ICameraDevice::
open()

open
/* 通过之前注册好的g_
jumpTableHAL3结构体函数指针调
到camxhal3.cpp中的open函数*/

pHAL3->open

StrToUL(pCameraIdAPI, &pNameEnd,
10) /* 上层下发的cameraID字符串转为整形数
据*/

GetCHIAppCallbacks()->chi_remap_
camera_id /* 根据之前注册的chi里的
callback接口函数调到chi-cdk里的相应接口*/ chi_remap_camera_id RemapCameraId

GenerateExtendOpenData FillTokenList
/* 将定制化的settings一一填充到tokenlist里*/

ProcessCameraOpen

m_ChiAppCallbacks.chi_extend_open pExtensionModule->ExtendOpen

GetActiveResourceCost

PerfLockManager::Create pPerfLockManager->Initialize

MappingConfigSettings
/* 初始化了m_pConfigSettings数组*/

更新记录camera open和close状态

SetCameraStatus

DisplayConfig::Initialize IDisplayConfig::getService

DisplayConfig::SetCameraStatus mDisplayConfig->setCameraLaunchStatus

GenerateModifySettingsData /* 将之前初始
化存在pStaticSettings里的配置值写入到刚定
义的setting结构体数组里*/

GetCHIAppCallbacks()->chi_modify_
settings /* 循环遍历setting结构体数组，将每
个配置值都写入m_pConfigSettings数组或者
其它全局成员变量里*/ pExtensionModule->ModifySettings

HALDevice::Create pHALDevice->Initialize

m_cameraId = cameraId
/* 将前面remap后的cameraId赋给成员变量
m_cameraId*/

将camxhal3entry.cpp中的close函数注册到
m_camera3Device里

将g_camera3DeviceOps结构体地址注册给
pDeviceOps

将入参HwModule* pHwModule赋给
pModule

将ProcessCaptureResult函数注册给m_
HALCallbacks

将Notify函数注册给m_HALCallbacks

ClearFrameworkRequestBuffer()

CalculateSizeAllMeta

pHwEnvironment->
GetStaticMetadataKeysInfo(&
requestKeysInfo,
RequestAvailableRequestKeys) Titan17xContext::
GetStaticMetadataKeysInfo

pHwEnvironment->
GetStaticMetadataKeysInfo(&
resultKeysInfo,
RequestAvailableResultKeys) Titan17xContext::
GetStaticMetadataKeysInfo

VendorTagManager::
PopulateControlVendorTagId()

HAL3MetadataUtil::CreateMetadata

HALDevice::
ConstructDefaultRequestSettings

pHALDevice->GetCameraDevice()
/* 拿到初始化过程中注册的m_
camera3Device并转化成camera3_device_t
指针*/

*ppHwDeviceAPI = &pCamera3Device->
common; /* 将初始化过程中填充的m_
camera3Device.hwDevice信息回给上层的
ppHwDeviceAPI接口*/