Document about How to Run DVR A/P(for 5864)
(support Intersil 5864 encoder)

(I).Requirements for Platform of Compiler

1.System Platform

X86 Linux is required, Linux system of 32 bits, linux kernel version >= 2.6,any distribution version is ok(fedora,ubuntu,centos …). Tool for Cross Compiler should be matched with system platform.

2.Requirement for Compiler

gcc or gcc cross-compiler, the version of gcc >= 3.3.4

3.Requirement for C Library

Under environment of compiler, glibc or uclibc is ok, newlib is not tested.

4.Libraries needed

Posix’s libpthread (-lpthread), libm(-lm), libevent(version:1.4.x, version 2.x is not tested), libev(any version)

5.How to execute the A/P

Currently, the link (for A/P) is static, it is not needed for the final compiled executable A/P to load any dynamic libraries. Its purpose is to avoid bringing any dynamic libraries to any targeted platform. In the future, to decrease the size of the executable A/P, you can consider to add dynamic libraries.

About other consideration about compiler, Please refer to “Document about A/P and Compiler for SDK”

(II).Requirements for Targeted Platform

1.Requirement for Internal Kernel of System

Version of linux kernel >= 2.6, the version should be matched with the one of driver version.

2.Requirement for Driver Version

The version should be matched with the one of linux kernel.

3.Requirement for Authority of Execution of A/P

User should be granted to have authority of root.

4.Requirement for Path of A/P

Write, read and execution rights are granted for the current path

5.Requirement for encoder and decoder chips

Support intesil 5864 encoder chip now

6.Requirement for System Environment

There are /dev, /proc, /sys on the targeted environment. rm, mkdir, mknod can be executed on it. Mount, unmount and mkfs.xx(ext3, ext2, etc.) are also needed for file access. Port 10001~10003 should be opened for communication(TCP).

(III).Requirements for Communication with Client A/P (NetClient, Center)

1.Protocol of communication

Mainly by TCP, TCP or UDP for streaming transmission

2.Port

The default port of TCP communication for DVR A/P is 10001, port 10002 for data. It is needed for NetClient, Center to set control port for server, 10001.

3.The Procedures

Run DVR A/P on targeted platform first, then run Client A/P on PC.

