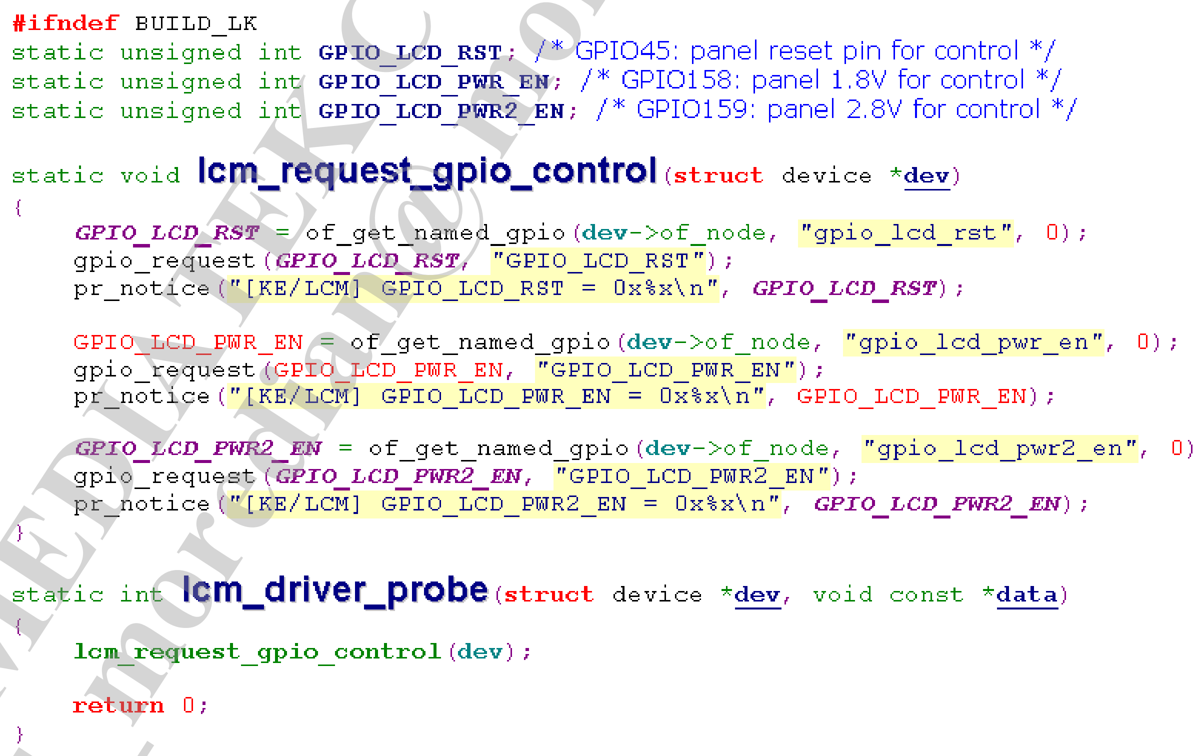
Third, add lcm_platform_of_match[] and lcm_platform_probe(). The compatible must same as that in project.dts. After these two compatible match, will excute lcm_platform_probe().



```
static const struct of_device_id lcm_platform_of_match[] = {
	{
		.compatible = "tpv,otm1901a",
		.data = 0,
	}, {
		/* sentinel */
	}
};

MODULE_DEVICE_TABLE(of, platform_of_match);
static int lcm_platform_probe(struct platform_device *pdev)
{
	const struct of_device_id *id;

	id = of_match_node(lcm_platform_of_match, pdev->dev.of_node);
	if (!id)
		return -ENODEV;
	return lcm_driver_probe(&pdev->dev, id->data);
}
```

Figure 4-15. Implement compatible

Forth, request GPIO. Then you can use them directly.



```
#ifndef BUILD_LK
static unsigned int GPIO_LCD_RST; /* GPIO45: panel reset pin for control */
static unsigned int GPIO_LCD_PWR_EN; /* GPIO158: panel 1.8V for control */
static unsigned int GPIO_LCD_PWR2_EN; /* GPIO159: panel 2.8V for control */

static void lcm_request_gpio_control(struct device *dev)
{
	GPIO_LCD_RST = of_get_named_gpio(dev->of_node, "gpio_lcd_rst", 0);
	gpio_request(GPIO_LCD_RST, "GPIO_LCD_RST");
	pr_notice("[KE/LCM] GPIO_LCD_RST = 0x%x\n", GPIO_LCD_RST);

	GPIO_LCD_PWR_EN = of_get_named_gpio(dev->of_node, "gpio_lcd_pwr_en", 0);
	gpio_request(GPIO_LCD_PWR_EN, "GPIO_LCD_PWR_EN");
	pr_notice("[KE/LCM] GPIO_LCD_PWR_EN = 0x%x\n", GPIO_LCD_PWR_EN);

	GPIO_LCD_PWR2_EN = of_get_named_gpio(dev->of_node, "gpio_lcd_pwr2_en", 0)
	gpio_request(GPIO_LCD_PWR2_EN, "GPIO_LCD_PWR2_EN");
	pr_notice("[KE/LCM] GPIO_LCD_PWR2_EN = 0x%x\n", GPIO_LCD_PWR2_EN);
}

static int lcm_driver_probe(struct device *dev, void const *data)
{
	lcm_request_gpio_control(dev);

	return 0;
}
```

Figure 4-16. Request GPIO

4.9 Add <lcm config> to <project_defconfig>

Path: kernel-4.4\arch\arm64\configs

Take < project=aiv8183m1_64_bsp> for example, the project_defconfig and project_debug_defconfig file is aiv8183m1_64_bsp_defconfig and aiv8183m1_64_bsp_debug_defconfig.

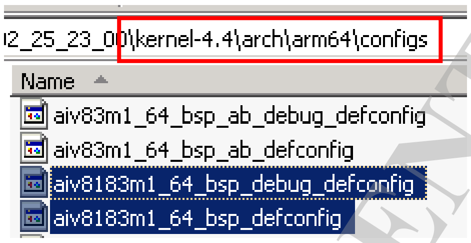

```
i2_25_23_00\kernel-4.4\arch\arm64\configs
Name
aiv83m1_64_bsp_ab_debug_defconfig
aiv83m1_64_bsp_ab_defconfig
aiv8183m1_64_bsp_debug_defconfig
aiv8183m1_64_bsp_defconfig
```

Figure 4-17. Path of defconfig file

MTK_LCM_PHYSICAL_ROTATION: Configuration lcm rotation, it can be: 0/90/180/270

CONFIG_CUSTOM_KERNEL_LCM: Enable lcm configuration. If the case is single LCM, mask previous <lcm configuration> and add yours here. If is multiple LCMs, add other <lcm configuration> after previous one with " " between them. In figure 3-24, there configure 3 LCMs. This means 3 LCMs are build into binary, you can choose one of them by which panel is used.
Pleased be noted that, there is a " " between lcm configuration.

