everyday genius

MT8788 Camera Bring Up SOP

Version:	1.0
Release date:	2019-10-16

Specifications are subject to change without notice.

Document Revision History

Revision	Date	Description
1.0	2019-10-08	Initial Draft

Contents

1. Introduction

In this document, we'll introduce MT8183 camera sensor porting.

1.1 Purpose

This document is to guide customers on how to porting camera sensor in mtk platform.

1.2 Definitions, Acronyms and Abbreviations

$(project)

Take MTK's turnkey solution as an example, $(project) correspond to tb8183m1_64_bsp.

$(kernel_version)

kernel-4.14

1.3 References

N/A

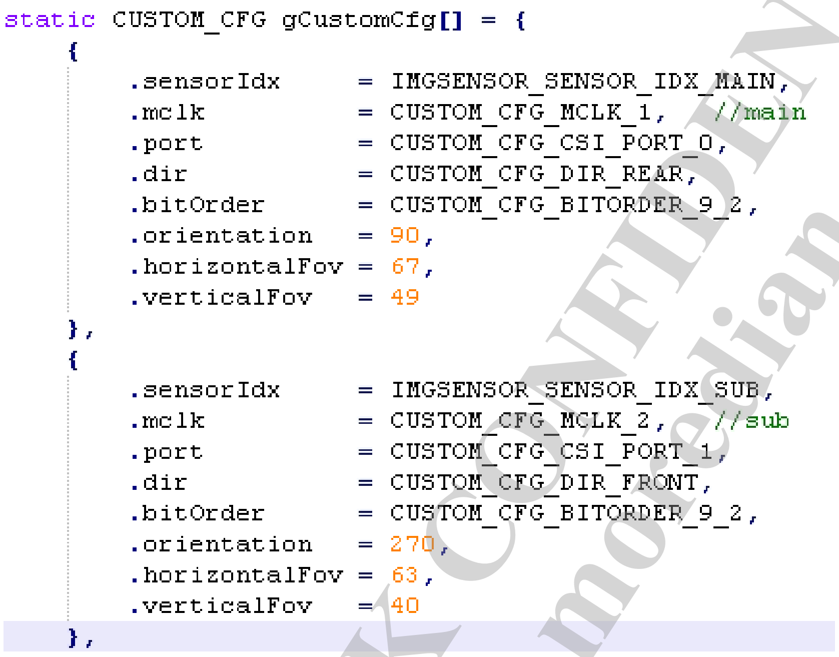
1.4 Overview

N/A

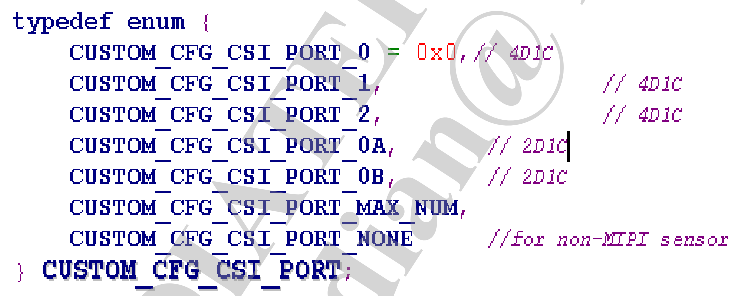
2. Camera sensor porting

2.1 Mipi Port Connection Customization

-- modify the customization setting by your hw layout.
-- cfg_setting_imgsensor.cpp(\custom\$project$\hal\imgsensor_src\)(file Priority: project > platform > common)。



```
static CUSTOM_CFG gCustomCfg[] = {
    {
        .sensorIdx      = IMGSENSOR_SENSOR_IDX_MAIN,
        .mclk           = CUSTOM_CFG_MCLK_1,    //main
        .port           = CUSTOM_CFG_CSI_PORT_0,
        .dir            = CUSTOM_CFG_DIR_REAR,
        .bitOrder       = CUSTOM_CFG_BITORDER_9_2,
        .orientation    = 90,
        .horizontalFov = 67,
        .verticalFov    = 49
    },
    {
        .sensorIdx      = IMGSENSOR_SENSOR_IDX_SUB,
        .mclk           = CUSTOM_CFG_MCLK_2,    //sub
        .port           = CUSTOM_CFG_CSI_PORT_1,
        .dir            = CUSTOM_CFG_DIR_FRONT,
        .bitOrder       = CUSTOM_CFG_BITORDER_9_2,
        .orientation    = 270,
        .horizontalFov = 63,
        .verticalFov    = 40
    },
```



```
typedef enum {
    CUSTOM_CFG_CSI_PORT_0 = 0x0,// 4D1C
    CUSTOM_CFG_CSI_PORT_1,              // 4D1C
    CUSTOM_CFG_CSI_PORT_2,              // 4D1C
    CUSTOM_CFG_CSI_PORT_0A,       // 2D1C
    CUSTOM_CFG_CSI_PORT_0B,       // 2D1C
    CUSTOM_CFG_CSI_PORT_MAX_NUM,
    CUSTOM_CFG_CSI_PORT_NONE      //for non-MIPI sensor
} CUSTOM_CFG_CSI_PORT;
```

camera_custom_imgsensor_cfg.h

2.2 Mclk Connection Customization

-- modify the customization setting by your hw layout.
-- cfg_setting_imgsensor.cpp(vendor\mediatek\proprietary\custom\$project$\hal\imgsensor_src)

```
static CUSTOM_CFG gCustomCfg[] = {
    {
        .sensorIdx      = IMGSENSOR_SENSOR_IDX_MAIN,
        .mclk           = CUSTOM_CFG_MCLK_1,    //main
        .port           = CUSTOM_CFG_CSI_PORT_0,
        .dir            = CUSTOM_CFG_DIR_REAR,
        .bitOrder       = CUSTOM_CFG_BITORDER_9_2,
        .orientation    = 90,
        .horizontalFov = 67,
        .verticalFov    = 49
    },
    {
        .sensorIdx      = IMGSENSOR_SENSOR_IDX_SUB,
        .mclk           = CUSTOM_CFG_MCLK_2,    //sub
        .port           = CUSTOM_CFG_CSI_PORT_1,
        .dir            = CUSTOM_CFG_DIR_FRONT,
        .bitOrder       = CUSTOM_CFG_BITORDER_9_2,
        .orientation    = 270,
        .horizontalFov = 63,
        .verticalFov    = 40
    },
```

```
typedef enum {
    CUSTOM_CFG_MCLK_1 = 0x0,      //mclk1
    CUSTOM_CFG_MCLK_2,   //mclk2
    CUSTOM_CFG_MCLK_3,   //mclk3
    CUSTOM_CFG_MCLK_4,
    CUSTOM_CFG_MCLK_5,
    CUSTOM_CFG_MCLK_MAX_NUM,
    CUSTOM_CFG_MCLK_NONE
} CUSTOM_CFG_MCLK;
```

