Camera Setting SOP

Overview

- 1. HW spec & Reference design
  - mipi port
  - Mclk
  - add new sensor
- 2. DWS/DTS
- 3. MOL
- 4. 修正feature table(请务必执行，会减少大量CTS bug)
- 5. 必看FAQ

Mipi Port Customization

- 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)。后面介绍 的文件，都涉及到路径优先级问题，不再特别强调

```
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

Mclk customization(1/2)

- 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

Mclk customization(2/2)

- Mclk On/Off control

  -- Customization mclk on/off control in power on sequence

  (kernel-4.4\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},
  },
 },
```

Camera Driver File Path

- Kernel driver()
  - kernel-4.4\drivers\misc\mediatek\imgsensor\src\{platform}
  - kernel-4.4\drivers\misc\mediatek\imgsensor\inc\{platform}
  - hal driver()
  - vendor\mediatek\proprietary\custom\{project}\hal

How to add a new sensor

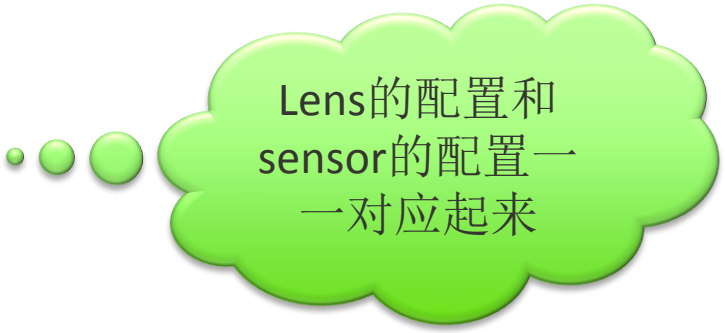
- Step1 device\mediateksample\$project$\ProjectConfig.mk
- 修改imgsensor相关
- eg:main(后摄) imx135_mipi_raw, sub (前摄)ov5648_mipi_raw)
  - CUSTOM_HAL_IMGSENSOR = **imx135_mipi_raw ov5648_mipi_raw**
  - CUSTOM_KERNEL_IMGSENSOR = **imx135_mipi_raw ov5648_mipi_raw**
  - CUSTOM_HAL_MAIN_IMGSENSOR = **imx135_mipi_raw**
  - CUSTOM_HAL_SUB_IMGSENSOR = **ov5648_mipi_raw**
  - CUSTOM_KERNEL_MAIN_IMGSENSOR = **imx135_mipi_raw**
  - CUSTOM_KERNEL_SUB_IMGSENSOR = **ov5648_mipi_raw**

要 build进来的所有的sensor都填上,两颗 sensor name 之间用空格间隔,兼容的sensor继续在后面填

How to add a new sensor

- 修改lens相关

- 没有AF ,设置dummy_lens; YUV sensor带af设置为sensordrive; RAW sensor设置为相应型号(eg:fm50af ,ov8825af)
- imx135_mipi_raw搭AF, sub没有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**



How to add a new sensor

- 修改flashlight相关

- 支持Flashlight设置为constant_flashlight, 不支持设置为dummy_flashlight
  - CUSTOM_HAL_FLASHLIGHT = **constant_flashlight**
  - CUSTOM_KERNEL_FLASHLIGHT = **constant_flashlight**

How to add a new sensor

- Step2
  - \device\mediatek\common\kernel-headers\kd_imgsensor.h
  - **\kernel-4.4\drivers\misc\mediatek\imgsensor\inc\kd_imgsensor.h**

    - #define OV5648_SENSOR_ID 0x5648
      - **Config sensor ID**

    - #define SENSOR_DRVNAME_OV5648_MIPI_RAW “ov5648mipiraw“
      - Define sensor device driver name

How to add a new sensor

- Step3

-- \kernel-4.4\drivers\misc\mediatek\imgsensor\src\common\v1_1\imgsensor_sensor_list.h

- UINT32 **OV5648_MIPI_RAW_SensorInit**(PSENSOR_FUNCTION_STRUCT *pfFunc);

-- \kernel-4.4\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

与ov5648mipi_sensor.c中的函数名对应

How to add a new sensor

- sensorlist.cpp 裡的 SensorList[] 與 imgsensor_sensor_list.c 的 kdSensorList[]裡面的順序必須相同,否則user space & kernel space的id等等就會對不起來
- 且建议按照resolution从大到小的顺序依次排列下来

```
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,EEPROMGetCal
#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
```

```
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
```

How to add a new sensor

- Step4
- Power On/Off

\kernel-4.4\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(1/4)

```
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... ?
```

