## Software decode

### Cut of FFMPEG open source library

In order to realize real-time decoding of H264 video source on rt1050 development board and reduce code redundancy and improve compilation efficiency, FFMPEG library needs to be cut and ported. Specific steps are as follows:

1: Insall yasm

yasm can be installed through the ubuntu package, or it can be installed by compiling the source code.

**sudo apt-get install yasm**

2: Download and Extract the FFMPEG library

Download and unzip the FFMPEG library(Optional version)

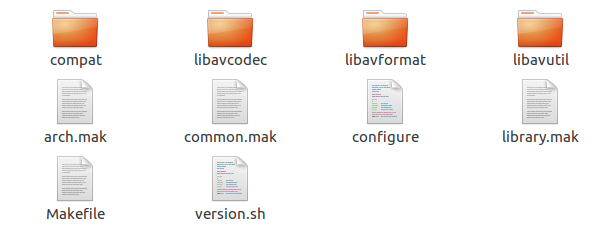
[https://www.FFMPEG.org/](https://www.ffmpeg.org/)

**Latitude:~/name$ tar -xvf ffmpeg-3.0.11.tar.bz2**

**Latitude:~/name$ cd ffmpeg-3.0.11/**

3: cut redundant folders and related functions in FFMPEG library

Considering the application requirements and the convenience of cut, modify the [makefile](https://github.com/xinyigao/H.264-video-decoding) function before cut and selectively delete unnecessary folders and functions in the FFMPEG library. The deleted file directory in Figure 5:



1. FFMPEG deleted file directory

4: Configuration and compilation

The configuration parameters of FFMPEG can be viewed by following commands:

**Latitude:~/name/ffmpeg-3.0.11$ ./configure --help**

According to the different function modules, configure relevant parameters such as:

**Latitude:~/name/ ffmpeg-3.0.11$ ./configure --prefix=host --disable-gpl --disable-nonfree --enable-shared --disable-static --disable-asm --disable-pthreads --disable-doc --disable-htmlpages --disable-manpages --disable-podpages --disable-txtpages --disable-encoders --disable-muxers --disable-decoders --enable-decoder=h264 --disable-parsers --enable-parser=h264 --disable-demuxers --enable-demuxer=h264 --disable-bsfs --disable-faan --disable-fft --disable-dct --disable-devices --disable-network --disable-filters --disable-programs --disable-protocols --enable-protocol=file --disable-avfilter --disable-postproc --disable-swscale --disable-swresample --disable-avdevice --disable-avresample --disable-w32threads --disable-os2threads --disable-symver --disable-runtime-cpudetect --disable-swscale-alpha --disable-d3d11va --disable-dxva2 --disable-vaapi --disable-vda --disable-vdpau --disable-videotoolbox --disable-iconv --disable-sdl --disable-zlib --disable-xlib --disable-libxcb-shape --disable-libxcb-shm --disable-libxcb-xfixes**

Then execute:

**Latitude:~/name/ ffmpeg-3.0.11$ make -j16 & make install**

After compiling FFMPEG, there are dynamic libraries and header files needed for engineering in host directory.

5: API test

Download the H.264 decoding function xxx.c from the Internet and pack it with the H.264 video source in the host/test directory.

**Latitude:~/name/ ffmpeg-3.0.11/host/test$**

environment variable

**Latitude:~/name/ ffmpeg-3.0.11/host/test$ export LD\_LIBRARY\_PATH=../lib/**

**compile**

**Latitude:~/gxy/ ffmpeg-3.0.11/host/test$ g++ -I ../include/ xxx.c -o xxx -L../lib/ -lavcodec -lavuti**

link dynamic libraries

**Latitude:~/gxy/ ffmpeg-3.0.11/host/test$ ldd ./xxx**

**Latitude:~/gxy/ ffmpeg-3.0.11/host/test$./xxx**

6:Cut

After running successfully, keep the corresponding .c and .h functions (makefile, .v reservation) and delete other functions according to the .o function in the libavcodec and libavutil folders

7: Complement Dependency Function

**Latitude:~/name/ ffmpeg-3.0.11/$make**

Repeat this step to fill in dependent. c and. h functions as prompted.

8: Cut library test

The cut function library needs further validation and modification. It can be validated by writing its own program or by using the existed [ffmpeg-h264](https://github.com/xinyigao/H.264-video-decoding) test project.

### FFMPEG porting

The H.264 video decoding running on the i.MX RT1050 EVK board does not have an operating system, so many system functions cannot be found, and the file system of the board is based on fatfs 32, so the cut FFMPEG library needs to be ported. More details in the [diff](https://github.com/xinyigao/H.264-video-decoding) file.