camera_custom_imgsensor_cfg.h

2.3 Mclk On/Off control

-- Customization mclk on/off control in power on sequence
($(kernel_version)\drivers\misc\mediatek\imgsensor\src\mt6771\camera_hw\imgsensor_cfg_table.c)

```
#if defined(OV13855_MIPI_RAW)
        {
         SENSOR_DRVNAME_OV13855_MIPI_RAW,
         {
          {SensorMCLK, Vol_High, 0},
          {PDN, Vol_Low, 0},
          {RST, Vol_Low, 0},
          {DOVDD, Vol_1800, 1},
          {AVDD, Vol_2800, 1},
          {DVDD, Vol_1200, 5},
          {AFVDD, Vol_2800, 1},
          {PDN, Vol_High, 1},
          {RST, Vol_High, 2}
          },
         },
#endif
```

```
struct IMGSENSOR_HW_CFG imgsensor_custom_config[] = {
        {
         IMGSENSOR_SENSOR_IDX_MAIN,
         IMGSENSOR_I2C_DEV_0,
         {
          {IMGSENSOR_HW_PIN_MCLK, IMGSENSOR_HW_ID_MCLK},
          {IMGSENSOR_HW_PIN_AVDD, IMGSENSOR_HW_ID_REGULATOR},
          {IMGSENSOR_HW_PIN_DOVDD, IMGSENSOR_HW_ID_REGULATOR},
          {IMGSENSOR_HW_PIN_DVDD, IMGSENSOR_HW_ID_GPIO},
          {IMGSENSOR_HW_PIN_PDN, IMGSENSOR_HW_ID_GPIO},
          {IMGSENSOR_HW_PIN_RST, IMGSENSOR_HW_ID_GPIO},
          {IMGSENSOR_HW_PIN_NONE, IMGSENSOR_HW_ID_NONE},
          },
         },
        {
         IMGSENSOR_SENSOR_IDX_SUB,
         IMGSENSOR_I2C_DEV_1,
         {
          {IMGSENSOR_HW_PIN_MCLK, IMGSENSOR_HW_ID_MCLK},
          {IMGSENSOR_HW_PIN_AVDD, IMGSENSOR_HW_ID_REGULATOR},
          {IMGSENSOR_HW_PIN_DOVDD, IMGSENSOR_HW_ID_REGULATOR},
          {IMGSENSOR_HW_PIN_DVDD, IMGSENSOR_HW_ID_REGULATOR},
          {IMGSENSOR_HW_PIN_PDN, IMGSENSOR_HW_ID_GPIO},
          {IMGSENSOR_HW_PIN_RST, IMGSENSOR_HW_ID_GPIO},
          {IMGSENSOR_HW_PIN_NONE, IMGSENSOR_HW_ID_NONE},
          },
         },
```

2.4 Camera Driver File Path

Kernel driver()
$(kernel_version)\drivers\misc\mediatek\imgsensor\src\{platform}
$(kernel_version)\drivers\misc\mediatek\imgsensor\inc\{platform}

hal driver()
vendor\mediatek\proprietary\custom\{project}\hal

2.5 Add a new sensor

2.5.1.1 Modify imgsensor configuration

2.5.1.1.1 Step1 device\mediateksample\$project$\ProjectConfig.mk

Modify imgsensor configuration

eg:main(rear camera) imx135_mipi_raw, sub (front camera)ov5648_mipi_raw)

```
CUSTOM_HAL_IMGSENSOR = imx135_mipi_raw ov5648_mipi_raw
CUSTOM_KERNEL_IMGSENSOR = imx135_mipi_raw ov5648_mipi_raw
CUSTOM_HAL_MAIN_IMGSENSOR = imx135_mipi_raw
CUSTOM_HAL_SUB_IMGSENSOR = ov5648_mipi_raw
CUSTOM_KERNEL_MAIN_IMGSENSOR = imx135_mipi_raw
CUSTOM_KERNEL_SUB_IMGSENSOR = ov5648_mipi_raw
```

Modify lens

if have no AF set it as dummy_lens; YUV sensor has af set it as sensordrive; RAW sensor has af set it as lens name (eg:fm50af ,ov8825af)

imx135_mipi_raw has AF, sub sensor has no af

```
CUSTOM_HAL_LENS = dw9714af dummy_lens
CUSTOM_KERNEL_LENS = dw9714af dummy_lens
CUSTOM_HAL_MAIN_LENS = d29714af
CUSTOM_HAL_SUB_LENS = dummy_lens
CUSTOM_KERNEL_MAIN_LENS = dw9714af
CUSTOM_KERNEL_SUB_LENS = dummy_lens
```

Modify flashlight

If it has Flashlight set it as constant_flashlight, if no set it as dummy_flashlight

```
CUSTOM_HAL_FLASHLIGHT = constant_flashlight
CUSTOM_KERNEL_FLASHLIGHT = constant_flashlight
```

2.5.1.1.2 Step2 \device\mediatek\common\kernel-headers\kd_imgsensor.h

\kernel-4.4\drivers\misc\mediatek\imgsensor\inc\kd_imgsensor.h

#define OV5648_SENSOR_ID 0x5648Config sensor ID

#define SENSOR_DRVNAME_OV5648_MIPI_RAW “ov5648mipiraw“
Define sensor device driver name

2.5.1.1.3 Step3

$(kernel_version)\drivers\misc\mediatek\imgsensor\src\common\v1_1\imgsensor_sensor_list.h

UINT32 OV5648_MIPI_RAW_SensorInit(PSENSOR_FUNCTION_STRUCT *pfFunc);