CONFIG_MTK_LCM=y: For module build

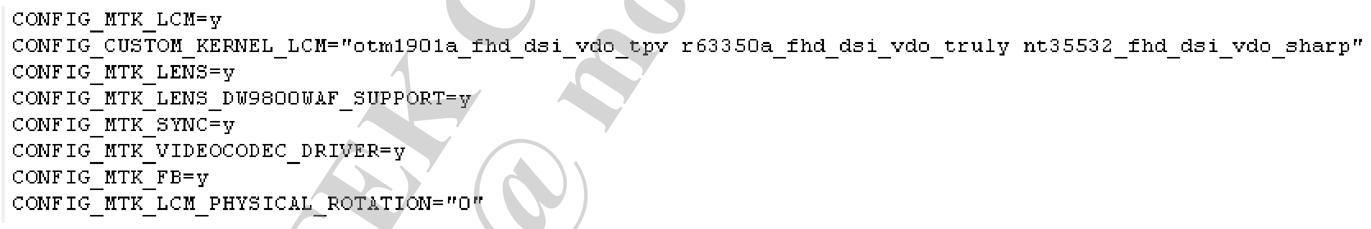

```
CONFIG_MTK_LCM=y
CONFIG_CUSTOM_KERNEL_LCM="otm1901a_fhd_dsi_vdo_tpv r63350a_fhd_dsi_vdo_truly nt35532_fhd_dsi_vdo_sharp"
CONFIG_MTK_LENS=y
CONFIG_MTK_LENS_DW9800WAF_SUPPORT=y
CONFIG_MTK_SYNC=y
CONFIG_MTK_VIDEOCODEC_DRIVER=y
CONFIG_MTK_FB=y
CONFIG_MTK_LCM_PHYSICAL_ROTATION="0"
```

Figure 4-18. Example of LCM config

5. Add multiple panels

5.1 Add multiple <LCM_driver> in lk

For aiv8183_64_bsp project, there configure 3 LCMs: otm1901a_fhd_dsi_vdo_tpv, r63350a_fhd_dsi_vdo_truly, nt35532_fhd_dsi_vdo_sharp. Please refer to Chapter 3 to add these 3 LCM drivers of lk one by one.

5.2 Add multiple <LCM_driver> in kernel

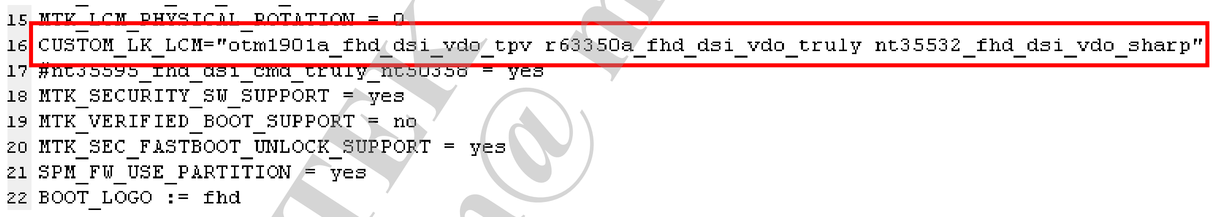
For aiv8183_64_bsp project, there configure 3 LCMs: otm1901a_fhd_dsi_vdo_tpv, r63350a_fhd_dsi_vdo_truly, nt35532_fhd_dsi_vdo_sharp. Please refer to Chapter 4 to add these 3 LCM drivers of kernel one by one.

5.3 Add multiple LCM config to <project.mk>

Path: vendor\mediatek\propreitary\bootable\bootloader\lk\project

CUSTOM_LK_LCM: Enable lcm configuration. If the case is single LCM, mask previous <lcm configuration> and add yours here. If is multiple LCMs, add other <lcm configuration> after previous one with " " between them. In figure 3-24, there configure 3 LCMs. This means 3 LCMs are build into binary, you can choose one of them by which panel is used.
Pleased be noted that, there is a " " between lcm configuration.

Take < project=aiv8183m1_64_bsp> for example, the project.mk file is aiv8183m1_64_bsp.mk



```
MTK_LCM_PHYSICAL_ROTATION = 0
CUSTOM_LK_LCM="otm1901a_fhd_dsi_vdo_tpv r63350a_fhd_dsi_vdo_truly nt35532_fhd_dsi_vdo_sharp"
#nt35595_fhd_dsi_cmd_truly_nt50358 = yes
MTK_SECURITY_SW_SUPPORT = yes
MTK_VERIFIED_BOOT_SUPPORT = no
MTK_SEC_FASTBOOT_UNLOCK_SUPPORT = yes
SPM_FW_USE_PARTITION = yes
BOOT_LOGO := fhd
```

Figure 5-1. Add multiple LCM configuration

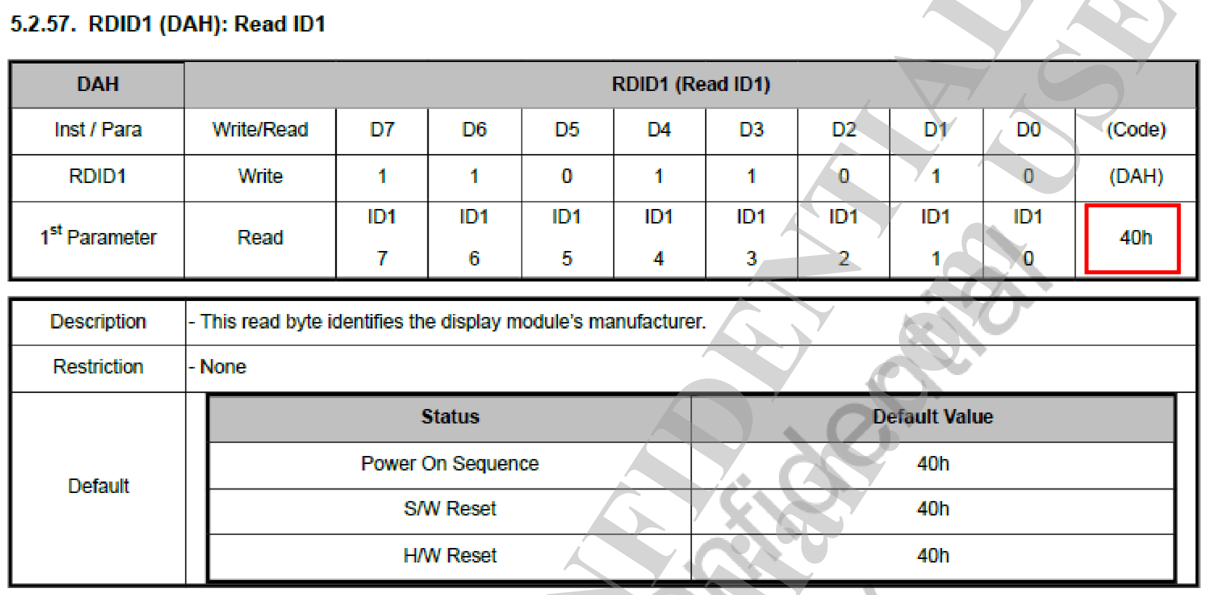
5.4 Add and implement compare_id() function to <LCM_driver.c> of lk

We only connect one panel on aiv8183 project at a time, so we have to distinguish which panel is connected. We can use panel ID to distinguish different panels.

Add lcm_compare_id() function for each panel driver, and return read ID result. We can use this result to distinguish panel.

This compare identification function is doing in lk lcm. After lk get panel info, will transfer panel info to kernel lcm probe function, so needn't do this again in kernel.

Take “otm1901a_fhd_dsi_vdo_tpv panel” for example, its panel ID is in lcm registers of 0xDAh, 0xDBh, 0xDCh from panel spec, you can get ID by reading these 3 registers. Panel ID is： 0xDAh = 0x40, 0xDBh = 0x00, 0xDCh = 0x00.