How to add a new sensor

- Step5 Add Sensor driver
  - 如果已有现成的sensor driver, 可以直接放入相应路径下，需要添加的文件有

(1)kernel driver (kernel-4.4\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)

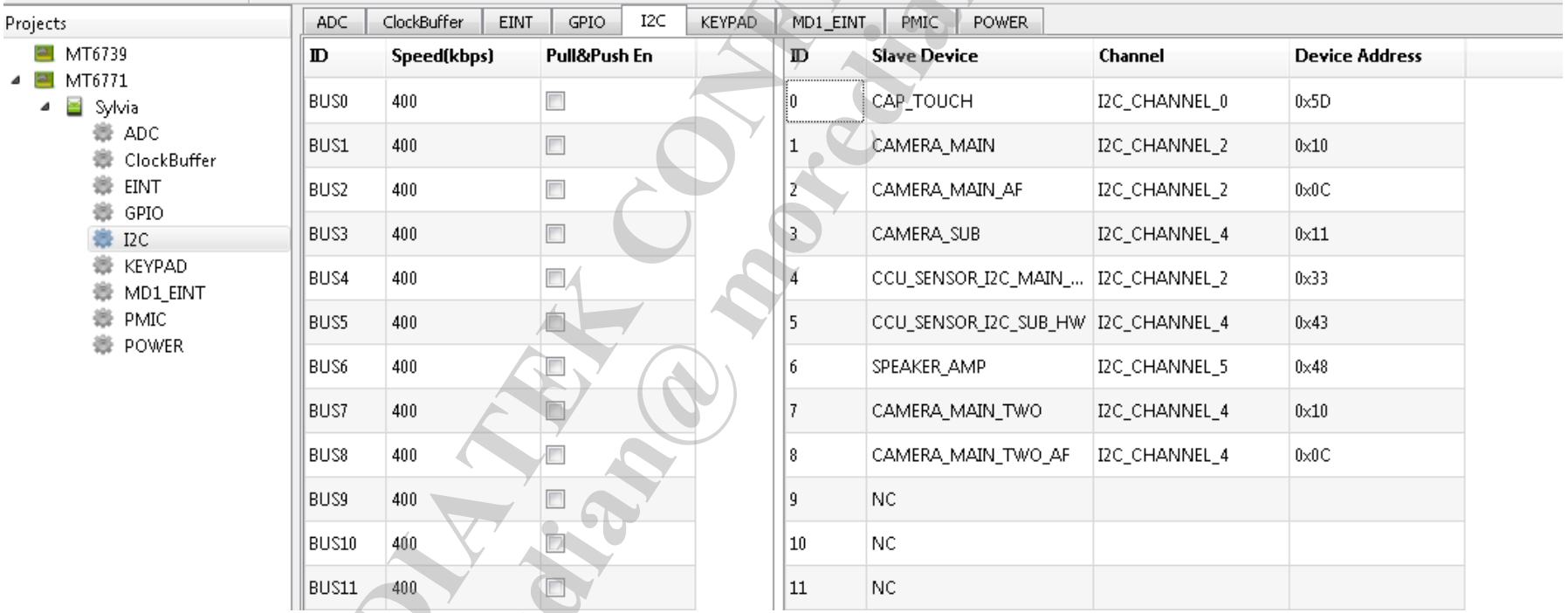
备注: 一般QVL 下载的code 中并没有提供metadata 文件，可自行从metadata 目录下把其他sensor的metadata 文件名字修改成当前的senosr即可使用。