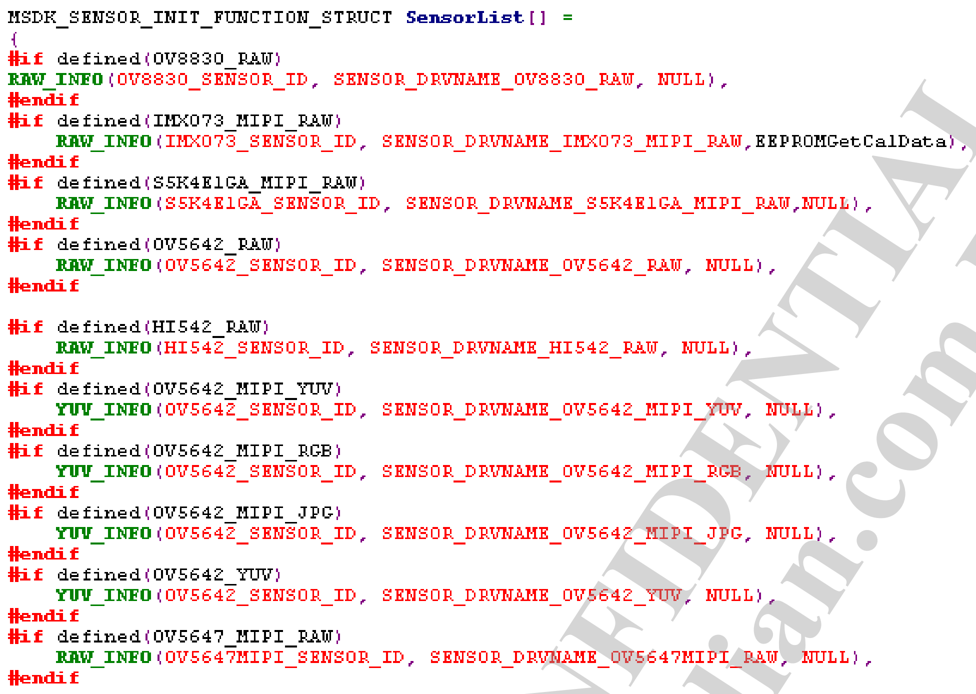
$(kernel_version)\drivers\misc\mediatek\imgsensor\src\common\v1_1\imgsensor_sensor_list.c

```
kdSensorList[]
#if defined(OV5648_MIPI_RAW)
 {OV5648_SENSOR_ID,SENSOR_DRVNAME_OV5648_MIPI_RAW,OV5648_MIPI_RAW_SensorInit},
#endif
```

\vendor\mediatek\proprietary\custom\mt6771\hal\imgsensor_src\sensorlist.cpp

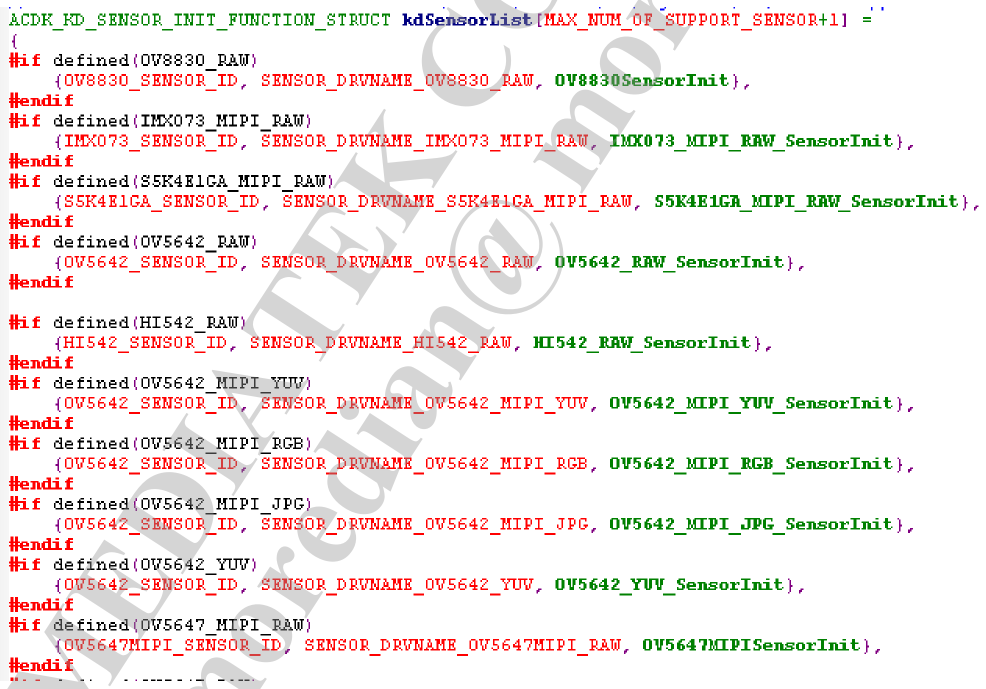
```
SensorList[]
#if defined(OV5648_MIPI_RAW)
RAW_INFO(OV5648_SENSOR_ID, SENSOR_DRVNAME_OV5648_MIPI_RAW, NULL),
#endif
```