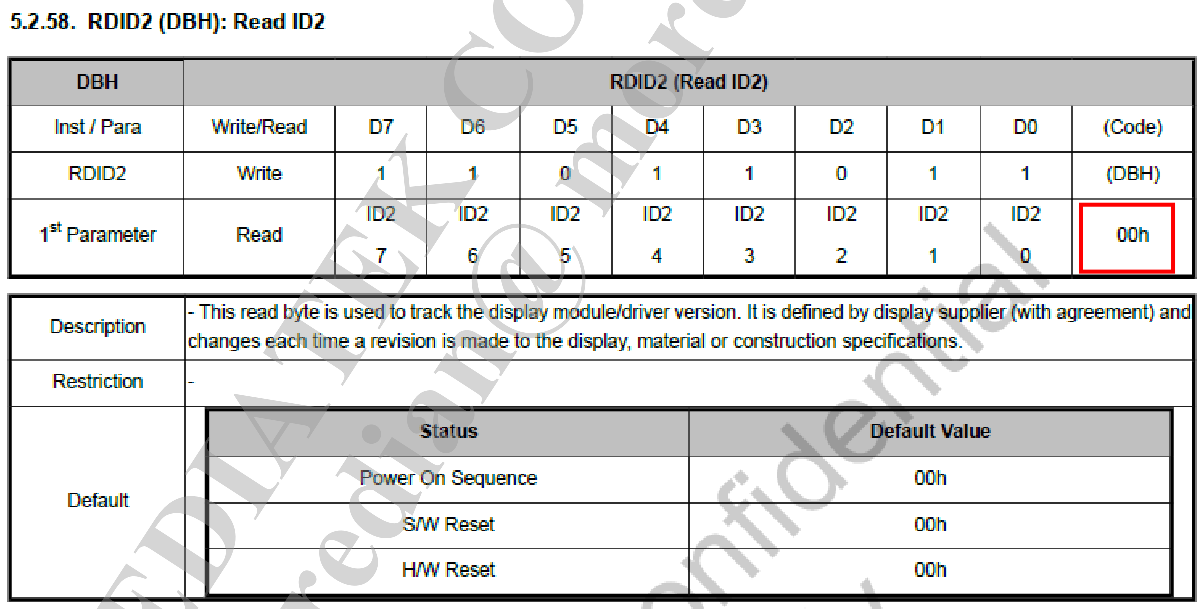


5.2.57. RDID1 (DAH): Read ID1

DAH	RDID1 (Read ID1)									
Inst / Para	Write/Read	D7	D6	D5	D4	D3	D2	D1	D0	(Code)
RDID1	Write	1	1	0	1	1	0	1	0	(DAH)
1st Parameter	Read	ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	40h

Description	- This read byte identifies the display module’s manufacturer.	
Restriction	- None	
Default	**Status**	**Default Value**
	Power On Sequence	40h
	S/W Reset	40h
	H/W Reset	40h

Figure 5-2. Panel ID1 register of otm1901a_fhd_dsi_vdo_tpv



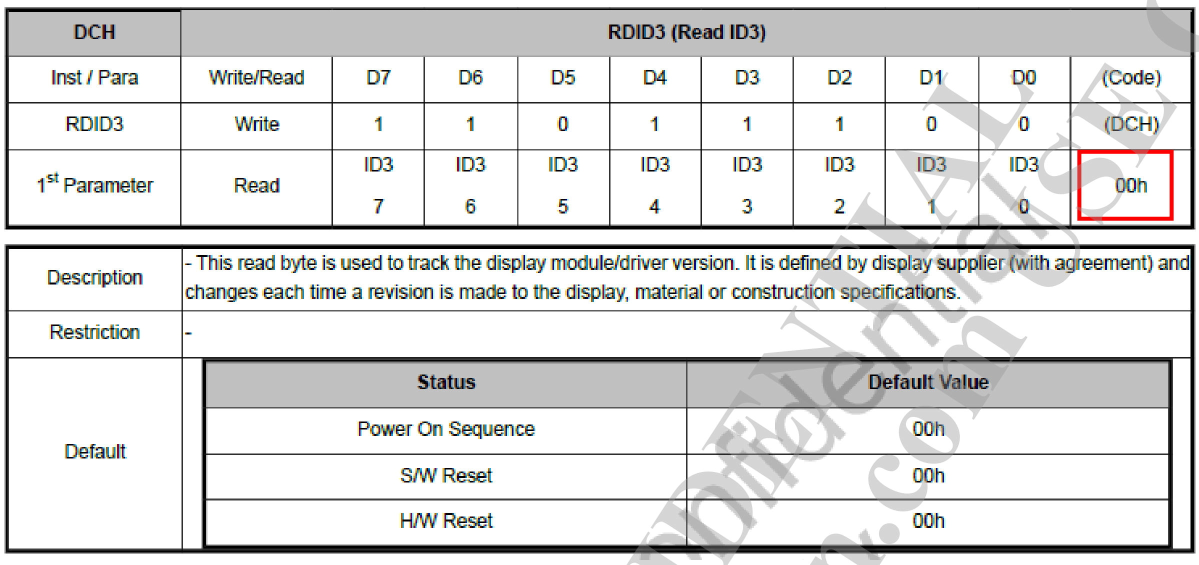
5.2.58. RDID2 (DBH): Read ID2

DBH	RDID2 (Read ID2)									
Inst / Para	Write/Read	D7	D6	D5	D4	D3	D2	D1	D0	(Code)
RDID2	Write	1	1	0	1	1	0	1	1	(DBH)
1st Parameter	Read	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	00h

Description	- This read byte is used to track the display module/driver version. It is defined by display supplier (with agreement) and changes each time a revision is made to the display, material or construction specifications.	
Restriction	-	
Default	**Status**	**Default Value**
	Power On Sequence	00h
	S/W Reset	00h
	H/W Reset	00h

Figure 5-3. Panel ID2 register of otm1901a_fhd_dsi_vdo_tpv

5.2.59. RDID3 (DCH): Read ID3



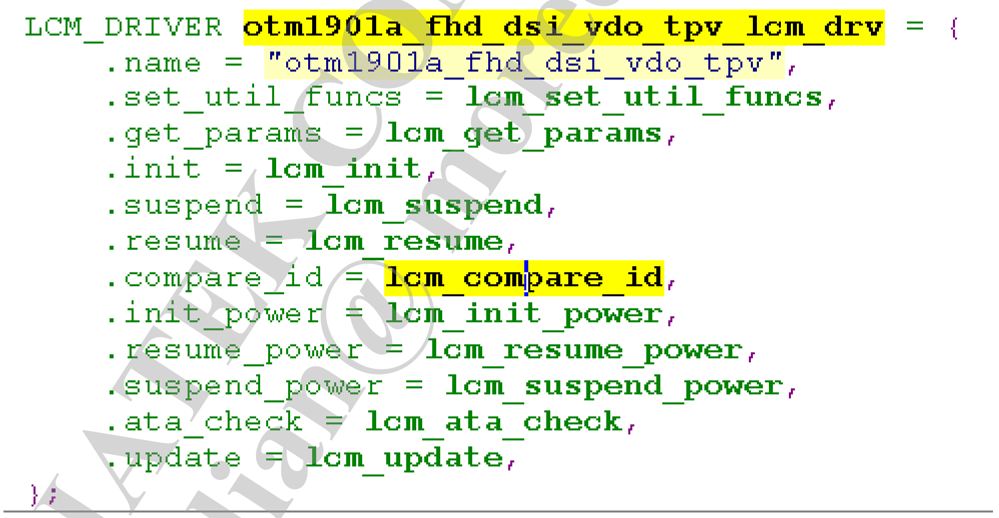
DCH	RDID3 (Read ID3)									
Inst / Para	Write/Read	D7	D6	D5	D4	D3	D2	D1	D0	(Code)
RDID3	Write	1	1	0	1	1	1	0	0	(DCH)
1st Parameter	Read	ID3 7	ID3 6	ID3 5	ID3 4	ID3 3	ID3 2	ID3 1	ID3 0	00h

Description	- This read byte is used to track the display module/driver version. It is defined by display supplier (with agreement) and changes each time a revision is made to the display, material or construction specifications.
Restriction	-
Default	

