MTK Sensors Customer Document

Version:	1.0.6
Release date:	2020-03-12

Specifications are subject to change without notice.

Document Revision History

Revision	Date	Author	Description
1.0	2017-11-24	MTK	
1.0.1	2017-12-07	MTK	Sensor Calibration loadable
1.0.2	2017-12-10	MTK	Tinysis driver , CHRE APP
1.0.3	2017-12-13	MTK	SCP AP
1.0.4	2018-01-08	MTK	SPI
1.0.5	2018-01-16	MTK	AP/SCP uart switch SCP exception debug
1.0.6	2018-01-22	MTK	vendor lib

1

1.1

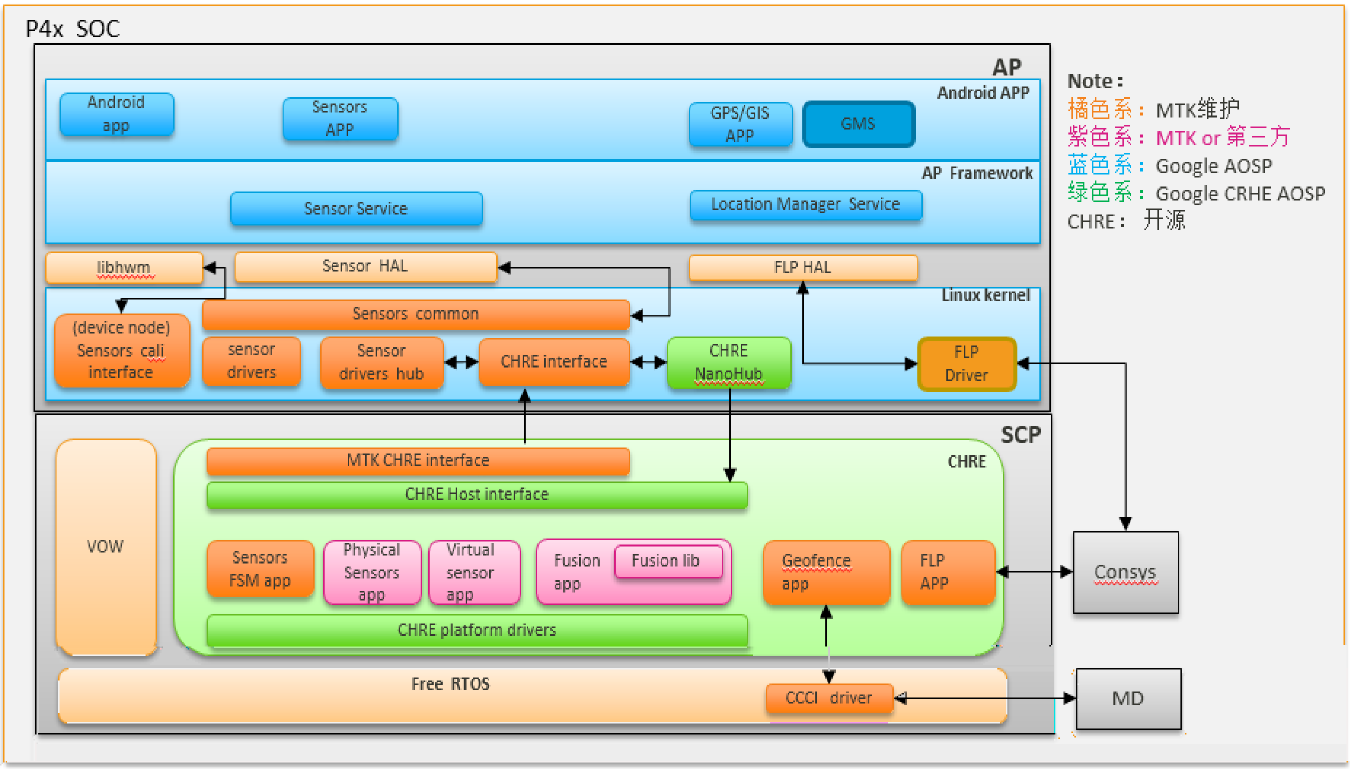
MTK Sensor porting Guild API build option debug

AP side SCP side

MTK P40

2 Architecture overview

2.1 Architecture



MTK Sensor AP SCP AP (CA5x CA7x) SCP CM4
Sensor SCP AP sensor

3 AP sensors introduction

HAL kernel AP sensor porting Guild

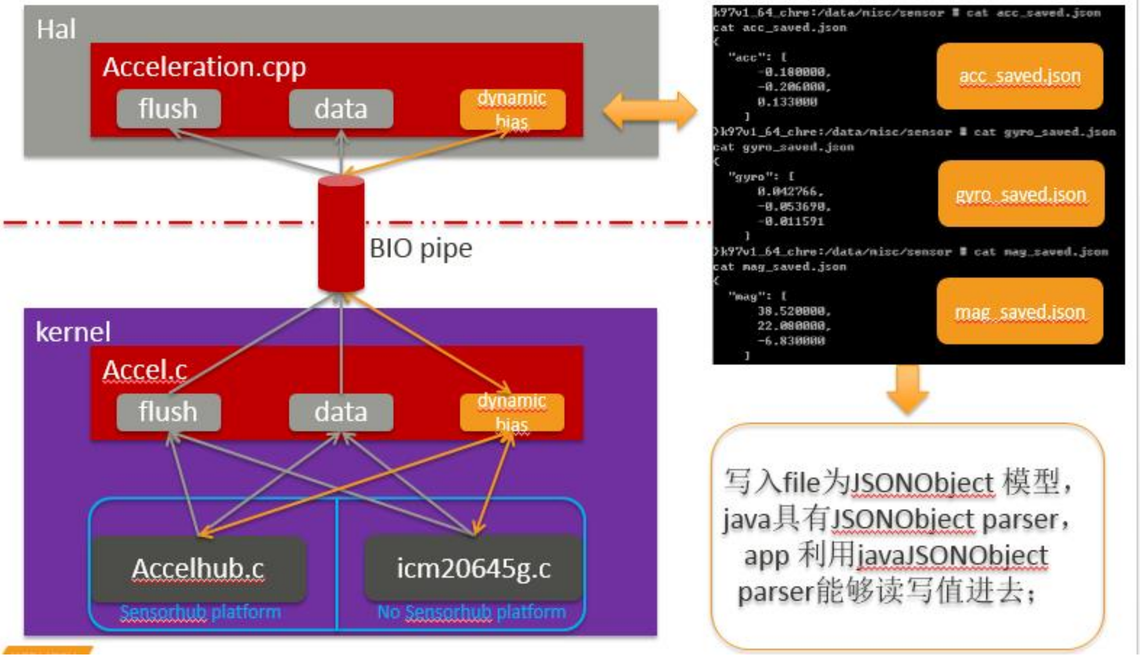
3.1 Linux sensors drivers interface

MTK kernel common sensor Interface porting guiled

3.1.1 MTK sensor common layer introduction

common porting

common MTK HAL MTK Interface (BIO pipe) input event



```
k97v1_64_chre:/data/misc/sensor # cat acc_saved.json
cat acc_saved.json
{
  "acc": [
      -0.180000,
      -0.206000,
      0.133000
    ]
}k97v1_64_chre:/data/misc/sensor # cat gyro_saved.json
cat gyro_saved.json
{
  "gyro": [
      0.042766,
      -0.053690,
      -0.011591
    ]
}k97v1_64_chre:/data/misc/sensor # cat mag_saved.json
cat mag_saved.json
{
  "mag": [
      38.520000,
      22.080000,
      -6.830000
    ]
```

写入file为JSONObject 模型，java具有JSONObject parser，app 利用javaJSONObject parser能够读写值进去；

3.1.2 Common layer API

common API

API

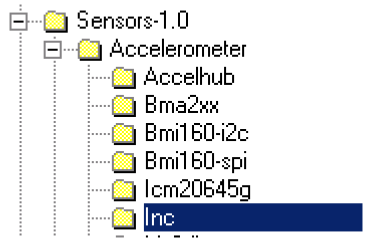
1. **Accel.h,** Android data flow contrl flow API
2. **Acc_factory.h** MTK data flow contrl flow API

Accel.h API Android

API Name	Parameter description
acc_driver_add(struct acc_init_info* obj)	acc_init_info sensor common driver sensor auto detect
acc_register_data_path(struct acc_data_path *data)	acc_data_path sensor function sensor raw data

	common sensor driver MTK sensor
acc_register_control_path(struct acc_control_path *ctl)	acc_control_path 3 function sensor enable/disable setDelay
acc_data_report()	
Acc_flush_report()	flush

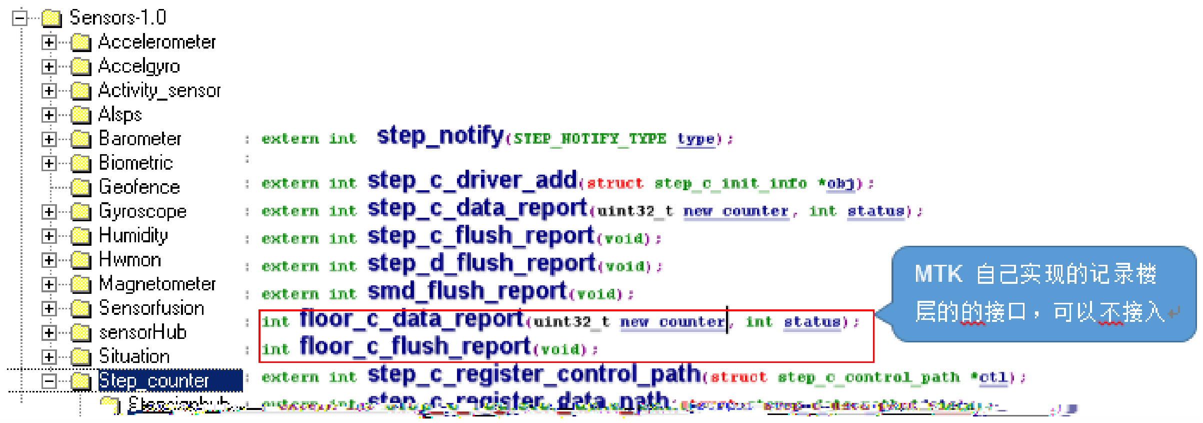
API driver



3.2 Step counter 接入

acc step counter fodler

acc common



```
extern int step_notify(STEP_NOTIFY_TYPE type);

extern int step_c_driver_add(struct step_c_init_info *obj);
extern int step_c_data_report(uint32_t new_counter, int status);
extern int step_c_flush_report(void);
extern int step_d_flush_report(void);
extern int smd_flush_report(void);
int floor_c_data_report(uint32_t new_counter, int status);
int floor_c_flush_report(void);
extern int step_c_register_control_path(struct step_c_control_path *ctl);
extern int step_c_register_data_path(
```

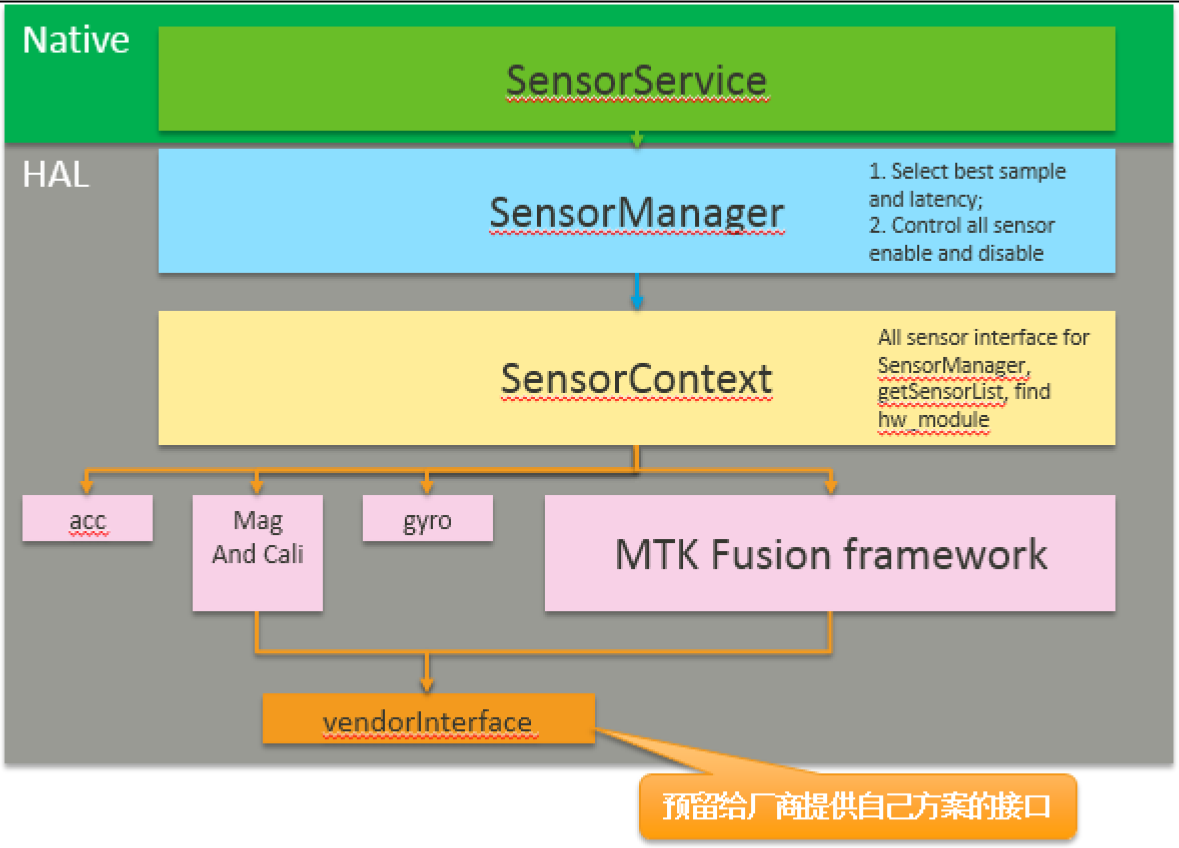
MTK 自己实现的记录楼层的的接口，可以不接入

3.3 Sensor HAL Interface

MTK HAL native fusion

virtual gyro

MTK HAL


Native
SensorService
HAL
SensorManager
1. Select best sample and latency;
2. Control all sensor enable and disable
SensorContext
All sensor interface for SensorManager, getSensorList, find hw_module
acc
Mag And Cali
gyro
MTK Fusion framework
vendorInterface
预留给厂商提供自己方案的接口

3.3.1 SensorManger user manual

- Get sensormanager interface

  mSensorManager = SensorManager::getInstance();

- Create sensor connection

  mSensorConnection = SensorManager::createSensorConnection(magnetic);

- Enable sensor

  mSensorManager->batch(mSensorConnection, ID_ACCELEROMETER, 20000000, 0);

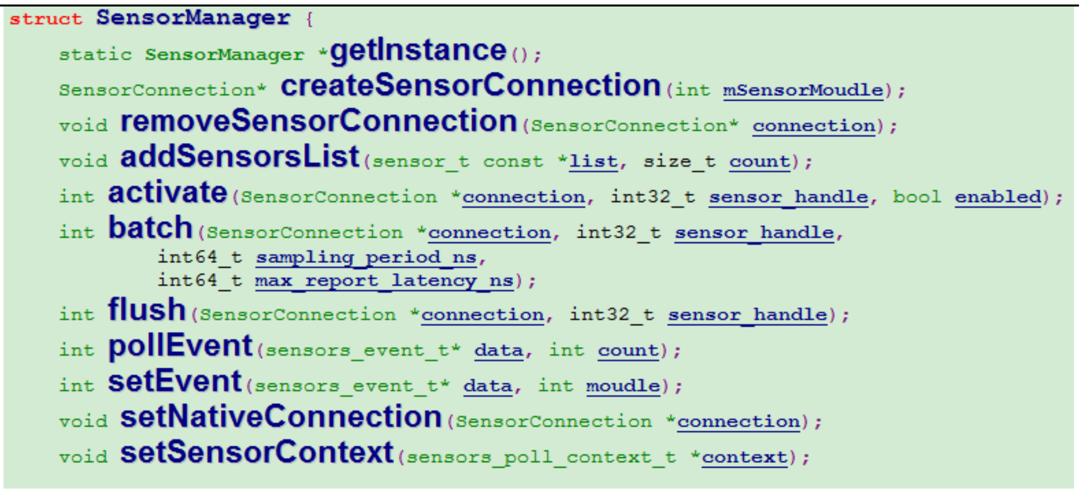
  mSensorManager->activate(mSensorConnection, ID_ACCELEROMETER, true);

- Disable sensor

  mSensorManager->activate(mSensorConnection, ID_ACCELEROMETER, false);

- Remove sensor connection

  mSensorManager->removeSensorConnection(mSensorConnection);

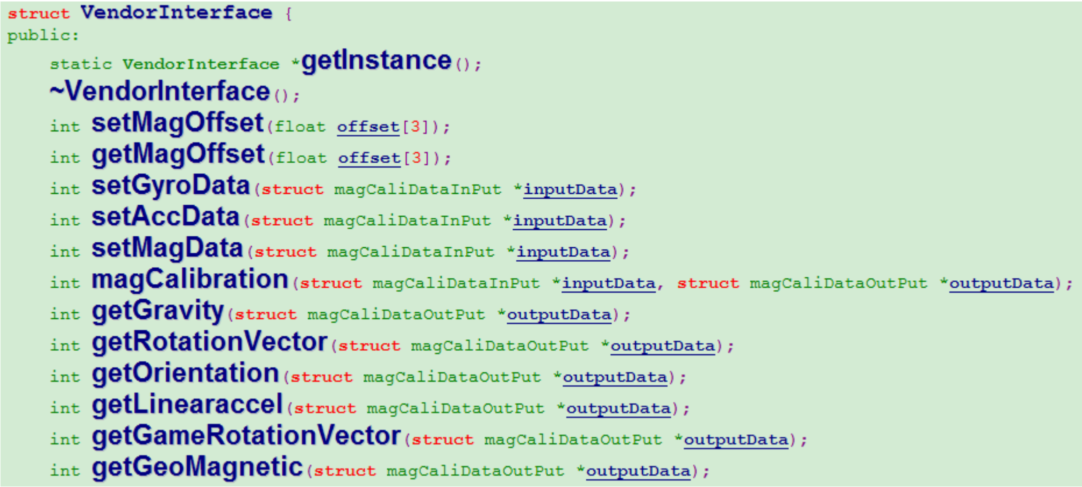


```
struct SensorManager {
    static SensorManager *getInstance();
    SensorConnection* createSensorConnection(int mSensorMoudle);
    void removeSensorConnection(SensorConnection* connection);
    void addSensorsList(sensor_t const *list, size_t count);
    int activate(SensorConnection *connection, int32_t sensor_handle, bool enabled);
    int batch(SensorConnection *connection, int32_t sensor_handle,
            int64_t sampling_period_ns,
            int64_t max_report_latency_ns);
    int flush(SensorConnection *connection, int32_t sensor_handle);
    int pollEvent(sensors_event_t* data, int count);
    int setEvent(sensors_event_t* data, int moudle);
    void setNativeConnection(SensorConnection *connection);
    void setSensorContext(sensors_poll_context_t *context);
```

3.3.2 Vendor Interface user manual

- Get vendor interface

mVendorInterface = VendorInterface::getInstance();



```
struct VendorInterface {
public:
    static VendorInterface *getInstance();
    ~VendorInterface();
    int setMagOffset(float offset[3]);
    int getMagOffset(float offset[3]);
    int setGyroData(struct magCaliDataInPut *inputData);
    int setAccData(struct magCaliDataInPut *inputData);
    int setMagData(struct magCaliDataInPut *inputData);
    int magCalibration(struct magCaliDataInPut *inputData, struct magCaliDataOutPut *outputData);
    int getGravity(struct magCaliDataOutPut *outputData);
    int getRotationVector(struct magCaliDataOutPut *outputData);
    int getOrientation(struct magCaliDataOutPut *outputData);
    int getLinearaccel(struct magCaliDataOutPut *outputData);
    int getGameRotationVector(struct magCaliDataOutPut *outputData);
    int getGeoMagnetic(struct magCaliDataOutPut *outputData);
```

3.4 Sensors Calibration

MTK Android APP Calibration ACC Gyro Proximity threshold MTK APK API

3.4.1 Debug

accel and gyroscope calibration cmd

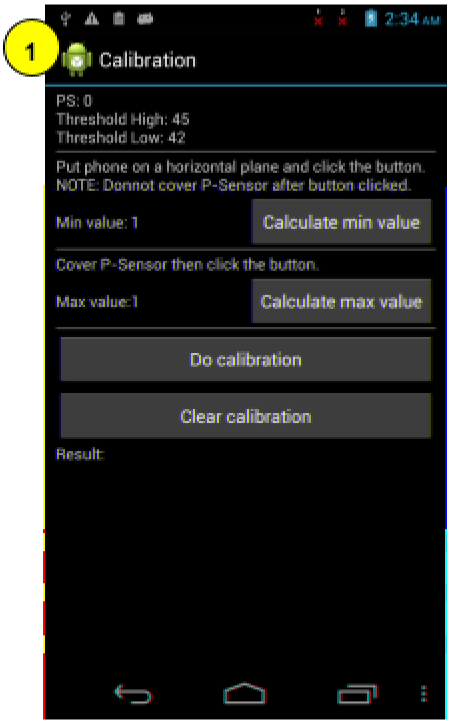
1 adb root

2 adb shell

3 debug Accel calibration cd sys/bus/platform/drivers/gsensor

debug Gyro calibration cd sys/bus/platform/drivers/gyroscope

4 Calibration echo 1 > test_cali



1 Calibration

PS: 0
Threshold High: 45
Threshold Low: 42

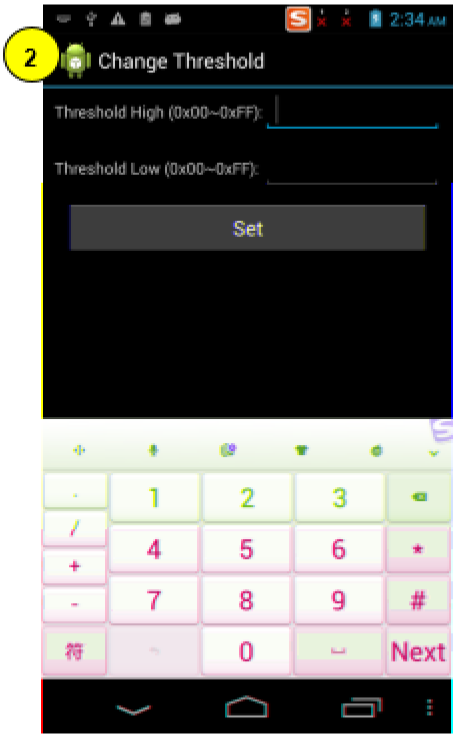
Put phone on a horizontal plane and click the button.
NOTE: Donnot cover P-Sensor after button clicked.