The order of the SensorList[] in sensorlist.cpp and the kdSensorList[] in imgsensor_sensor_list.c must be the same, otherwise the id of user space & kernel space will not be able to match.



```
MSDK_SENSOR_INIT_FUNCTION_STRUCT SensorList[] =
{
#if defined(OV8830_RAW)
RAW_INFO(OV8830_SENSOR_ID, SENSOR_DRVNAME_OV8830_RAW, NULL),
#endif
#if defined(IMX073_MIPI_RAW)
    RAW_INFO(IMX073_SENSOR_ID, SENSOR_DRVNAME_IMX073_MIPI_RAW,EEPROMGetCalData),
#endif
#if defined(S5K4E1GA_MIPI_RAW)
    RAW_INFO(S5K4E1GA_SENSOR_ID, SENSOR_DRVNAME_S5K4E1GA_MIPI_RAW,NULL),
#endif
#if defined(OV5642_RAW)
    RAW_INFO(OV5642_SENSOR_ID, SENSOR_DRVNAME_OV5642_RAW, NULL),
#endif

#if defined(HI542_RAW)
    RAW_INFO(HI542_SENSOR_ID, SENSOR_DRVNAME_HI542_RAW, NULL),
#endif
#if defined(OV5642_MIPI_YUV)
    YUV_INFO(OV5642_SENSOR_ID, SENSOR_DRVNAME_OV5642_MIPI_YUV, NULL),
#endif
#if defined(OV5642_MIPI_RGB)
    YUV_INFO(OV5642_SENSOR_ID, SENSOR_DRVNAME_OV5642_MIPI_RGB, NULL),
#endif
#if defined(OV5642_MIPI_JPG)
    YUV_INFO(OV5642_SENSOR_ID, SENSOR_DRVNAME_OV5642_MIPI_JPG, NULL),
#endif
#if defined(OV5642_YUV)
    YUV_INFO(OV5642_SENSOR_ID, SENSOR_DRVNAME_OV5642_YUV, NULL),
#endif
#if defined(OV5647_MIPI_RAW)
    RAW_INFO(OV5647MIPI_SENSOR_ID, SENSOR_DRVNAME_OV5647MIPI_RAW, NULL),
#endif
```

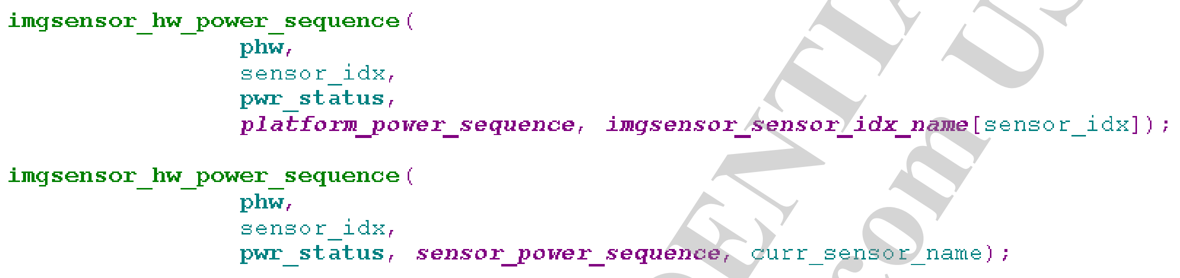
And it is recommended to arrange the resolutions in descending order.



```
ACDK_KD_SENSOR_INIT_FUNCTION_STRUCT kdSensorList[MAX_NUM_OF_SUPPORT_SENSOR+1] =
{
#if defined(OV8830_RAW)
    {OV8830_SENSOR_ID, SENSOR_DRVNAME_OV8830_RAW, OV8830SensorInit},
#endif
#if defined(IMX073_MIPI_RAW)
    {IMX073_SENSOR_ID, SENSOR_DRVNAME_IMX073_MIPI_RAW, IMX073_MIPI_RAW_SensorInit},
#endif
#if defined(S5K4E1GA_MIPI_RAW)
    {S5K4E1GA_SENSOR_ID, SENSOR_DRVNAME_S5K4E1GA_MIPI_RAW, S5K4E1GA_MIPI_RAW_SensorInit},
#endif
#if defined(OV5642_RAW)
    {OV5642_SENSOR_ID, SENSOR_DRVNAME_OV5642_RAW, OV5642_RAW_SensorInit},
#endif

#if defined(HI542_RAW)
    {HI542_SENSOR_ID, SENSOR_DRVNAME_HI542_RAW, HI542_RAW_SensorInit},
#endif
#if defined(OV5642_MIPI_YUV)
    {OV5642_SENSOR_ID, SENSOR_DRVNAME_OV5642_MIPI_YUV, OV5642_MIPI_YUV_SensorInit},
#endif
#if defined(OV5642_MIPI_RGB)
    {OV5642_SENSOR_ID, SENSOR_DRVNAME_OV5642_MIPI_RGB, OV5642_MIPI_RGB_SensorInit},
#endif
#if defined(OV5642_MIPI_JPG)
    {OV5642_SENSOR_ID, SENSOR_DRVNAME_OV5642_MIPI_JPG, OV5642_MIPI_JPG_SensorInit},
#endif
#if defined(OV5642_YUV)
    {OV5642_SENSOR_ID, SENSOR_DRVNAME_OV5642_YUV, OV5642_YUV_SensorInit},
#endif
#if defined(OV5647_MIPI_RAW)
    {OV5647MIPI_SENSOR_ID, SENSOR_DRVNAME_OV5647MIPI_RAW, OV5647MIPISensorInit},
#endif
```