Status	Default Value
Power On Sequence	00h
S/W Reset	00h
H/W Reset	00h

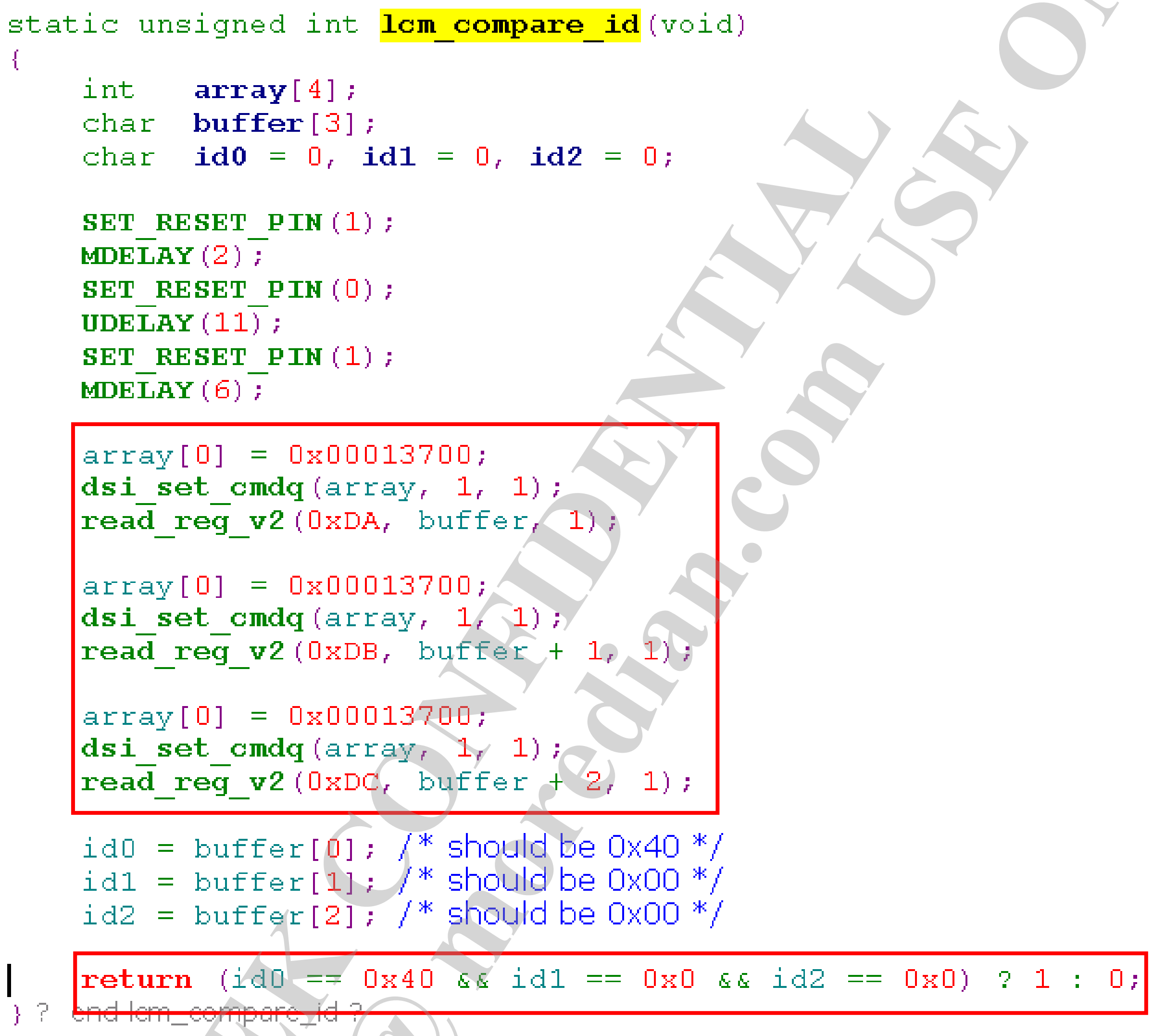
Figure 5-4. Panel ID3 register of otm1901a_fhd_dsi_vdo_tpv

Implement lcm_compare_id() function of “otm1901a_fhd_dsi_vdo_tpv” panel in lk otm1901a_fhd_dsi_vdo_tpv.c.



```
LCM_DRIVER otm1901a_fhd_dsi_vdo_tpv_lcm_drv = {
    .name = "otm1901a_fhd_dsi_vdo_tpv",
    .set_util_funcs = lcm_set_util_funcs,
    .get_params = lcm_get_params,
    .init = lcm_init,
    .suspend = lcm_suspend,
    .resume = lcm_resume,
    .compare_id = lcm_compare_id,
    .init_power = lcm_init_power,
    .resume_power = lcm_resume_power,
    .suspend_power = lcm_suspend_power,
    .ata_check = lcm_ata_check,
    .update = lcm_update,
};
```

Figure 5-5. Add lcm_compare_id() to lcm main structure

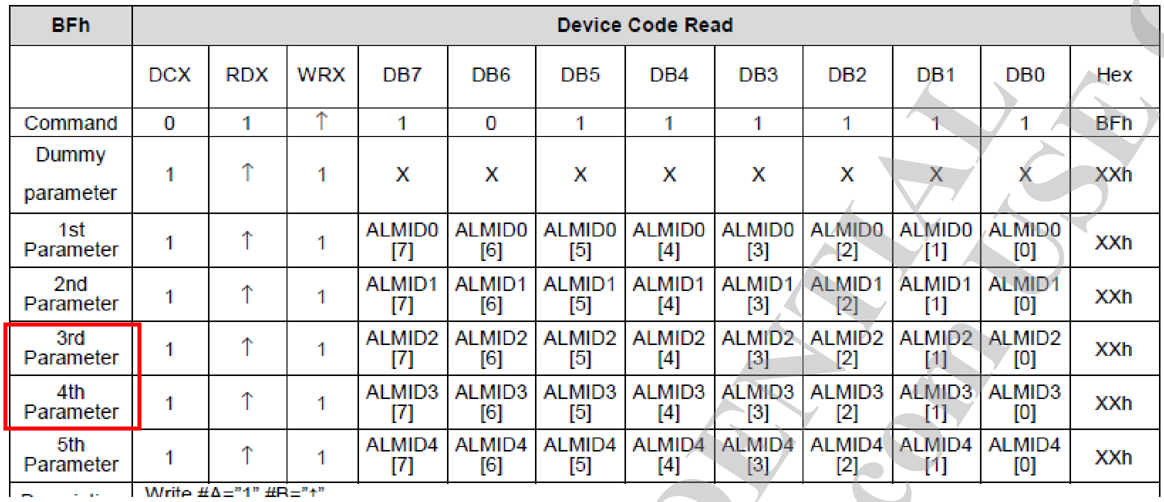


```
static unsigned int lcm_compare_id(void)
{
    int   array[4];
    char  buffer[3];
    char  id0 = 0, id1 = 0, id2 = 0;

    SET_RESET_PIN(1);
    MDELAY(2);
    SET_RESET_PIN(0);
    UDELAY(11);
    SET_RESET_PIN(1);
    MDELAY(6);

    array[0] = 0x00013700;
    dsi_set_cmdq(array, 1, 1);
    read_reg_v2(0xDA, buffer, 1);

    array[0] = 0x00013700;
    dsi_set_cmdq(array, 1, 1);
    read_reg_v2(0xDB, buffer + 1, 1);

    array[0] = 0x00013700;
    dsi_set_cmdq(array, 1, 1);
    read_reg_v2(0xDC, buffer + 2, 1);

    id0 = buffer[0]; /* should be 0x40 */
    id1 = buffer[1]; /* should be 0x00 */
    id2 = buffer[2]; /* should be 0x00 */

    return (id0 == 0x40 && id1 == 0x0 && id2 == 0x0) ? 1 : 0;
} ? end lcm_compare_id ?
```