Min value: 1

Cover P-Sensor then click the button.

Max value:1

Result:



2 Change Threshold

Threshold High (0x00~0xFF):

Threshold Low (0x00~0xFF):

Native API vendor\mediatek\proprietary\external\sensor-tools\include\libhwm.h

int get_psensor_data(void)

✓ psensor raw data

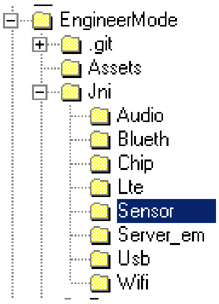
Int set_psensor_threshold(int high, int low)

✓ high low nvram driver

Native psensor Android native API

3.4.4 *#*#3646633#*#*

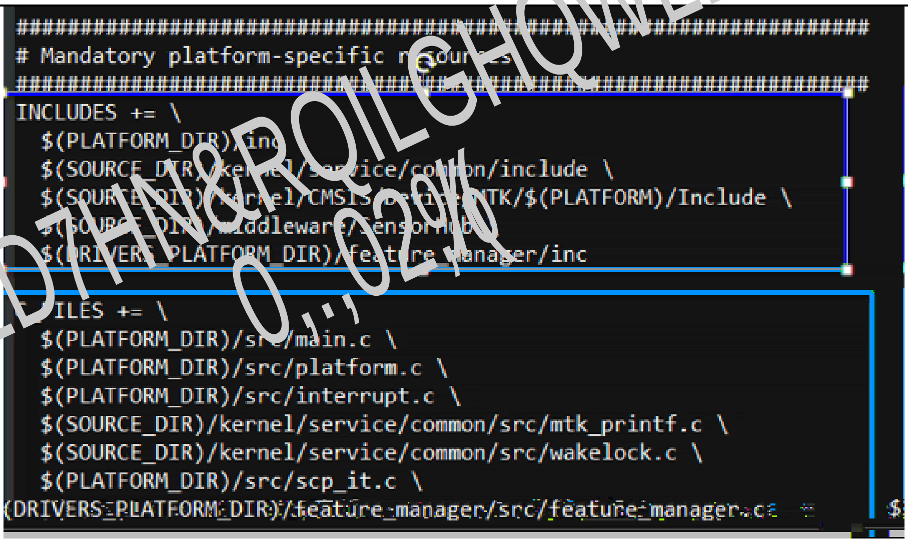
- Native Layer interface : JIN APP



```
static jint calculatePsensorMaxValue(JNIEnv *, jclass) {
    ALOGD("Enter calculatePsensorMaxValue()\n");
    int ret = calculate_psensor_max_value();
    ALOGD("calculatePsensorMaxValue() returned %d\n", ret);

    return ret;
}
```

vendor\mediatek\proprietary\external\apps\engineerMode

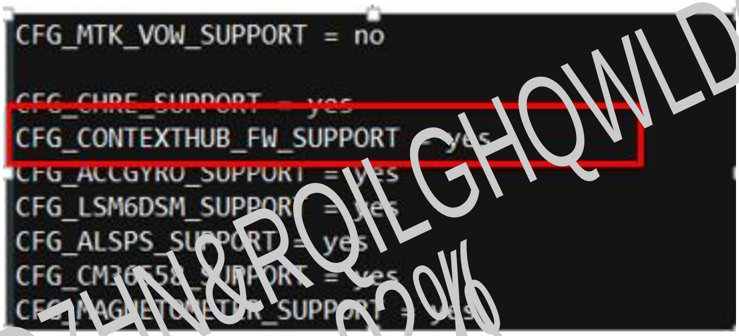
3.5 Device tree introduction



```
################################################################################
# Mandatory platform-specific resources
################################################################################
INCLUDES += \
  $(PLATFORM_DIR)/inc \
  $(SOURCE_DIR)/kernel/service/common/include \
  $(SOURCE_DIR)/kernel/CMSIS/Device/MTK/$(PLATFORM)/Include \
  $(SOURCE_DIR)/middleware/SensorHub \
  $(DRIVERS_PLATFORM_DIR)/feature_manager/inc

C_FILES += \
  $(PLATFORM_DIR)/src/main.c \
  $(PLATFORM_DIR)/src/platform.c \
  $(PLATFORM_DIR)/src/interrupt.c \
  $(SOURCE_DIR)/kernel/service/common/src/mtk_printf.c \
  $(SOURCE_DIR)/kernel/service/common/src/wakelock.c \
  $(PLATFORM_DIR)/src/scp_it.c \
(DRIVERS_PLATFORM_DIR)/feature_manager/src/feature_manager.c
```

2. **Project Configuration file**

ProjectConfig.mk will **overwriting** options in platform.mk



```
CFG_MTK_VOW_SUPPORT = no

CFG_CHRE_SUPPORT = yes
CFG_CONTEXTHUB_FW_SUPPORT = yes
CFG_ACCGYRO_SUPPORT = yes
CFG_LSM6DSM_SUPPORT = yes
CFG_ALSPS_SUPPORT = yes
CFG_CM36558_SUPPORT = yes
CFG_MAGNETOMETER_SUPPORT = yes
```

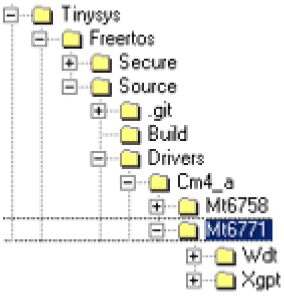
4.1.3 如何在 Tinysis 下添加一新 freeRTOS driver

1 put the code in the appropriate folder

For driver code, put your file at following path:

drivers/$(PROCESSOR)/$(PLATFORM)/(New_Driver_Folder)

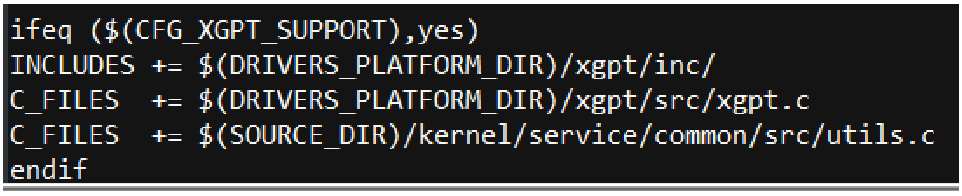
Example: drivers/CM4_A/mt6771/XGPT



2. add a new compiler option in platform.mk

project/$(PROCESSOR)/$(PLATFORM)/platform.mk

example project/mt6771/platform.mk



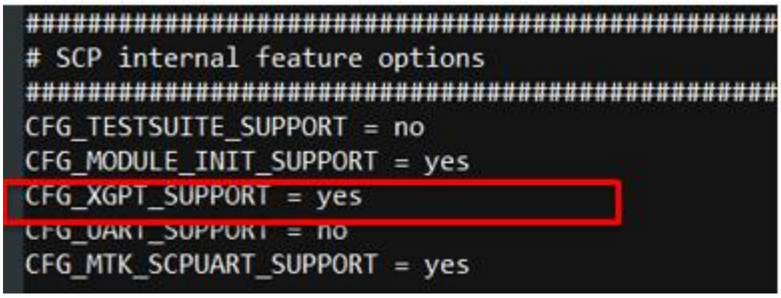
```
ifeq ($(CFG_XGPT_SUPPORT),yes)
INCLUDES += $(DRIVERS_PLATFORM_DIR)/xgpt/inc/
C_FILES  += $(DRIVERS_PLATFORM_DIR)/xgpt/src/xgpt.c
C_FILES  += $(SOURCE_DIR)/kernel/service/common/src/utils.c
endif
```

3. add a new configuration in platform.mk or ProjectConfig.mk

project/$(PROCESSOR)/$(PLATFORM)/platform.mk

project/$(PROCESSOR)/$(PLATFORM)/$(PROJECT)/ProjectConfig.mk

example project/mt6771/platform.mk

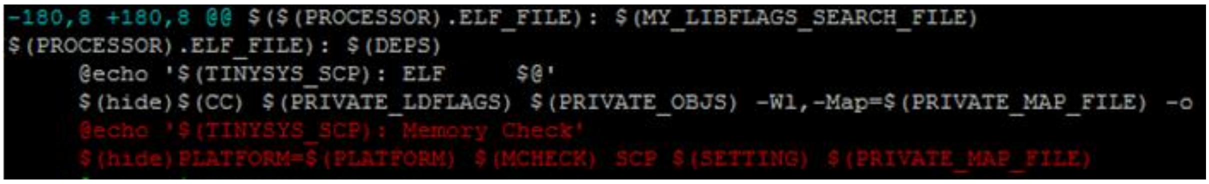


```
#################################################
# SCP internal feature options
#################################################
CFG_TESTSUITE_SUPPORT = no
CFG_MODULE_INIT_SUPPORT = yes
CFG_XGPT_SUPPORT = yes
CFG_UART_SUPPORT = no
CFG_MTK_SCPUART_SUPPORT = yes
```

4.1.4 SCP code size

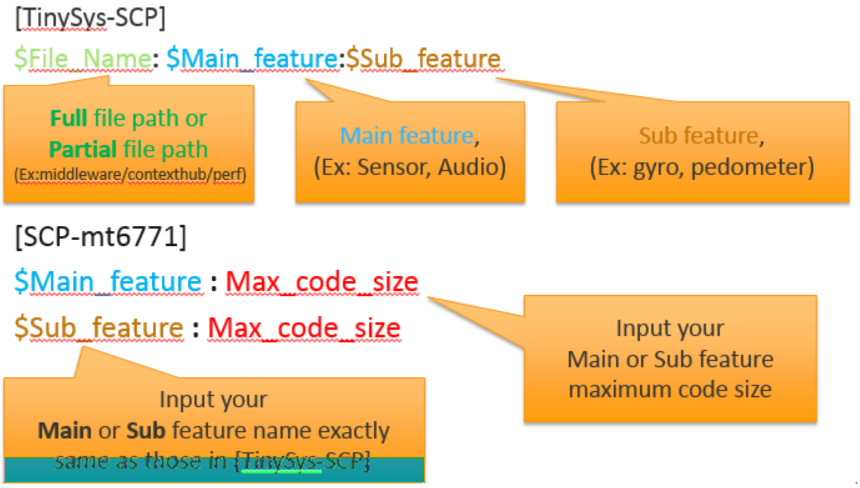
1. memoryReport.py is a script which use to limit code size at the build time.
   If code size over your settings, it will cause build errors.
   (script: vendor/mediatek/proprietary/tinysys/freertos/source/tools/**memoryReport.py**)
2. This script is hooked by tinysys scp make file:

vendor/mediatek/proprietary/tinysys/freertos/source/build/**config.mk**



```
-180,8 +180,8 @@ $($(PROCESSOR).ELF_FILE): $(MY_LIBFLAGS_SEARCH_FILE)
$(PROCESSOR).ELF_FILE): $(DEPS)
     @echo '$(TINYSYS_SCP): ELF     $@'
     $(hide)$(CC) $(PRIVATE_LDFLAGS) $(PRIVATE_OBJS) -Wl,-Map=$(PRIVATE_MAP_FILE) -o
     @echo '$(TINYSYS_SCP): Memory Check'
     $(hide)PLATFORM=$(PLATFORM) $(MCHECK) SCP $(SETTING) $(PRIVATE_MAP_FILE)
```

3. Configuration file at following path :
   vendor/mediatek/proprietary/tinysys/freertos/source/project/CM4_A/**$PLATFORM**/platform/**Setting.ini**
4. **Setting.ini**



```
[TinySys-SCP]
$File_Name: $Main_feature:$Sub_feature
```

Full file path or **Partial** file path (Ex:middleware/contexthub/perf)

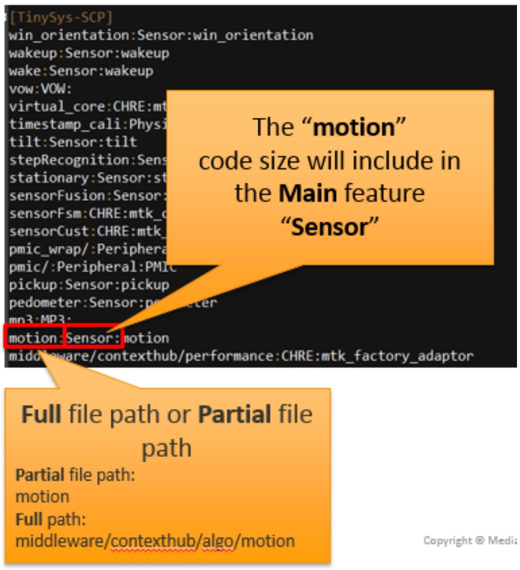
Main feature, (Ex: Sensor, Audio)

Sub feature, (Ex: gyro, pedometer)

```
[SCP-mt6771]
$Main_feature : Max_code_size
$Sub_feature : Max_code_size
```

Input your **Main** or **Sub** feature name exactly same as those in [TinySys-SCP]

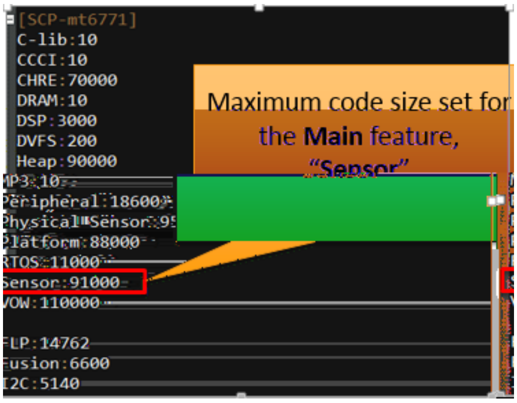
Input your Main or Sub feature maximum code size



```
[TinySys-SCP]
win_orientation:Sensor:win_orientation
wakeup:Sensor:wakeup
wake:Sensor:wakeup
vow:VOW:
virtual_core:CHRE:mt
timestamp_cali:Physi
tilt:Sensor:tilt
stepRecognition:Sens
stationary:Sensor:st
sensorFusion:Sensor:
sensorFsm:CHRE:mtk_c
sensorCust:CHRE:mtk_
pmic_wrap/:Periphera
pmic/:Peripheral:PMIC
pickup:Sensor:pickup
pedometer:Sensor:pe    ter
mp3:MP3:
motion:Sensor:motion
midd    ware/contexthub/performance:CHRE:mtk_factory_adaptor
```

The "**motion**" code size will include in the **Main** feature "**Sensor**"

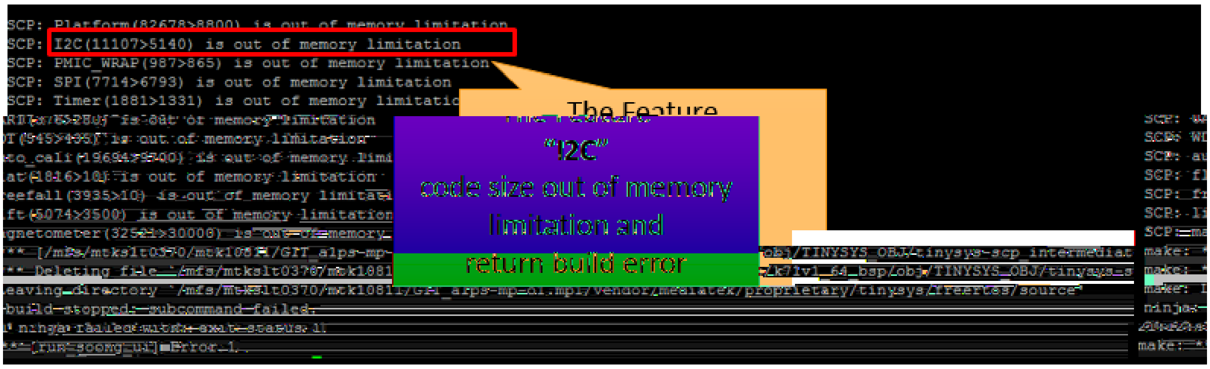
Full file path or **Partial** file path
Partial file path:
motion
Full path:
middleware/contexthub/algo/motion



```
[SCP-mt6771]
C-lib:10
CCCI:10
CHRE:70000
DRAM:10
DSP:3000
DVFS:200
Heap:90000
MP3:10
Peripheral:18600
Physical Sensor:95
Platform:88000
RTOS:11000
Sensor:91000
VOW:110000

FLP:14762
Fusion:6600
I2C:5140
```

Maximum code size set for the **Main** feature, "Sensor"

size build error



```
SCP: Platform(82678>8800) is out of memory limitation
SCP: I2C(11107>5140) is out of memory limitation
SCP: PMIC_WRAP(987>865) is out of memory limitation
SCP: SPI(7714>6793) is out of memory limitation
SCP: Timer(1881>1331) is out of memory limitatio
ARI(x/65280) is out of memory limitation
DT(545>435) is out of memory limitation
to_cali(19694>9700) is out of memory limi
at(1816>10) is out of memory limitation
reefall(3935>10) is out of memory limitati
ift(5074>3500) is out of memory limitation
gnetometer(32521>30000) is out of memory
*** [/mfs/mtks1t0370/mtk10811/GIT_alps-mp-         obj/TINYSYS_OBJ/tinysys-scp_intermediat
*** Deleting file `/mfs/mtks1t0370/mtk1081         /k71v1_64_bsp/obj/TINYSYS_OBJ/tinysys-s
eaving directory `/mfs/mtks1t0370/mtk10811/GIT_alps-mp-o1.mp1/vendor/mediatek/proprietary/tinysys/freertos/source'
build stopped: subcommand failed.
! ninja failed with: exit status: 1
** [run_soong_ui] Error 1
```

The Feature
"I2C"
code size out of memory
limitation and
return build error

5 CHRE sensors introduction

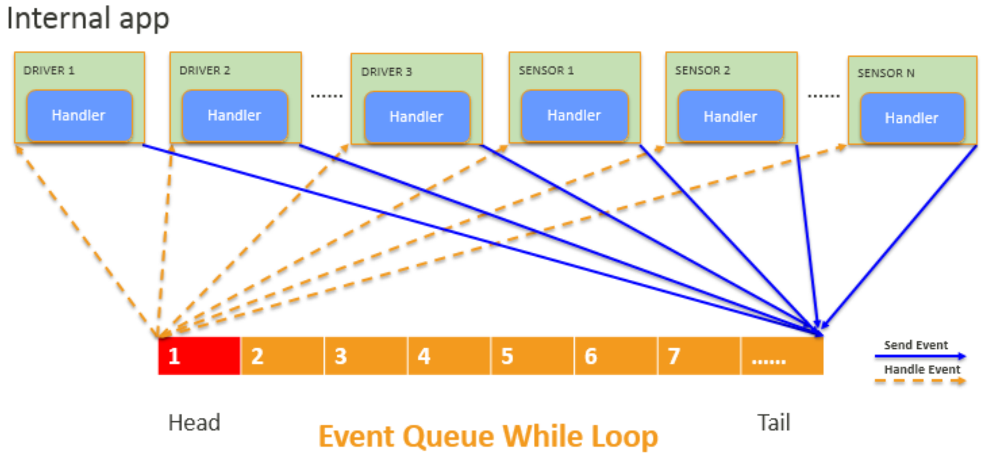
5.1 CHRE

SCP MTK sensor hub feature Google CHRE .
CHRE (Context Hub Runtime Environment) OS

Event Queue CHRE while Event Queue
Event Queue interrupt Event Queue CHRE
512 Event Queue EventQueue

