

# [FAQ12441][NvRam] Normal mode下，sensor校准数据，恢复出厂设置，不丢失

## [DESCRIPTION]

目前支持的ACC，GYRO，PS 三种sensor的校准数据，是保存在nvram中，并且也有加入到nvram binregion中，可以实现恢复出厂设置后，sensor校准数据的恢复。

但是nvram binregion中的sensor数据是要在产线上做校准并保存的，而且永远都是这个固定值。

但是客户实际上是在normal mode下，需要随时做校准，而且需要保存最近的校准值不丢失。

因nvram binregion和sensor daemon源码不开放，所以无法将sensor nvram保存在pro\_info中。

现提供对sensor nvram的backup的功能，来实现sensor数据的保存。

## [SOLUTION]

### 1. 确认打开 MTK\_PRODUCT\_INFO\_SUPPORT

修改alps/mediatek/config/\$project/ProjectConfig.mk：

MTK\_PRODUCT\_INFO\_SUPPORT=yes

MTK\_PRODUCT\_INFO\_SUPPORT 也必须存在于 AUTO\_ADD\_GLOBAL\_DEFINE\_BY\_NAME 中。

### 2. 定义所需要保存的结构体

这里定义了 ACC，GYRO，PS 三种结构体到一个backup的数据中，可以根据需要做添加或删除，文件名字可以自定义，或者添加到已有的文件中。

\mediatek\custom\project\cgen\cfgfileinc\CFG\_XXXX\_File.h

//the new record for the hwmon sensor backup data

#defineCFG\_HWMON\_BACKUP\_RESERVED\_SIZE(1024-CFG\_FILE\_HWMON\_ACC\_REC\_SIZE-CFG\_FILE\_HWMON\_GYRO\_REC\_SIZE-CFG\_FILE\_HWMON\_PS\_REC\_SIZE)

typedef struct{

NVRAM\_HWMON\_ACC\_STRUCThwmon\_acc;

NVRAM\_HWMON\_GYRO\_STRUCThwmon\_gyro;

NVRAM\_HWMON\_PS\_STRUCThwmon\_ps;

```

unsigned char reserved[CFG_HWMON_BACKUP_RESERVED_SIZE];
}HWMON_BACKUP_STRUCT;

#defineCFG_FILE_HWMON_BACKUP_SIZEsizeof(HWMON_BACKUP_STRUCT)
#defineCFG_FILE_HWMON_BACKUP_TOTAL1

```

//此结构体数据的default值

```

\mediatek\custom\${project}\cgen\cfgdefault\CFG_XXXX_Default.h

```

```

HWMON_BACKUP_STRUCT st_hwmon_backup_default =
{
//acc
{0},
//gyro
{0},
//ps
{0},
//reserved
{0}
};

```

### 3. 添加新的nvram数据的定义

```

\mediatek\custom\${project}\cgen\inc\Custom_NvRam_LID.h

```

//添加LID定义

```

AP_CFG_RDCL_HWMON_BACKUP_LID, //new add
AP_CFG_CUSTOM_FILE_MAX_LID,
} CUSTOM_CFG_FILE_LID;

```

//添加LID版本信息

```

#define AP_CFG_RDCL_HWMON_BACKUP_LID_VERNO "000"

```

//注意此table中添加的位置，需要和 CUSTOM\_CFG\_FILE\_LID 中的位置对应。

```

\mediatek\custom\${project}\cgen\inc\CFG_file_info_custom.h

```

```

const TCFG_FILE g_akCFG_File_Custom[] =
{

....

{ "/data/nvram/APCFG/APRDCL/HWMON_BACKUP",
VER(AP_CFG_RDCL_HWMON_BACKUP_LID), CFG_FILE_HWMON_BACKUP_SIZE,
CFG_FILE_HWMON_BACKUP_TOTAL, SIGNLE_DEFAULT_REC , (char
*)&st_hwmon_backup_default, DataReset , NULL
},
};

```

//Meta tool需要读取的信息

```

\mediatek\custom\${project}\cgen\inc\Custom_NvRam_data_item.h

```

```

LID_BIT VER_LID(AP_CFG_RDCL_HWMON_BACKUP_LID)
HWMON_BACKUP_STRUCT *CFG_FILE_HWMON_BACKUP_TOTAL
{

};