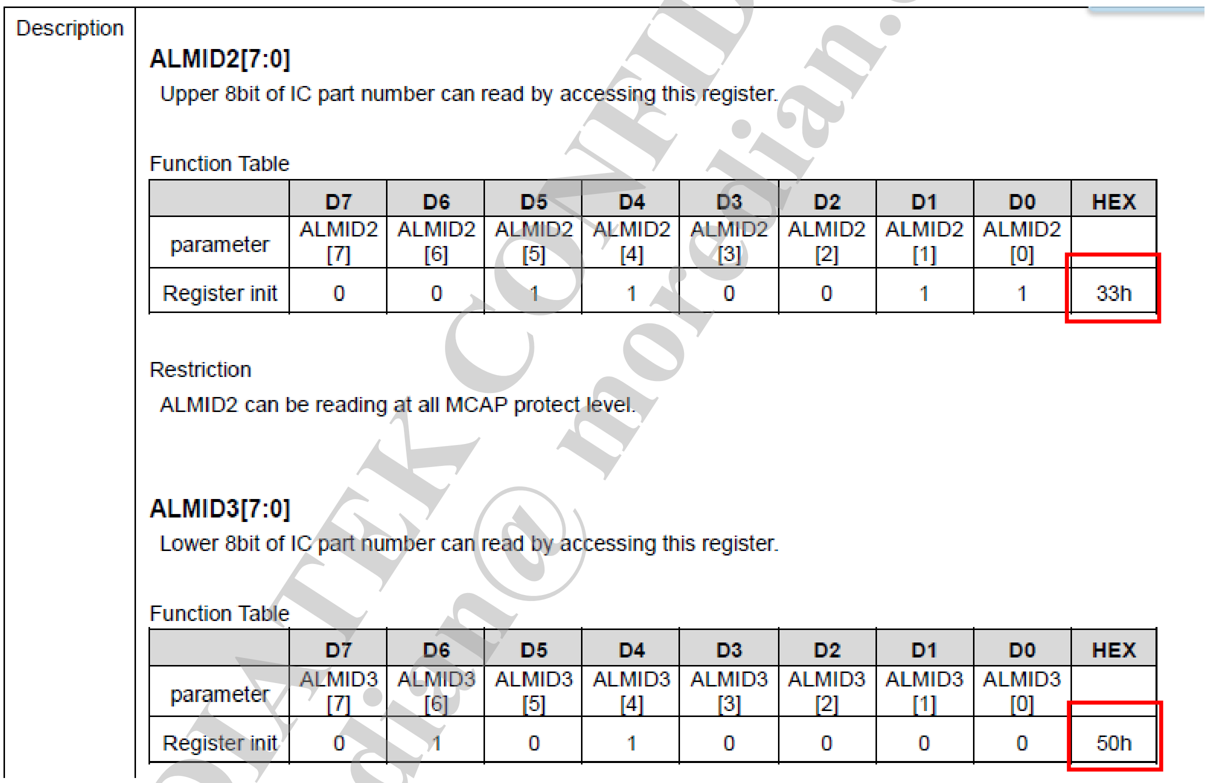
Figure 5-6. Implement lcm_compare_id() function

Take “r63350a_fhd_dsi_vdo_truly” panel for example, its panel ID is in lcm registers of 0xBFh from panel spec, you can get ID by reading this registers. Panel ID is the third and forth parameters of 0xBFh. The third parameter is 0x 33, the forth parameter is 0x50.

Device Code Read: BFh



BFh	Device Code Read										
	DCX	RDX	WRX	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Hex
Command	0	1	↑	1	0	1	1	1	1	1	1	BFh
Dummy parameter	1	↑	1	X	X	X	X	X	X	X	X	XXh
1st Parameter	1	↑	1	ALMID0 [7]	ALMID0 [6]	ALMID0 [5]	ALMID0 [4]	ALMID0 [3]	ALMID0 [2]	ALMID0 [1]	ALMID0 [0]	XXh
2nd Parameter	1	↑	1	ALMID1 [7]	ALMID1 [6]	ALMID1 [5]	ALMID1 [4]	ALMID1 [3]	ALMID1 [2]	ALMID1 [1]	ALMID1 [0]	XXh
3rd Parameter	1	↑	1	ALMID2 [7]	ALMID2 [6]	ALMID2 [5]	ALMID2 [4]	ALMID2 [3]	ALMID2 [2]	ALMID2 [1]	ALMID2 [0]	XXh
4th Parameter	1	↑	1	ALMID3 [7]	ALMID3 [6]	ALMID3 [5]	ALMID3 [4]	ALMID3 [3]	ALMID3 [2]	ALMID3 [1]	ALMID3 [0]	XXh
5th Parameter	1	↑	1	ALMID4 [7]	ALMID4 [6]	ALMID4 [5]	ALMID4 [4]	ALMID4 [3]	ALMID4 [2]	ALMID4 [1]	ALMID4 [0]	XXh
	Write #A=”1” #B=”↑”											



Description

ALMID2[7:0]

Upper 8bit of IC part number can read by accessing this register.

Function Table

	D7	D6	D5	D4	D3	D2	D1	D0	HEX
parameter	ALMID2 [7]	ALMID2 [6]	ALMID2 [5]	ALMID2 [4]	ALMID2 [3]	ALMID2 [2]	ALMID2 [1]	ALMID2 [0]	
Register init	0	0	1	1	0	0	1	1	33h

Restriction

ALMID2 can be reading at all MCAP protect level.

ALMID3[7:0]

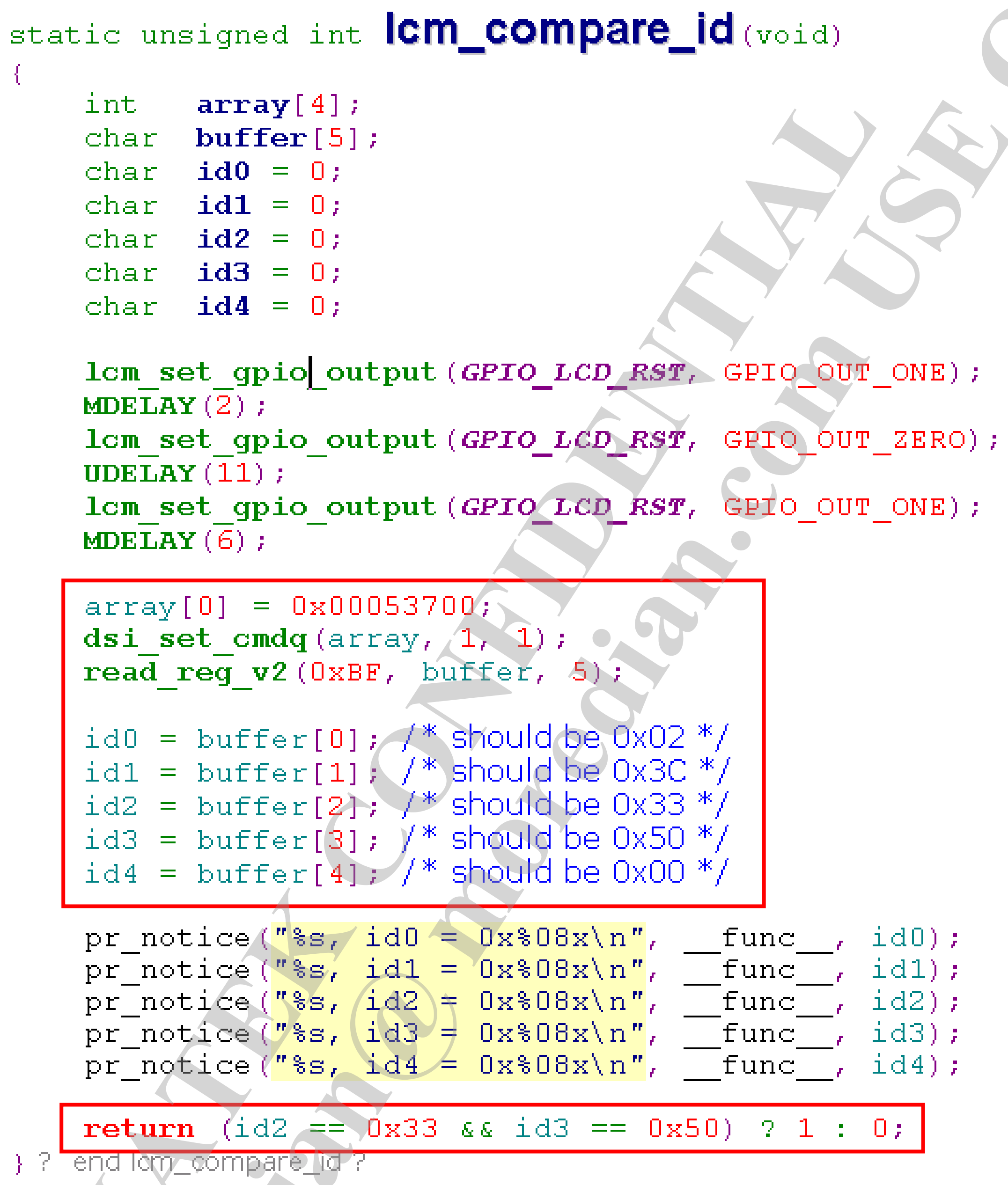
Lower 8bit of IC part number can read by accessing this register.