CHRE driver Google nano hub app
nano hub app

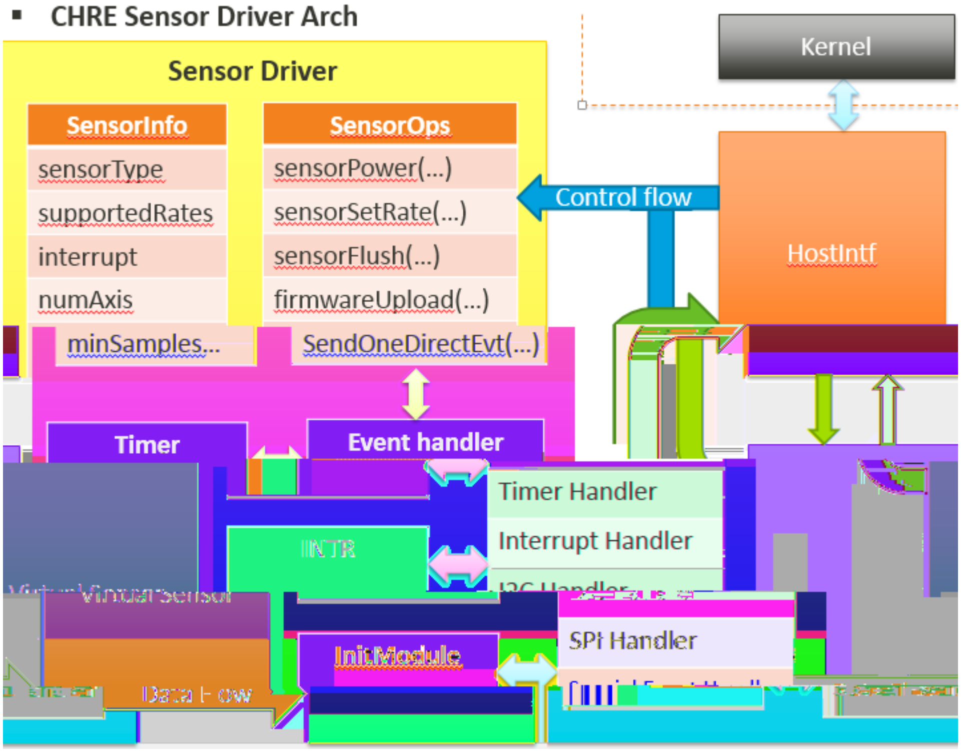
CHRE



5.2 MTK CHRE Sensors Common Layer

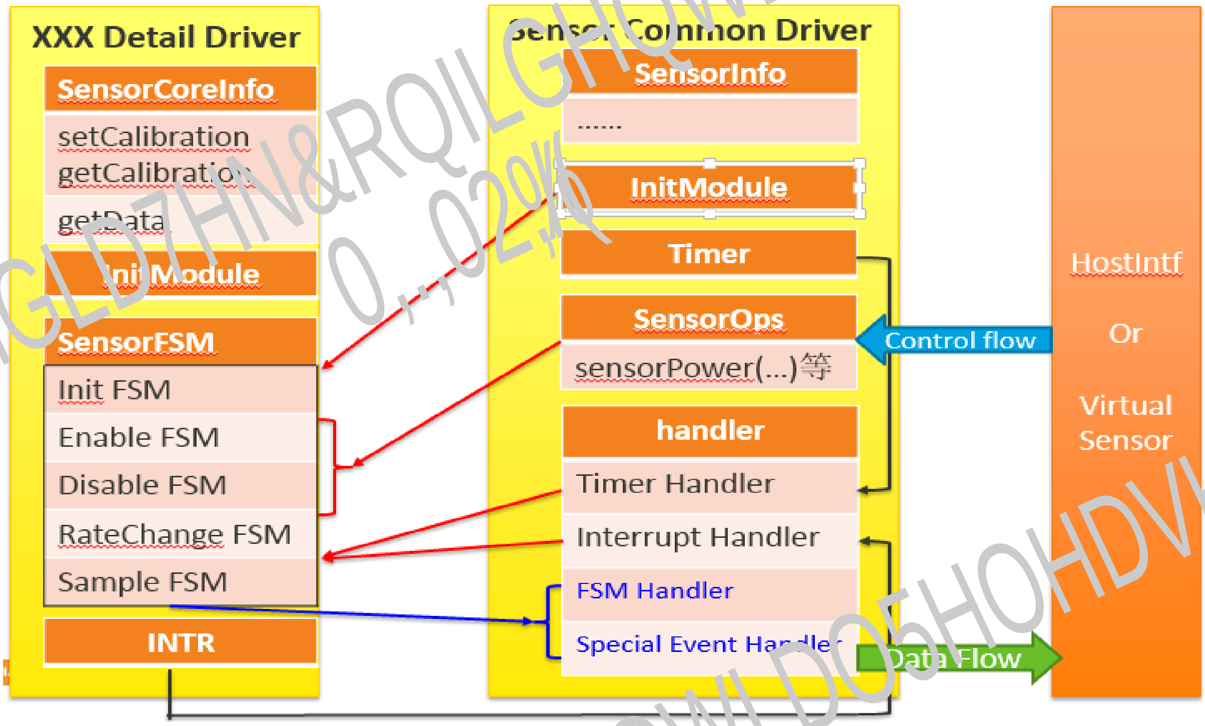
CHRE app coding CHRE app

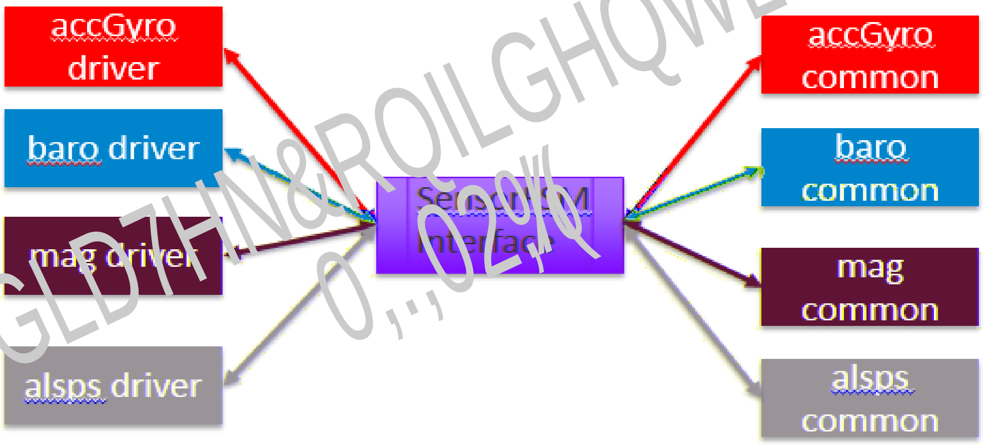
- CHRE Sensor Driver Arch



App hostintf hostintf AP SensorManager SensorService HAL
control flow data flow hostintf

porting bug MTK sensor
sensorFSM sensorFSM





common

- Init auto detect)
- sensorFSM
- SensorCoreInfo

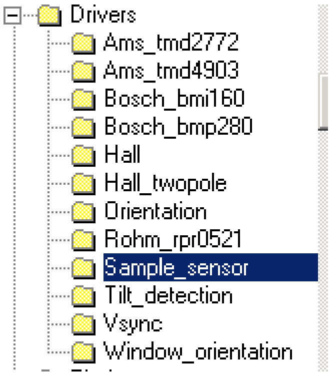
5.3 CHRE APP

MTK sensor CHRE APP CHRE APP sensor sensor APP APP

sensor type flat type

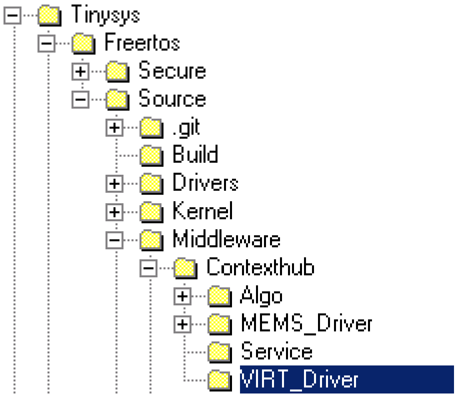
5.3.1 Copy demo driver CHRE APP

vendor\mediatek\proprietary\hardware\contexthub\firmware\src\drivers\sample_sensor\sample_sensor.cpp



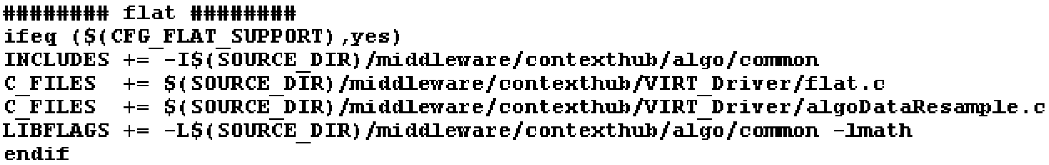
Copy

vendor\mediatek\proprietary\tinysis\freertos\souce\middleware\virt_driver\



5.3.2

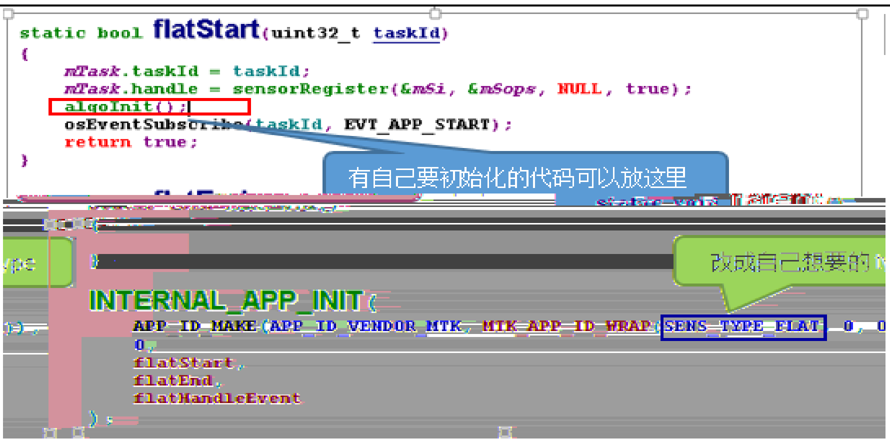
project/CM4_a/mt6771/platform/feature_config/chre.mk



```
######## flat ########
ifeq ($(CFG_FLAT_SUPPORT),yes)
INCLUDES += -I$(SOURCE_DIR)/middleware/contexthub/algo/common
C_FILES  += $(SOURCE_DIR)/middleware/contexthub/VIRT_Driver/flat.c
C_FILES  += $(SOURCE_DIR)/middleware/contexthub/VIRT_Driver/algoDataResample.c
LIBFLAGS += -L$(SOURCE_DIR)/middleware/contexthub/algo/common -lmath
endif
```

5.3.3 APP

1. CHRE APP

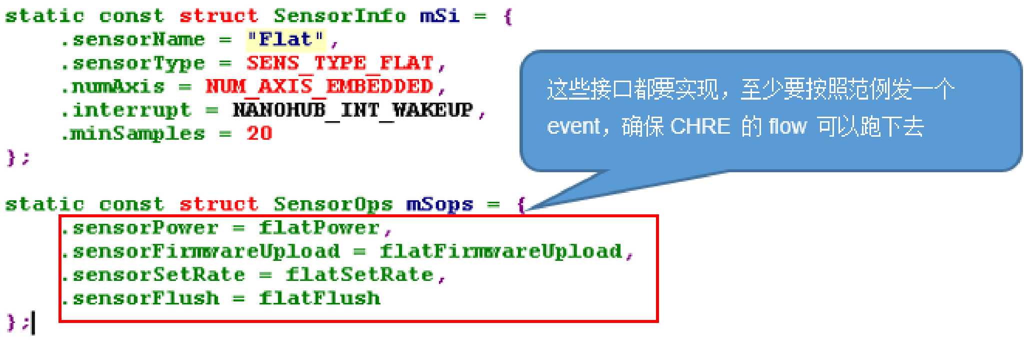


```
static bool flatStart(uint32_t taskId)
{
    mTask.taskId = taskId;
    mTask.handle = sensorRegister(&mSi, &mSops, NULL, true);
    algoInit();
    osEventSubscribe(taskId, EVT_APP_START);
    return true;
}
```

有自己要初始化的代码可以放这里

```
INTERNAL_APP_INIT(
    APP_ID_MAKE(APP_ID_VENDOR_MTK, MTK_APP_ID_WRAP(SENS_TYPE_FLAT, 0, 0
    0,
    flatStart,
    flatEnd,
    flatHandleEvent
);
```

2. APP sensor

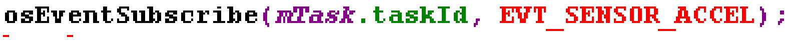


```
static const struct SensorInfo mSi = {
    .sensorName = "Flat",
    .sensorType = SENS_TYPE_FLAT,
    .numAxis = NUM_AXIS_EMBEDDED,
    .interrupt = NANOHUB_INT_WAKEUP,
    .minSamples = 20
};

static const struct SensorOps mSops = {
    .sensorPower = flatPower,
    .sensorFirmwareUpload = flatFirmwareUpload,
    .sensorSetRate = flatSetRate,
    .sensorFlush = flatFlush
};
```

这些接口都要实现，至少要按照范例发一个event，确保 CHRE 的 flow 可以跑下去

3.

Flat ACC raw data

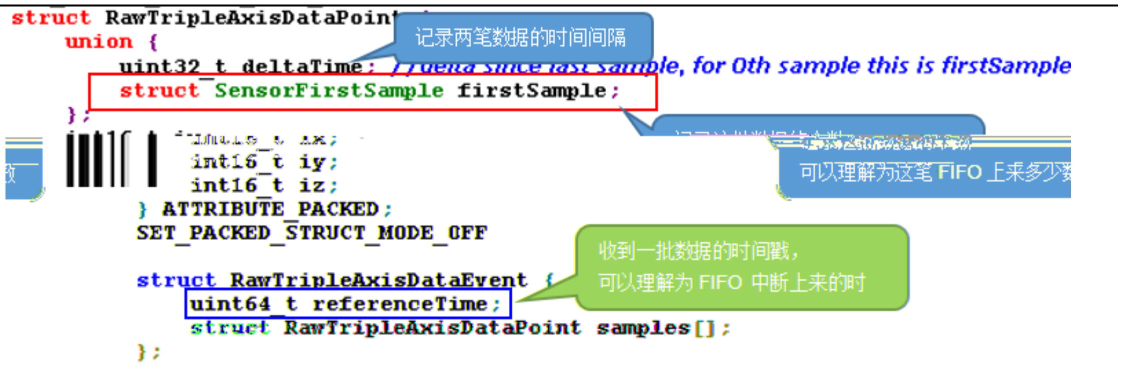


```
osEventSubscribe(mTask.taskId, EVT_SENSOR_ACCEL);
```

4.

5. CHRE sensor

vendor\mediatek\proprietary\hardware\contexthub\firmware\inc\sensors.h



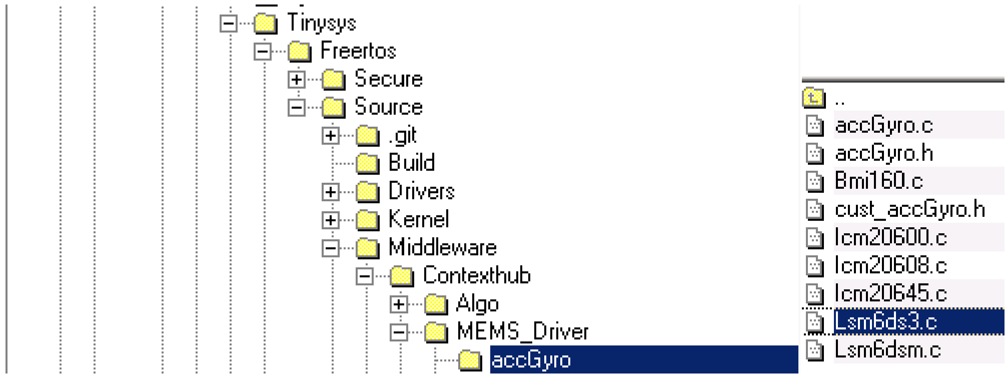
```
struct RawTripleAxisDataPoint {
    union {
        uint32_t deltaTime; //delta since last sample, for 0th sample this is firstSample
        struct SensorFirstSample firstSample;
    };
            int16_t ix;
            int16_t iy;
            int16_t iz;
        } ATTRIBUTE_PACKED;
        SET_PACKED_STRUCT_MODE_OFF

        struct RawTripleAxisDataEvent {
            uint64_t referenceTime;
            struct RawTripleAxisDataPoint samples[];
        };
```

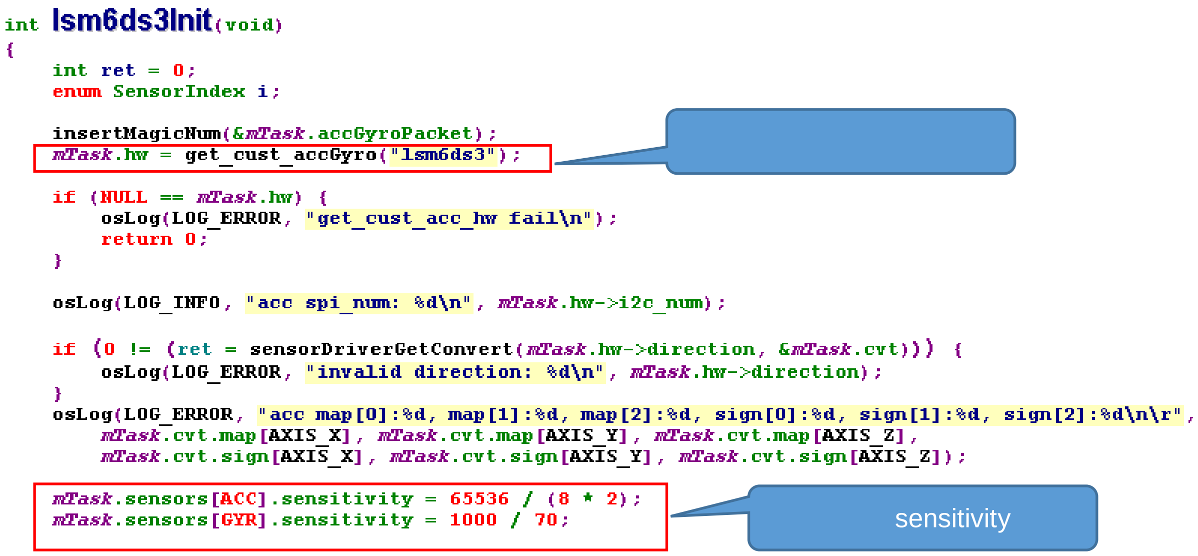
5.4 A+G driver porting guide

5.4.1 A+G initialization

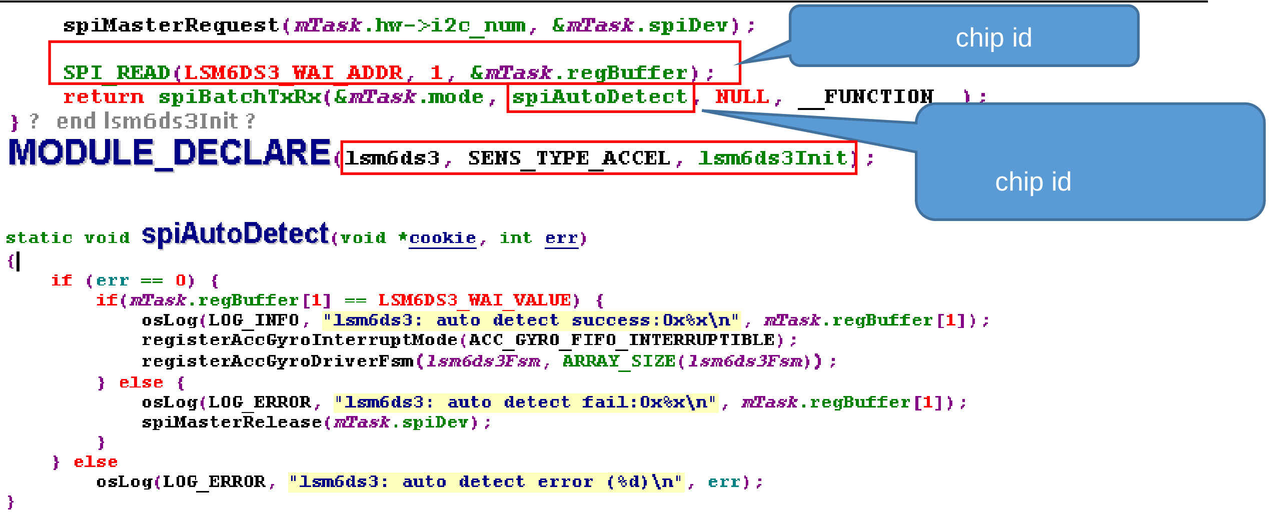
ST driver

Copy driver copy





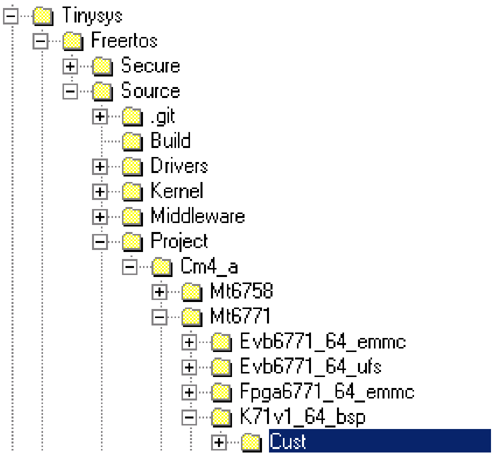
```
int lsm6ds3Init(void)
{
    int ret = 0;
    enum SensorIndex i;

    insertMagicNum(&mTask.accGyroPacket);
    mTask.hw = get_cust_accGyro("lsm6ds3");

    if (NULL == mTask.hw) {
        osLog(LOG_ERROR, "get_cust_acc_hw fail\n");
        return 0;
    }

    osLog(LOG_INFO, "acc spi_num: %d\n", mTask.hw->i2c_num);

    if (0 != (ret = sensorDriverGetConvert(mTask.hw->direction, &mTask.cvt))) {
        osLog(LOG_ERROR, "invalid direction: %d\n", mTask.hw->direction);
    }
    osLog(LOG_ERROR, "acc map[0]:%d, map[1]:%d, map[2]:%d, sign[0]:%d, sign[1]:%d, sign[2]:%d\n\r",
        mTask.cvt.map[AXIS_X], mTask.cvt.map[AXIS_Y], mTask.cvt.map[AXIS_Z],
        mTask.cvt.sign[AXIS_X], mTask.cvt.sign[AXIS_Y], mTask.cvt.sign[AXIS_Z]);

    mTask.sensors[ACC].sensitivity = 65536 / (8 * 2);
    mTask.sensors[GYR].sensitivity = 1000 / 70;
```