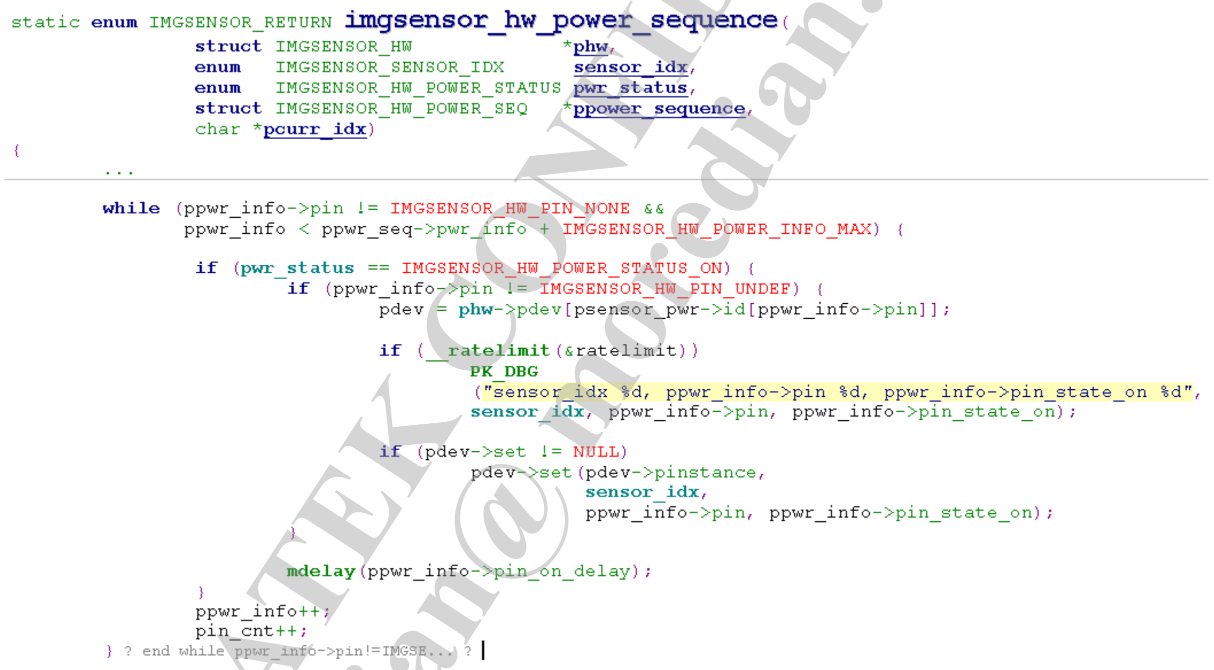
2.6 Modify Power On/Off

$(kernel_version)\drivers\misc\mediatek\imgsensor\src\common\v1_1\imgsensor_hw.c



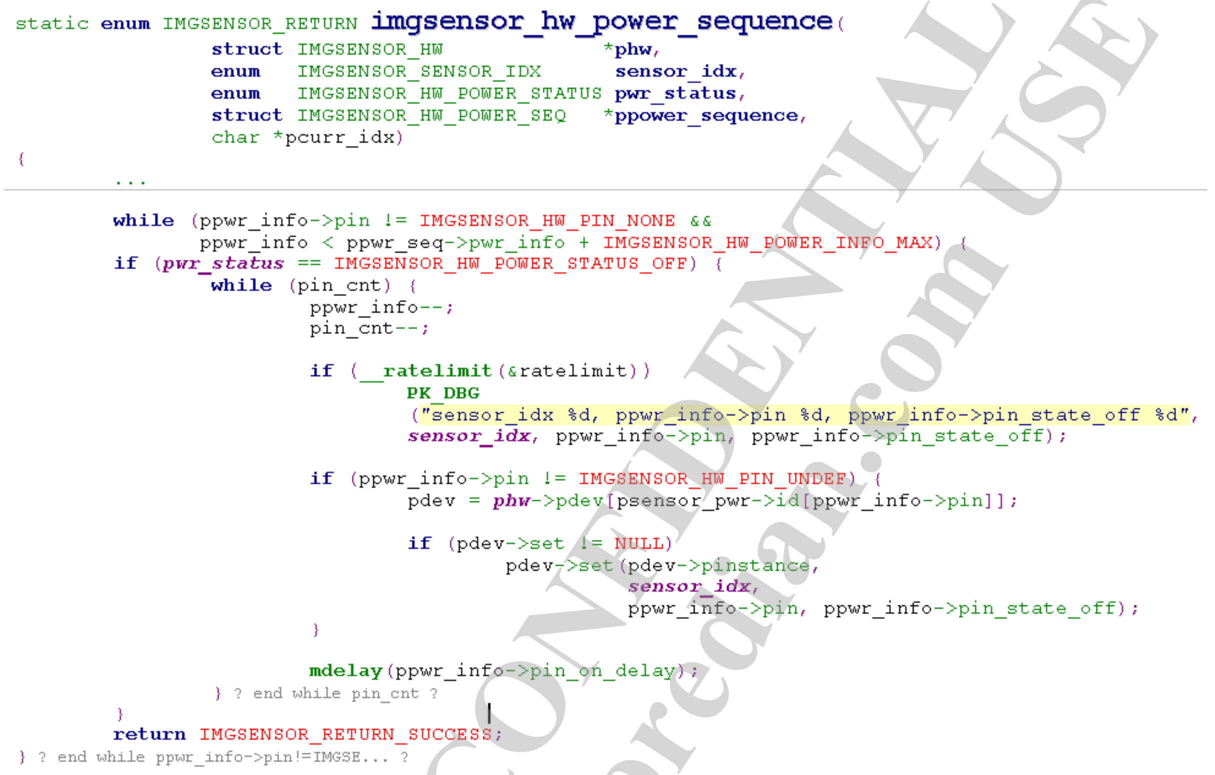
```
imgsensor_hw_power_sequence(
                phw,
                sensor_idx,
                pwr_status,
                platform_power_sequence, imgsensor_sensor_idx_name[sensor_idx]);

imgsensor_hw_power_sequence(
                phw,
                sensor_idx,
                pwr_status, sensor_power_sequence, curr_sensor_name);
```

POWER ON



```
static enum IMGSENSOR_RETURN imgsensor_hw_power_sequence(
            struct IMGSENSOR_HW             *phw,
            enum   IMGSENSOR_SENSOR_IDX      sensor_idx,
            enum   IMGSENSOR_HW_POWER_STATUS pwr_status,
            struct IMGSENSOR_HW_POWER_SEQ   *ppower_sequence,
            char *pcurr_idx)
{
        ...

        while (ppwr_info->pin != IMGSENSOR_HW_PIN_NONE &&
               ppwr_info < ppwr_seq->pwr_info + IMGSENSOR_HW_POWER_INFO_MAX) {

                if (pwr_status == IMGSENSOR_HW_POWER_STATUS_ON) {
                        if (ppwr_info->pin != IMGSENSOR_HW_PIN_UNDEF) {
                                pdev = phw->pdev[psensor_pwr->id[ppwr_info->pin]];

                                if (__ratelimit(&ratelimit))
                                        PK_DBG
                                        ("sensor_idx %d, ppwr_info->pin %d, ppwr_info->pin_state_on %d",
                                        sensor_idx, ppwr_info->pin, ppwr_info->pin_state_on);

                                if (pdev->set != NULL)
                                        pdev->set(pdev->pinstance,
                                                  sensor_idx,
                                                  ppwr_info->pin, ppwr_info->pin_state_on);
                        }

                        mdelay(ppwr_info->pin_on_delay);
                }
                ppwr_info++;
                pin_cnt++;
        } ? end while ppwr_info->pin!=IMGSE... ? |
```