4. The Procedures to Operate Client A/P

Please reference associated manuals

(IV). The Structure of Files and the Description of Source Code of DVR A/P

1. The Structure of 1st layer of Directories of DVR A/P

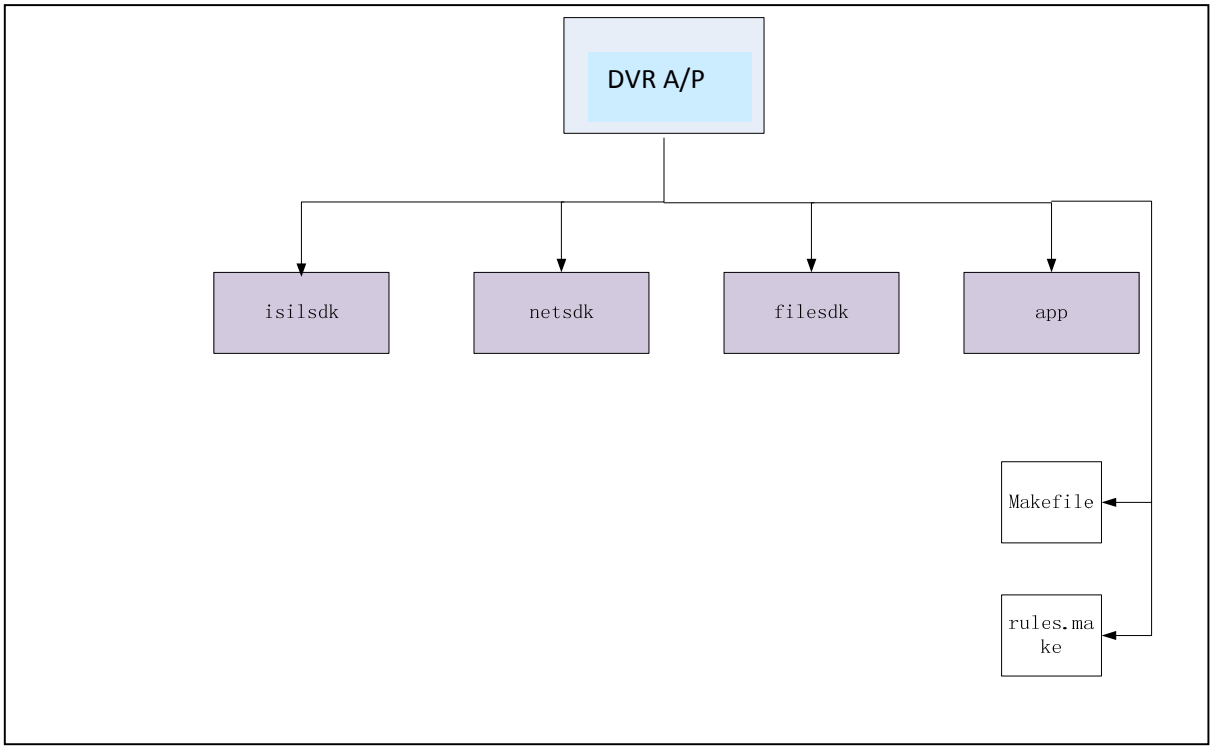

Figure 1: 1st layer of directories of DVR A/P

The description of 1st layer of directories:

a. All of SDK about audio and video encoding are packaged in isilsdk. Header files and source codes are included in it. It will need libevent (1.4.x). It Provides API for development(Please reference MediaSDK API(V1.2)_20111207). It can be compiled by developer.

b. All of SDK about network are packaged in netsdk. TCP and UDP are supported. Header files and source codes are included in it. It will need libev (version：4.04). It Provides API for development.

c. All of SDK about file access and file format are packaged in filesdk.

It supports functions of read/write for audio/video files. Header files and source codes are included in it.

d. All of A/Ps to implement the above functions, associated source codes and header files, compiled executable files are stored in app. compiled executable files are put under the directory, src which is under app. All of executable files are with the name, dvr in the beginning.

e. The files needed for compile (make) are under Makefile.

f. files containing variable settings of compiler environment are under rules.make.

2. The Description of intersil media SDK, isilsdk

Under the directory of isilsdk, the structure of files is as Figure 2.(only important files or directories shown)

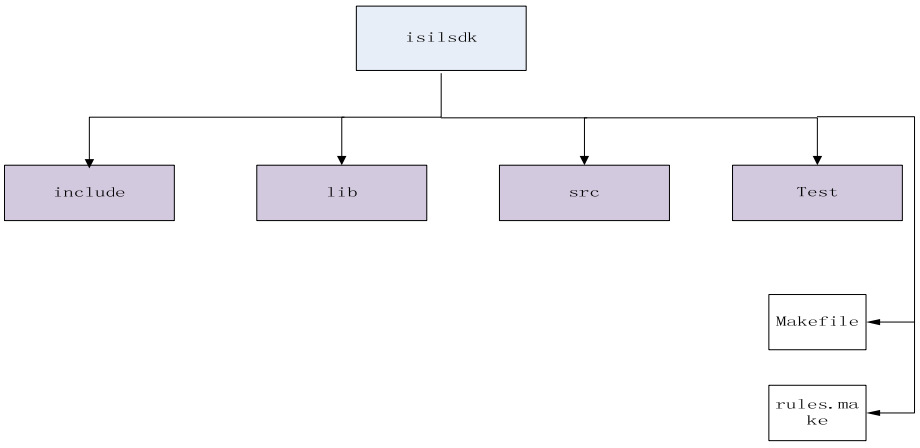

Figure 2: The Structure of files of isilsdk

The description of the directories and files:

a. All of header files(.h) of isilsdk are stored in include. It provides API for development(Please reference MediaSDK API(V1.2)_20111207).