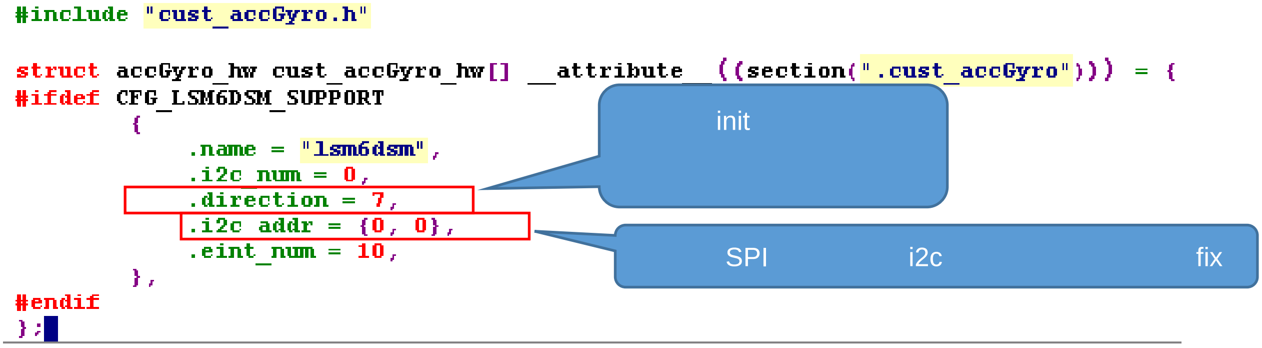
```
    spiMasterRequest(mTask.hw->i2c_num, &mTask.spiDev);

    SPI_READ(LSM6DS3_WAI_ADDR, 1, &mTask.regBuffer);
    return spiBatchTxRx(&mTask.mode, spiAutoDetect, NULL, __FUNCTION__);
} ? end lsm6ds3Init ?
MODULE_DECLARE(lsm6ds3, SENS_TYPE_ACCEL, lsm6ds3Init);


static void spiAutoDetect(void *cookie, int err)
{
    if (err == 0) {
        if(mTask.regBuffer[1] == LSM6DS3_WAI_VALUE) {
            osLog(LOG_INFO, "lsm6ds3: auto detect success:0x%x\n", mTask.regBuffer[1]);
            registerAccGyroInterruptMode(ACC_GYRO_FIFO_INTERRUPTIBLE);
            registerAccGyroDriverFsm(lsm6ds3Fsm, ARRAY_SIZE(lsm6ds3Fsm));
        } else {
            osLog(LOG_ERROR, "lsm6ds3: auto detect fail:0x%x\n", mTask.regBuffer[1]);
            spiMasterRelease(mTask.spiDev);
        }
    } else
        osLog(LOG_ERROR, "lsm6ds3: auto detect error (%d)\n", err);
}
```

chip id

chip id



```
⊟ Tinysys
  ⊟ Freertos
    ⊞ Secure
    ⊟ Source
      ⊞ .git
        Build
      ⊞ Drivers
      ⊞ Kernel
      ⊞ Middleware
      ⊟ Project
        ⊟ Cm4_a
          ⊞ Mt6758
          ⊟ Mt6771
            ⊞ Evb6771_64_emmc
            ⊞ Evb6771_64_ufs
            ⊞ Fpga6771_64_emmc
            ⊟ K71v1_64_bsp
              ⊞ Cust
```



```
#include "cust_accGyro.h"

struct accGyro_hw cust_accGyro_hw[] __attribute__((section(".cust_accGyro"))) = {
#ifdef CFG_LSM6DSM_SUPPORT
        {
            .name = "lsm6dsm",
            .i2c_num = 0,
            .direction = 7,
            .i2c_addr = {0, 0},
            .eint_num = 10,
        },
#endif
};
```

init

SPI i2c fix

1. SensorFSM

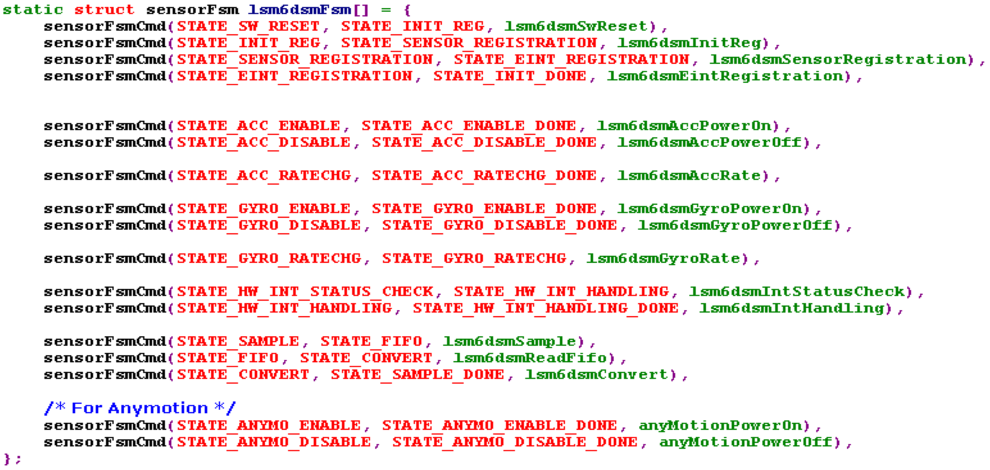
STAT_ENABLE STAT_SAMPLE

driver Event ENABLE_DONE

DISABLE_DONE RATECHG_DONE, SAMPLE_DONE event

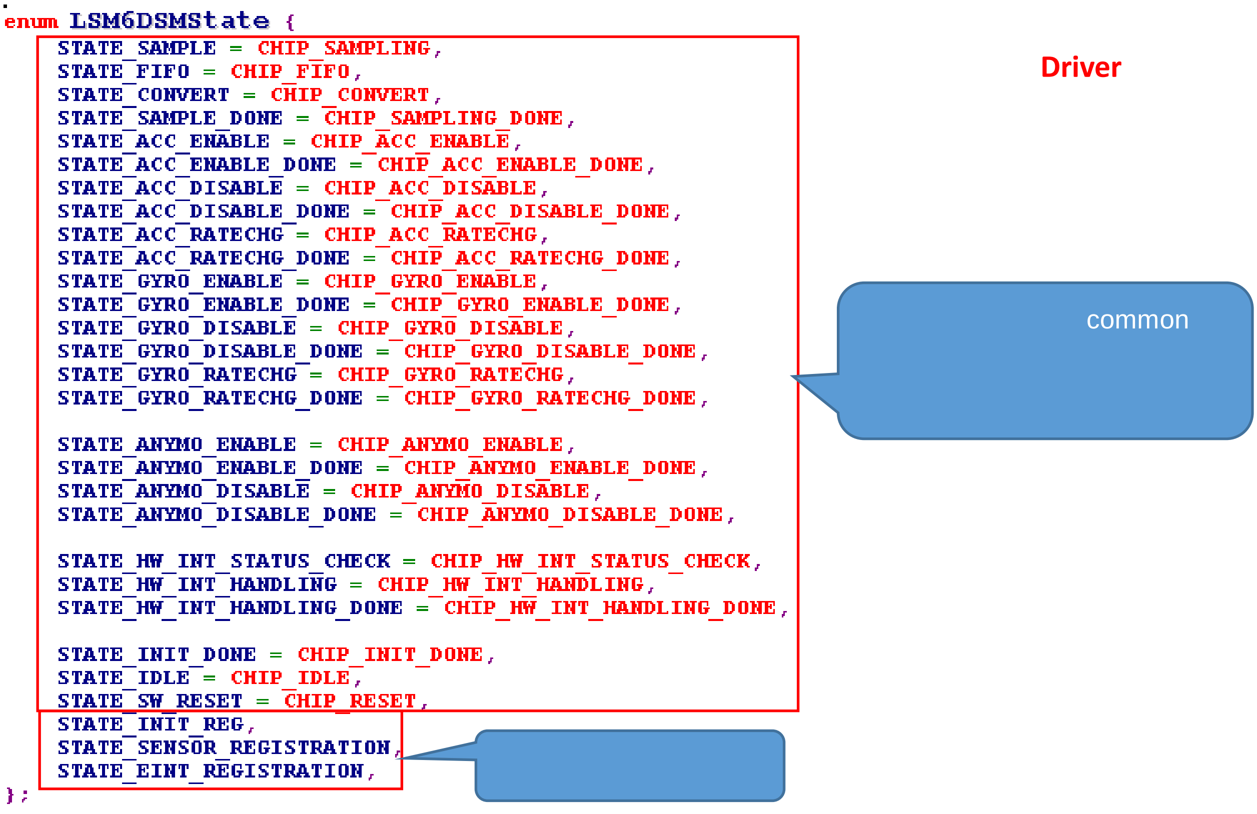
, hw spec i2c/SPI transfer state

:



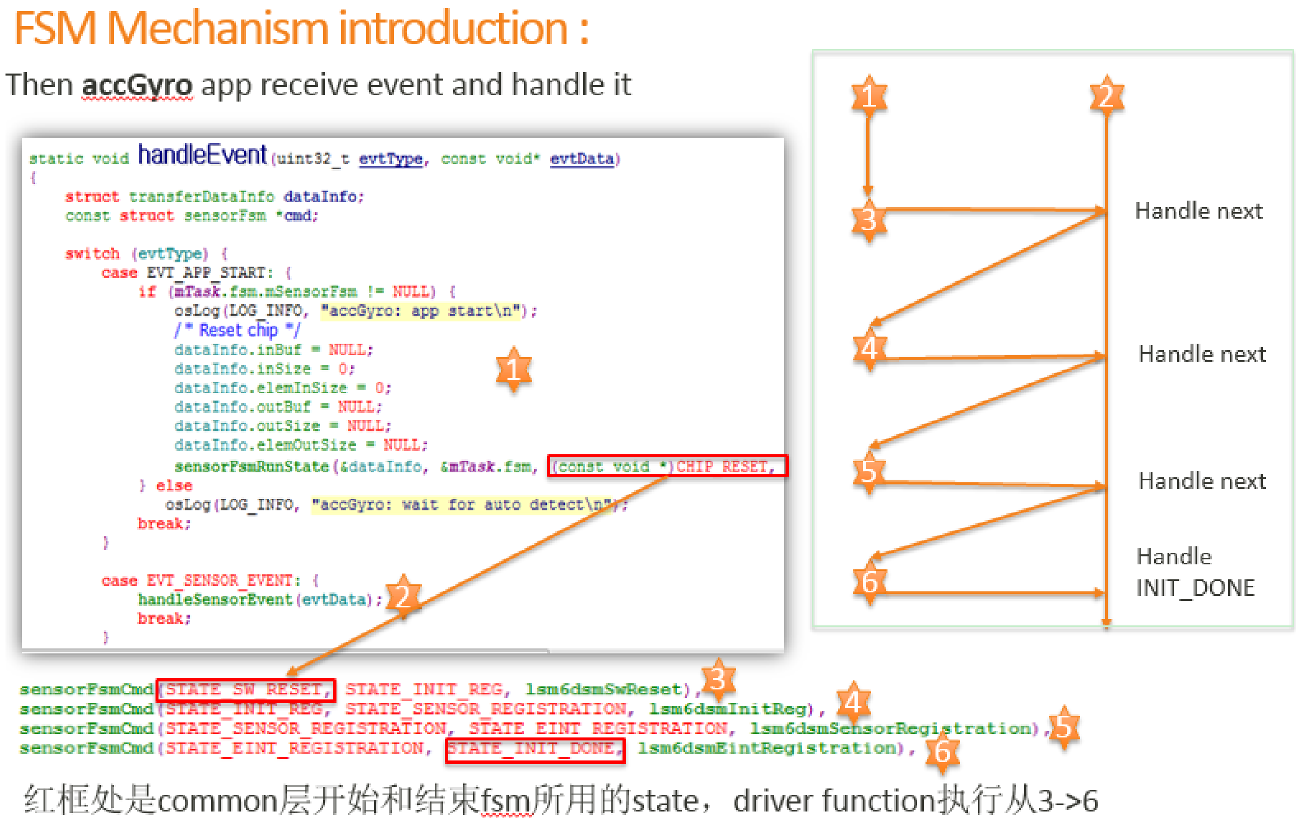
```
static struct sensorFsm lsm6dsmFsm[] = {
    sensorFsmCmd(STATE_SW_RESET, STATE_INIT_REG, lsm6dsmSwReset),
    sensorFsmCmd(STATE_INIT_REG, STATE_SENSOR_REGISTRATION, lsm6dsmInitReg),
    sensorFsmCmd(STATE_SENSOR_REGISTRATION, STATE_EINT_REGISTRATION, lsm6dsmSensorRegistration),
    sensorFsmCmd(STATE_EINT_REGISTRATION, STATE_INIT_DONE, lsm6dsmEintRegistration),


    sensorFsmCmd(STATE_ACC_ENABLE, STATE_ACC_ENABLE_DONE, lsm6dsmAccPowerOn),
    sensorFsmCmd(STATE_ACC_DISABLE, STATE_ACC_DISABLE_DONE, lsm6dsmAccPowerOff),

    sensorFsmCmd(STATE_ACC_RATECHG, STATE_ACC_RATECHG_DONE, lsm6dsmAccRate),

    sensorFsmCmd(STATE_GYRO_ENABLE, STATE_GYRO_ENABLE_DONE, lsm6dsmGyroPowerOn),
    sensorFsmCmd(STATE_GYRO_DISABLE, STATE_GYRO_DISABLE_DONE, lsm6dsmGyroPowerOff),

    sensorFsmCmd(STATE_GYRO_RATECHG, STATE_GYRO_RATECHG, lsm6dsmGyroRate),

    sensorFsmCmd(STATE_HW_INT_STATUS_CHECK, STATE_HW_INT_HANDLING, lsm6dsmIntStatusCheck),
    sensorFsmCmd(STATE_HW_INT_HANDLING, STATE_HW_INT_HANDLING_DONE, lsm6dsmIntHandling),

    sensorFsmCmd(STATE_SAMPLE, STATE_FIFO, lsm6dsmSample),
    sensorFsmCmd(STATE_FIFO, STATE_CONVERT, lsm6dsmReadFifo),
    sensorFsmCmd(STATE_CONVERT, STATE_SAMPLE_DONE, lsm6dsmConvert),

    /* For Anymotion */
    sensorFsmCmd(STATE_ANYMO_ENABLE, STATE_ANYMO_ENABLE_DONE, anyMotionPowerOn),
    sensorFsmCmd(STATE_ANYMO_DISABLE, STATE_ANYMO_DISABLE_DONE, anyMotionPowerOff),
};
```

.



```
enum LSM6DSMState {
    STATE_SAMPLE = CHIP_SAMPLING,
    STATE_FIFO = CHIP_FIFO,
    STATE_CONVERT = CHIP_CONVERT,
    STATE_SAMPLE_DONE = CHIP_SAMPLING_DONE,
    STATE_ACC_ENABLE = CHIP_ACC_ENABLE,
    STATE_ACC_ENABLE_DONE = CHIP_ACC_ENABLE_DONE,
    STATE_ACC_DISABLE = CHIP_ACC_DISABLE,
    STATE_ACC_DISABLE_DONE = CHIP_ACC_DISABLE_DONE,
    STATE_ACC_RATECHG = CHIP_ACC_RATECHG,
    STATE_ACC_RATECHG_DONE = CHIP_ACC_RATECHG_DONE,
    STATE_GYRO_ENABLE = CHIP_GYRO_ENABLE,
    STATE_GYRO_ENABLE_DONE = CHIP_GYRO_ENABLE_DONE,
    STATE_GYRO_DISABLE = CHIP_GYRO_DISABLE,
    STATE_GYRO_DISABLE_DONE = CHIP_GYRO_DISABLE_DONE,
    STATE_GYRO_RATECHG = CHIP_GYRO_RATECHG,
    STATE_GYRO_RATECHG_DONE = CHIP_GYRO_RATECHG_DONE,

    STATE_ANYMO_ENABLE = CHIP_ANYMO_ENABLE,
    STATE_ANYMO_ENABLE_DONE = CHIP_ANYMO_ENABLE_DONE,
    STATE_ANYMO_DISABLE = CHIP_ANYMO_DISABLE,
    STATE_ANYMO_DISABLE_DONE = CHIP_ANYMO_DISABLE_DONE,

    STATE_HW_INT_STATUS_CHECK = CHIP_HW_INT_STATUS_CHECK,
    STATE_HW_INT_HANDLING = CHIP_HW_INT_HANDLING,
    STATE_HW_INT_HANDLING_DONE = CHIP_HW_INT_HANDLING_DONE,

    STATE_INIT_DONE = CHIP_INIT_DONE,
    STATE_IDLE = CHIP_IDLE,
    STATE_SW_RESET = CHIP_RESET,
    STATE_INIT_REG,
    STATE_SENSOR_REGISTRATION,
    STATE_EINT_REGISTRATION,
};
```

Init



FSM Mechanism introduction :

Then accGyro app receive event and handle it

```
static void handleEvent(uint32_t evtType, const void* evtData)
{
    struct transferDataInfo dataInfo;
    const struct sensorFsm *cmd;

    switch (evtType) {
        case EVT_APP_START: {
            if (mTask.fsm.mSensorFsm != NULL) {
                osLog(LOG_INFO, "accGyro: app start\n");
                /* Reset chip */
                dataInfo.inBuf = NULL;
                dataInfo.inSize = 0;
                dataInfo.elemInSize = 0;
                dataInfo.outBuf = NULL;
                dataInfo.outSize = NULL;
                dataInfo.elemOutSize = NULL;
                sensorFsmRunState(&dataInfo, &mTask.fsm, (const void *)CHIP_RESET,
            } else
               osLog(LOG_INFO, "accGyro: wait for auto detect\n");
            break;
        }

        case EVT_SENSOR_EVENT: {
            handleSensorEvent(evtData);
            break;
        }
```

```
sensorFsmCmd(STATE_SW_RESET, STATE_INIT_REG, lsm6dsmSwReset),
sensorFsmCmd(STATE_INIT_REG, STATE_SENSOR_REGISTRATION, lsm6dsmInitReg),
sensorFsmCmd(STATE_SENSOR_REGISTRATION, STATE_EINT_REGISTRATION, lsm6dsmSensorRegistration),
sensorFsmCmd(STATE_EINT_REGISTRATION, STATE_INIT_DONE, lsm6dsmEintRegistration),
```

红框处是common层开始和结束fsm所用的state，driver function执行从3->6

5.4.2 Enable/Disable

```
sensorFsmCmd(STATE_ACC_ENABLE, STATE_ACC_ENABLE_DONE, lsm6dsmAccPowerOn),
sensorFsmCmd(STATE_ACC_DISABLE, STATE_ACC_DISABLE_DONE, lsm6dsmAccPowerOff),

sensorFsmCmd(STATE_GYRO_ENABLE, STATE_GYRO_ENABLE_DONE, lsm6dsmGyroPowerOn),
sensorFsmCmd(STATE_GYRO_DISABLE, STATE_GYRO_DISABLE_DONE, lsm6dsmGyroPowerOff),
```

FIFO FIFO FIFO

```
registerAccGyroFifoInfo((mTask.sensors[ACC].hwRate == 0) ? 0 : 1024000000000 / mTask.sensors[ACC].hwRate,
    (mTask.sensors[GYR].hwRate == 0) ? 0 : 1024000000000 / mTask.sensors[GYR].hwRate);
```

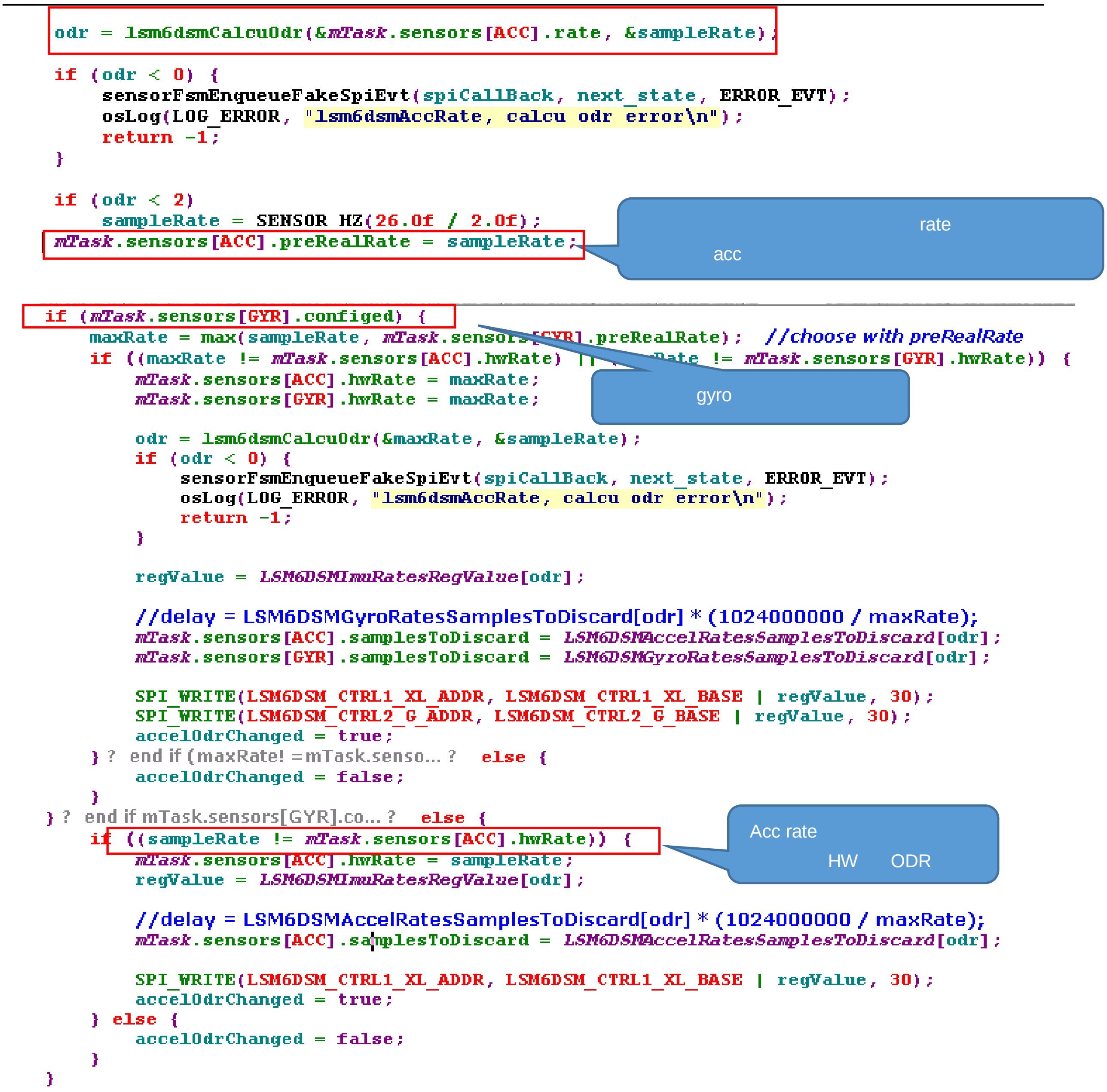
data sheet

5.4.3 Report rate

A+G driver rate Acc Gyro FIFO

```
static int lsm6dsmAccRate(I2cCallbackF i2cCallBack, SpiCbkF spiCallBack, void *next_state,
                          void *inBuf, uint8_t inSize, uint8_t elemInSize,
                          void *outBuf, uint8_t *outSize, uint8_t *elemOutSize)
{
```

