CONFIDENTIAL B

Sensors Bringup SOP for Android O

MEDIATEK CONFIDENTIAL
mtk@moredian.com USE

Outline

- keywords
- sensor types
- Mediatek Sensor system Architecture
- platform instruction
- sensor porting guide
  - configs
  - sensor specific drivers
  - sensor dts node
  - sensor dws file

MEDIATEK CONFIDENTIAL
moredan@moredan.com USE

Key words

- <proj>
  - project name, e.g. tb8183m1_64_bsp
- <platform>
  - platform, e.g. mt8183
- <kernel_ver>
  - linux kernel version , e.g. kernel-4.4
- <arm_ver>:
  - arm or arm64
- <xxx_sensor>:
  - sensor driver name， e.g. BMA222E_NEW, S62X

MEDIATEK CONFIDENTIAL
moredian@moredian.com USE

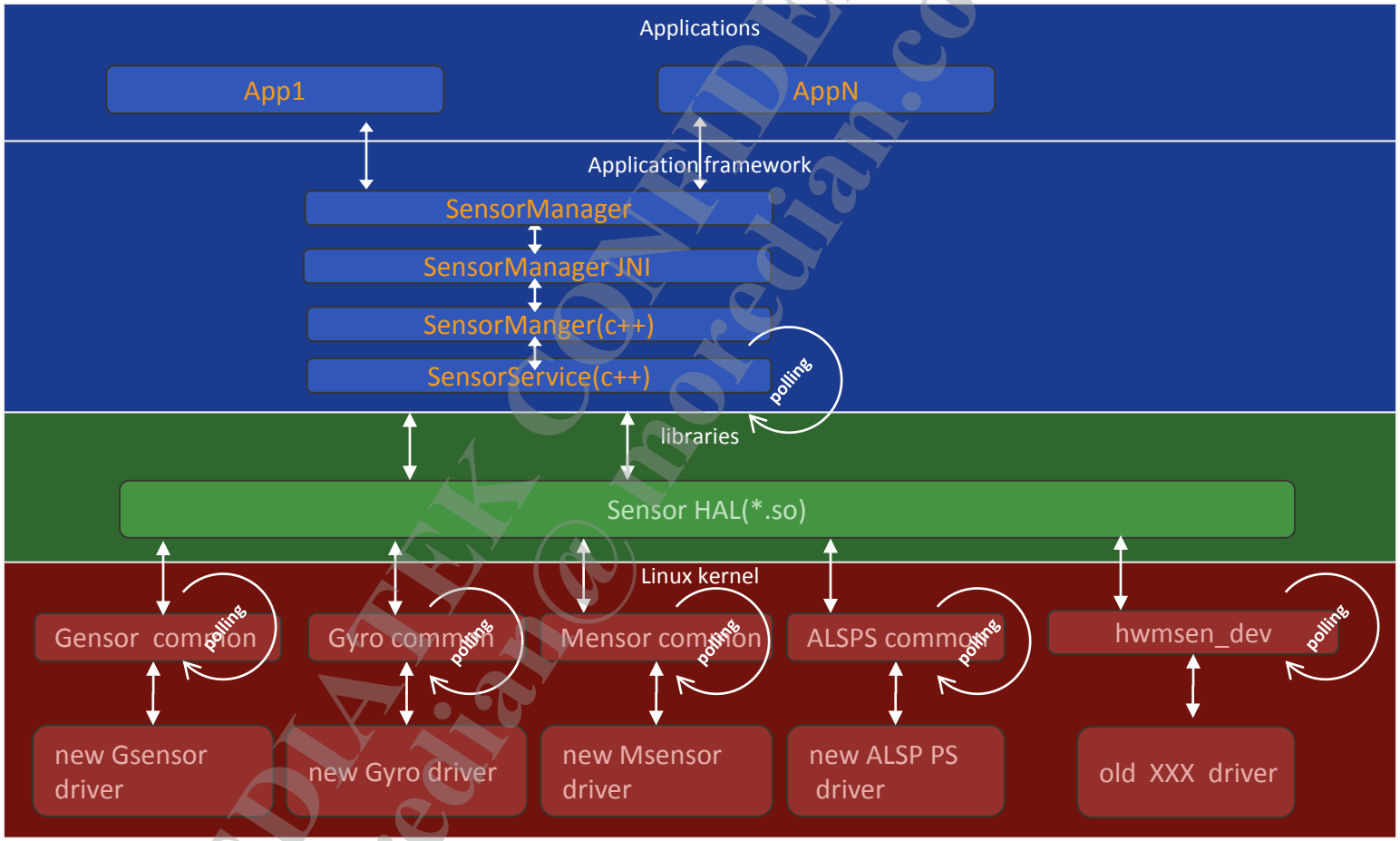
sensor types

- please refer to google for other sensors.