POWER OFF



```
static enum IMGSENSOR_RETURN imgsensor_hw_power_sequence(
                struct IMGSENSOR_HW             *phw,
                enum   IMGSENSOR_SENSOR_IDX      sensor_idx,
                enum   IMGSENSOR_HW_POWER_STATUS pwr_status,
                struct IMGSENSOR_HW_POWER_SEQ   *ppower_sequence,
                char *pcurr_idx)
{
        ...

        while (ppwr_info->pin != IMGSENSOR_HW_PIN_NONE &&
               ppwr_info < ppwr_seq->pwr_info + IMGSENSOR_HW_POWER_INFO_MAX) {
        if (pwr_status == IMGSENSOR_HW_POWER_STATUS_OFF) {
                while (pin_cnt) {
                        ppwr_info--;
                        pin_cnt--;

                        if (__ratelimit(&ratelimit))
                                PK_DBG
                                ("sensor_idx %d, ppwr_info->pin %d, ppwr_info->pin_state_off %d",
                                sensor_idx, ppwr_info->pin, ppwr_info->pin_state_off);

                        if (ppwr_info->pin != IMGSENSOR_HW_PIN_UNDEF) {
                                pdev = phw->pdev[psensor_pwr->id[ppwr_info->pin]];

                                if (pdev->set != NULL)
                                        pdev->set(pdev->pinstance,
                                                sensor_idx,
                                                ppwr_info->pin, ppwr_info->pin_state_off);
                        }

                        mdelay(ppwr_info->pin_on_delay);
                } ? end while pin_cnt ?
        }
        return IMGSENSOR_RETURN_SUCCESS;
} ? end while ppwr_info->pin!=IMGSE... ?
```

2.7 Add Sensor Driver

If you have a ready-prepared sensor driver, you can directly put it in the corresponding path. The files you need to add are:

1. kernel driver ($(kernel_version)\drivers\misc\mediatek\imgsensor\src\{platform}\)
2. tuning file(vendor\mediatek\proprietary\custom\{platform}\hal\imgsensor\)
3. ftb(vendor\mediatek\proprietary\custom\{platform}\hal\senindepfeature\)
4. metadata(vendor\mediatek\proprietary\custom\{platform}\hal\imgsensor_metadata)

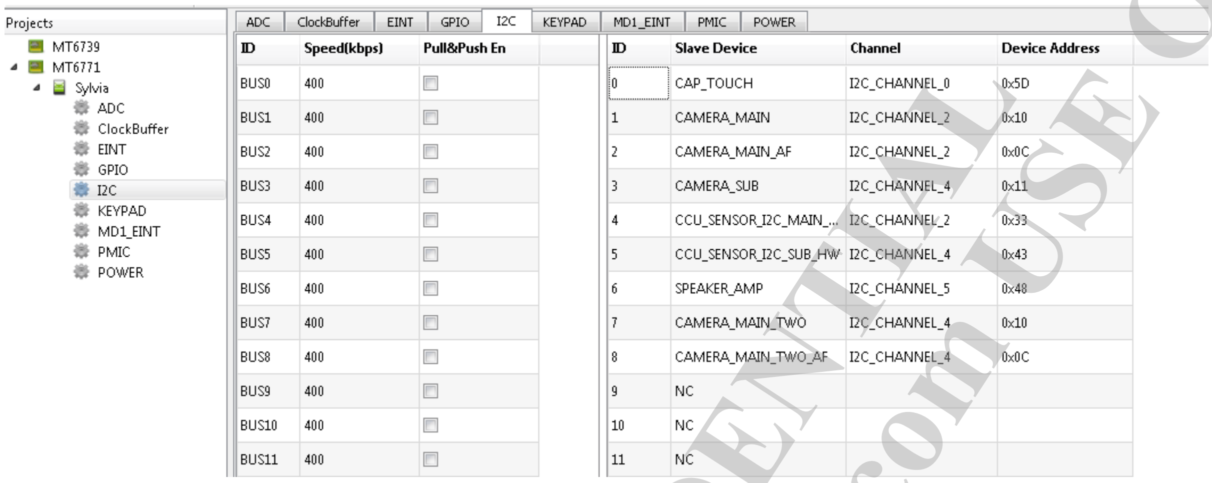
NOTE: The metadata file is not provided in the QVL download code. You can modify the name of the other sensor's metadata file to the current senosr from the metadata directory.

Currently only the facing, orientation, and flashlight configurations in metadata will be used.

2.8 DWS/DTS

Path：$(kernel_version)\drivers\misc\mediatek\dws\mt6771\{project}.dws

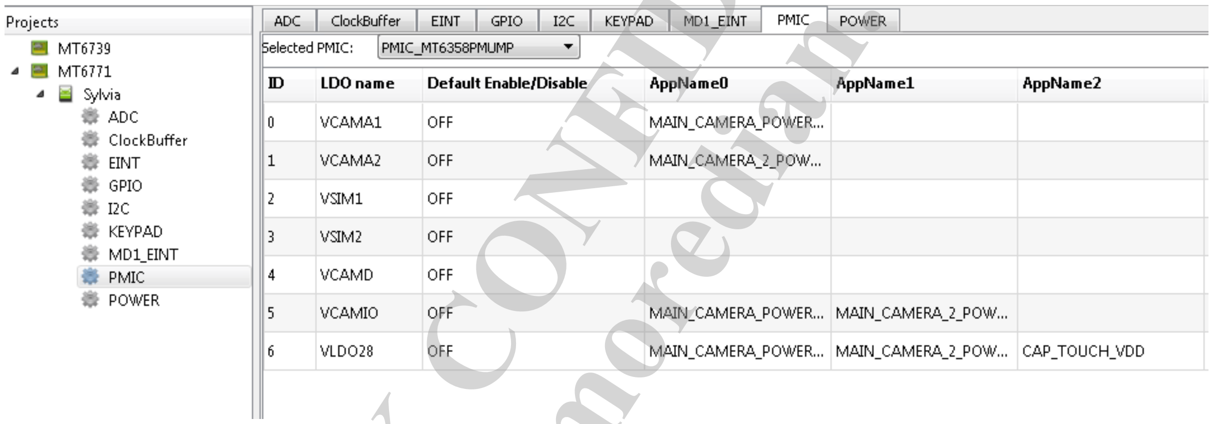
camera I2C config



ID	Speed(kbps)	Pull&Push En
BUS0	400	☐
BUS1	400	☐
BUS2	400	☐
BUS3	400	☐
BUS4	400	☐
BUS5	400	☐
BUS6	400	☐
BUS7	400	☐
BUS8	400	☐
BUS9	400	☐
BUS10	400	☐
BUS11	400	☐