```

#### 4. 添加nvram到product\_info

```

JB : \mediatek\custom\common\cgen\CFG_file_info.c

```

```

KK : \mediatek\external\nvram\libcustom_nvram\CFG_file_info.c

```

```

const TABLE_FOR_SPECIAL_LID g_new_nvram_lid[] =
{
{ AP_CFG_REEB_PRODUCT_INFO_LID, 0, 1024 * 1024 },
{ AP_CFG_RDCL_HWMON_BACKUP_LID, 1024 * 1024, 1024 * 1024 },
};

```

注意：这里LID的struct和offset size的定义，在使用nand和emmc有所不同，请参考后面note中的FAQ注意事项。

#### 5. 实现备份还原的代码

```

\mediatek\external\nvram\libnvram_daemon_callback\libnvram_daemon_callback
.c

```

```

int my_callback(void)
{
    ALOGD("nvram daemon callback will run!!!");
    my_nvram_daemon(); //add
    return 0;
}

int my_nvram_daemon(void)
{
    int err = 0, ret = 0;
    F_ID fd;
    int rec_size, rec_num;
    HWMON_BACKUP_STRUCT hwmon_backup;
    NVRAM_LOG("ready to read hwmon backup\n");

    /* read the hwmon backup data */
    fd = NVM_GetFileDesc(AP_CFG_RDCL_HWMON_BACKUP_LID, &rec_size, &rec_num,
        ISREAD);
    if(fd.iFileDesc < 0)
    {
        NVRAM_LOG("null hwmon backup file handle pointer: %d!\n", fd.iFileDesc);
        err = -EFAULT;
        return err;
    }
    ret = read(fd.iFileDesc, &hwmon_backup, (rec_size * rec_num));
    if(ret < 0)
    {
        NVRAM_LOG("read hwmon backup file error: %d!\n", ret);
        err = -EFAULT;
    }

    NVM_CloseFileDesc(fd);
    NVRAM_LOG("hwmon backup data, hwmon_acc:%d-%d-%d\n",
        hwmon_backup.hwmon_acc.offset[0], hwmon_backup.hwmon_acc.offset[1],
        hwmon_backup.hwmon_acc.offset[2]);
    /* set the hwmon acc data */
    fd = NVM_GetFileDesc(AP_CFG_RDCL_HWMON_ACC_LID, &rec_size, &rec_num,
        ISWRITE);
    ret = write(fd.iFileDesc, &hwmon_backup.hwmon_acc, (rec_size * rec_num));
    NVM_CloseFileDesc(fd);
    NVRAM_LOG("hwmon backup data, hwmon_gyro:%d-%d-%d\n",
        hwmon_backup.hwmon_gyro.offset[0], hwmon_backup.hwmon_gyro.offset[1],
        hwmon_backup.hwmon_gyro.offset[2]);
    /* set the hwmon gyro data */
    fd = NVM_GetFileDesc(AP_CFG_RDCL_HWMON_GYRO_LID, &rec_size, &rec_num,
        ISWRITE);
    ret = write(fd.iFileDesc, &hwmon_backup.hwmon_gyro, (rec_size * rec_num));
    NVM_CloseFileDesc(fd);

```

```
NVRAM_LOG("hwmon backup data, hwmon_ps:%d-%d-%d\n",  
hwmon_backup.hwmon_ps.ps_cali[0], hwmon_backup.hwmon_ps.ps_cali[1],  
hwmon_backup.hwmon_ps.ps_cali[2]);  
/* set the hwmon ps data */  
fd = NVM_GetFileDesc(AP_CFG_RDCL_HWMON_PS_LID, &rec_size, &rec_num,  
ISWRITE);  
ret = write(fd.iFileDesc, &hwmon_backup.hwmon_ps, (rec_size * rec_num));  
NVM_CloseFileDesc(fd);  
return err;  
}
```

## 7. sensor校准客制化

这里实现了对sensor 校准数据的backup，但是在具体sensor的校准数据保存的地方，需要将数据同时保存到 AP\_CFG\_RDCL\_HWMON\_BACKUP\_LID 上，否则无效。

Note：

ID: FAQ03920 [NvRAM] NvRAM product info的客制化以及注意事项