Sensor types	Service define	Driver define
Accelerometer	TYPE_ACCELEROMETER	SENSOR_TYPE_ACCELEROMETER
Magnetic Field	TYPE_MAGNETIC_FIELD	SENSOR_TYPE_MAGNETIC_FIELD
Orientation	TYPE_ORIENTATION	SENSOR_TYPE_ORIENTATION
Gyroscope	TYPE_GYROSCOPE	SENSOR_TYPE_GYROSCOPE
Light	TYPE_LIGHT	SENSOR_TYPE_LIGHT
Pressure	TYPE_PRESSURE	SENSOR_TYPE_PRESSURE
Temperature	TYPE_TEMPERATURE	SENSOR_TYPE_TEMPERATURE
Proximity	TYPE_PROXIMITY	SENSOR_TYPE_PROXIMITY

MEDIATEK CONFIDENTIAL B
moredian@moredian.com USE

Sensor system Architecture



MEDIATEK CONFIDENTIAL
moredian@moredian.com

Platform Instruction

MEDIATEK CONFIDENTIAL
moredian@ moredian.com USE

Platform instruction

- GPIO /EINT/POWER
  - After got the HW information, you should apply your hardware interface to device tree(".dts"), such as EINT/RST/Power … etc.
- I2C
  - Mediatek platform I2C support：
    - FIFO mode： read/write 8 Bytes one time
    - DMA mode： only read/write: 65532Byte;
      write and read: write 255 Byte, read 31Byte

NOTE: For DMA 255x255Byte: The low 8-bit is "trans_len".
The high 8-bit is "trans_num"

```
trans_len = (msg->len) & 0xFF;
trans_num = (msg->len >> 8) & 0xFF;
```

MEDIATEK CONFIDENTIAL
moredian@moredian.com USE

I2C interface-Read with FIFO mode

- read with FIFO mode, max 8 bytes one time
  - write and read mode: I2C_WR_FLAG
  - without stop condition after register address write: I2C_RS_FLAG

Multiple Read


```
static s32 i2c_read_nondma(struct i2c_client *client,
    u8 addr, u8 *rxbuf, int len)
{
    int ret;
    struct i2c_msg msg;

    memset(&msg, 0, sizeof(struct i2c_msg));
    rxbuf[0] = addr;
    msg.addr = client->addr & I2C_MASK_FLAG;
    msg.flags = 0;
    msg.len = ((len & 0x1f) << 8) | 1;
    msg.buf = rxbuf;
    msg.ext_flag = I2C_WR_FLAG | I2C_RS_FLAG;
    msg.timing = GSLTP_I2C_MASTER_CLOCK;
    ret = i2c_transfer(client->adapter, &msg, 1);
    return ret;
}
```

if stop condition need after register addr, please remove I2C_RS_FLAG

MEDIATEK CONFIDENTIAL B
MEDIA TEK CONFIDENTIAL
moredian@moredian.com

I2C interface—Write with FIFO mode

- write with FIFO mode
  - max write 8 bytes one time

```
static s32 i2c_write_nondma(struct i2c_client *client, u8 addr, u8 *txbuf, int len)
{
    int ret;
    int retry = 0;
    struct i2c_msg msg;
    u8 wrBuf[RPR_FIFO_MAX_WR_SIZE + 1];

    if ((txbuf == NULL) && len > 0)
        return -1;

    memset(&msg, 0, sizeof(struct i2c_msg));
    memset(wrBuf, 0, RPR_FIFO_MAX_WR_SIZE + 1);
    wrBuf[0] = addr;
    if (txbuf)
        memcpy(wrBuf + 1, txbuf, len);

    msg.flags = 0;
    msg.buf = wrBuf;
    msg.len = 1 + len;
    msg.addr = (client->addr & I2C_MASK_FLAG);
    msg.ext_flag = (client->ext_flag | I2C_ENEXT_FLAG);
    msg.timing = RPR_I2C_MASTER_CLOCK;

    for (retry = 0; retry < 5; ++retry) {
        ret = i2c_transfer(client->adapter, &msg, 1);
        if (ret < 0)
            continue;
        return 0;
    }
    RPR0521_ERR("Dma I2C Write Error: 0x%04X, %d bytes, err-code: %d\n", addr, len, ret);
    return ret;
} ? end i2c_write_nondma ?
```

```
#define RPR_FIFO_MAX_RD_SIZE C_I2C_FIFO_SIZE
#define RPR_FIFO_MAX_WR_SIZE C_I2C_FIFO_SIZE - RPR_REG_ADDR_LEN
```

Parameter of i2c_write_nond

MEDIATEK CONFIDENTIAL
moredian@moredian.com

I2C interface—read with DMA WRRD mode

- read with DMA mode: I2C_DMA_FLAG
  - with write and read mode: I2C_WR_FLAG, max 31bytes
  - without stop condition after register address write: I2C_RS_FLAG

```
static s32 i2c_dma_read(struct i2c_client *client, u8 addr, u8
{
    int ret;
    struct i2c_msg msg;

    memset(&msg, 0, sizeof(struct i2c_msg));
    *g_dma_buff_va = addr;
    msg.addr = client->addr & I2C_MASK_FLAG;
    msg.flags = 0;
    msg.len = ((len & 0x1f) << 8) | 1;
    msg.buf = g_dma_buff_pa;
    msg.ext_flag = I2C_WR_FLAG | I2C_RS_FLAG | I2C_DMA_FLAG;
    msg.timing = GSLTP_I2C_MASTER_CLOCK;

    ret = i2c_transfer(client->adapter, &msg, 1);
    memcpy(rxbuf, g_dma_buff_va, len);
    return ret;
}
```

