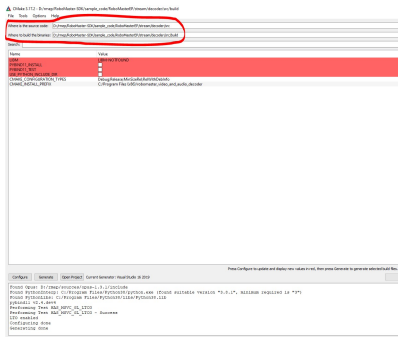
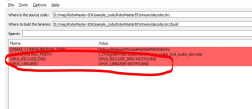


Compiling H264 under windows

1. Install Visual Studio 2019 (with only C++ development options)
2. download and install cmake and python <https://www.python.org/ftp/python/3.8.3/python-3.8.3.exe>
3. download <https://archive.mozilla.org/pub/opus/opus-1.3.1.tar.gz>
4. unzip opus and go into the folder where you unzip opus and go into the subdirectory win32/vc2015 in that folder and double click on opus.sln
5. make sure x64 and then build opus. "Build->Build Solution"
6. after build, make sure that is opus.lib in win32/vc2015/x65/Debug/
7. download boost https://dl.bintray.com/boostorg/release/1.73.0/source/boost_1_73_0.zip
8. build boost.
 - a) bootstrap
 - b) ./b2take note of :
The following directory should be added to compiler include paths:
D:\rmep\sources\boost_1_73_0
The following directory should be added to linker library paths:
D:\rmep\sources\boost_1_73_0\stage\lib
9. download the following and unzip:
 - a) pybind11 <https://github.com/pybind/pybind11/archive/v2.5.0.zip>
 - b) ffmpeg <https://ffmpeg.zeranoe.com/builds/>
ffmpeg dev : E.g. Win10 64bit <https://ffmpeg.zeranoe.com/builds/win64/dev/> (Select the one is for your OS and architecture) (Win64 - <https://ffmpeg.zeranoe.com/builds/win64/dev/ffmpeg-latest-win64-dev.zip>)
ffmpeg shared : E.g. Win10 64bit <https://ffmpeg.zeranoe.com/builds/win64/shared/> <https://ffmpeg.zeranoe.com/builds/win64/shared/ffmpeg-latest-win64-shared.zip>
 - c) opus-v0.9-win32 libraries <https://archive.mozilla.org/pub/opus/win32/opusfile-v0.9-win32.zip>
10. Use git to clone ("download") <https://github.com/dji-sdk/RoboMaster-SDK.git> (git clone <https://github.com/dji-sdk/RoboMaster-SDK.git>)
11. run cmake.
12. Make sure "Where is your source code?" is pointed to the folder where you download the Robomaster-SDK and the sample_code/RoboMasterEP/stream/decoder/src.
13. Make sure "Where to build the binaries" is pointed to the folder where you download the Robomaster-SDK and the same_code/RoboMasterEP/stream/decoder/src/build



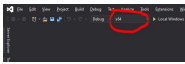
14. click on configure.
15. When you encountered error on opus libraries such as the below:



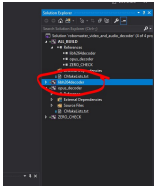
Edit the value of OPUS_INCLUDE_DIRS to the folder where you unzip opus-1.3.1 and the subdirectory include in that folder and edit value of OPUS_LIBRARIES to point to the location where VC compile and stored the opus.lib (step 5)

Once edited, clicked configure and generate again.

18. Go into the folder that you extract opus-1.3.1 and the include folder in it. Create a new folder opus in that include folder and copy all the files the include into the new folder opus.
19. Go into Robomaster-SDK/sample_code/RoboMasterEP/stream/decoder/src/build directory and double click on ALL_BUILD.vcxproj. Visual Studio will be launched.
21. Ensure that the architecture is as per the platform on which you are going to use the libraries.



21. click on the lib264decoder in the solution explorer pane and then enter "ALT+ENTER" to go into properties of lib264decoder

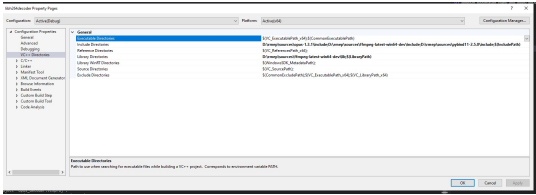


22. Make sure the "Include Directories" under VC++ Directories contains the path to:

- A) The include folder in the folder containing opus-1.3.1
- B) the include folder in the folder containing ffmpeg-latest-win64-dev
- c) the include folder in the folder containing pybind11-2.5.0

23. Make sure the "Library Directories" under VC++ Directories contains the directory path to:

- A) The lib folder in the folder containing ffmpeg-latest-win64



24. click ok.

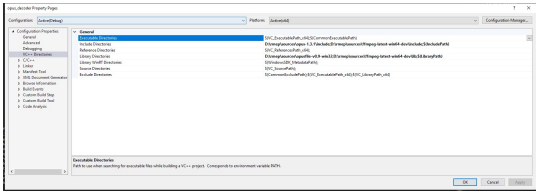
25. Click on opus_decoder in the solution explorer and enter "ALT+ENTER" to go into properties of opus_decoder.

26. Make sure the "Include Directories" in VC++ Directories contains the following path. If not, edit to add these path.

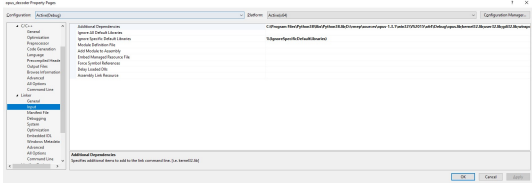
- a) the include directories in the folder containing opus-1.3.1
- b) the include directories in the folder containing ffmpeg-latest-win64-dev
- c) the include folder in the folder containing pybind11-2.5.0

27. Make sure the "Library Directories" contains the following path. If not, edit to add these path.

- a) the folder containing opusfile-v0.9-win32
- b) the lib directory in the folder containing ffmpeg-latest-win64-dev

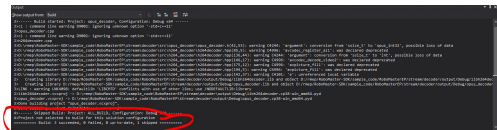


28) Go Linker->Input properties page, and add the path to the opus.lib file (created in step 5) into the "Additional Dependencies"



29. Click ok.

30. Click Build->Build Solution to build. When successful, you will have libh264decoder.cp38-win_amd64.pyd and opus_decoder.cp38-win_amd64.pyd in the RoboMaster-SDK\sample_code\RoboMasterEP\stream\decoder\output\Debug directory.



31. Copy all files in bin directory in ffmpeg-latest-win64-shared into the directory containing the two pyd files.