

【转载】Android Camera HAL V3 Vendor Tag及V1 V3参数转换

发表于2016-03-27 类别: 技术(57) 标签: 转载(72) Android(23) Camera(9)

(注：转载于为知笔记)

转眼一看，上一次发博文都快是三年之前了，惭愧！主要是三年前找的这份工作，虽然是世界500强的技术大牛公司，但是工作可一点都不高大上，非常的忙，一天不但要处理各种camera的bug，还要开发camera的各种feature和sensor驱动，还要和内部、外部的人各种扯皮，你懂的。忙的三年了才有闲心来发表这篇博文。

牢骚已完，言归正传。

在Android 5.0 上，Google正式的将Camera HAL 3.0作为一个标准配置进行了发行，当然Camera HAL V1也是作为兼容标准是可以用的。但是很多有实力的芯片厂商都第一时间切换到了HAL V3。HAL V3与V1最大的本质区别，我认为就是把帧的参数和帧的图像数据绑定到了一起，比如V1的时候一张preview上来的YUV帧，APP是不知道这个YUV帧采用的Gain和曝光时间究竟是多少。但是在V3里面，每一帧都有一个数据结构来描述，其中包括了帧的参数和帧的数据，当APP发送一个 request的时候是需要指定使用什么样的参数，到request返回的时候，返回数据中就有图像数据和相关的参数配置。

在V1里面，如果想增加一些厂商特定的参数，比如增加Saturation的设置，最简单的方法就是直接使用CameraParameters.set来实现，即在APP中，

```
Camera.Parameters mParameters = mCamera.getParameters();
mParameters.set("saturation", 10);
mCamera.setParameters(parameters);
```

由于HAL V1的参数传递是通过字符串来完成的，也就是最后传到HAL的字符串里面会有“saturation=10”这样的字符串，在HAL直接解析这些字符串就OK了。

但是到了HAL V3，从framework到hal的参数传递都是通过metadata的方式来传递，简单的说就是每一个设置现在都变成了一个参数对，比如假设要设置 AE mode为auto，以前V1可能是“AE mode=auto”这样的字符串，在V3就是比如AE mode的功能序号是10，参数auto为1，传下来的其实是类似（10,1）这样的参数对。在HAL里面需要通过10这个参数，去获取设置值1。因此比如原来在V1里面有saturation这种设置的话，在V3就需要添加新的处理来实现。

Google 考虑到各大芯片厂商都可能有自己的特定参数需要设置，因此在metadata里面定义了vendor tag的数据范围来让vendor可以添加自己的特定操作。比如上面说到的saturation由于不是google的标准参数，就可以通过vendor tag来实现。

首先，需要在 camera_metadata_tags.h 里面定义自己的vendor tag序号值，比如

```
typedef enum camera_metadata_tag {
    ANDROID_SYNC_START,
    ANDROID_SYNC_MAX_LATENCY,    // enum      | public
    ANDROID_SYNC_END,

+   VENDOR_TAG_SATURATION =
+   VENDOR_SECTION_START,

    .....

} camera_metadata_tag_t;
```

Google规定，Vendor Tag都需要在 VENDOR_SECTION_START 后面添加，此处添加了 VENDOR_TAG_SATURATION。在HAL里面如果需要处理Vendor Tag，一个是需要camera module的版本是2.2以上，因为Google在这个版本之后才稳定支持vendor tag。一个是需要vendor tag的operations函数。如下面所述。

```
camera_module_t HAL_MODULE_INFO_SYM = {
    NAMED_FIELD_INITIALIZER(common) {
        NAMED_FIELD_INITIALIZER(tag) HARDWARE_MODULE_TAG,
-       NAMED_FIELD_INITIALIZER(module_api_version) CAMERA_MODULE_API_VERSION_2_0,
+       NAMED_FIELD_INITIALIZER(module_api_version) CAMERA_MODULE_API_VERSION_2_2,
        .....
-       NAMED_FIELD_INITIALIZER(get_vendor_tag_ops) NULL,
+       NAMED_FIELD_INITIALIZER(get_vendor_tag_ops) get_vendor_tag_ops,
        .....
    };
};
```

get_vendor_tag_ops 是需要自己实现的，其中主要有以下几个接口，可以参照hardware/libhardware/modules /camera下面的VendorTags.cpp和VendorTags.h来实现。主要的功能如以下描述

```
static void get_vendor_tag_ops(vendor_tag_ops_t* ops)
+{
+   ALOGE("%s : ops=%p", __func__, ops);
+   ops->get_tag_count      = get_tag_count;
+   ops->get_all_tags       = get_all_tags;
+   ops->get_section_name   = get_section_name;
+   ops->get_tag_name       = get_tag_name;
+   ops->get_tag_type       = get_tag_type;
+}
```