MEDIATEK CONFIDENTIAL
moreqian@moredian.com USE

I2C interface—read only with DMA mode

- read only with DMA mode: have I2C_DMA_FLAG but no I2C_WR_FLAG
- there will be a stop condition between msg[0] and msg[1]
- max read length is 65532 bytes

```
static s32 i2c_dma_non_wrrd_read(struct i2c_client *client, u
{
    int ret;
    struct i2c_msg msg[2];

    memset(&msg, 0, 2 * sizeof(struct i2c_msg));
    msg[0].addr = client->addr & I2C_MASK_FLAG;
    msg[0].flags = 0;
    msg[0].len = GSLTP_REG_ADDR_LEN;
    msg[0].buf = &addr;
    msg[0].ext_flag = I2C_DMA_FLAG;
    msg[0].timing = GSLTP_I2C_MASTER_CLOCK;

    msg[1].addr = client->addr & I2C_MASK_FLAG;
    msg[1].flags = 0;
    msg[1].len = len;
    msg[1].buf = g_dma_buff_pa;
    msg[1].ext_flag = I2C_DMA_FLAG;
    msg[1].timing = GSLTP_I2C_MASTER_CLOCK;

    ret = i2c_transfer(client->adapter, &msg, 2);
    memcpy(rxbuf, g_dma_buff_va, len);
    return ret;
} ? end i2c_dma_non_wrrd_read ?
```

MEDIATEK CONFIDENTIAL
moredian@moredian.com

I2C interface—Write with DMA mode

- write with DMA mode, max 65532 bytes

```
static s32 i2c_dma_write(struct i2c_client *client, u8 addr, u8
{
    int ret;
    struct i2c_msg msg;

    memset(&msg, 0, sizeof(struct i2c_msg));
    *g_dma_buff_va = addr;
    msg.addr = (client->addr & I2C_MASK_FLAG);
    msg.flags = 0;
    msg.buf = g_dma_buff_pa;
    msg.len = 1 + len;
    msg.ext_flag = (client->ext_flag | I2C_ENEXT_FLAG \
            | I2C_DMA_FLAG);
    msg.timing = GSLTP_I2C_MASTER_CLOCK;

    memcpy(g_dma_buff_va + 1, txbuf, len);
    ret = i2c_transfer(client->adapter, &msg, 1);
    return ret;
}
```

MEDIATEK CONFIDENTIAL B
moredian@moredian.com

Sensor Proting Guide

MEDIATEK CONFIDENTIAL
moredian@ moredian.com USE

Sensor porting guide

- 1. add or modify sensor information in projectconfig.mk
- 2. add or modify sensor info in kernel config
- 3. add or modify sensor driver in kernel driver
- 4. add or modify sensor customized info in dts file and dws file

MEDIATEK CONFIDENTIAL
moredian@moredian.com

Project Config

- /device/mediatek/<proj>/ProjectConfig.mk
  - MTK_SENSOR_SUPPORT = yes
  - MTK_SENSORS_1_0 = yes
  - CUSTOM_KERNEL_MAGNETOMETER = yes
  - CUSTOM_KERNEL_ACCELEROMETER = yes
  - CUSTOM_KERNEL_ALSPS = no
  - CUSTOM_KERNEL_GYROSCOPE = yes
  - CUSTOM_HAL_SENSORS = sensor
- for the sensor supported, say yes.
- say no or not set for the not supported sensors

MEDIATEK CONFIDENTIAL
mordaian@mordaian.com USE

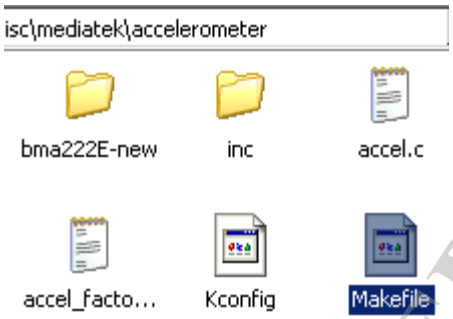
Kernel config(1/2)

- path: <kernel_ver>/arch/<arm_ver>/configs/<proj>_debug_defconfig and <proj>_defconfig
- configuration items, e.g. Gsensor, Msensor, alsps…:
  - CONFIG_MTK_SENSOR_SUPPORT=y
  - CONFIG_MTK_SENSORS_1_0=y
  - CONFIG_CUSTOM_KERNEL_ACCELEROMETER=y
  - CONFIG_MTK_<g_sensor>=y
    - <g_sensor> means your Gsensor, e.g. BMA222E_NEW
  - CONFIG_CUSTOM_KERNEL_GYROSCOPE=y
  - CONFIG_MTK_<gyro_sensor>=y
  - CONFIG_CUSTOM_KERNEL_ALSPS=y
  - CONFIG_MTK_<alsps_sensor>=y
    - e.g. CONFIG_MTK_EPL2182_NEW=y
  - CONFIG_CUSTOM_KERNEL_MAGNETOMETER=y
    - CONFIG_MTK_<m_sensor>=y
- if the sensor is not supported, please set as below or remove the items
  - # CONFIG_CUSTOM_KERNEL_GYROSCOPE= is not set
  - # CONFIG_MTK_<gyro_sensor>= is not set

MEDIATEK CONFIDENTIAL
moredian.com USE

Kernel config config(2/2)

- there is no early suspend, please remove CONFIG_HAS_EARLYSUSPEND
- the name of <g_sensor>, <m_sensor>… should match the information in makefile of the relative sensor driver
- take gsensor as an example
  - kernel config is: CONFIG_MTK_BMA222E_NEW=y
  - make file informaiton :
    - path: <kernel_ver>/drivers/misc/mediatek/sensors-1.0/accelerometer/makefile




```
# In case the platform does NOT support this type of sensors
ccflags-y += -I$(srctree)/drivers/misc/mediatek/hwmon/include

obj-y   +=  accel.o accel_factory.o

obj-$(CONFIG_MTK_K2DH)    +=  k2dh/
obj-$(CONFIG_MTK_BMA050)    +=  bma050/
obj-$(CONFIG_MTK_BMA2XX)    +=  bma2xx/
obj-$(CONFIG_MTK_BMA050_NEW)    +=  bma050-new/
obj-$(CONFIG_MTK_BMA222E)    +=  bma222E/
obj-$(CONFIG_MTK_BMA222E_NEW)    +=  bma222E-new/
```