rate
report rate

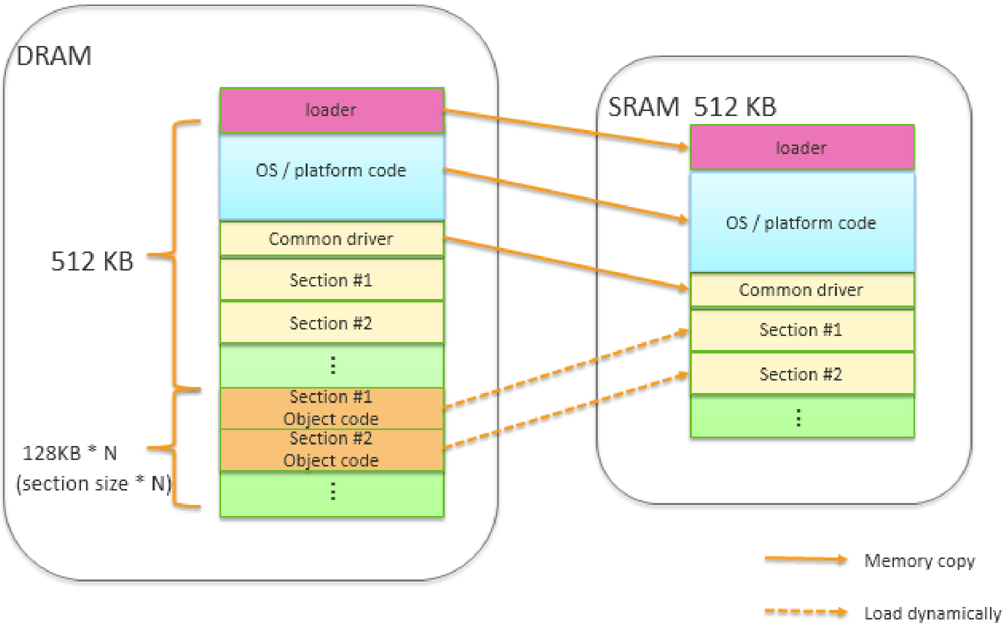


```
    odr = lsm6dsmCalcuOdr(&mTask.sensors[ACC].rate, &sampleRate);

    if (odr < 0) {
        sensorFsmEnqueueFakeSpiEvt(spiCallBack, next_state, ERROR_EVT);
        osLog(LOG_ERROR, "lsm6dsmAccRate, calcu odr error\n");
        return -1;
    }

    if (odr < 2)
        sampleRate = SENSOR_HZ(26.0f / 2.0f);
    mTask.sensors[ACC].preRealRate = sampleRate;

   if (mTask.sensors[GYR].configed) {
       maxRate = max(sampleRate, mTask.sensors[GYR].preRealRate);  //choose with preRealRate
       if ((maxRate != mTask.sensors[ACC].hwRate) || (maxRate != mTask.sensors[GYR].hwRate)) {
           mTask.sensors[ACC].hwRate = maxRate;
           mTask.sensors[GYR].hwRate = maxRate;

           odr = lsm6dsmCalcuOdr(&maxRate, &sampleRate);
           if (odr < 0) {
               sensorFsmEnqueueFakeSpiEvt(spiCallBack, next_state, ERROR_EVT);
               osLog(LOG_ERROR, "lsm6dsmAccRate, calcu odr error\n");
               return -1;
           }

           regValue = LSM6DSMImuRatesRegValue[odr];

           //delay = LSM6DSMGyroRatesSamplesToDiscard[odr] * (1024000000 / maxRate);
           mTask.sensors[ACC].samplesToDiscard = LSM6DSMAccelRatesSamplesToDiscard[odr];
           mTask.sensors[GYR].samplesToDiscard = LSM6DSMGyroRatesSamplesToDiscard[odr];

           SPI_WRITE(LSM6DSM_CTRL1_XL_ADDR, LSM6DSM_CTRL1_XL_BASE | regValue, 30);
           SPI_WRITE(LSM6DSM_CTRL2_G_ADDR, LSM6DSM_CTRL2_G_BASE | regValue, 30);
           accelOdrChanged = true;
       } ? end if (maxRate! =mTask.senso... ?  else {
           accelOdrChanged = false;
       }
   } ? end if mTask.sensors[GYR].co... ?  else {
       if ((sampleRate != mTask.sensors[ACC].hwRate)) {
           mTask.sensors[ACC].hwRate = sampleRate;
           regValue = LSM6DSMImuRatesRegValue[odr];

           //delay = LSM6DSMAccelRatesSamplesToDiscard[odr] * (1024000000 / maxRate);
           mTask.sensors[ACC].samplesToDiscard = LSM6DSMAccelRatesSamplesToDiscard[odr];

           SPI_WRITE(LSM6DSM_CTRL1_XL_ADDR, LSM6DSM_CTRL1_XL_BASE | regValue, 30);
           accelOdrChanged = true;
       } else {
           accelOdrChanged = false;
       }
   }
```

5.5 Sensor driver overlay

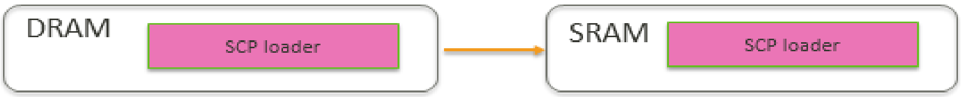
Purpose: sensor SCP driver auto detect SCP SRAM driver DRAM SCP load

- Load overlay scp image : emmc -> dram - > sram

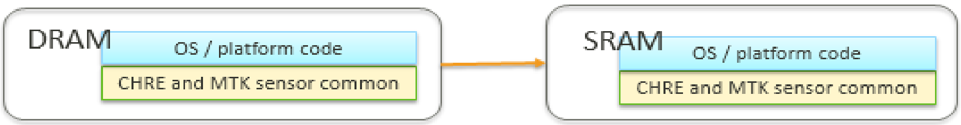


Overlay load flow

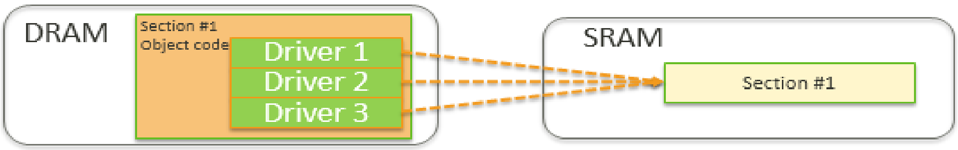
1) Memory copy loader code from dram to SRAM, then SCP run loader



2) SCP loader copy Tinysys common code from dram to SRAM, os run and sensor driver init



3) OverlayRemap copy sensor driver 1 to sram section and hw verify , if fail ,copy driver 2 and verify , if success , remap next sensor type



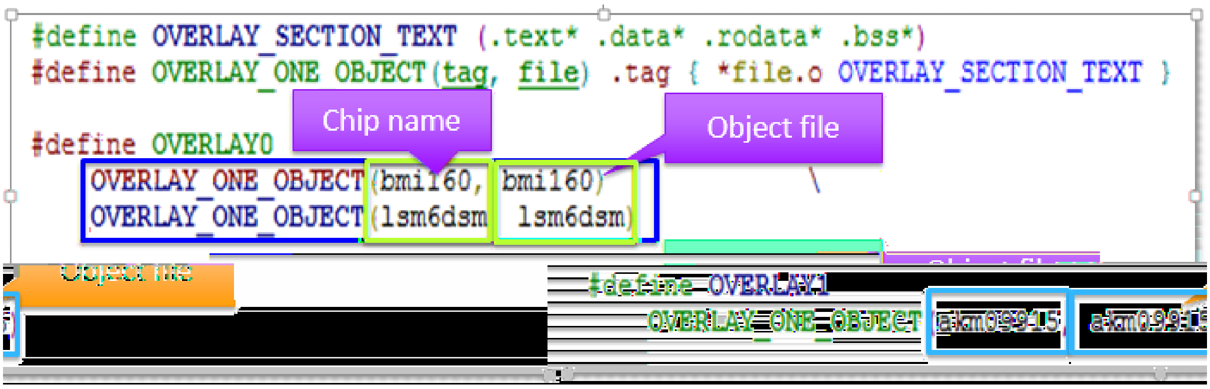
One section represent one sensor type ,may have multiple drivers (object code)

5.5.1 overlay driver

1) ADD object in linker script

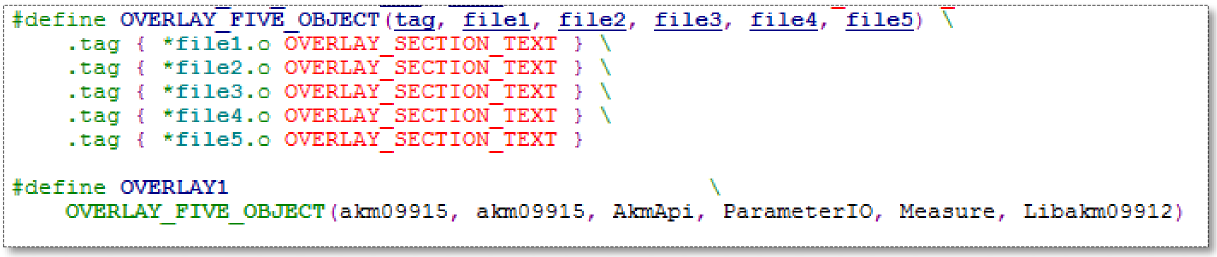
vendor\mediatek\proprietary\tinysys\freertos\source\project\CM4_A\#PLATFORM\$PROJECT\inc\overlay_sensor.h

For A+G sensor type , may have bmi160 chip and lsm6dsm chip, add driver object file in overlay scp image, and for mag sensor type ,may have akm09915 chip.



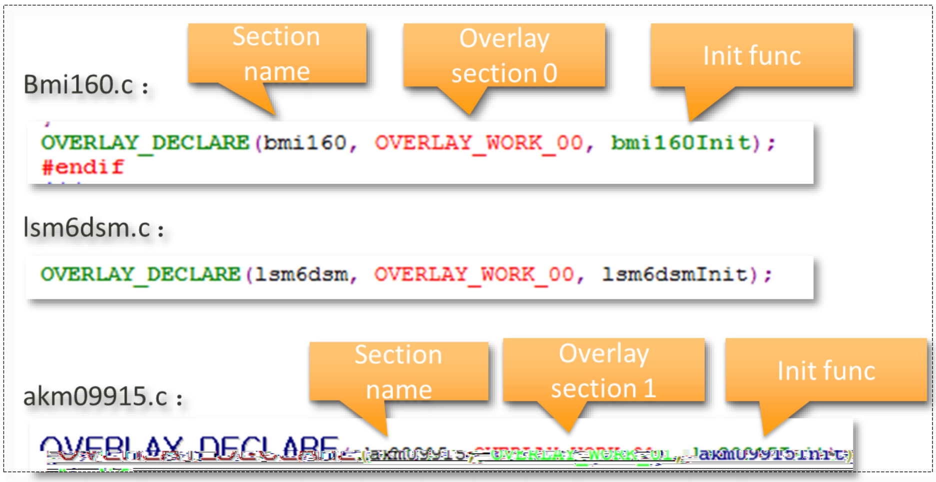
```
#define OVERLAY_SECTION_TEXT (.text* .data* .rodata* .bss*)
#define OVERLAY_ONE_OBJECT(tag, file) .tag { *file.o OVERLAY_SECTION_TEXT }

#define OVERLAY0                                                       \
    OVERLAY_ONE_OBJECT(bmi160, bmi160)
    OVERLAY_ONE_OBJECT(lsm6dsm  lsm6dsm)

#define OVERLAY1
    OVERLAY_ONE_OBJECT(akm09915, akm09915
```

For mag sensor type , there are several library file



```
#define OVERLAY_FIVE_OBJECT(tag, file1, file2, file3, file4, file5) \
    .tag { *file1.o OVERLAY_SECTION_TEXT } \
    .tag { *file2.o OVERLAY_SECTION_TEXT } \
    .tag { *file3.o OVERLAY_SECTION_TEXT } \
    .tag { *file4.o OVERLAY_SECTION_TEXT } \
    .tag { *file5.o OVERLAY_SECTION_TEXT }

#define OVERLAY1                                   \
    OVERLAY_FIVE_OBJECT(akm09915, akm09915, AkmApi, ParameterIO, Measure, Libakm09912)
```

2) ADD overlay declare in your overlay object (driver)



Bmi160.c :

```
OVERLAY_DECLARE(bmi160, OVERLAY_WORK_00, bmi160Init);
#endif
```

lsm6dsm.c :

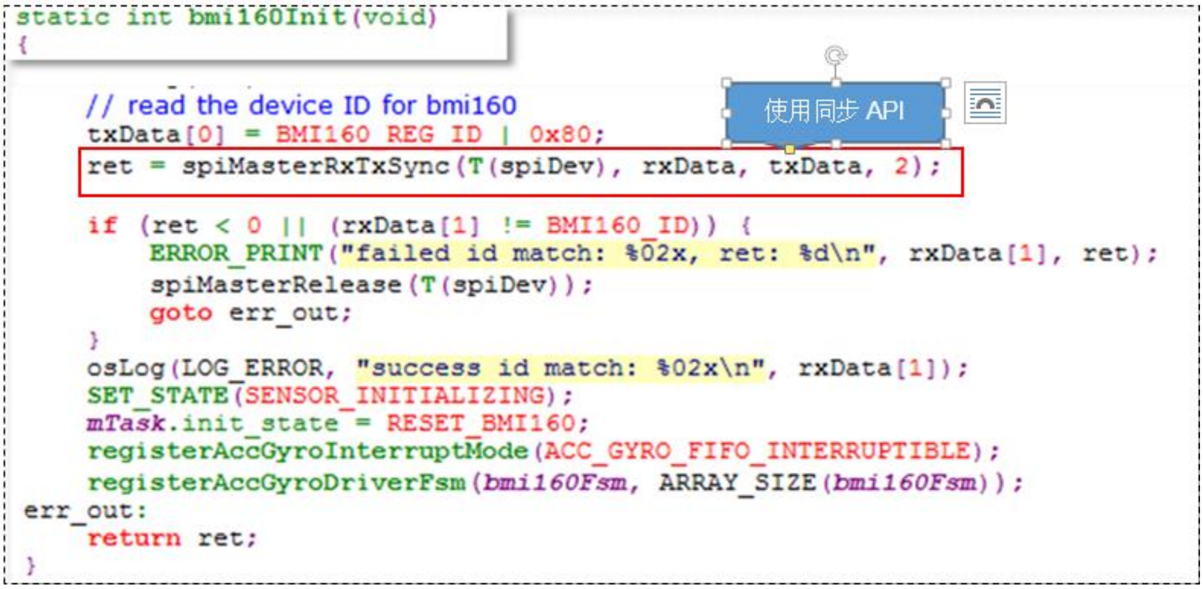
```
OVERLAY_DECLARE(lsm6dsm, OVERLAY_WORK_00, lsm6dsmInit);
```

akm09915.c :

```
OVERLAY_DECLARE(akm09915, OVERLAY_WORK_01, akm09915Init);
```

3) Driver SPI or i2c API init (driver)

vendor\mediatek\proprietary\tinysys\freertos\source\middleware\contexthub\MEMS_Driver\

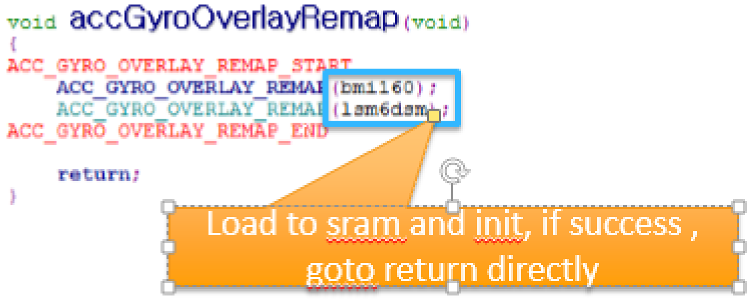


```
static int bmi160Init(void)
{

    // read the device ID for bmi160
    txData[0] = BMI160_REG_ID | 0x80;
    ret = spiMasterRxTxSync(T(spiDev), rxData, txData, 2);

    if (ret < 0 || (rxData[1] != BMI160_ID)) {
        ERROR_PRINT("failed id match: %02x, ret: %d\n", rxData[1], ret);
        spiMasterRelease(T(spiDev));
        goto err_out;
    }
    osLog(LOG_ERROR, "success id match: %02x\n", rxData[1]);
    SET_STATE(SENSOR_INITIALIZING);
    mTask.init_state = RESET_BMI160;
    registerAccGyroInterruptMode(ACC_GYRO_FIFO_INTERRUPTIBLE);
    registerAccGyroDriverFsm(bmi160Fsm, ARRAY_SIZE(bmi160Fsm));
err_out:
    return ret;
}
```

使用同步 API

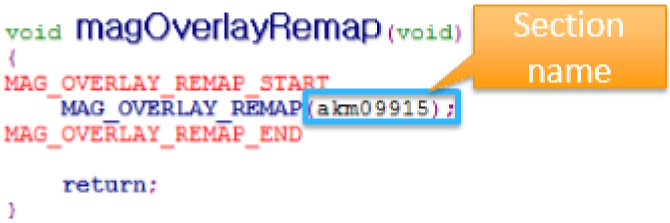
4) ADD overlay remap for load and init in overlay.c

vendor\mediatek\proprietary\tinysys\freertos\source\project\CM4_A\$PLATFORM\$PROJECT\cust\overlay\



```
void accGyroOverlayRemap(void)
{
ACC_GYRO_OVERLAY_REMAP_START
    ACC_GYRO_OVERLAY_REMAP(bmi160);
    ACC_GYRO_OVERLAY_REMAP(lsm6dsm);
ACC_GYRO_OVERLAY_REMAP_END

    return;
}
```

Load to sram and init, if success , goto return directly



```
void magOverlayRemap(void)
{
MAG_OVERLAY_REMAP_START
    MAG_OVERLAY_REMAP(akm09915);
MAG_OVERLAY_REMAP_END

    return;
}
```

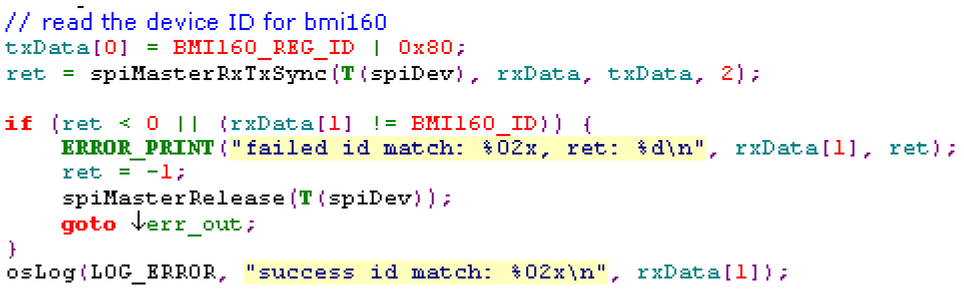
Section name

5) overlay 的 feature 开关

vendor\mediatek\proprietary\tinysys\freertos\source\project\CM4_A\$PLATFORM\$PROJECT\Projectconfig.mk

```
CFG_OVERLAY_INIT_SUPPORT = yes
CFG_OVERLAY_DEBUG_SUPPORT = yes
```

SPI AP



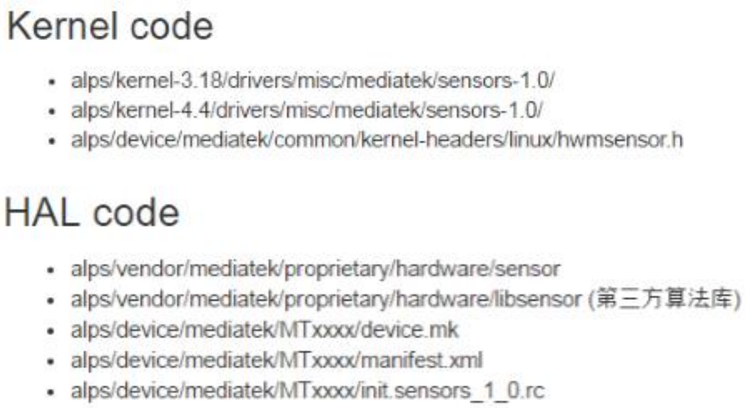
```
// read the device ID for bmi160
txData[0] = BMI160_REG_ID | 0x80;
ret = spiMasterRxTxSync(T(spiDev), rxData, txData, 2);

if (ret < 0 || (rxData[1] != BMI160_ID)) {
    ERROR_PRINT("failed id match: %02x, ret: %d\n", rxData[1], ret);
    ret = -1;
    spiMasterRelease(T(spiDev));
    goto ↓err_out;
}
osLog(LOG_ERROR, "success id match: %02x\n", rxData[1]);
```

CHRE API

```
 // perform soft reset and wait for 100ms
 SPI_WRITE(BMI160_REG_CMD, 0xb6, 100000);
 // dummy reads after soft reset, wait 100us
 SPI_READ(BMI160_REG_MAGIC, 1, &mTask.dataBuffer, 100);
spiBatchTxRx(&mTask.mode, sensorSpiCallback, &mTask, "sensorInit RESET" );
```

6 Build and Debug

6.1 Source Code Structure & File Description



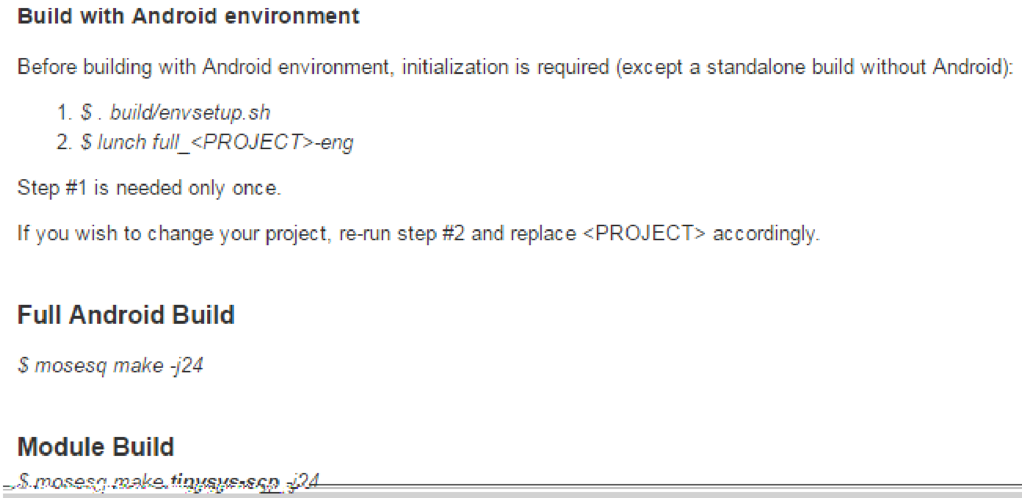
Kernel code

- alps/kernel-3.18/drivers/misc/mediatek/sensors-1.0/
- alps/kernel-4.4/drivers/misc/mediatek/sensors-1.0/
- alps/device/mediatek/common/kernel-headers/linux/hwmsensor.h

HAL code

- alps/vendor/mediatek/proprietary/hardware/sensor
- alps/vendor/mediatek/proprietary/hardware/libsensor (第三方算法库)
- alps/device/mediatek/MTxxxx/device.mk
- alps/device/mediatek/MTxxxx/manifest.xml
- alps/device/mediatek/MTxxxx/init.sensors_1_0.rc

6.2 How to build SCP



Build with Android environment

Before building with Android environment, initialization is required (except a standalone build without Android):

1. *$. build/envsetup.sh*
2. *$ lunch full_<PROJECT>-eng*

Step #1 is needed only once.