get_tag_count 返回vendor tag的个数，有多少个返回多少个

get_all_tags 把所有vendor tag挨个放在service传下来的 uint32_t * tag_array 里面，这样上层就知道每一个tag对应的序号值了

get_section_name 获取vendor tag的section对应的section名称，比如可以把某几个tag放在一个section里面，其它的放在其它的section里面。 查看metadata.h里面的定义很好理解，如果你想增加自己的section，就可以在 VENDOR_SECTION = 0x8000 后面添加自己的section。本人由于参数较少，也没有分类的必要，所以就使用默认的 VENDOR_SECTION。

```
typedef enum camera_metadata_section {
    ANDROID_COLOR_CORRECTION,
    .....
    ANDROID_SECTION_COUNT,
    VENDOR_SECTION = 0x8000
} camera_metadata_section_t;

typedef enum camera_metadata_section_start {
    ANDROID_COLOR_CORRECTION_START = ANDROID_COLOR_CORRECTION << 16,
    .....
    VENDOR_SECTION_START           = VENDOR_SECTION           << 16
} camera_metadata_section_start_t;
```

get_tag_name 用于获取每一个tag的名称，比如我这个地方返回“saturation”就可以了

get_tag_type 这个函数返回tag对应的设置数据的类型，可以用TYPE_INT32，TYPE_FLOAT等多种数据格式，取决于需求，我这个只要是INT32就行了。

这样CameraService.cpp在启动的时候就会调用onFirstRef里面的下面代码来加载我们所写的vendor tag

```
if (mModule->common.module_api_version >= CAMERA_MODULE_API_VERSION_2_2) {
    ALOGE("yangsy CameraService %s %d", __FUNCTION__, __LINE__);
    setUpVendorTags();
}
```

那vendor tag如何设置，在HAL里面如何处理呢？由于我这个saturation设置是在以前V1的APP里面需要使用，因此首先需要实现V1和V3参数的转换，Google在services/camera/libcameraservice/api1/client2/Parameters.cpp实现相应的转换，因此首先需要在status_t Parameters::set(const String8& paramString)里面获取V1 APP传下来的saturation的值，其中的paramString就是V1的参数设置的字符串

```
mSaturation = newParams.getInt("saturation");
```

由于V3的参数都是在request frame的时候一起下发的，因此需要讲mSaturation的值在Parameters::updateRequest(CameraMetadata*request)里面下发到HAL，即

```
+ res = request->update(VENDOR_TAG_SATURATION, &mSaturation, 1);
+ ALOGE("yangsy request update saturation result:%d", res);
```

这样就将saturation的vendor tag和其设置值发送到了HAL V3。

那在HAL V3里面如何获得saturation的值呢？这个就比较简单了，使用CameraMetadata::find(uint32_t tag)的find函数，在处理request的metadata的代码里面添加就OK了

```
+ tag = VENDOR_TAG_SATURATION; //需要查找的tag
+ ALOGE("yangsy HAL settings effect setting");
+ entry = settings.find(tag); //调用metada.find
+ if (entry.count == 1) { //找到了tag
+     ALOGE("yangsy HAL try saturation %d", entry.data.i32[0]); //entry.data.i32[0]就是相应的设置值
+     mSensor->setSaturation(entry.data.i32[0]);
+ }
```

[← 新一篇](#)
[旧一篇 →](#)

相关文章

- 2018-01-23 **【记录】** Win8安装adb
- 2017-08-26 **【原创】** Android x86上的Camera HAL(2)
- 2017-08-20 **【原创】** Steps of Camera HAL Module Searching
- 2017-08-16 **【转载】** Android Camera Overview
- 2017-08-05 **【转载】** Android x86睡眠触发时间修改
- 2017-07-27 **【原创】** Camera不同版本API与HAL流程
- 2017-07-18 **【原创】** 在Github上建立clean study
- 2017-07-17 **【原创】** Android x86上Camera流程
- 2017-06-27 **【原创】** 写了一个Camera App：GoldCam
- 2017-06-14 **【原创】** Android x86屏幕旋转设定apk
- 2017-06-13 **【记录】** USB硬盘安装Android x86
- 2017-06-09 **【记录】** Ubuntu 16.04安装Android Studio

Copyright © Clean Li 2020