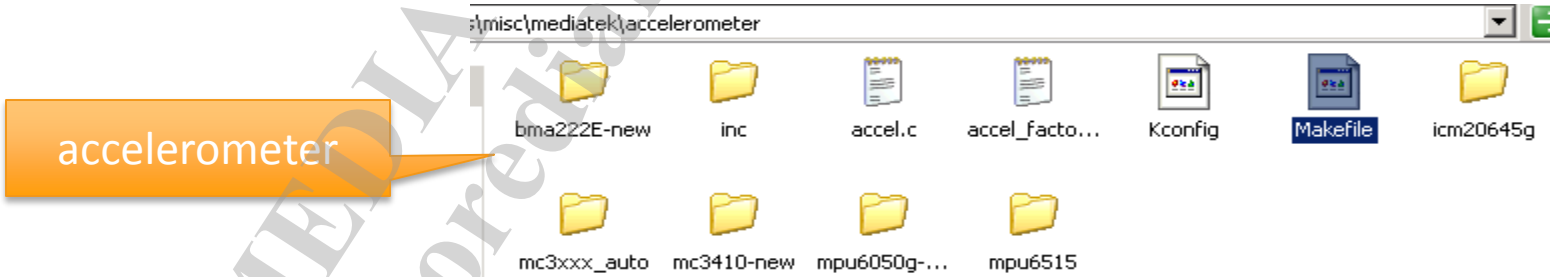
kernel config item

Driver folder name

MEDIATEK CONFIDENTIAL B
mordian@mordian.com

Sensor drivers

- If the sensor is supported in MTK platform, the driver location is
  - <kernel_ver>/drivers/misc/mediatek/sensors-1.0/accelerometer/<g_sensor>
  - <kernel_ver>/drivers/misc/mediatek/sensors-1.0/alsps/<alsps_sensor>
  - <kernel_ver>/drivers/misc/mediatek/sensors-1.0/gyroscope/<gyro_sensor>
  - <kernel_ver>/drivers/misc/mediatek/sensors-1.0/magnetometer/<m_sensor>
- Note:
  - 1). if there was no driver code in the above path, but the driver is in MTK QVL, please ask MTK to release it.
  - 2). we'll take Gsensor as an example, other sensors are just similar with it



MEDIATEK CONFIDENTIAL B
moredian@moredian.com USE

G-sensor drivers add(1/3)

- step1: add driver source code
  - e.g. add bma222E-new to accelerometer

- step2: check and modify Kconfig and Makefile
  - 1). makefile in accelerometer folder

```
# In case the platform does NOT support this type of sensors
ccflags-y += -I$(srctree)/drivers/misc/mediatek/hwmon/include

obj-y   +=  accel.o accel_factory.o

obj-$(CONFIG_MTK_K2DH)    +=  k2dh/
obj-$(CONFIG_MTK_BMA050)    +=  bma050/
obj-$(CONFIG_MTK_BMA2XX)    +=  bma2xx/
obj-$(CONFIG_MTK_BMA050_NEW)    +=  bma050-new/
obj-$(CONFIG_MTK_BMA222E)    +=  bma222E/
obj-$(CONFIG_MTK_BMA222E_NEW)    +=  bma222E-new/
```

MEDIATEK CONFIDENTIAL
mobian@inoregan.com

G-sensor drivers add(2/3)

- 2). Kconfig in specific driver folder

```
config MTK_BMA222E_NEW
  bool "MTK_BMA222E_NEW for MediaTek package"
  default n
  help
    It support different accelerometer sensor.
          If this option is set,
    it will support
          MTK_BMA222E_NEW Accelerometer.
```

match with kernel config

```
config BMA222_LOWPASS
  bool "MTK_BMA222E_NEW for Mediatek package"
  default n
  help
  Say Y here if you have MTK_BMA222E_NEW touch panel BMA222_LOWPASS.

  If unsure, say N.

  To compile this dirver as a module, choose M here: the
  module will be called.
```

only config when you need

misc\mediatek\accelerometer\bma222E-new

- bma222E.c
- bma222E.h
- Kconfig
- Makefile

- 3). makefile in specific driver folder

```
ccflags-y += -I$(srctree)/drivers/misc/mediatek/accelerometer/inc
ccflags-y += -I$(srctree)/drivers/misc/mediatek/hwmon/include
ccflags-y += -I$(srctree)/drivers/misc/mediatek/include/mt-plat/
ccflags-y += -I$(srctree)/drivers/misc/mediatek/include/mt-plat/$(MTK_PLATFORM)/include/
ccflags-y += -I$(srctree)/drivers/misc/mediatek/include/mt-plat/$(MTK_PLATFORM)/include/mach/

obj-y :=  bma222E.o
```

MEDIATEK CONFIDENTIAL B
moredian@moredian.com

G-sensor drivers add(3/3)

- step3: specific driver modification(e.g. bma222E.c)
  - 1). use get_accel_dts_func() to get customized information in dts file

```
static int __init bma222_init(void)
{
  GSE_FUN();
  hw = get_accel_dts_func(COMPATIABLE_NAME, hw);
```

  - 2). of_match_table is needed

```
static const struct of_device_id accel_of_match[] = {
  {.compatible = "mediatek,gsensor"},
  {},
};

static struct i2c_driver bma222_i2c_driver = {
  .driver = {
      .name = BMA222_DEV_NAME,
      .of_match_table = accel_of_match,
      },
```

```
static int gsensor_local_init(void)
{
  GSE_FUN();

  BMA222_power(hw, 1);
  if (i2c_add_driver(&bma222_i2c_driver)) {
```