Function Table

	D7	D6	D5	D4	D3	D2	D1	D0	HEX
parameter	ALMID3 [7]	ALMID3 [6]	ALMID3 [5]	ALMID3 [4]	ALMID3 [3]	ALMID3 [2]	ALMID3 [1]	ALMID3 [0]	
Register init	0	1	0	1	0	0	0	0	50h

Figure 5-7. Panel ID of r63350a_fhd_dsi_vdo_truly

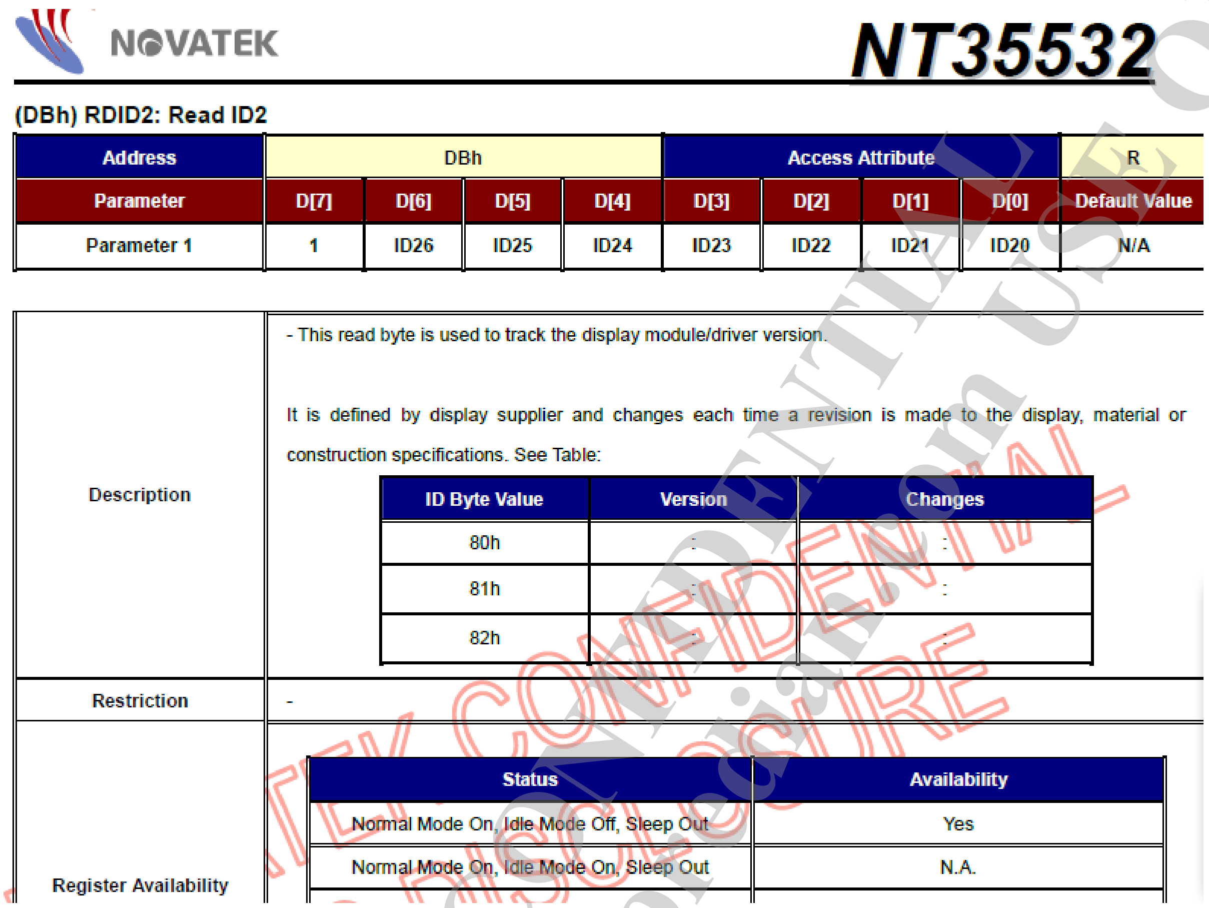
Implement lcm_compare_id() function of “r63350a_fhd_dsi_vdo_truly” panel in lk r63350a_fhd_dsi_vdo_truly.c.



```
static unsigned int lcm_compare_id(void)
{
	int	array[4];
	char	buffer[5];
	char	id0 = 0;
	char	id1 = 0;
	char	id2 = 0;
	char	id3 = 0;
	char	id4 = 0;

	lcm_set_gpio_output(GPIO_LCD_RST, GPIO_OUT_ONE);
	MDELAY(2);
	lcm_set_gpio_output(GPIO_LCD_RST, GPIO_OUT_ZERO);
	UDELAY(11);
	lcm_set_gpio_output(GPIO_LCD_RST, GPIO_OUT_ONE);
	MDELAY(6);

	array[0] = 0x00053700;
	dsi_set_cmdq(array, 1, 1);
	read_reg_v2(0xBF, buffer, 5);

	id0 = buffer[0]; /* should be 0x02 */
	id1 = buffer[1]; /* should be 0x3C */
	id2 = buffer[2]; /* should be 0x33 */
	id3 = buffer[3]; /* should be 0x50 */
	id4 = buffer[4]; /* should be 0x00 */

	pr_notice("%s, id0 = 0x%08x\n", __func__, id0);
	pr_notice("%s, id1 = 0x%08x\n", __func__, id1);
	pr_notice("%s, id2 = 0x%08x\n", __func__, id2);
	pr_notice("%s, id3 = 0x%08x\n", __func__, id3);
	pr_notice("%s, id4 = 0x%08x\n", __func__, id4);

	return (id2 == 0x33 && id3 == 0x50) ? 1 : 0;
} ? end lcm_compare_id ?
```

Figure 5-8. Implement lcm_compare_id() function

Take “nt35532_fhd_dsi_vdo_sharp” panel for example, its panel ID is in lcm registers of 0xDBh from panel spec, you can get ID by reading this registers. Panel ID is 0x80.