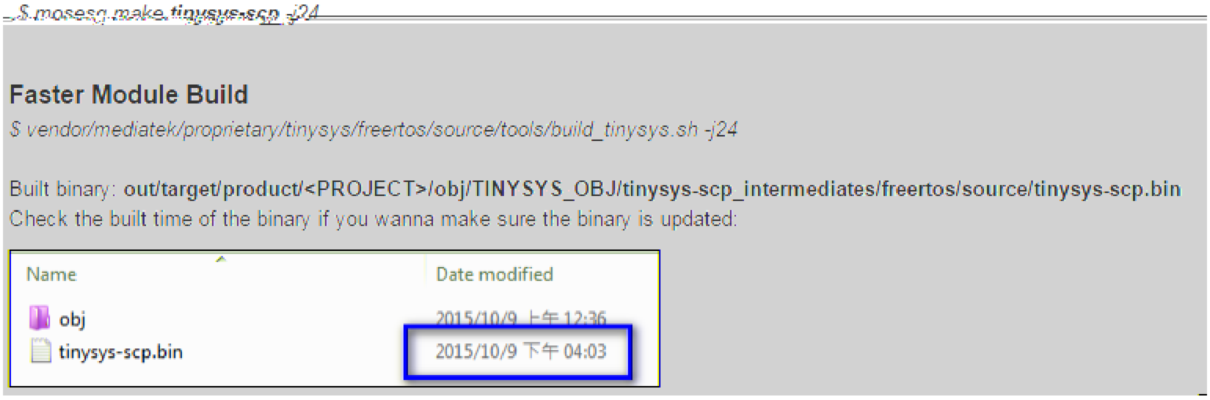
If you wish to change your project, re-run step #2 and replace <PROJECT> accordingly.

Full Android Build

$ mosesq make -j24

Module Build

$ mosesq make tinysys-scp -j24



Faster Module Build

$ vendor/mediatek/proprietary/tinysys/freertos/source/tools/build_tinysys.sh -j24

Built binary: **out/target/product/<PROJECT>/obj/TINYSYS_OBJ/tinysys-scp_intermediates/freertos/source/tinysys-scp.bin**

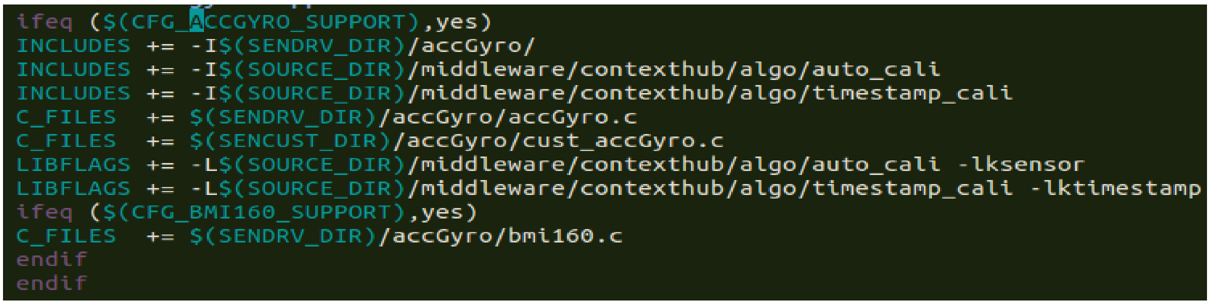
Check the built time of the binary if you wanna make sure the binary is updated:

Name	Date modified
obj	2015/10/9 上午 12:36
tinysys-scp.bin	2015/10/9 下午 04:03

6.3 How to build a sensor driver to binary

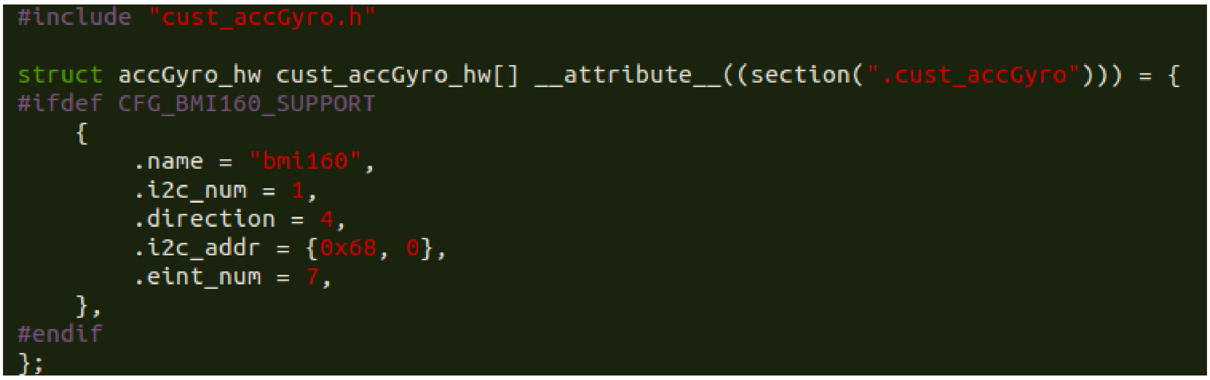
6.3.1 AccGyro

vendor/mediatek/proprietary/tinysys/freertos/source/project/CM4_A/mt6758/platform/feature_config/chre.mk



```
ifeq ($(CFG_ACCGYRO_SUPPORT),yes)
INCLUDES += -I$(SENDRV_DIR)/accGyro/
INCLUDES += -I$(SOURCE_DIR)/middleware/contexthub/algo/auto_cali
INCLUDES += -I$(SOURCE_DIR)/middleware/contexthub/algo/timestamp_cali
C_FILES  += $(SENDRV_DIR)/accGyro/accGyro.c
C_FILES  += $(SENCUST_DIR)/accGyro/cust_accGyro.c
LIBFLAGS += -L$(SOURCE_DIR)/middleware/contexthub/algo/auto_cali -lksensor
LIBFLAGS += -L$(SOURCE_DIR)/middleware/contexthub/algo/timestamp_cali -lktimestamp
ifeq ($(CFG_BMI160_SUPPORT),yes)
C_FILES  += $(SENDRV_DIR)/accGyro/bmi160.c
endif
endif
```

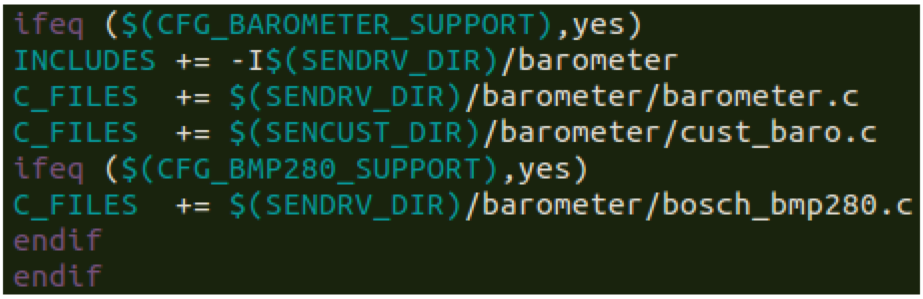
vendor/mediatek/proprietary/tinysys/freertos/source/project/CM4_A/mt6758/{PROJECT}/cust/accGyro/cust_accGyro.c



```
#include "cust_accGyro.h"

struct accGyro_hw cust_accGyro_hw[] __attribute__((section(".cust_accGyro"))) = {
#ifdef CFG_BMI160_SUPPORT
    {
        .name = "bmi160",
        .i2c_num = 1,
        .direction = 4,
        .i2c_addr = {0x68, 0},
        .eint_num = 7,
    },
#endif
};
```

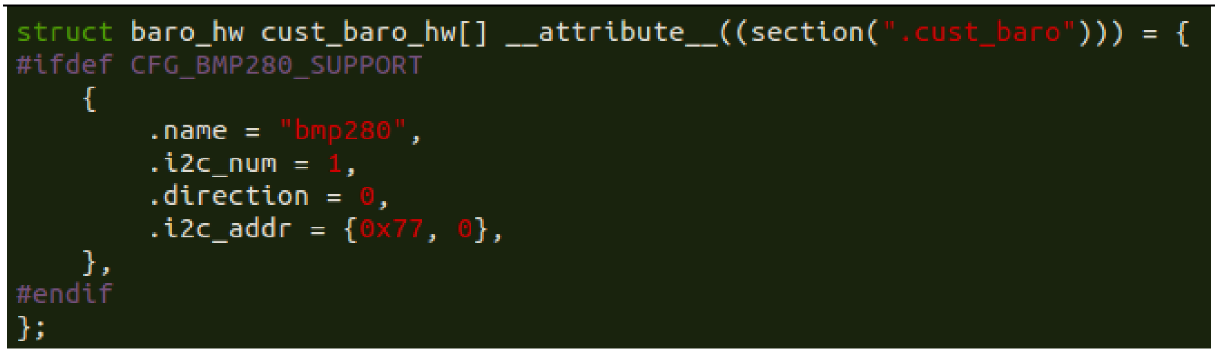
6.3.2 Barometer

vendor/mediatek/proprietary/tinysys/freertos/source/project/CM4_A/mt6758/platform/feature_config/chre.mk



```
ifeq ($(CFG_BAROMETER_SUPPORT),yes)
INCLUDES += -I$(SENDRV_DIR)/barometer
C_FILES  += $(SENDRV_DIR)/barometer/barometer.c
C_FILES  += $(SENCUST_DIR)/barometer/cust_baro.c
ifeq ($(CFG_BMP280_SUPPORT),yes)
C_FILES  += $(SENDRV_DIR)/barometer/bosch_bmp280.c
endif
endif
```

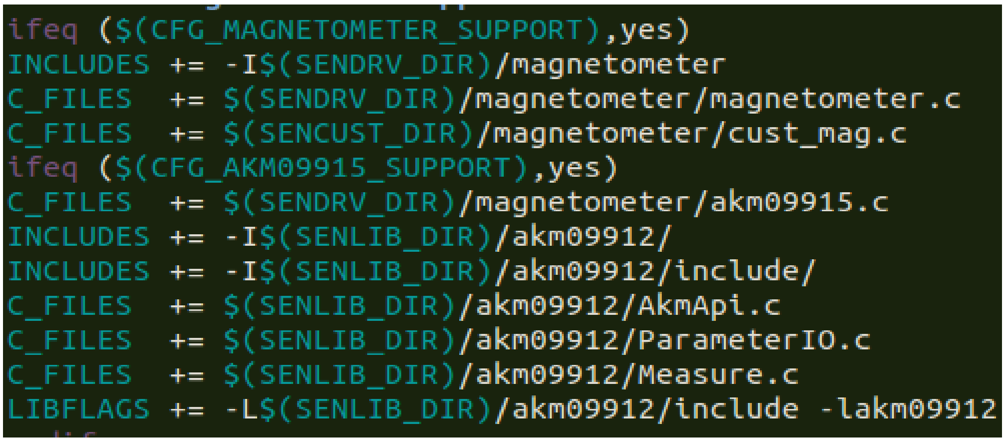
vendor/mediatek/proprietary/tinysys/freertos/source/project/CM4_A/mt6758/{PROJECT}/cust/barometer/cust_baro.c



```
struct baro_hw cust_baro_hw[] __attribute__((section(".cust_baro"))) = {
#ifdef CFG_BMP280_SUPPORT
    {
        .name = "bmp280",
        .i2c_num = 1,
        .direction = 0,
        .i2c_addr = {0x77, 0},
    },
#endif
};
```

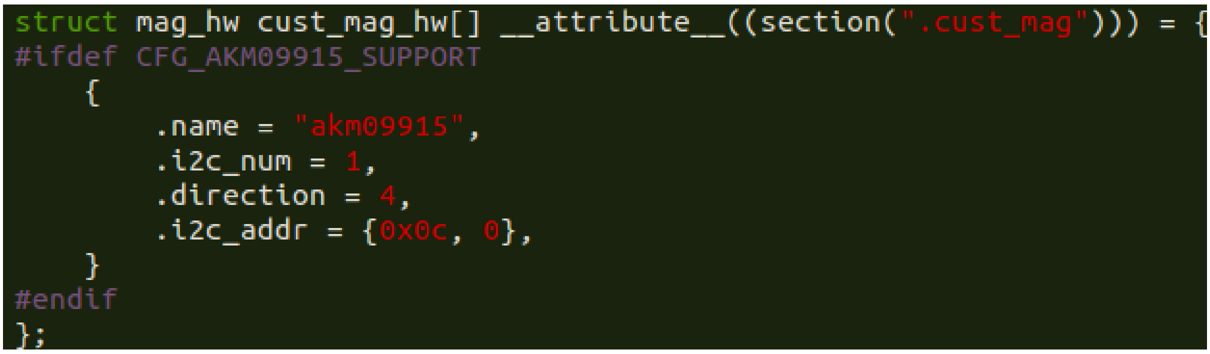
6.3.3 Magnetometer

vendor/mediatek/proprietary/tinysys/freertos/source/project/CM4_A/mt6758/platform/feature_config/chre.mk



```
ifeq ($(CFG_MAGNETOMETER_SUPPORT),yes)
INCLUDES += -I$(SENDRV_DIR)/magnetometer
C_FILES  += $(SENDRV_DIR)/magnetometer/magnetometer.c
C_FILES  += $(SENCUST_DIR)/magnetometer/cust_mag.c
ifeq ($(CFG_AKM09915_SUPPORT),yes)
C_FILES  += $(SENDRV_DIR)/magnetometer/akm09915.c
INCLUDES += -I$(SENLIB_DIR)/akm09912/
INCLUDES += -I$(SENLIB_DIR)/akm09912/include/
C_FILES  += $(SENLIB_DIR)/akm09912/AkmApi.c
C_FILES  += $(SENLIB_DIR)/akm09912/ParameterIO.c
C_FILES  += $(SENLIB_DIR)/akm09912/Measure.c
LIBFLAGS += -L$(SENLIB_DIR)/akm09912/include -lakm09912
```

vendor/mediatek/proprietary/tinysys/freertos/source/project/CM4_A/mt6758/{PROJECT}/cust/alsps/cust_mag.c



```
struct mag_hw cust_mag_hw[] __attribute__((section(".cust_mag"))) = {
#ifdef CFG_AKM09915_SUPPORT
    {
        .name = "akm09915",
        .i2c_num = 1,
        .direction = 4,
        .i2c_addr = {0x0c, 0},
    }
#endif
};
```

6.4 Build Option

6.4.1 Common build option

1. **Device config**

   Patch: /device/mediatekprojects/$project/ProjectConfig.mk

   MTK_TINYSYS_SCP_SUPPORT=yes

   MTK_SENSOR_SUPPORT =yes

```
CUSTOM_KERNEL_SENSORHUB=yes
MTK_SENSORS_1_0=yes
```

2. **Kernel config**

Path:
/kernel-4.4/arch/$TARGET_ARCH/configs/$project_defconfig
/kernel-4.4/arch/$TARGET_ARCH/configs/$project_debug_defconfig

```
 CONFIG_MTK_TINYSYS_SCP_SUPPORT=y
 CONFIG_MTK_HWMON=y
 CONFIG_MTK_SENSOR_SUPPORT=y
  CONFIG_CUSTOM_KERNEL_SENSORHUB=y
 CONFIG_NANOHUB_MTK_IPI=y
 CONFIG_MTK_SENSORS_1_0=y
 CONFIG_NANOHUB=y
 CONFIG_IIO=y
```

3. **SCP config**

Patch: /vendor/mediatek/proprietary/tinysys/freertos/source/Project/cm4_a/mt6758/
/projectconfig.mk

```
       CFG_CHRE_SUPPORT =yes
CFG_CONTEXTHUB_FW_SUPPORT =yes
```

4. **LK config**

Path: /vendor/mediatek/proprietary/bootable/bootloader/lk/project/$(project)

```
 MTK_TINYSYS_SCP_SUPPORT=no
  MTK_TINYSYS_SCP_SUPPORT=yes
```

6.4.2 Physical sensor build option

1. **Device config**

Patch: /device/mediatek/$project/ProjectConfig.mk

2. **Kernel config**

Path:
/kernel-4.4/arch/$TARGET_ARCH/configs/$project_defconfig
/kernel-4.4/arch/$TARGET_ARCH/configs/$project_debug_defconfig

3. **SCP config**

Patch: /vendor/mediatek/proprietary/tinysys/freertos/source/Project/cm4_a/mt6758/
/projectconfig.mk

6.4.3 Fusion sensors build option

1. **Device config**

Patch: /device/mediatek/$project/ProjectConfig.mk

2. **Kernel config**

Path:

/kernel-4.4/arch/$TARGET_ARCH/configs/$project_defconfig
/kernel-4.4/arch/$TARGET_ARCH/configs/$project_debug_defconfig

3. **SCP config**

Patch: /vendor/mediatek/proprietary/tinysys/freertos/source/Project/cm4_a/mt6758/ /projectconfig.mk

CFG_FUSION_SUPPORT=yes

6.4.4 Pedometer build option

1. **Device config**

Patch: /device/mediatek/$project/ProjectConfig.mk

2. **Kernel config**

Path:

/kernel-4.4/arch/$TARGET_ARCH/configs/$project_defconfig

/kernel-4.4/arch/$TARGET_ARCH/configs/$project_debug_defconfig

3. **SCP config**

Patch: /vendor/mediatek/proprietary/tinysys/freertos/source/Project/cm4_a/mt6758/ /projectconfig.mk

6.4.5 Situation & Gesture build option

1. **Device config**

Patch: /device/mediatek/$project/ProjectConfig.mk

2. **Kernel config**
   Path:
   /kernel-4.4/arch/$TARGET_ARCH/configs/$project_defconfig
   /kernel-4.4/arch/$TARGET_ARCH/configs/$project_debug_defconfig

3. **SCP config**
   Patch: /vendor/mediatek/proprietary/tinysys/freertos/source/Project/cm4_a/mt6758/ /projectconfig.mk

6.4.6 Activity build option

1. **Device config**
   Patch: /device/mediatek/$project/ProjectConfig.mk
   CUSTOM_KERNEL_ACTIVITY_SENSOR=yes
2. **Kernel config**
   Path:
   /kernel-4.4/arch/$TARGET_ARCH/configs/$project_defconfig
   /kernel-4.4/arch/$TARGET_ARCH/configs/$project_debug_defconfig
   CONFIG_CUSTOM_KERNEL_ACTIVITY_SENSOR=y
   CONFIG_MTK_ACTIVITYHUB=y
3. **SCP config**
   Patch: /vendor/mediatek/proprietary/tinysys/freertos/source/Project/cm4_a/mt6758/ /projectconfig.mk

```
CFG_ACTIVITY_NO_BARO_SUPPORT=yes
CFG_ACTIVITY_BARO_SUPPORT=yes
```

6.4.7 Vendor lib AKM M-sensor

SCP config

Patch: /vendor/mediatek/proprietary/tinysys/freertos/source/Project/cm4_a/mt67xx/
/projectconfig.mk

1. gyro

   CFG_AKM_FUSION_SUPPORT = yes
   CFG_FUSION_SUPPORT = no // MTK fusion virtual gyro AKMfusion
   CFG_AKM_ALGO_ALLINONE_SUPPORT = YES// gyro gyro
   gyro gyro

2. gyro gyro config

   CFG_AKM_FUSION_SUPPORT = yes
   CFG_FUSION_SUPPORT = no // MTK fusion virtual gyro AKMfusion
   CFG_AKM_ALGO_ALLINONE_SUPPORT = no // gyro gyro

   CFG_FAST_CALIBRATION_SUPPORT = yes
   gyro
   CFG_VIRTUAL_GYRO_SUPPORT = yes

6.4.8 SCP MTK fusion gyro

Patch: /vendor/mediatek/proprietary/tinysys/freertos/source/Project/cm4_a/mt67xx/
/projectconfig.mk

CFG_VIRTUAL_GYRO_SUPPORT = no // MTK virtual gyro
CFG_AKM_FUSION_SUPPORT = no
CFG_FUSION_SUPPORT = yes // MTK fusion

MTK fusion vendor

6.5 Debug

6.5.1 SCP Uart

vendor/mediatek/proprietary/tinysys/freertos/source/project/CM4_A/mt67xx/platform/platform.mk

```
CFG_UART_SUPPORT = yes
```

QA uart performance

6.5.2 SCP uart AP uart

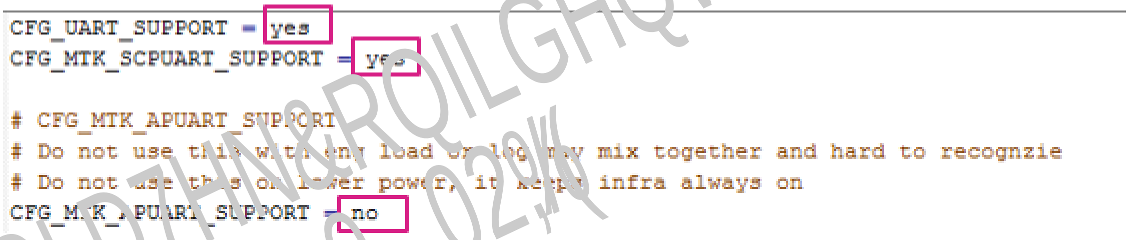
Enable by modify config

project/CM4_A/mt6771/platform/platform.mk

- Warning

  DO NOT apply this change to ENG build, because AP and SCP log will mix together and hard to recognize.