  - 3). use architecture of accel.c for data and control path

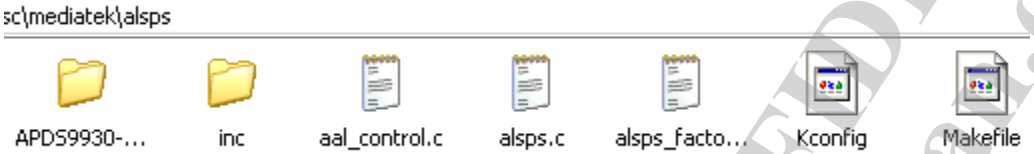
```
static int bma222_i2c_probe(struct i2c_client *client, const struct i2c_device_id
{
  struct i2c_client *new_client;
  struct bma222_i2c_data *obj;
  struct acc_control_path ctl = { 0 };
  struct acc_data_path data = { 0 };
```

don't use hwmsen_object method

MEDIATEK CONFIDENTIAL B
mediatek@meridian.com

other sensor driver code

- alsps sensor



- m-sensor

- gyroscope sensor

MEDIATEK CONFIDENTIAL
moredian@moredian.com

alsps sensor driver code

- alsps sensor data path and control path

```
static int APDS9930_i2c_probe(struct i2c_client *clien
{
  struct APDS9930_priv *obj;
  /*struct hwmsen_object obj_ps, obj_als;*/
  struct als_control_path als_ctl = {0};
  struct als_data_path als_data = {0};
  struct ps_control_path ps_ctl = {0};
  struct ps_data_path ps_data = {0};
```

- please disable IRQ as soon as interrupt happen when use level trigger type

```
static irqreturn_t alsps_interrupt_handler(int irq, void *dev_id)
{
  struct APDS9930_priv *obj = g_APDS9930_ptr;

  if (!obj)
    return IRQ_HANDLED;
  disable_irq_nosync(alsps_irq);
  int_top_time = sched_clock();
  schedule_work(&obj->irq_work);
  return IRQ_HANDLED;
}
```

MEDIATEK CONFIDENTIAL
moredian@moredian.com

Msensor driver code

- msensor data and control path

```
static int s62x_i2c_probe(struct i2c_client *client,
        const struct i2c_device_id *id)
{
  struct i2c_client *new_client;
  struct s62x_i2c_data *data;
  int err = 0;
  struct mag_control_path ctl = {0};
  struct mag_data_path mag_data = {0};

  ctl.is_use_common_factory = false;
  ctl.m_enable = s62x_m_enable;
  ctl.m_set_delay  = s62x_m_set_delay;
  ctl.m_open_report_data = s62x_m_open_report_data;
  ctl.o_enable = s62x_o_enable;
  ctl.o_set_delay  = s62x_o_set_delay;
  ctl.o_open_report_data = s62x_o_open_report_data;
  ctl.is_report_input_direct = false;
  ctl.is_support_batch = data->hw->is_batch_supported;

  err = mag_register_control_path(&ctl);

   mag_data.div_m = CONVERT_M_DIV;
   mag_data.div_o = CONVERT_O_DIV;
   mag_data.get_data_o = s62x_o_get_data;
   mag_data.get_data_m = s62x_m_get_data;

   err = mag_register_data_path(&mag_data);
```

MEDIATEK CONFIDENTIAL
moredian@moredian.com

Msensor

- Please ask vendor to provide daemon for msensor
- FAQ of MTK online system
  - [FAQ06113][sensor]如何编译和启动自己porting的daemon
  - [FAQ04551][sensor] Msensor Daemon该如何检查
  - [FAQ05813]Msensor Daemon的路径
  - [FAQ13345]Android L版本上指南针apk读取不到sensor数据的原因分析

MEDIATEK CONFIDENTIAL
morgian@ morgian.com USE

Sensor Customization

- both modify <proj>.dts file and <proj>.dws file for your project
- File path:
  - dws file: <kernel_ver>/drivers/misc/mediatek/dws/<platfrom>/<proj>.dws
  - dts file: <kernel_ver>/arch/<arm_ver>/boot/dts/<proj>.dts

MEDIATEK CONFIDENTIAL
moredian@moredian.com USE

G-Sensor Customization dts node(1/4)

- Gsensor customized information is set in <proj>.dts

```
cust_accel@0 {
    compatible = "mediatek,bma222e_new";
    i2c_num = <2>;
    i2c_addr = <0x18 0 0 0>;
    direction = <4>;
    power_id = <0xffff>;
    power_vol = <0>;
    firlen = <16>;
    is_batch_supported = <0>;
};
```

Gsensor note name

Compatible info, must same as the one in SW

```
#define COMPATIABLE_NAME "mediatek,bma222e_new"

static int __init bma222_init(void)
{
    GSE_FUN();
    hw = get_accel_dts_func(COMPATIABLE_NAME, hw);
```

sensor_dts.c

```
struct acc_hw *get_accel_dts_func(const char *name, struct acc_hw *hw)
{
    node = of_find_compatible_node(NULL, NULL, name);
    if (node) {
        ret = of_property_read_u32_array(node , "i2c_num", i2c_num, ARRAY_SIZE(i2c_num));
        if (ret == 0)
            hw->i2c_num = i2c_num[0];

        ret = of_property_read_u32_array(node , "i2c_addr", i2c_addr, ARRAY_SIZE(i2c_addr));
```