NOVATEK NT35532

(DBh) RDID2: Read ID2

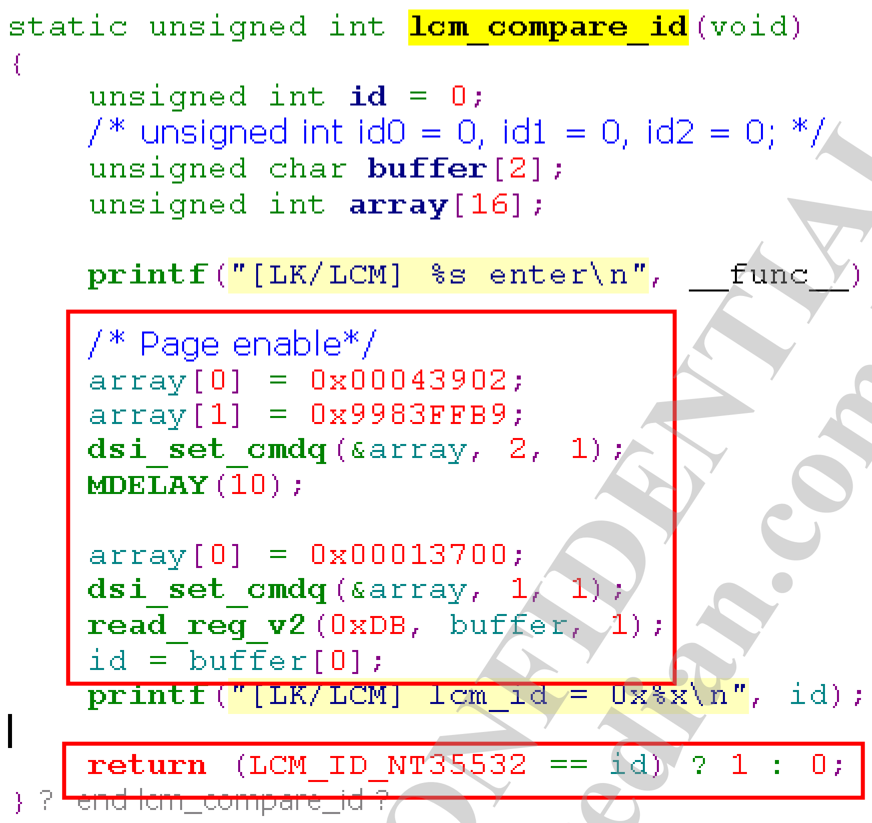
Address	DBh				Access Attribute				R
Parameter	D[7]	D[6]	D[5]	D[4]	D[3]	D[2]	D[1]	D[0]	Default Value
Parameter 1	1	ID26	ID25	ID24	ID23	ID22	ID21	ID20	N/A

Description	- This read byte is used to track the display module/driver version. It is defined by display supplier and changes each time a revision is made to the display, material or construction specifications. See Table:
Restriction	-
Register Availability	

ID Byte Value	Version	Changes
80h	-	-
81h	-	-
82h	-	-

Status	Availability
Normal Mode On, Idle Mode Off, Sleep Out	Yes
Normal Mode On, Idle Mode On, Sleep Out	N.A.

Figure 5-9. Panel ID of nt35532_fhd_dsi_vdo_sharp



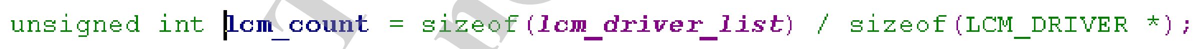
```
static unsigned int lcm_compare_id(void)
{
    unsigned int id = 0;
    /* unsigned int id0 = 0, id1 = 0, id2 = 0; */
    unsigned char buffer[2];
    unsigned int array[16];

    printf("[LK/LCM] %s enter\n", __func__);

    /* Page enable*/
    array[0] = 0x00043902;
    array[1] = 0x9983FFB9;
    dsi_set_cmdq(&array, 2, 1);
    MDELAY(10);

    array[0] = 0x00013700;
    dsi_set_cmdq(&array, 1, 1);
    read_reg_v2(0xDB, buffer, 1);
    id = buffer[0];
    printf("[LK/LCM] lcm_id = 0x%x\n", id);

    return (LCM_ID_NT35532 == id) ? 1 : 0;
} ? end lcm_compare_id ?
```

Figure 5-10. Implement lcm_compare_id() function

5.5 Implement distinguish panel in disp_lcm_probe() of lk

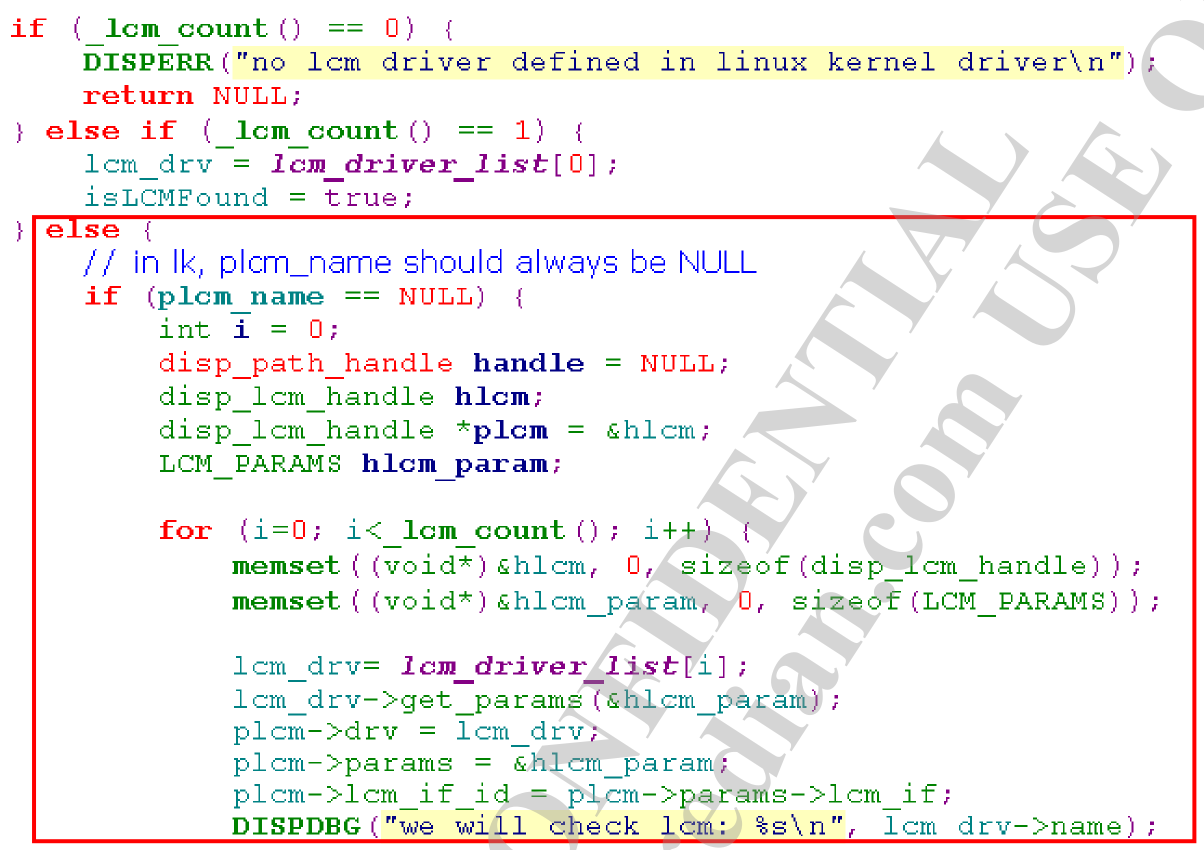
Path: Vendor\mediatek\proprietary\bootable\bootloader\lk\platform\mt6771\disp_lcm.c

In disp_lcm_probe() function, as we have configure 3 LCM driver, so lcm_count is 3. And LCM is first initialize in lk, so plcm_name is NULL. Will go to else condition to distinguish panel by panel ID.