b. All of libraries generated by isilsdk are stored in lib. The name of static or dynamic libraries at compiling isilsdk is libisilmediasdk.a(so). When you develop your application, you need to link it by using -lisilmediasdk command in the Make file. You can refer to the Makefile in the app\dvr\app\src folder.
Now, only static libraries supported, dynamic libraries can be considered if necessary.

c. all of source coded(.c) of isilsdk are stored in src.

d. All of test examples of isilsdk are stored in Test. The file, isil_sdk_test1.c can be called and by some associated APIs, audio and video encoding are implemented. In the future, more test examples will be added in it.

e. Files containing variable settings for compiling isilsdk are stored in rules.make. The files are only needed for compiling isilsdk. They are not needed for compiling DVR a/p.

3. The Description of SDK for Supporting Communication, netsdk

All of TCP and UDP handling functions are packaged in netsdk. Centralized trigger mechanism for network events (libev) is packaged in it also. Under it, there are include, lib, src

a. All of header files are included in Include

b. The libraries generated by compiling netsdk are stored in Lib. The name of library is libnetevcore.a(static library)

c. All of source codes(.c) of netsdk are included in src.

4. The Description of SDK for File Access, filesdk

filesdk implements most of file access functions, including storage management of files, formating of disk, unique file format for access, supporting read frame by frame and also offering APIs for them. The structure of filesdk is similar to the one of netsdk. The library generated by compiling filesdk is libfilesdk.a (static library)

5. The Description of app

All of application layers to implement most functions about the communication with netclient or Center, are stored in app, including preview of audio/video, parameter settings for audio/video, channel management, local storage for files, supporting audio/video streaming under TCP, UDP, local file playback for files decoded from intersil 5866 chip, loopback for video streaming

The structure of the files of app is as figure 3.

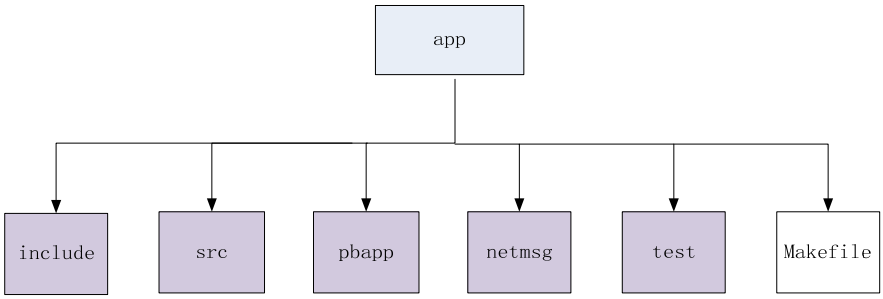

Figure 3: The Structure of files of app

The description:

a. All of header files of app are stored in include.

b. All of source codes of app are included in src.

c. pbapp is decoder module, implementing file playback and loopback, mainly for intersil 5866.

d. netmsg is module for communication with Netclient or Center, implementing analyzing, packaging. Please be noted, the messages

for communication is little-endian. It should be specified under rules.make, which is under DVR directory.

e. All of test examples of app are stored in test. There is only one file now. Its name is test_enc_1.c. It can be called and by some associated APIs, provided by isilsdk, audio and video encoding are implemented.

(V).Miscellaneous

1.Suport Multiple Chips

A/Ps, including isilmediasdk, support multiple chips of 5864 to be run simultaneously. A/Ps will check all of chips automatically and do the necessary initialization for the chips. To consider the requirements of users, you can find the file, app/include/isil_channel_map.h and you can modify one statement, in it,

```
#define TEST_CHIP_CNT 1
It is to define how many chips will be run simultaneously.
For example, 1 for one chip, 2 for 2 chips
```

2.How to Enable File Access Function

The default of A/P is disablement of file access functions. If it is needed, file_record_init function of main.c need be opened. Also, user need add Hard Drive, the mount point is /mnt/sda1