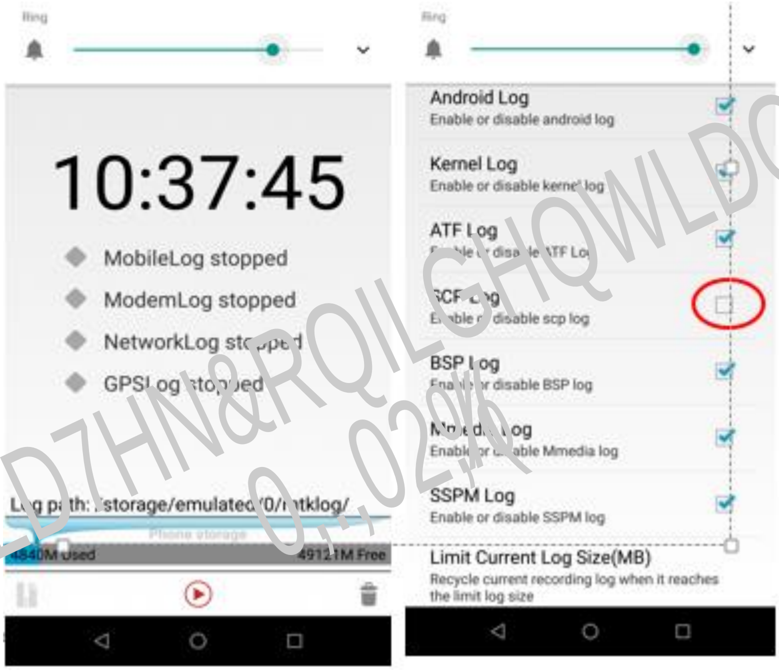
DO NOT use this when measure suspend power, it keeps infra always on.



```
CFG_UART_SUPPORT = yes
CFG_MTK_SCPUART_SUPPORT = yes

# CFG_MTK_APUART_SUPPORT
# Do not use this with eng load or log may mix together and hard to recognzie
# Do not use this on lower power, it keeps infra always on
CFG_MTK_APUART_SUPPORT = no
```

6.5.3 usb SCP log

1. Make sure mobile log (SCP part) is disabled
   1) All mobile log disable
   2) Or Disable SCP log



10:37:45

- MobileLog stopped
- ModemLog stopped
- NetworkLog stopped
- GPSLog stopped

Log path: /storage/emulated/0/mtklog/
Phone storage
4840M Used 49121M Free

Android Log
Enable or disable android log ☑

Kernel Log
Enable or disable kernel log ☑

ATF Log
Enable or disable ATF Log ☑

SCP Log
Enable or disable scp log □

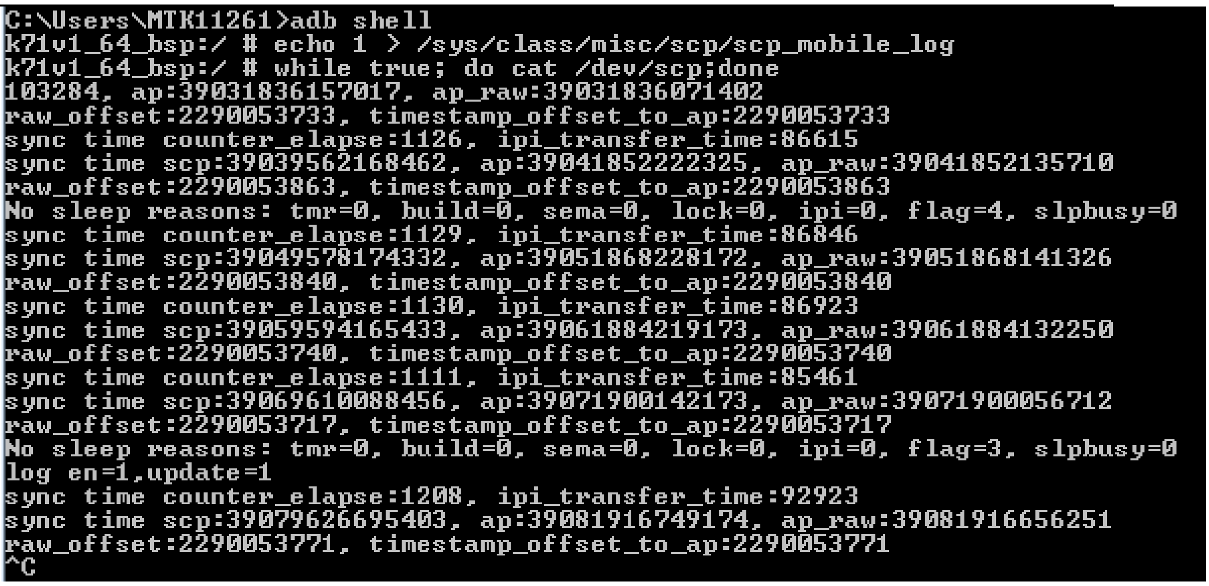
BSP Log
Enable or disable BSP log ☑

Mmedia Log
Enable or disable Mmedia log ☑

SSPM Log
Enable or disable SSPM log ☑

Limit Current Log Size(MB)
Recycle current recording log when it reaches the limit log size

2. Enter adb shell and then
   1) echo 1 > /sys/class/misc/scp/scp_mobile_log
   2) while true; do cat /dev/scp;done



```
C:\Users\MTK11261>adb shell
k71v1_64_bsp:/ # echo 1 > /sys/class/misc/scp/scp_mobile_log
k71v1_64_bsp:/ # while true; do cat /dev/scp;done
103284, ap:39031836157017, ap_raw:39031836071402
raw_offset:2290053733, timestamp_offset_to_ap:2290053733
sync time counter_elapse:1126, ipi_transfer_time:86615
sync time scp:39039562168462, ap:39041852222325, ap_raw:39041852135710
raw_offset:2290053863, timestamp_offset_to_ap:2290053863
No sleep reasons: tmr=0, build=0, sema=0, lock=0, ipi=0, flag=4, slpbusy=0
sync time counter_elapse:1129, ipi_transfer_time:86846
sync time scp:39049578174332, ap:39051868228172, ap_raw:39051868141326
raw_offset:2290053840, timestamp_offset_to_ap:2290053840
sync time counter_elapse:1130, ipi_transfer_time:86923
sync time scp:39059594165433, ap:39061884219173, ap_raw:39061884132250
raw_offset:2290053740, timestamp_offset_to_ap:2290053740
sync time counter_elapse:1111, ipi_transfer_time:85461
sync time scp:39069610088456, ap:39071900142173, ap_raw:39071900056712
raw_offset:2290053717, timestamp_offset_to_ap:2290053717
No sleep reasons: tmr=0, build=0, sema=0, lock=0, ipi=0, flag=3, slpbusy=0
log en=1,update=1
sync time counter_elapse:1208, ipi_transfer_time:92923
sync time scp:39079626695403, ap:39081916749174, ap_raw:39081916656251
raw_offset:2290053771, timestamp_offset_to_ap:2290053771
^C
```

6.5.4 SCP open EE DB

- 1 . AEE
  - -

    3 3
- 2. SCP db dump
  - adb command:
  - 1) adb shell cat /sys/class/misc/scp/scp_A_db_test (dumping SCP A db)
  - 2 db

    sdcard/mtklog/aee_exp/data/aee_exp/

6.5.5 SCP AP

Force enable KE when SCP EE occur

Default Status: DISABLE

How to switch: write the control node to turn on/off

- How to use explain in next page

When Enable:

Reset scope: Whole system (KE)

Debug info: Full RAM Dump (takes a long time) and mobile log

- db = db.xx.EE & db.fatal.xx.KE

When Disable:

Reset scope: SCP only (EE)

Debug info: SCP db and mobile log

- db = db.xx.EE

1. Control node:

Path: /sys/class/misc/scp/scp_ee_force_ke

Enable

- echo 1 > /sys/class/misc/scp/scp_ee_force_ke

Disable

- echo 0 > /sys/class/misc/scp/scp_ee_force_ke

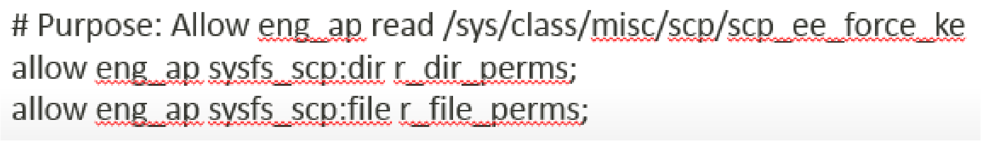
2. Selinux

Must allow to access the path: /sys/class/misc/scp/scp_ee_force_ke

eng_app access sysfs_scp

device/mediatek/sepolicy/basic/non_plat/eng_app.te

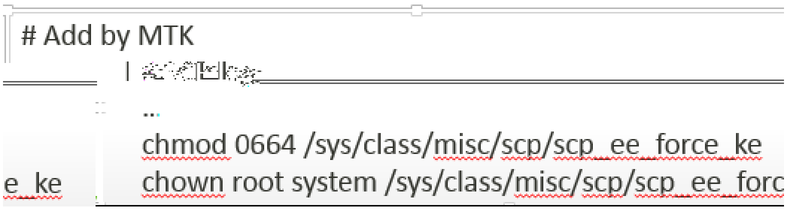
xxx.te eng_app



```
# Purpose: Allow eng_ap read /sys/class/misc/scp/scp_ee_force_ke
allow eng_ap sysfs_scp:dir r_dir_perms;
allow eng_ap sysfs_scp:file r_file_perms;
```

3. Property

device/mediatek/mt6771/init.mt6771.rc



```
# Add by MTK
    [illegible]
    ...
    chmod 0664 /sys/class/misc/scp/scp_ee_force_ke
    chown root system /sys/class/misc/scp/scp_ee_forc
e_ke
```

6.5.6 Dynamic AP/SCP UART Switch

- Default enable AP uart support, switch with Fastboot cmd line
- Warning!! Limitation & side effect

extra code size require(+560 bytes)

Must disable AP uart log self

Timing impact, for debug only, can not use it on stress test

Power impact

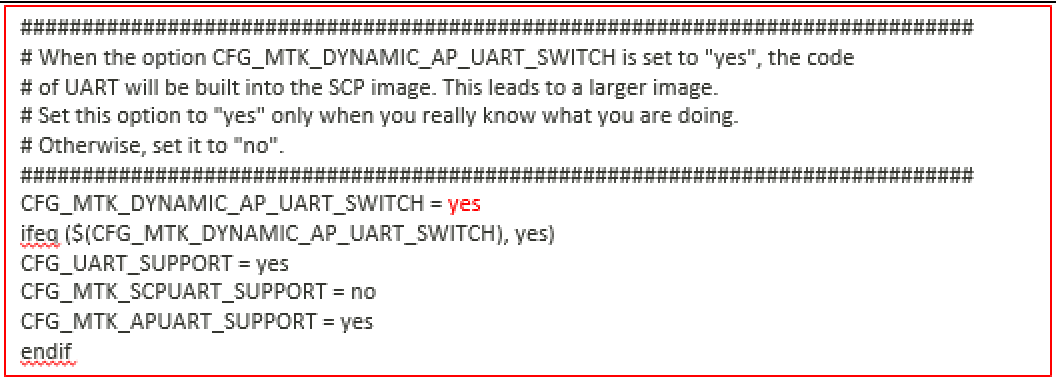
(DO NOT apply this change to ENG build, because AP and SCP log will mix together and hard to recognize)

(DO NOT use this when measure suspend power, it keeps infra always on

1. How to use:

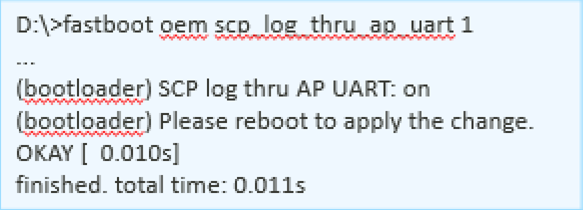
Set CFG_MTK_DYNAMIC_AP_UART_SWITCH = yes

(@ project/CM4_A/mt6771/platform/platform.mk)



```
#####################################################################################
# When the option CFG_MTK_DYNAMIC_AP_UART_SWITCH is set to "yes", the code
# of UART will be built into the SCP image. This leads to a larger image.
# Set this option to "yes" only when you really know what you are doing.
# Otherwise, set it to "no".
#####################################################################################
CFG_MTK_DYNAMIC_AP_UART_SWITCH = yes
ifeq ($(CFG_MTK_DYNAMIC_AP_UART_SWITCH), yes)
CFG_UART_SUPPORT = yes
CFG_MTK_SCPUART_SUPPORT = no
CFG_MTK_APUART_SUPPORT = yes
endif
```

2. How to dynamic switch
   1) enter lk fastboot
      use adb reboot bootloader (@adb shell)
      or booting menu (Power key + Volume- boot up) -> fastboot
   2) switch with fastboot cmd
      enable: fastboot oem scp_log_thru_ap_uart 1
      disable: fastboot oem scp_log_thru_ap_uart 0

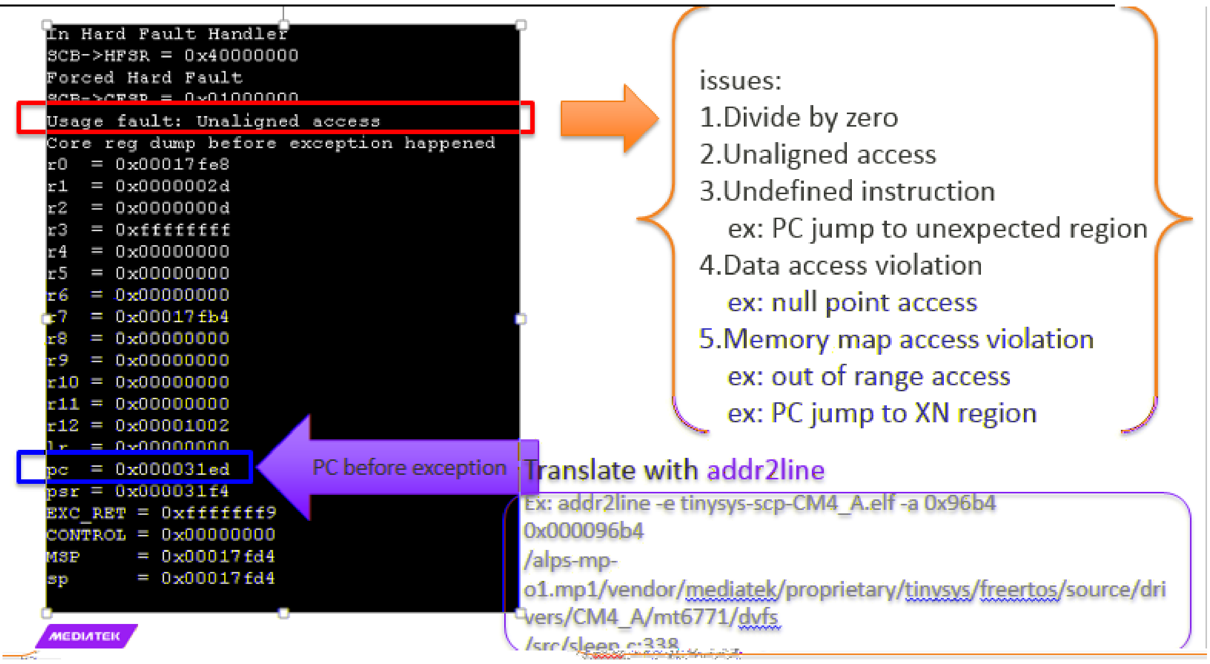


```
D:\>fastboot oem scp_log_thru_ap_uart 1
...
(bootloader) SCP log thru AP UART: on
(bootloader) Please reboot to apply the change.
OKAY [  0.010s]
finished. total time: 0.011s
```

   3) reboot (remember disable AP uart)

6.5.7 SCP Exception debug

1. **How to get exception log**
   1) From uart/mobile log
   2) From db
      i. Extract SCP EE DB, it can see SYS_SCP_DUMP
2. **Exception category introduction**
   If get exception log already



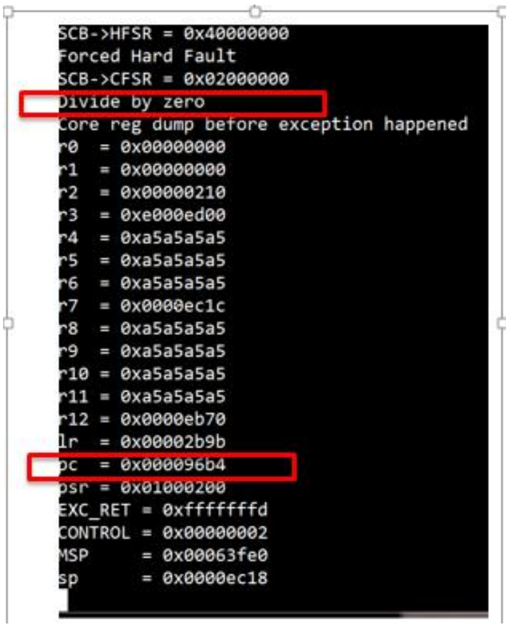
```
In Hard Fault Handler
SCB->HFSR = 0x40000000
Forced Hard Fault
SCB->CFSR = 0x01000000
Usage fault: Unaligned access
Core reg dump before exception happened
r0  = 0x00017fe8
r1  = 0x0000002d
r2  = 0x0000000d
r3  = 0xffffffff
r4  = 0x00000000
r5  = 0x00000000
r6  = 0x00000000
r7  = 0x00017fb4
r8  = 0x00000000
r9  = 0x00000000
r10 = 0x00000000
r11 = 0x00000000
r12 = 0x00001002
lr  = 0x00000000
pc  = 0x000031ed
psr = 0x000031f4
EXC_RET = 0xfffffff9
CONTROL = 0x00000000
MSP     = 0x00017fd4
sp      = 0x00017fd4
```

issues:
1.Divide by zero
2.Unaligned access
3.Undefined instruction
ex: PC jump to unexpected region
4.Data access violation
ex: null point access
5.Memory map access violation
ex: out of range access
ex: PC jump to XN region

Translate with addr2line

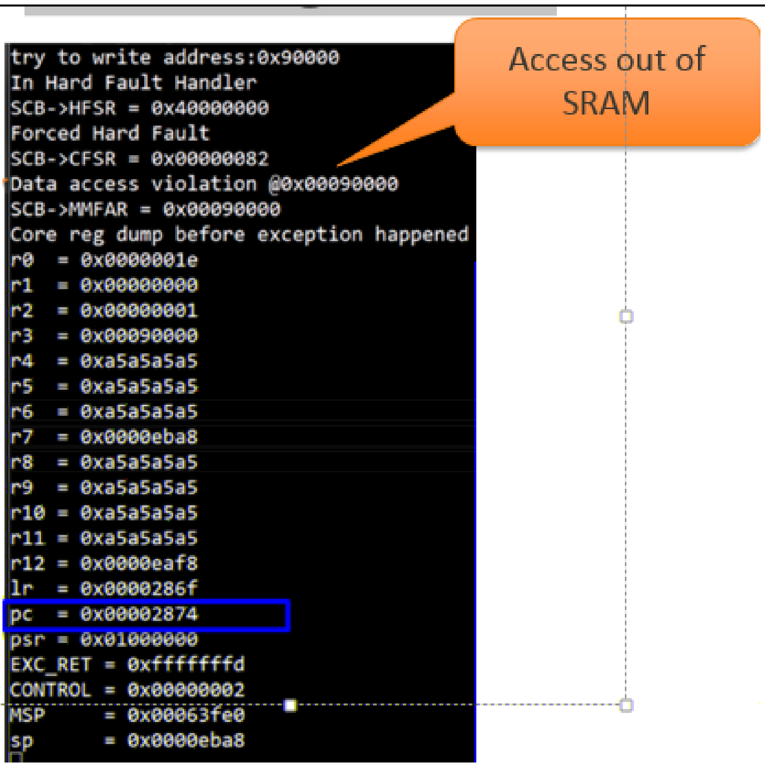
```
Ex: addr2line -e tinysys-scp-CM4_A.elf -a 0x96b4
0x000096b4
/alps-mp-
o1.mp1/vendor/mediatek/proprietary/tinysys/freertos/source/dri
vers/CM4_A/mt6771/dvfs
/src/sleep.c:338
```

1) Divde by zero



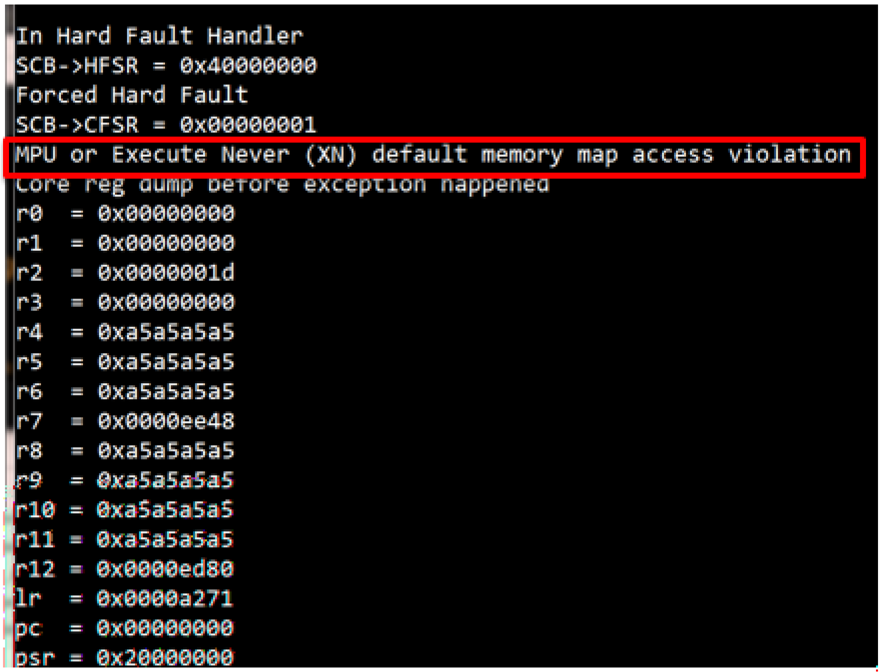
```
SCB->HFSR = 0x40000000
Forced Hard Fault
SCB->CFSR = 0x02000000
Divide by zero
Core reg dump before exception happened
r0  = 0x00000000
r1  = 0x00000000
r2  = 0x00000210
r3  = 0xe000ed00
r4  = 0xa5a5a5a5
r5  = 0xa5a5a5a5
r6  = 0xa5a5a5a5
r7  = 0x0000ec1c
r8  = 0xa5a5a5a5
r9  = 0xa5a5a5a5
r10 = 0xa5a5a5a5
r11 = 0xa5a5a5a5
r12 = 0x0000eb70
lr  = 0x00002b9b
pc  = 0x000096b4
psr = 0x01000200
EXC_RET = 0xfffffffd
CONTROL = 0x00000002
MSP     = 0x00063fe0
sp      = 0x0000ec18
```

