编译过程可参考：

<http://blog.csdn.net/tttyd/article/details/39253281>

1. mingw 32位版本安装

下载mingw线上安装程序。双击，根据提示安装。

安装完成后，找到文件C:\MinGW\msys\1.0\msys.bat，在文件开头添加：

call "C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\bin\amd64\vcvars64.bat"

参考教程：http://www.cnblogs.com/ode/archive/2011/08/01/2143541.html

1. 安装cmake，用于编译x265

下载win32版本：<https://cmake.org/download/#previous>

安装。

在mingw命令行窗口中设置环境变量

export PATH=”$PATH: /c/Program Files (x86)/CMake/bin”

1. 安装pkg-config，由于编译x265

参考：<http://blog.csdn.net/chaosllgao/article/details/3984462>

下载pkg-config-0.23-2.zip及glib\_2.18.4-1\_win32.zip

http://ftp.gnome.org/pub/gnome/binaries/win32/dependencies/pkg-config\_0.23-3\_win32.zip

http://ftp.gnome.org/pub/gnome/binaries/win32/glib/2.18/glib\_2.18.4-1\_win32.zip

把glib\_2.18.4-1\_win32.zip中的libglib-2.0-0.dll与pkg-config.exe放在C:\MinGW\msys\1.0\bin.

1. 编译yasm，用于编译x264, x265, ffmpeg.

wget http://www.tortall.net/projects/yasm/releases/yasm-1.3.0.tar.gz

tar xzvf yasm-1.3.0.tar.gz

cd yasm-1.3.0

./configure --prefix="$HOME/ffmpeg/build" --bindir="$HOME/ffmpeg/bin"

make

make install

1. 编译SDL

下载SDL-1.2.15

cd SDL-1.2.15

./configure --prefix="$HOME/ffmpeg/build" --bindir="$HOME/ffmpeg/bin"

make

make install

1. 编译x264

wget http://download.videolan.org/pub/x264/snapshots/last\_x264.tar.bz2

tar xjvf last\_x264.tar.bz2

cd x264-snapshot\*

PATH="$HOME/ffmpeg/bin:$PATH" ./configure --prefix="$HOME/ffmpeg\_build" --bindir="$HOME/ffmpeg/bin" --enable-static --disable-opencl

PATH="$HOME/ffmpeg/bin:$PATH" make

make install

参考：<http://blog.csdn.net/naaaa/article/details/44061839>

执行完后就会在MinGW\msys\1.0\local下生成dll文件跟头文件。

lib文件只需将local\lib下的libx264.dll.a文件直接改名为libx264.lib即可。

<http://berritus.blog.163.com/blog/static/10528425201410284403940/> dll定位问题

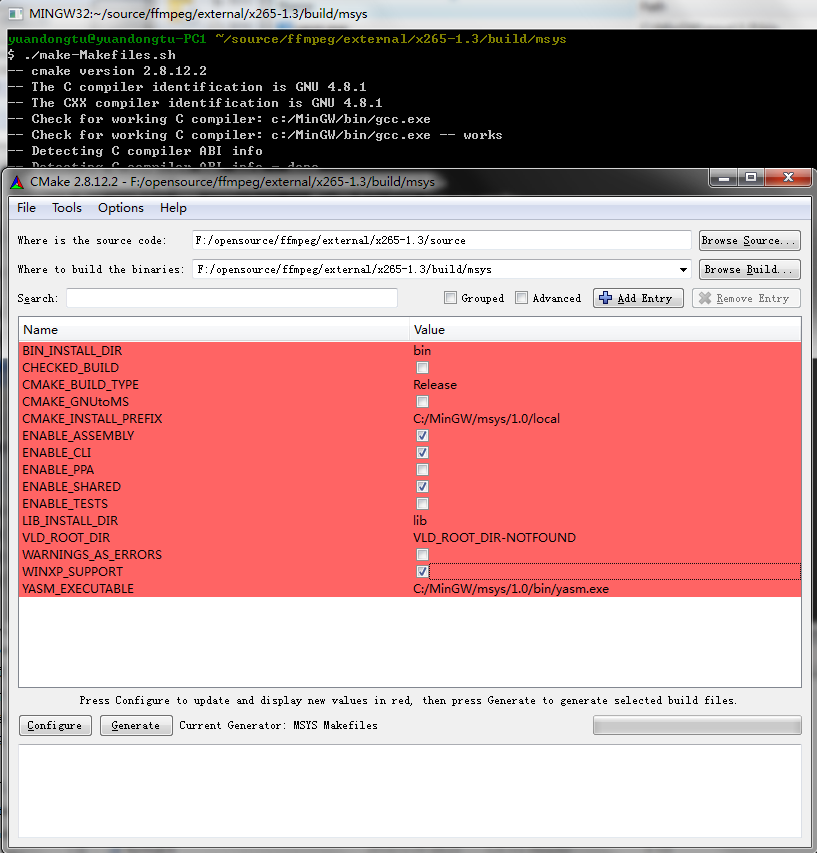
1. 编译x265

编译过程可参考：<http://blog.csdn.net/tttyd/article/details/39253281>

下载代码：hg clone <https://bitbucket.org/multicoreware/x265>

cd ~/ffmpeg/source/x265/build/msys

sh make-Makefiles.sh



修改CMAKE\_INSTALL\_PREFIX为: C:\MinGW\msys\1.0\home\g7626\ffmpeg\build

ENABLE\_SHARED去掉。

WINXP\_SUPPORT选中。

先点下面的configure，然后点generate即可，然后关闭，回到命令行。

make

make install

1. 编译ffmpeg
   1. x265链接库的准备

修改文件x265/source/x265.cpp 下面这一行：

SetThreadExecutionState(ES\_CONTINUOUS | ES\_SYSTEM\_REQUIRED | ES\_AWAYMODE\_REQUIRED);