ID	Slave Device	Channel	Device Address
0	CAP_TOUCH	I2C_CHANNEL_0	0x5D
1	CAMERA_MAIN	I2C_CHANNEL_2	0x10
2	CAMERA_MAIN_AF	I2C_CHANNEL_2	0x0C
3	CAMERA_SUB	I2C_CHANNEL_4	0x11
4	CCU_SENSOR_I2C_MAIN_...	I2C_CHANNEL_2	0x33
5	CCU_SENSOR_I2C_SUB_HW	I2C_CHANNEL_4	0x43
6	SPEAKER_AMP	I2C_CHANNEL_5	0x48
7	CAMERA_MAIN_TWO	I2C_CHANNEL_4	0x10
8	CAMERA_MAIN_TWO_AF	I2C_CHANNEL_4	0x0C
9	NC		
10	NC		
11	NC		

camera PMIC power supply

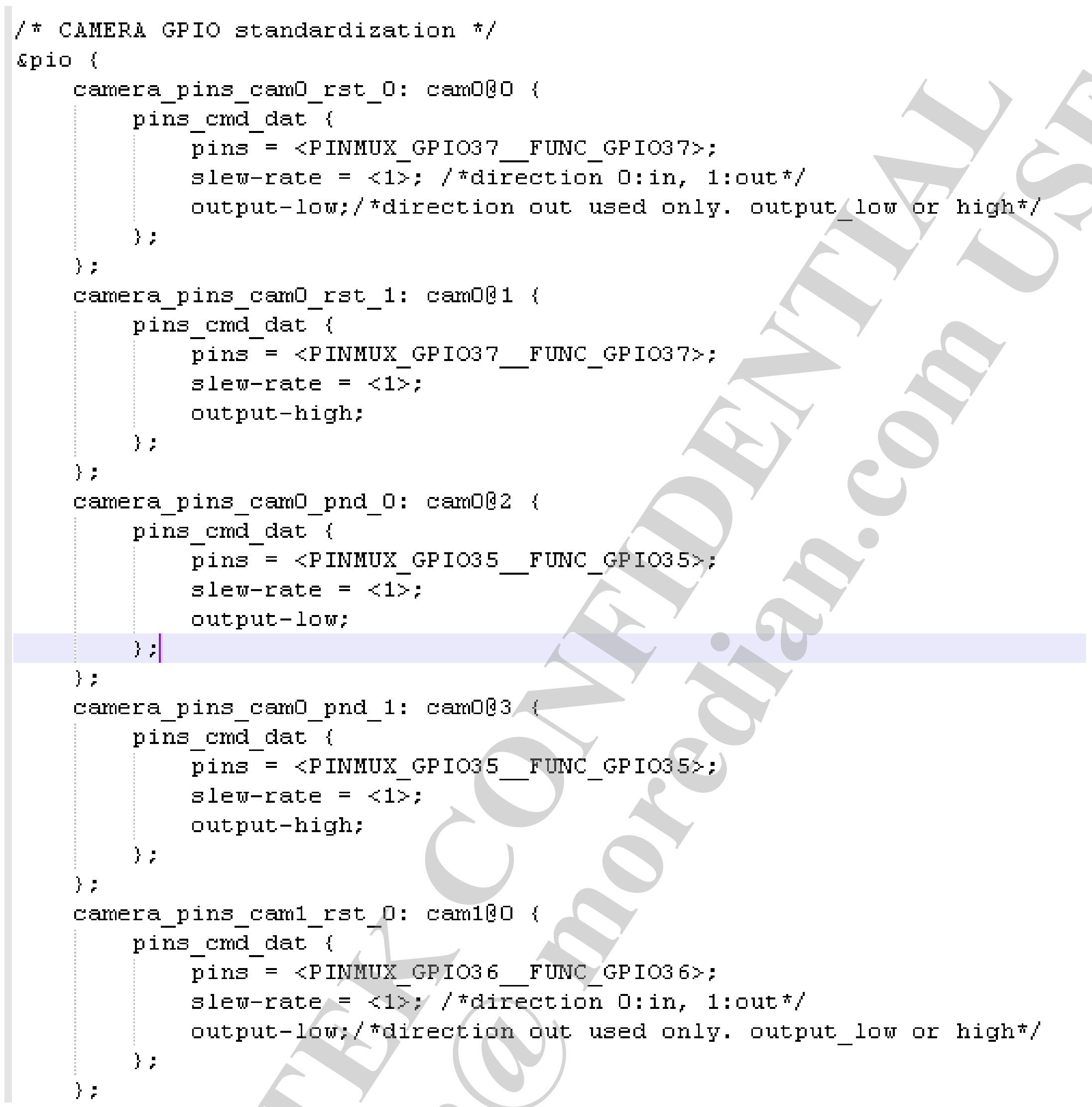


Selected PMIC: PMIC_MT6358PMUMP

ID	LDO name	Default Enable/Disable	AppName0	AppName1	AppName2
0	VCAMA1	OFF	MAIN_CAMERA_POWER...		
1	VCAMA2	OFF	MAIN_CAMERA_2_POW...		
2	VSIM1	OFF			
3	VSIM2	OFF			
4	VCAMD	OFF			
5	VCAMIO	OFF	MAIN_CAMERA_POWER...	MAIN_CAMERA_2_POW...	
6	VLDO28	OFF	MAIN_CAMERA_POWER...	MAIN_CAMERA_2_POW...	CAP_TOUCH_VDD