MEDIATEK CONFIDENTIAL
mordian@mordian.com USE

G-Sensor Customization dts node(2/4)

- i2c number and i2c slave address:

```
i2c_num = <2>;
i2c_addr = <0x18 0 0 0>;
```

  - Customer can define the I2C number used by G-sensor, the value could be defined as 0 ~ 2
  - if driver have auto probe function, please add other slave address here
- gsensor power information

```
power_id = <0xffff>;
power_vol = <0>;
```

- SW low pass filter

```
firlen = <16>;
```

  - Customer can define the filter length of SW low pass filter.
  - The value could be defined as 0 ~ 32. 0 will disable the functionality
- Gsensor direction

```
direction = <4>;
```

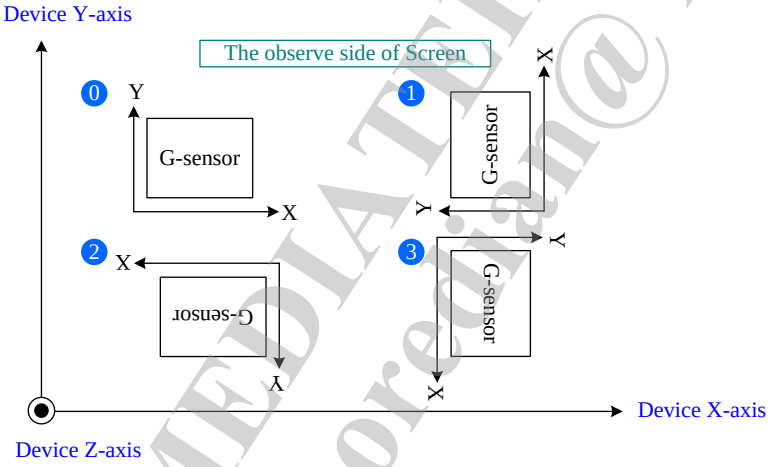
  - important, please modify it to make gsensor x,y,z data right direction. (refer to the next slide)

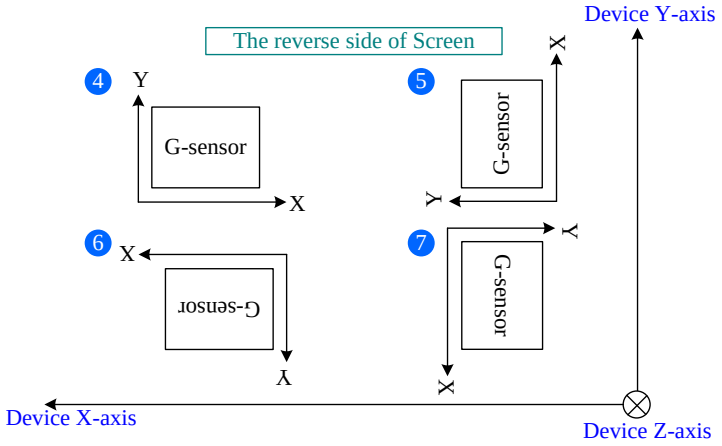
MEDIATEK CONFIDENTIAL
moredian@moredian.com

G-Sensor Customization (3/4)

- direction
  - Customer can define the device direction of g-sensor in device.
  - The value could be defined as 0 ~ 7

Value	Description
0	{x, y, z} => { x, y, z}
1	{x, y, z} => {-y, x, z}
2	{x, y, z} => {-x, -y, z}
3	{x, y, z} => { y, -x, z}
4	{x, y, z} => {-x, y, -z}
5	{x, y, z} => { y, x, -z}
6	{x, y, z} => { x, -y, -z}
7	{x, y, z} => {-y, -x, -z}





MEDIATEK CONFIDENTIAL B
moredian@moredian.com USE

G-Sensor Customization (4/4)

- Eint information
  - if you use EINT for Gsensor, EINT information is needed
  - if there was a dws file in kernel, EINT information will set in dws file, and dws will generate cust.dtsi
  - if there was no dws file in kernel, EINT info will need to set in <platform>.dts file.
- Root node in <platform>.dtsi

```
/* dummy nodes for cust_eint */
gse_1: gse_1 {
  compatible = "mediatek, gse_1-eint";
  status = "disabled";
};
```

- attach node in <proj>.dtsi (if have dws file, it is in cust.dtsi which is generated by dws file, and not need to set in <platform>.dts)

```
&gse_1 {
  interrupt-parent = <&eintc>;
  interrupts = <66 IRQ_TYPE_LEVEL_LOW>;
  debounce = <66 0>;
  status = "okay";
};
```

> EINT pin info, trigger type such as IRQ_TYPE_EDGE_FALLING should be the one say in next slide