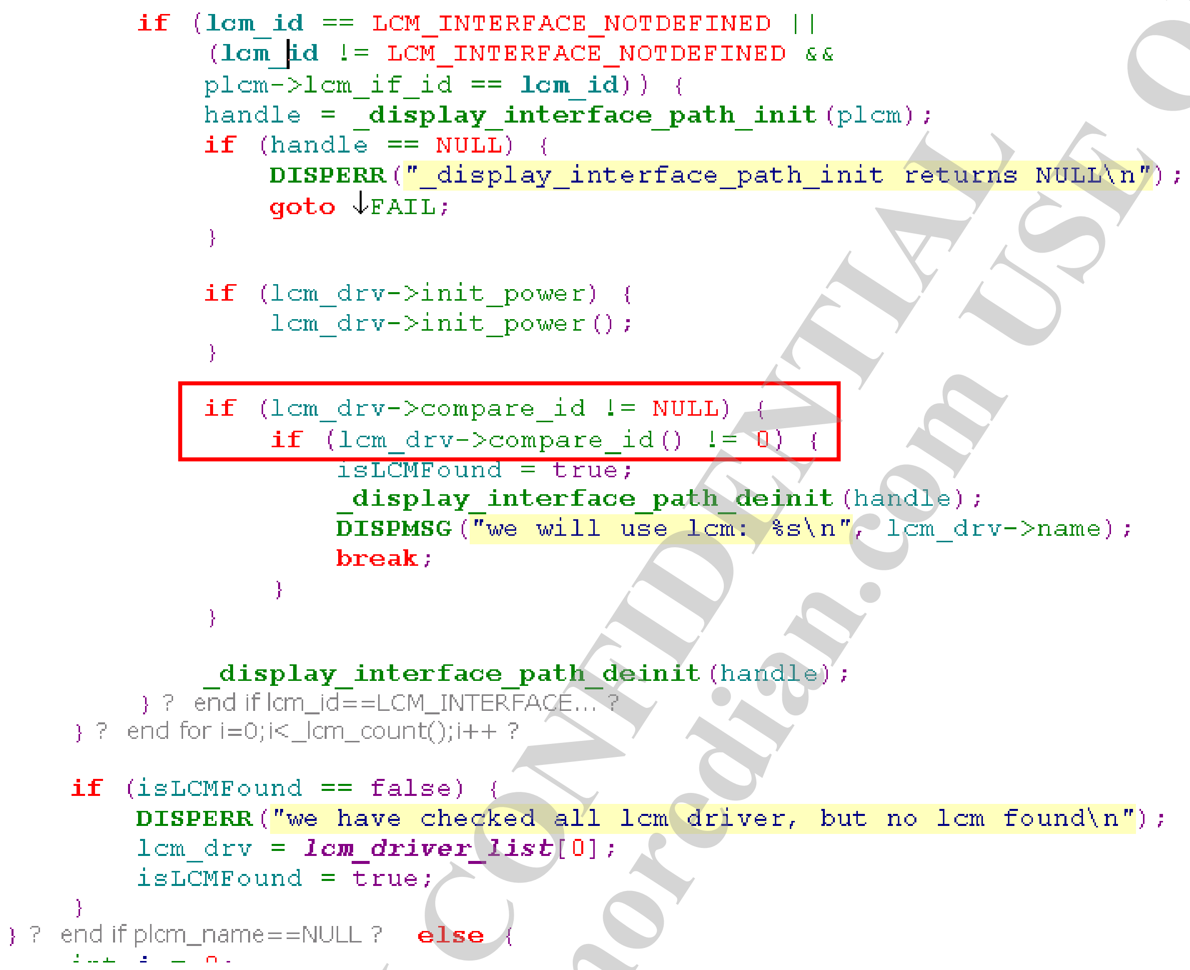
```
unsigned int lcm_count = sizeof(lcm_driver_list) / sizeof(LCM_DRIVER *);
```

Figure 5-11.How t calculate lcm_count



```
if (_lcm_count() == 0) {
    DISPERR("no lcm driver defined in linux kernel driver\n");
    return NULL;
} else if (_lcm_count() == 1) {
    lcm_drv = lcm_driver_list[0];
    isLCMFound = true;
} else {
    // in lk, plcm_name should always be NULL
    if (plcm_name == NULL) {
        int i = 0;
        disp_path_handle handle = NULL;
        disp_lcm_handle hlcm;
        disp_lcm_handle *plcm = &hlcm;
        LCM_PARAMS hlcm_param;

        for (i=0; i<_lcm_count(); i++) {
            memset((void*)&hlcm, 0, sizeof(disp_lcm_handle));
            memset((void*)&hlcm_param, 0, sizeof(LCM_PARAMS));

            lcm_drv= lcm_driver_list[i];
            lcm_drv->get_params(&hlcm_param);
            plcm->drv = lcm_drv;
            plcm->params = &hlcm_param;
            plcm->lcm_if_id = plcm->params->lcm_if;
            DISPDBG("we will check lcm: %s\n", lcm_drv->name);
```

Figure 5-12. disp_lcm_probe() of lk



```
        if (lcm_id == LCM_INTERFACE_NOTDEFINED ||
            (lcm_id != LCM_INTERFACE_NOTDEFINED &&
            plcm->lcm_if_id == lcm_id)) {
            handle = _display_interface_path_init(plcm);
            if (handle == NULL) {
                DISPERR("_display_interface_path_init returns NULL\n");
                goto ↓FAIL;
            }

            if (lcm_drv->init_power) {
                lcm_drv->init_power();
            }

            if (lcm_drv->compare_id != NULL) {
                if (lcm_drv->compare_id() != 0) {
                    isLCMFound = true;
                    _display_interface_path_deinit(handle);
                    DISPMSG("we will use lcm: %s\n", lcm_drv->name);
                    break;
                }
            }

            _display_interface_path_deinit(handle);
        } ? end if lcm_id==LCM_INTERFACE... ?
    } ? end for i=0;i<_lcm_count();i++ ?

    if (isLCMFound == false) {
        DISPERR("we have checked all lcm driver, but no lcm found\n");
        lcm_drv = lcm_driver_list[0];
        isLCMFound = true;
    }
} ? end if plcm_name==NULL ?  else {
```

Figure 5-13. disp_lcm_probe() of lk

5.6 Distinguish panel Log

Below log is for panel distinguish ID. This part is done in lk.

[443] [DISP]func|disp_lcm_probe

[444] [DISP]we will check lcm: r63350a_fhd_dsi_vdo_truly ///First configure panel

[LK/LCM][r63350a] lcm_init_power() enter

[r63350a] lcm_compare_id, id0 = 0x00000000 ///Read ID NG

[r63350a] lcm_compare_id, id1 = 0x00000000

[r63350a] lcm_compare_id, id2 = 0x00000000

[r63350a] lcm_compare_id, id3 = 0x00000000

[r63350a] lcm_compare_id, id4 = 0x00000000

```
[613] [DISP]we will check lcm: otm1901a_fhd_dsi_vdo_tpv ///Second configure panel

[LK/LCM][otm1901a] lcm_init_power() enter

[otm1901a] lcm_compare_id, id0 = 0x00000000 ///Read ID NG

[otm1901a] lcm_compare_id, id1 = 0x00000080

[otm1901a] lcm_compare_id, id2 = 0x00000000

[786] [DISP]we will check lcm: nt35532_fhd_dsi_vdo_sharp_lcm_drv ///Third configure panel

[LK/LCM][nt35532] lcm_compare_id enter

[LK/LCM][nt35532] lcm_id = 0x80 ///Read ID OK

[937] [DISP]we will use lcm: nt35532_fhd_dsi_vdo_sharp_lcm_drv ///Use third panel to power on

[938] [DISPCHECK]******** dump lcm driver information ********

[939] [DISPCHECK][LCM], name: nt35532_fhd_dsi_vdo_sharp_lcm_drv
```

6. Conclusion

If you have any question about how to bring up a DSI panel on aiv8183, please feel free to connect me.