Path：$(kernel_version)\arch\arm64\boot\dts\mediatek\{project}.dts

Pinctrl configuration



```
/* CAMERA GPIO standardization */
&pio {
    camera_pins_cam0_rst_0: cam0@0 {
        pins_cmd_dat {
            pins = <PINMUX_GPIO37__FUNC_GPIO37>;
            slew-rate = <1>; /*direction 0:in, 1:out*/
            output-low;/*direction out used only. output_low or high*/
        };
    };
    camera_pins_cam0_rst_1: cam0@1 {
        pins_cmd_dat {
            pins = <PINMUX_GPIO37__FUNC_GPIO37>;
            slew-rate = <1>;
            output-high;
        };
    };
    camera_pins_cam0_pnd_0: cam0@2 {
        pins_cmd_dat {
            pins = <PINMUX_GPIO35__FUNC_GPIO35>;
            slew-rate = <1>;
            output-low;
        };
    };
    camera_pins_cam0_pnd_1: cam0@3 {
        pins_cmd_dat {
            pins = <PINMUX_GPIO35__FUNC_GPIO35>;
            slew-rate = <1>;
            output-high;
        };
    };
    camera_pins_cam1_rst_0: cam1@0 {
        pins_cmd_dat {
            pins = <PINMUX_GPIO36__FUNC_GPIO36>;
            slew-rate = <1>; /*direction 0:in, 1:out*/
            output-low;/*direction out used only. output_low or high*/
        };
    };
```

```
&kd_camera_hw1 {
    pinctrl-names = "default",
            "cam0_rst0", "cam0_rst1",
            "cam0_pnd0", "cam0_pnd1",
            "cam1_rst0", "cam1_rst1",
            "cam1_pnd0", "cam1_pnd1",
            "cam2_rst0", "cam2_rst1",
            "cam2_pnd0", "cam2_pnd1",
            "cam_ldo_vcamd_0", "cam_ldo_vcamd_1",
            "cam_ldo_main2_vcamd_0", "cam_ldo_main2_vcamd_1",
            "cam0_mclk_off", "cam0_mclk_on",
            "cam1_mclk_off", "cam1_mclk_on",
            "cam2_mclk_off", "cam2_mclk_on";
    pinctrl-0 = <&camera_pins_default>;
    pinctrl-1 = <&camera_pins_cam0_rst_0>;
    pinctrl-2 = <&camera_pins_cam0_rst_1>;
    pinctrl-3 = <&camera_pins_cam0_pnd_0>;
    pinctrl-4 = <&camera_pins_cam0_pnd_1>;
    pinctrl-5 = <&camera_pins_cam1_rst_0>;
    pinctrl-6 = <&camera_pins_cam1_rst_1>;
    pinctrl-7 = <&camera_pins_cam1_pnd_0>;
    pinctrl-8 = <&camera_pins_cam1_pnd_1>;
    pinctrl-9 = <&camera_pins_cam2_rst_0>;
    pinctrl-10 = <&camera_pins_cam2_rst_1>;
    pinctrl-11 = <&camera_pins_cam2_pnd_0>;
    pinctrl-12 = <&camera_pins_cam2_pnd_1>;
    pinctrl-13 = <&camera_pins_cam0_vcamd_0>;
    pinctrl-14 = <&camera_pins_cam0_vcamd_1>;
    pinctrl-15 = <&camera_pins_cam2_vcamd_0>;
    pinctrl-16 = <&camera_pins_cam2_vcamd_1>;
    pinctrl-17 = <&camera_pins_cam0_mclk_off>;
    pinctrl-18 = <&camera_pins_cam0_mclk_on>;
    pinctrl-19 = <&camera_pins_cam1_mclk_off>;
    pinctrl-20 = <&camera_pins_cam1_mclk_on>;
    pinctrl-21 = <&camera_pins_cam2_mclk_off>;
    pinctrl-22 = <&camera_pins_cam2_mclk_on>;
    status = "okay";
};
```

2.9 MOL

- http://online.mediatek.inc/Pages/eCourse.aspx?Tags=camera+driver

2.10 Feature Table

Related size determination

(1) The maximum value of picture-size-values (max(width*height)), the aspect ratio is as close as possible to the sensor resolution, and the width and height need 16 align

Note: In picture, if the resolution of the current sensor is greater than 1080P, be sure to include the size in picture size. ("1920x1080").

(2) The maximum value in preview-size-values and the maximum value in picture-size-values should be consistent or controlled within 0.01. It is recommended that the maximum preview-size should not exceed the resolution of the screen.

Note: ratio is also the aspect ratio of the scale.
(3) Video-size-values should include the following resolution as much as possible:
QCIF 176X144, QVGA 320X240 CIF 352X288,
480p 720x480, 720p 1280x720, 1080p 1920x1088
Specifically check /system/etc/permissions/media_profile.xml
If the above file does not exist, please find the answer at framworks/av/media/libmedia/MediaProfiles.cpp
And the preview-size-values must contain the values in video-size-values.

AF function determination

device/mediatek/<project>/android.hardware.camera.xml
if the platform has no lens, no auto focus, please delete it :
<feature name="android.hardware.camera.autofocus" />
KEY_FOCUS_MODE default set it as FOCUS_MODE_FIXED, Values set it as FOCUS_MODE_FIXED

Flashlight function determination

If the platform does not support flash, please delete:
<feature name="android.hardware.camera.flash" />
And configure KEY_FLASH_MODE as a null string in the feature table"
note: flashlight feature table (hal/../../../hal/flashlight/config.flashlight***)

Other function determination

(1) KEY_ANTIBANDING must have ANTIBANDING_AUTO mode
(2) Front camera does not support continuous shooting. Capture mode is removed.
(3) Fps range:
If the Sensor can support it, please modify the range in the feature table as follows:FTABLE_CONFIG_AS_TYPE_OF_USER(KEY_AS_(MtkCameraParameters::KEY_PREVIEW_FPS_RANGE),

```
 SCENE_AS_DEFAULT_SCENE(                    ITEM_AS_DEFAULT_( “5000,30000"),
ITEM_AS_USER_LIST_(
              "(15000,15000)",
              "(20000,20000)",
              "(5000,30000)",
              "(30000,30000)",
```

2.11 FAQ required to see

FAQ12869 KK to L, sensor driver modify

The FAQ has instructions to convert to the 64 bit chip sensor driver that needs to be modified.

FAQ18079 Analysis of common black screen problems

Change direction in FAQ14558 metadata

FAQ19451 pass1 deque fail

FAQ17668 camera feature table fps-range configuration