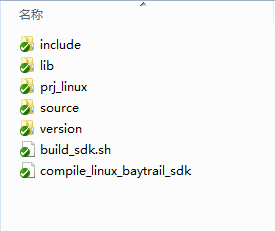
开发环境搭建：

把baytail.tgz压缩文件，解压在linux主机/opt目录下即可。

SDK的文件目录如下图：



其中：

SDK\baytrail\_sdk\source 下包含：kd\_comm\_ipc\_public.c和kd\_custom\_alg\_demo.c文件。

kd\_custom\_alg\_demo.c 供用户开发的接口文件，用户可直接修改此文件，或在此目录下继续增加其他文件。

SDK\baytrail\_sdk\include 下包含：头文件申明

SDK\baytrail\_sdk\lib 下包含相机运行的所需的库文件。

SDK\baytrail\_sdk\prj\_linux 下为makefile文件。若用户增加其他文件，修改makefile\_baytrail即可

主要开发工作：

1. 算法的实现

在kd\_custom\_alg\_demo.c中，我们已经封装好了对算法调用的主要接口。包括，算法的created，process，算法输出，以及对后续的H264码流或JPEG图片的后续处理等等。用户无需关心相机业务系统流程，仅结合api说明文档，填充各函数的功能实现，调用自开发的算法库即可。用户的算法若已封装成库的形式，把库文件放于SDK\baytrail\_sdk\lib下。

1. Web消息的处理

我们已经封装了相机端和web端的消息透明传输通道，用户可通过调用此通道，实现相机与上位机端的通信。函数custom\_demo\_msg\_process为web消息处理的封装函数。

此通道支持用户自定义消息格式或长度（最大为10K）等信息。

以设置算法参数为例：

API\_STATUS custom\_demo\_msg\_process(TIntelliMsgParam \*ptMsgHandle, void \*pContext)

{

API\_STATUS ret = RET\_OK;

struct TUserMsg tMsgTestSet; //与WEB端协定的结构体，用于存放该 //条消息的信息

switch(ptMsgHandle->eMsgtype)

{

case MSG\_algparameterset: //消息号，消息体结构等通信协议用户自定义

memcpy(&tMsgTestSet, ptMsgHandle->msgBuf, sizeof(TMsgTestGet))

kd\_call\_set\_param(……);

printf("custom\_demo msg base\n");

break;

case …..

case …..

default:

break;

}

return ret;

}

编译

用户编码完毕后，修改SDK\baytrail\_sdk\prj\_linux目录下makefile\_baytrail文件（若需要）。

运行SDK\baytrail\_sdk 目录下build\_sdk.sh脚本，进行工程编译。生成kd\_comm\_ipc\_public.so文件。

用户开发程序调试完毕，运行SDK\baytrail\_sdk 目录下compile\_linux\_baytrail\_sdk文件，生成IPC121-Ei7N-X100.pkg文件。用于相机的升级。