2) Out of range access



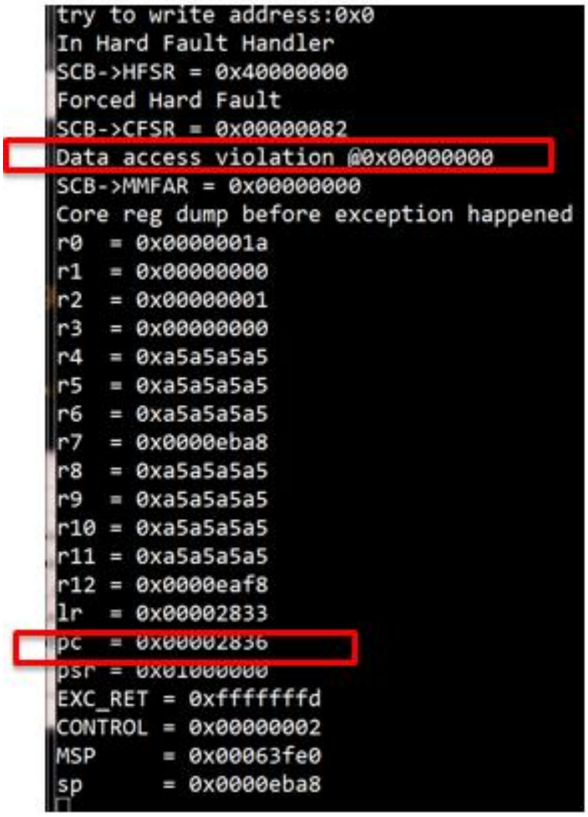
```
try to write address:0x90000
In Hard Fault Handler
SCB->HFSR = 0x40000000
Forced Hard Fault
SCB->CFSR = 0x00000082
Data access violation @0x00090000
SCB->MMFAR = 0x00090000
Core reg dump before exception happened
r0  = 0x0000001e
r1  = 0x00000000
r2  = 0x00000001
r3  = 0x00090000
r4  = 0xa5a5a5a5
r5  = 0xa5a5a5a5
r6  = 0xa5a5a5a5
r7  = 0x0000eba8
r8  = 0xa5a5a5a5
r9  = 0xa5a5a5a5
r10 = 0xa5a5a5a5
r11 = 0xa5a5a5a5
r12 = 0x0000eaf8
lr  = 0x0000286f
pc  = 0x00002874
psr = 0x01000000
EXC_RET = 0xfffffffd
CONTROL = 0x00000002
MSP     = 0x00063fe0
sp      = 0x0000eba8
```

3) Jump to XN region



```
In Hard Fault Handler
SCB->HFSR = 0x40000000
Forced Hard Fault
SCB->CFSR = 0x00000001
MPU or Execute Never (XN) default memory map access violation
Core reg dump before exception happened
r0  = 0x00000000
r1  = 0x00000000
r2  = 0x0000001d
r3  = 0x00000000
r4  = 0xa5a5a5a5
r5  = 0xa5a5a5a5
r6  = 0xa5a5a5a5
r7  = 0x0000ee48
r8  = 0xa5a5a5a5
r9  = 0xa5a5a5a5
r10 = 0xa5a5a5a5
r11 = 0xa5a5a5a5
r12 = 0x0000ed80
lr  = 0x0000a271
pc  = 0x00000000
psr = 0x20000000
```

4) Null pointer access



```
try to write address:0x0
In Hard Fault Handler
SCB->HFSR = 0x40000000
Forced Hard Fault
SCB->CFSR = 0x00000082
Data access violation @0x00000000
SCB->MMFAR = 0x00000000
Core reg dump before exception happened
r0  = 0x0000001a
r1  = 0x00000000
r2  = 0x00000001
r3  = 0x00000000
r4  = 0xa5a5a5a5
r5  = 0xa5a5a5a5
r6  = 0xa5a5a5a5
r7  = 0x0000eba8
r8  = 0xa5a5a5a5
r9  = 0xa5a5a5a5
r10 = 0xa5a5a5a5
r11 = 0xa5a5a5a5
r12 = 0x0000eaf8
lr  = 0x00002833
pc  = 0x00002836
psr = 0x01000000
EXC_RET = 0xfffffffd
CONTROL = 0x00000002
MSP     = 0x00063fe0
sp      = 0x0000eba8
```

3. **Debug ram dump with gdb**

GDB download

Get android ndk

https://developer.android.com/ndk/downloads/index.html

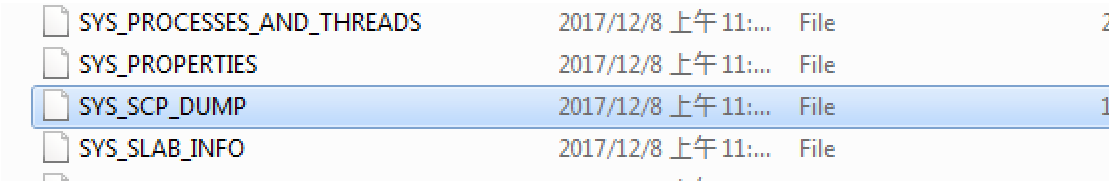
You can find it in prebuild folder

Ex: prebuild/linux-x86_64/bin/gdb

1) Get ramdump

i. Get SCP EE DB and extract it, it can see SYS_SCP_DUMP

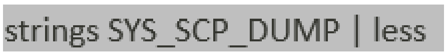
ii. If size of SYS_SCP_DUMP is 0, this issue is probably not an SCP issue



SYS_PROCESSES_AND_THREADS	2017/12/8 上午 11:...	File	2
SYS_PROPERTIES	2017/12/8 上午 11:...	File	
SYS_SCP_DUMP	2017/12/8 上午 11:...	File	1
SYS_SLAB_INFO	2017/12/8 上午 11:...	File	

2) See last log

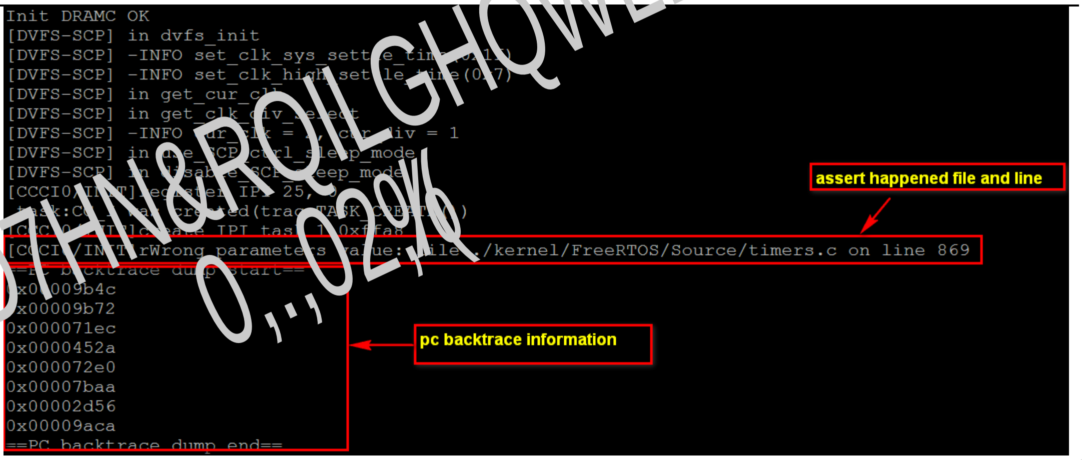
Run command, strings, to parse the ram dump:



```
strings SYS_SCP_DUMP | less
```

FreeRTOS/Source/timers.c:869.

We can know the failed point. The PC backtrace is also helpful. You can use addr2line to lo locate the problem



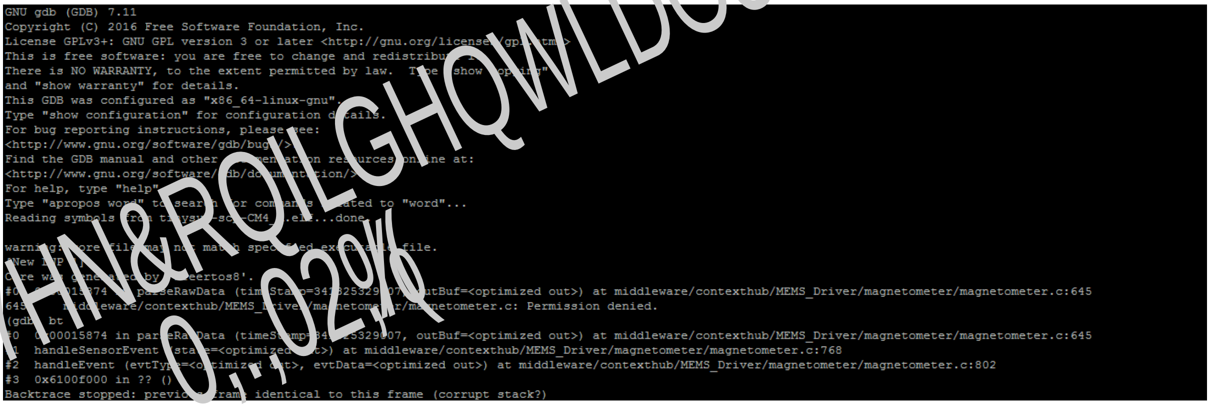
```
Init DRAMC OK
[DVFS-SCP] in dvfs_init
[DVFS-SCP] -INFO set_clk_sys_settle_time(0x11)
[DVFS-SCP] -INFO set_clk_high_settle_time(0x7)
[DVFS-SCP] in get_cur_clk
[DVFS-SCP] in get_clk_div_select
[DVFS-SCP] -INFO cur_clk = 2, cur_div = 1
[DVFS-SCP] in use_SCP_ctrl_sleep_mode
[DVFS-SCP] in disable_SCP_sleep_mode
[CCCI0/INIT]register IPI 25, 0
 task:CCCI was created(trace TASK_CREATE())
[CCCI0/INIT]create IPI task 1 0xffa8
[CCCI0/INIT]rWrong parameters value:file ./kernel/FreeRTOS/Source/timers.c on line 869
==PC backtrace dump start==
0x00009b4c
0x00009b72
0x000071ec
0x0000452a
0x000072e0
0x00007baa
0x00002d56
0x00009aca
==PC backtrace dump end==
```

3) **Debug ram dump with gdb**

[Shell]$ ./gdb tinysys-scp-CM4_A.elf SYS_SCP_DUMP

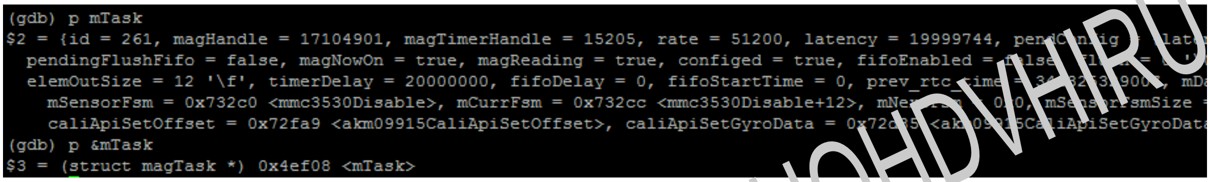
Further cmds please reference GDB guide

Back trace: bt



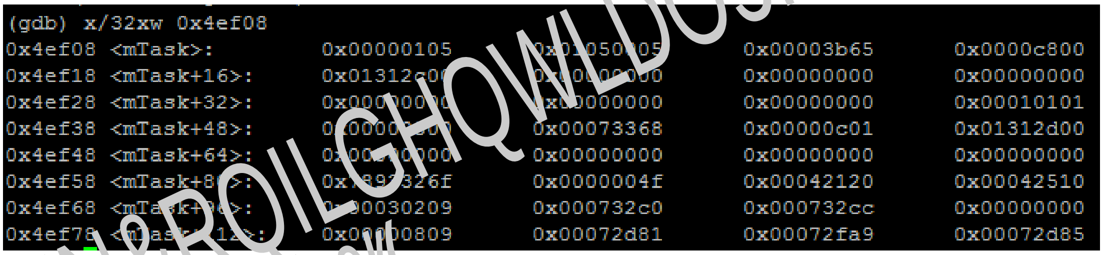
```
GNU gdb (GDB) 7.11
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from tinysys-scp-CM4_A.elf...done.

warning: core file may not match specified executable file.
[New LWP 1]
Core was generated by `freertos8'.
#0  0x00015874 in parseRawData (timeStamp=341825329007, outBuf=<optimized out>) at middleware/contexthub/MEMS_Driver/magnetometer/magnetometer.c:645
645     middleware/contexthub/MEMS_Driver/magnetometer/magnetometer.c: Permission denied.
(gdb) bt
#0  0x00015874 in parseRawData (timeStamp=341825329007, outBuf=<optimized out>) at middleware/contexthub/MEMS_Driver/magnetometer/magnetometer.c:645
#1  handleSensorEvent (state=<optimized out>) at middleware/contexthub/MEMS_Driver/magnetometer/magnetometer.c:768
#2  handleEvent (evtType=<optimized out>, evtData=<optimized out>) at middleware/contexthub/MEMS_Driver/magnetometer/magnetometer.c:802
#3  0x6100f000 in ?? ()
Backtrace stopped: previous frame identical to this frame (corrupt stack?)
```

GDB Basic cmds: p variable



```
(gdb) p mTask
$2 = {id = 261, magHandle = 17104901, magTimerHandle = 15205, rate = 51200, latency = 19999744, pendConfig = {late
  pendingFlushFifo = false, magNowOn = true, magReading = true, configed = true, fifoEnabled = false, flush = 0 '\
  elemOutSize = 12 '\f', timerDelay = 20000000, fifoDelay = 0, fifoStartTime = 0, prev_rtc_time = 341825329007, mDa
    mSensorFsm = 0x732c0 <mmc3530Disable>, mCurrFsm = 0x732cc <mmc3530Disable+12>, mNextFsm = 0x0, mSensorFsmSize =
    caliApiSetOffset = 0x72fa9 <akm09915CaliApiSetOffset>, caliApiSetGyroData = 0x72d85 <akm09915CaliApiSetGyroData
(gdb) p &mTask
$3 = (struct magTask *) 0x4ef08 <mTask>
```

Dump memory : x/FMT address



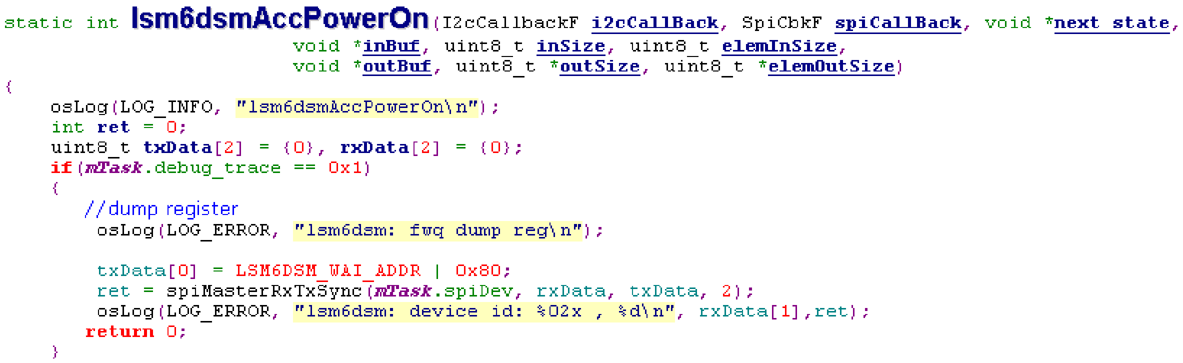
```
(gdb) x/32xw 0x4ef08
0x4ef08 <mTask>:        0x00000105      0x01050005      0x00003b65      0x0000c800
0x4ef18 <mTask+16>:     0x01312c00      0x00000000      0x00000000      0x00000000
0x4ef28 <mTask+32>:     0x00000000      0x00000000      0x00000000      0x00010101
0x4ef38 <mTask+48>:     0x00000c00      0x00073368      0x00000c01      0x01312d00
0x4ef48 <mTask+64>:     0x00000000      0x00000000      0x00000000      0x00000000
0x4ef58 <mTask+80>:     0x7893326f      0x0000004f      0x00042120      0x00042510
0x4ef68 <mTask+96>:     0x00030209      0x000732c0      0x000732cc      0x00000000
0x4ef78 <mTask+112>:    0x00000809      0x00072d81      0x00072fa9      0x00072d85
```

6.5.8 Sensor driver debug

1. **SPI driver debug**

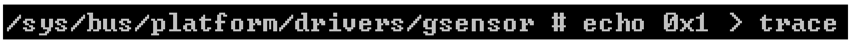
CHRE SPI SPI driver debug dump register sensor flow dump

driver power on flow debug trace debug trace 0x1,0 , debug_trace SPI api dump register



```
static int lsm6dsmAccPowerOn(I2cCallbackF i2cCallBack, SpiCbkF spiCallBack, void *next_state,
                             void *inBuf, uint8_t inSize, uint8_t elemInSize,
                             void *outBuf, uint8_t *outSize, uint8_t *elemOutSize)
{
    osLog(LOG_INFO, "lsm6dsmAccPowerOn\n");
    int ret = 0;
    uint8_t txData[2] = {0}, rxData[2] = {0};
    if(mTask.debug_trace == 0x1)
    {
      //dump register
       osLog(LOG_ERROR, "lsm6dsm: fwq dump reg\n");

       txData[0] = LSM6DSM_WAI_ADDR | 0x80;
       ret = spiMasterRxTxSync(mTask.spiDev, rxData, txData, 2);
       osLog(LOG_ERROR, "lsm6dsm: device id: %02x , %d\n", rxData[1],ret);
      return 0;
    }
```

adb

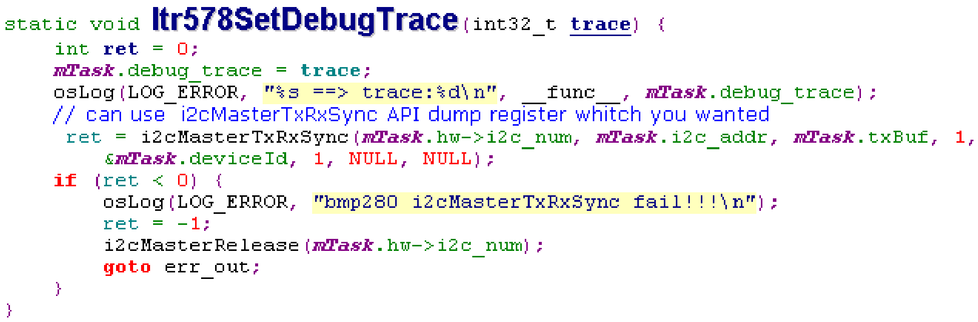


```
/sys/bus/platform/drivers/gsensor # echo 0x1 > trace
```

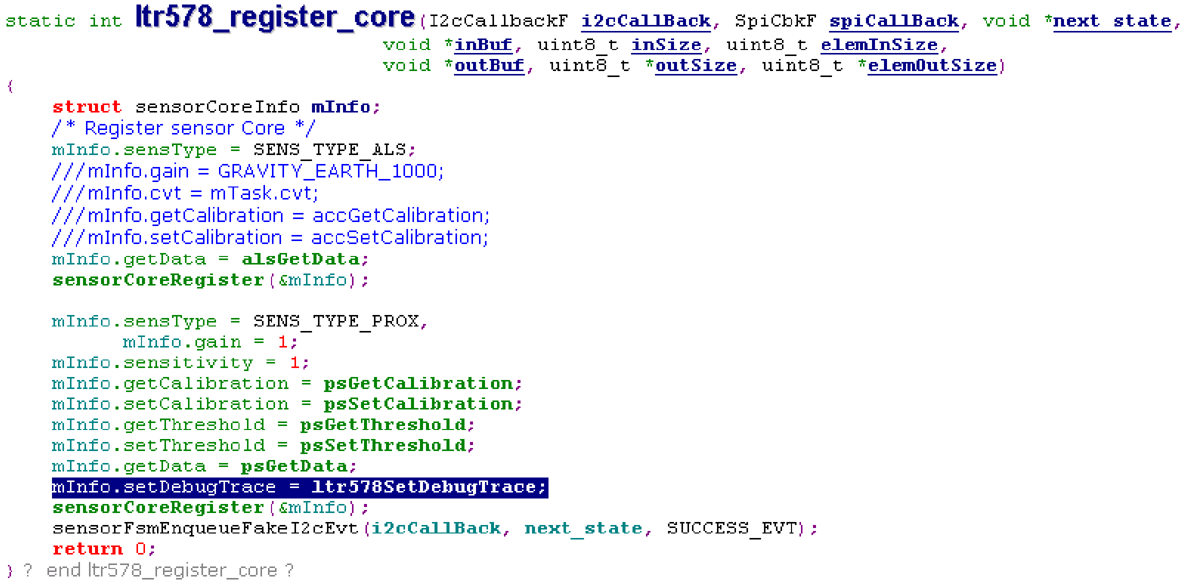
dump return 0 control flow sensor

2. **I2c driver bug**

call back i2c API



```
static void ltr578SetDebugTrace(int32_t trace) {
    int ret = 0;
    mTask.debug_trace = trace;
    osLog(LOG_ERROR, "%s ==> trace:%d\n", __func__, mTask.debug_trace);
    // can use  i2cMasterTxRxSync API dump register whitch you wanted
     ret = i2cMasterTxRxSync(mTask.hw->i2c_num, mTask.i2c_addr, mTask.txBuf, 1,
        &mTask.deviceId, 1, NULL, NULL);
    if (ret < 0) {
        osLog(LOG_ERROR, "bmp280 i2cMasterTxRxSync fail!!!\n");
        ret = -1;
        i2cMasterRelease(mTask.hw->i2c_num);
        goto err_out;
    }
}
```



```
static int ltr578_register_core(I2cCallbackF i2cCallBack, SpiCbkF spiCallBack, void *next_state,
                                void *inBuf, uint8_t inSize, uint8_t elemInSize,
                                void *outBuf, uint8_t *outSize, uint8_t *elemOutSize)
{
    struct sensorCoreInfo mInfo;
    /* Register sensor Core */
    mInfo.sensType = SENS_TYPE_ALS;
    ///mInfo.gain = GRAVITY_EARTH_1000;
    ///mInfo.cvt = mTask.cvt;
    ///mInfo.getCalibration = accGetCalibration;
    ///mInfo.setCalibration = accSetCalibration;
    mInfo.getData = alsGetData;
    sensorCoreRegister(&mInfo);

    mInfo.sensType = SENS_TYPE_PROX,
          mInfo.gain = 1;
    mInfo.sensitivity = 1;
    mInfo.getCalibration = psGetCalibration;
    mInfo.setCalibration = psSetCalibration;
    mInfo.getThreshold = psGetThreshold;
    mInfo.setThreshold = psSetThreshold;
    mInfo.getData = psGetData;
    mInfo.setDebugTrace = ltr578SetDebugTrace;
    sensorCoreRegister(&mInfo);
    sensorFsmEnqueueFakeI2cEvt(i2cCallBack, next_state, SUCCESS_EVT);
    return 0;
} ? end ltr578_register_core ?
```