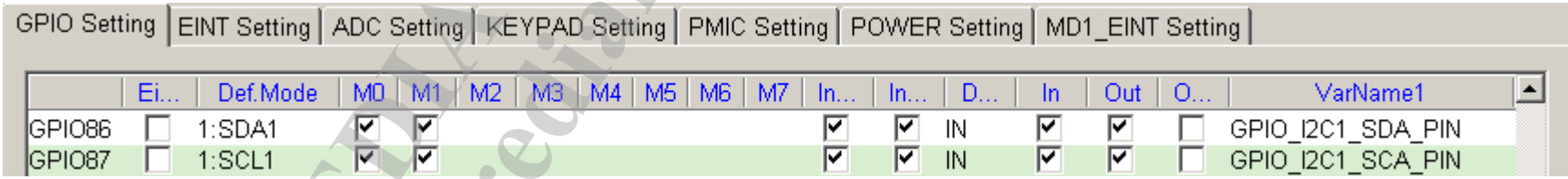
IRQ flags

- interrupt trigger type (e.g. IRQ_TYPE_EDGE_FALLING)should be the ones that defined int the following files
  - <kernel_ver>/include/dt-bindings/interrupt-controller/arm-gic.h
  - <kernel_ver>/include/dt-bindings/interrupt-controller/irq.h
- IRQ flags:
  - #define IRQ_TYPE_NONE 0
  - #define IRQ_TYPE_EDGE_RISING 1
  - #define IRQ_TYPE_EDGE_FALLING 2
  - #define IRQ_TYPE_EDGE_BOTH (IRQ_TYPE_EDGE_FALLING | IRQ_TYPE_EDGE_RISING)
  - #define IRQ_TYPE_LEVEL_HIGH 4
  - #define IRQ_TYPE_LEVEL_LOW 8

MEDIATEK CONFIDENTIAL
moredian@moredian.com USE

dws file setting(1/3)

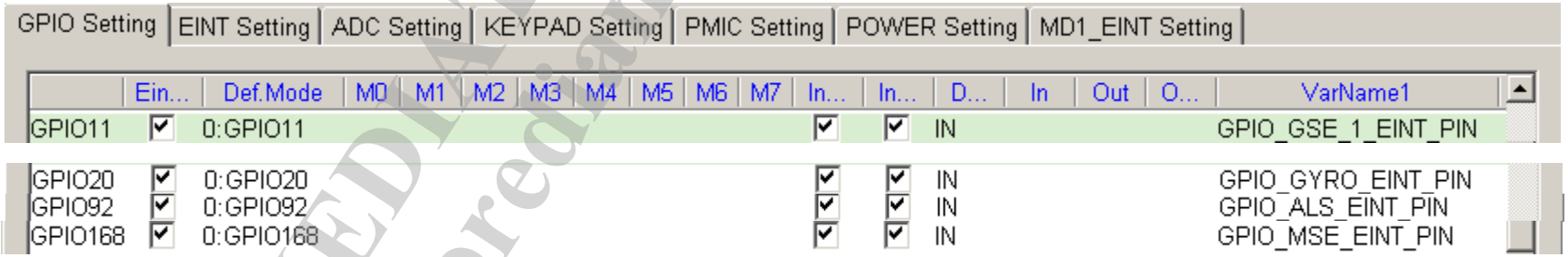
- check settings in dws file
- Check All I2C pin setting in dws file.
  - <kernel_ver>/drivers/misc/mediatek/dws/<platform>/<proj>.dws
  - for example : G-Sensor uses I2C id1
  - VarName1 should be
    - GPIO_I2C1_SDA_PIN
    - GPIO_I2C1_SDA_PIN



MEDIATEK CONFIDENTIAL B
moredian@moredian.com

dws file setting(2/3)

- EINT Pin
  - Note: if your sensor doesn't use interrupt mode, EINT pin is not need to set
  - Check EINT mode
  - VarName1 should be
    - GPIO_GSE_1_EINT_PIN (G-Sensor)
    - GPIO_GYRO_EINT_PIN (Gyro-Sensor)
    - GPIO_ALS_EINT_PIN (Light Sensor, P-Sensor)
    - GPIO_MSE_EINT_PIN (M-Sensor)



MEDIATEK CONFIDENTIAL B
moredian@moredian.com

dws file setting(3/3)

- EINT settings
  - EINT setting : Please reference vendor's sensor Spec.

GPIO Setting | EINT Setting | ADC Setting | KEYPAD Setting | PMIC Setting | POWER Setting | MD1_EINT Setting

	EINT Var	Debounce Time (ms)	Polarity	Sensitive_Level	Debounce En
EINT11	GSE_1	0	High	Level	Disable
EINT20	GYRO	0	High	Level	Disable
EINT92	ALS	0	Low	Level	Disable
EINT168	MSE	0	Low	Level	Disable

- cust.dtsi
  - dws file will generate cust.dtsi in the following path when build, you need to include it in <proj>.dts
    - out/target/<proj>/obj/KERNEL_OBJ/arch/< arm_ver >/boot/dts/cust.dtsi

```
#include "mt6735.dtsi"
#include "cust.dtsi"
```

MEDIATEK CONFIDENTIAL
moredian@moredian.com

sensor dts node

- Msensor node and SW get dts info method

```
cust_mag@0 {
  compatible = "mediatek,s62x";
  i2c_num = <2>;
  i2c_addr = <0x0e 0 0 0>;
  direction = <4>;
  power_id = <0xffff>;
  power_vol = <0>;
  is_batch_supported = <0>;
};
```

```
static int __init s62x_init(void)
{
  const char *name = "mediatek,s62x";

  hw = get_mag_dts_func(name, hw);
  if (!hw)
    pr_err("get dts info fail\n");

  mag_driver_add(&s62x_init_info);
  return 0;
}
```

```
struct mag_hw *get_mag_dts_func(const char *name, struct mag_hw *hw)
{
  node = of_find_compatible_node(NULL, NULL, name);
  if (node) {
    ret = of_property_read_u32_array(node , "i2c_num", i2c_num, ARRAY_SIZE(i2c_num));
    if (ret == 0)
      hw->i2c_num = i2c_num[0];

    ret = of_property_read_u32_array(node , "i2c_addr", i2c_addr, ARRAY_SIZE(i2c_addr));
    if (ret == 0) {
      for (i = 0; i < M_CUST_I2C_ADDR_NUM; i++)
        hw->i2c_addr[i]    = i2c_addr[i];
    }
```