目前metadata 中只有facing，orientation以及flashlight 的配置会被使用到。

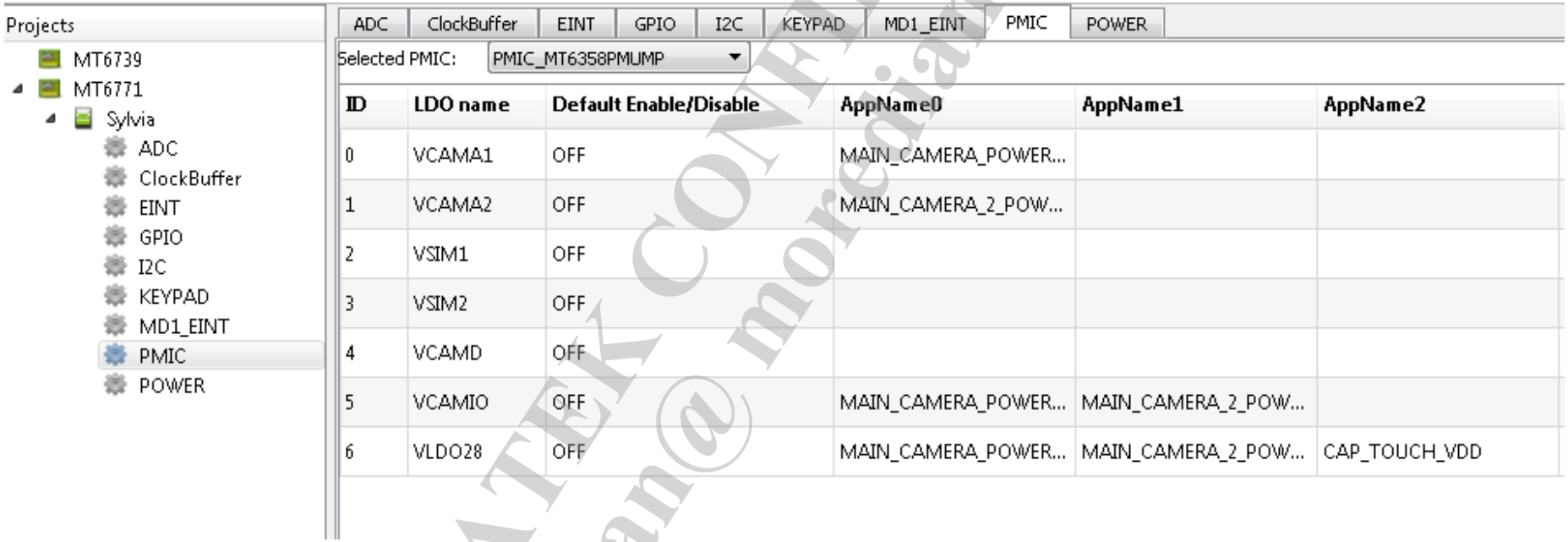
DWS/DTS

Path： kernel-4.4\drivers\misc\mediatek\dws\mt6771\{project}.dws

camera I2C config



camera PMIC power supply



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-4.4\arch\arm64\boot\dts\mediatek\{project}.dts

```
/* 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*/
        };
    };
```

Pinctrl配置

```
&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";
};
```

Pinctrl配置

MOL

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

Feature table

1、相关size确定。

(1) picture-size-values中最大值(max(width*height)), aspect ratio尽量和sensor resolution一致, 且width, height都需要16 align

注: 在picture 中，如果当前sensor的resolution大于1080P，请务必在picture size 中加入该size。(“1920x1080”).

(1) preview-size-values中最大值和picture-size-values中最大值aspect ratio应该一致或者控制在0.01内, 建议preview-size最大值一般不要超过屏幕的resolution

注: ratio 也就是比例的宽高比。

Feature table

(3) video-size-values尽量包含下面的resolution:
QCIF 176X144, QVGA 320X240 CIF 352X288,
480p 720x480, 720p 1280x720, 1080p 1920x1088
具体先check /system/etc/permissions/media_profile.xml
若上面的文件不存在, 请在framworks/av/media/libmedia/MediaProfiles.cpp寻找答案
并且preview-size-values务必要包含video-size-values中的值

Feature table

2、AF功能确定

device/mediatek/<project>/android.hardware.camera.xml
若平台没有lens, 不支持auto focus, 请删除：
<feature name="android.hardware.camera.autofocus" />
且KEY_FOCUS_MODE default 设为 FOCUS_MODE_FIXED, Values也仅支持 FOCUS_MODE_FIXED

3、flashlight功能确定

若平台不支持flash, 请删除:
<feature name="android.hardware.camera.flash" />
并且在feature table中把KEY_FLASH_MODE配置为null 字符串”“

备注：flashlight feature table (hal/../../../hal/flashlight/config.flashlight***)

Feature table

4、其他功能确定

(1) KEY_ANTIBANDING一定要有ANTIBANDING_AUTO模式
(2) Front camera 不支持连拍。Capture mode去除。
(3) Fps range:
Sensor 可以支持的话，请把feature table 中的range 修改如下：

```
FTABLE_CONFIG_AS_TYPE_OF_USER(      KEY_AS_(MtkCameraParameters::KEY_PREVIE
    W_FPS_RANGE),
    SCENE_AS_DEFAULT_SCENE(         ITEM_AS_DEFAULT_(“5000,30000"),
    ITEM_AS_USER_LIST_(
            "(15000,15000)",
            "(20000,20000)",
            "(5000,30000)",
            "(30000,30000)",
```

FAQ required to see

FAQ12869 KK转L，sensor driver modify
该FAQ 有指导转化到64 bit chip sensor driver 需要修改的

FAQ18079 常见黑屏问题分析

FAQ14558 metadata 中修改方向

FAQ19451 pass1 deque fail

FAQ17668 camera feature table的fps-range的配置