为：SetThreadExecutionState(ES\_CONTINUOUS | ES\_SYSTEM\_REQUIRED );

解决编译时找不到ES\_AWAYMODE\_REQUIRED的问题。

复制x265\build\msys\profile\PPA\libPPA.a 到$HOME/ffmpeg/build/lib下面，解决编译时找不到-lPPA的问题。

* 1. ffmpeg编译

PATH="$HOME/ffmpeg/bin:$PATH" PKG\_CONFIG\_PATH="$HOME/ffmpeg/build/lib/pkgconfig" ./configure --prefix="$HOME/ffmpeg/build" --pkg-config-flags="--static" --extra-cflags="-I$HOME/ffmpeg/build/include" --extra-ldflags="-L$HOME/ffmpeg/build/lib" --bindir="$HOME/ffmpeg/bin" --enable-gpl --enable-libx264 --enable-libx265 --enable-nonfree

PATH="$HOME/bin:$PATH" make

make install

1. libde265编译

linux编译：

git clone <https://github.com/strukturag/libde265>

cd libde265

./autogen.sh

export PATH="$HOME/ffmpeg/bin:$PATH"

./configure --prefix="$HOME/ffmpeg/build" --bindir="$HOME/ffmpeg/bin" --enable-static --disable-shared

make

make install

windows编译：打开vs工程直接编译

64位mingw安装：

参考<http://www.cnblogs.com/x_wukong/p/4833578.html>

安装mingw-w64-install.exe，设置环境变量。

安装MSYS-1.0.11.exe，设置fstab文件

编译ffmpeg，找不到pr.exe

下载工具包: <http://gnuwin32.sourceforge.net/packages/coreutils.htm>

覆盖到/msys/1.0/bin

Ffmpeg 3.3.3编译并支持硬件解码

参考https://trac.ffmpeg.org/wiki/HWAccelIntro

1. 准备硬件编码相关的头文件

下载dxva2api.h，下载地址<http://download.videolan.org/pub/contrib/dxva2api.h>，复制到C:\MinGW\include下面。

在windows机器上找到win7版本的d3d9.h，d3d9caps.h，d3d9types.h文件，覆盖C:\MinGW\include下面原始文件。Ffmpeg2版本不需要更新这3个文件。

1. 使用如下的配置编译和裁剪PC版本的ffmpeg

PATH="$HOME/ffmpeg/bin:$PATH" PKG\_CONFIG\_PATH="$HOME/ffmpeg/build/lib/pkgconfig" ./configure --prefix="$HOME/ffmpeg/build" --pkg-config-flags="--static" --extra-cflags="-I$HOME/ffmpeg/build/include" --extra-ldflags="-L$HOME/ffmpeg/build/lib -static-libstdc++ -static-libgcc" --bindir="$HOME/ffmpeg/bin" --enable-shared --disable-static --enable-gpl --enable-nonfree --disable-iconv --disable-programs --disable-ffmpeg --disable-ffplay --disable-ffprobe --disable-ffserver --disable-doc --disable-htmlpages --disable-manpages --disable-podpages --disable-txtpages --enable-dxva2 --disable-hwaccels --enable-hwaccel=h264\_dxva2 --enable-hwaccel=hevc\_dxva2 --disable-cuvid --disable-vaapi --disable-vda --disable-vdpau --disable-avdevice --disable-filters --disable-postproc --disable-avfilter \

--disable-encoders --enable-encoder=mjpeg \

--disable-decoders --enable-decoder=aac --enable-decoder=flv --enable-decoder=h264 --enable-decoder=hevc --enable-decoder=mp3\* \

--disable-muxers --enable-muxer=flv --enable-muxer=image2 \

--disable-demuxers --enable-demuxer=flv --enable-demuxer=hls --enable-demuxer=live\_flv --enable-demuxer=mov --enable-demuxer=mp3 --enable-demuxer=mpegts \

--disable-parsers --enable-parser=aac --enable-parser=h264 --enable-parser=hevc \

--disable-bsfs --enable-bsf=aac\_adtstoasc --enable-bsf=h264\_mp4toannexb --enable-bsf=hevc\_mp4toannexb --enable-bsf=mp3\_header\_decompress \

--enable-protocols --disable-protocol=bluray --disable-protocol=ffrtmpcrypt --enable-protocol=ffrtmphttp --disable-protocol=gopher --enable-protocol=librtmp\* --disable-protocol=libssh --disable-protocol=mmsh --disable-protocol=mmst --disable-protocol=pipe --enable-protocol=rtmp\* --enable-protocol=rtmp --enable-protocol=rtmpt --disable-protocol=rtp --disable-protocol=sctp --disable-protocol=srtp --disable-protocol=unix

其中，--enable-dxva2 --disable-hwaccels --enable-hwaccel=h264\_dxva2 --enable-hwaccel=hevc\_dxva2 用于打开h264硬件解码支持。没有第一步的头文件支持，由于ffmpeg的check，没办法从配置打开硬件解码进行编译。

--enable-encoder=mjpeg和-enable-muxer=image2两个选项是打开jpeg图片存储功能的最小要求。

--extra-ldflags="-L$HOME/ffmpeg/build/lib -static-libstdc++ -static-libgcc"用于减少libgcc的动态库链接，采用静态链接的方式

1. PC播放器编译的时候，引用libavutil/common.h会有编译错误，注释掉编译错误的代码就行。