MEDIATEK CONFIDENTIAL
mordian@ moredian.com USE

sensor dts node

- alsps sensor node in <platform>.dts file

```
cust_alsps@0 {
  compatible = "mediatek,epl2182";
  i2c_num = <2>;
  i2c_addr = <0x72 0x48 0x78 0x00>;
  polling_mode_ps = <0>;
  polling_mode_als = <1>;
  power_id = <0xffff>;
  power_vol = <0>;
  als_level = <0 1 1 7 15 15 100 1000 2000 3000 6000 10000 14000 18000 20000>;
  als_value = <40 40 90 90 160 160 225 320 640 1280 1280 2600 2600 2600 10240 10240>;
  ps_threshold_high = <900>;
  ps_threshold_low = <600>;
  is_batch_supported_ps = <0>;
  is_batch_supported_als = <0>;
};
&pio {
  alsps_intpin_cfg: alspspincfg {

      pins_cmd_dat {
      pins = <PINMUX_GPIO65__FUNC_GPIO65>;
      slew-rate = <0>;
      bias-pull-up = <00>;
    };
  };

  alsps_intpin_default: alspsdefaultcfg {

&alsps {
  pinctrl-names = "pin_default", "pin_cfg";
  pinctrl-0 = <&alsps_intpin_default>;
  pinctrl-1 = <&alsps_intpin_cfg>;
  status = "okay";
};
```

pin control information for alsps sensor EINT use

MEDIATEK CONFIDENTIAL B
MEDIATEK CONFIDENTIAL
moredian@moredian.com USE

sensor dts node

- Psensor will use interrupt mode, so EINT is needed
- alsps EINT information need to set in dws file, and it will generate cust.dtsi.
- interrupt trigger type (e.g. IRQ_TYPE_EDGE_FALLING)should be the ones that defined int the following files
  - <kernel_ver>/include/dt-bindings/interrupt-controller/arm-gic.h
  - <kernel_ver>/include/dt-bindings/interrupt-controller/irq.h
- root node of alsps EINT in <platform>.dtsi

```
als: als {
   compatible = "mediatek, als-eint";
};
```

- attach node of alsps eint in <proj>.dts (or cust.dtsi)

```
&als {
    interrupt-parent = <&eintc>;
    interrupts = <65 IRQ_TYPE_LEVEL_LOW>;
    debounce = <65 0>;
    status = "okay";
};
```

MEDIATEK CONFIDENTIAL B
mordan@mordian.com USE

remove sensor not supported for CTS

- device/mediatek/<proj>/device.mk
- or device/mediatek/<platform>/device.mk
- or device/mediatek/common/device.mk

```
ifneq ($(strip $(CUSTOM_KERNEL_ACCELEROMETER)),)  PRODUCT_COPY_FILES +=
frameworks/native/data/etc/android.hardware.sensor.accelerometer.xml:system/etc/permissions/a
ndroid.hardware.sensor.accelerometer.xmlendififneq ($(strip
$(CUSTOM_KERNEL_MAGNETOMETER)),)  PRODUCT_COPY_FILES +=
frameworks/native/data/etc/android.hardware.sensor.compass.xml:system/etc/permissions/androi
d.hardware.sensor.compass.xmlendififneq ($(strip $(CUSTOM_KERNEL_ALSPS)),)
PRODUCT_COPY_FILES +=
frameworks/native/data/etc/android.hardware.sensor.proximity.xml:system/etc/permissions/andro
id.hardware.sensor.proximity.xml  PRODUCT_COPY_FILES +=
frameworks/native/data/etc/android.hardware.sensor.light.xml:system/etc/permissions/android.ha
rdware.sensor.light.xmlelse  ifneq ($(strip $(CUSTOM_KERNEL_PS)),)    PRODUCT_COPY_FILES +=
frameworks/native/data/etc/android.hardware.sensor.proximity.xml:system/etc/permissions/andro
id.hardware.sensor.proximity.xml  endif  ifneq ($(strip $(CUSTOM_KERNEL_ALS)),)
PRODUCT_COPY_FILES +=
frameworks/native/data/etc/android.hardware.sensor.light.xml:system/etc/permissions/android.ha
rdware.sensor.light.xml  endifendififneq ($(strip $(CUSTOM_KERNEL_GYROSCOPE)),)
PRODUCT_COPY_FILES +=
frameworks/native/data/etc/android.hardware.sensor.gyroscope.xml:system/etc/permissions/andr
oid.hardware.sensor.gyroscope.xmlendififneq ($(strip $(CUSTOM_KERNEL_BAROMETER)),)
PRODUCT_COPY_FILES +=
frameworks/native/data/etc/android.hardware.sensor.barometer.xml:system/etc/permissions/andr
oid.hardware.sensor.barometer.xmlendif
```

MEDIATEK CONFIDENTIAL A
mediatek.com

remove sensor not supported for CTS

- copy handheld_core_hardware.xml from frameworks/native/data/etc to device/mediatek/<proj>
- delete sensor that didn't support
- e.g. there isn't msensor, please remove compass in handheld_core_hardware.xml

```
<feature name="android.hardware.location.network" />
<!--feature name="android.hardware.sensor.compass" /-->
<feature name="android.hardware.sensor.accelerometer" />
```

MEDIATEK CONFIDENTIAL
moredian@moredian.com

everyday genius

Copyright © MediaTek Inc. All rights reserved.

MEDIATEK CONFIDENTIAL
moredian@moredian.com USE