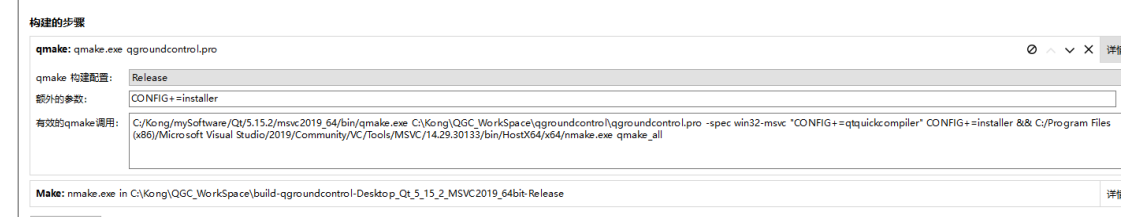
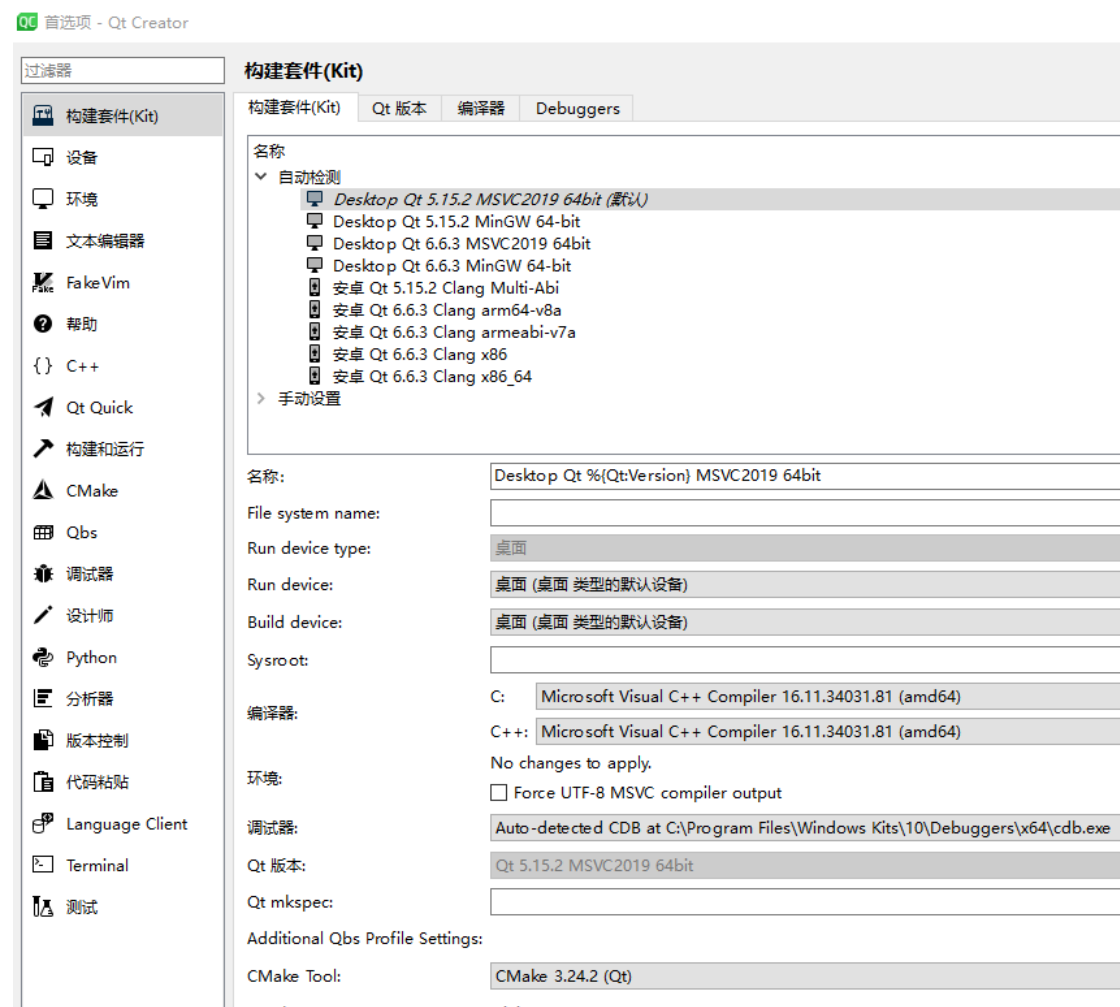


\*请先查看 QGC 官网源码构建说明（版本 Stable\_V4.4）：  
[https://docs.qgroundcontrol.com/Stable\\_V4.4/en/qgc-dev-guide/getting\\_started/](https://docs.qgroundcontrol.com/Stable_V4.4/en/qgc-dev-guide/getting_started/)

\*下面是我们二次开发 QGC(源码 V4.2.9)构建相关说明：

## 一、环境

1. Win10 系统
2. Qt 5.15.2
3. Visual Studio 2019 64 bit
4. 编译 PC 端，Qt Creator 中的配置显示：



5. 编译 Android 端：（JDK 11.0.23 、NDK21.3.6528147(默认使用) 和 NDK25.1.8937393)  
构建套件：安卓 Qt 5.15.2 Clang Multi-Abi

过滤器

构建套件(Kit)

设备

环境

文本编辑器

Fake Vim

帮助

C++

Qt Quick

构建和运行

CMake

Qbs

调试器

设计师

Python

分析器

版本控制

代码粘贴

Language Client

Terminal

测试

构建套件(Kit)

Qt 版本

编译器

Debuggers

名称

自动检测

Desktop Qt 5.15.2 MSVC2019 64bit (默认)

Desktop Qt 5.15.2 MinGW 64-bit

Desktop Qt 6.6.3 MSVC2019 64bit

Desktop Qt 6.6.3 MinGW 64-bit

安卓 Qt 5.15.2 Clang Multi-Abi

安卓 Qt 6.6.3 Clang arm64-v8a

安卓 Qt 6.6.3 Clang armeabi-v7a

安卓 Qt 6.6.3 Clang x86

安卓 Qt 6.6.3 Clang x86\_64

手动设置

名称:

安卓 Qt %{Qt:Version} Clang Multi-Abi

File system name:

Run device type:

Android设备

Run device:

Lenovo TB-J716F

Build device:

桌面 (桌面 类型的默认设备)

Sysroot:

编译器:

C:

Android Clang (C, arm, NDK 21.3.6528147)

C++:

Android Clang (C++, arm, NDK 21.3.6528147)

环境:

No changes to apply.

☐ Force UTF-8 MSVC compiler output

调试器:

安卓调试器 (Multi-Abi, NDK 21.3.6528147) GDB

Qt 版本:

Qt 5.15.2 for Android

Qt mkspec:

Additional Qbs Profile Settings:

CMake Tool:

CMake 3.24.2 (Qt)

过滤器

构建套件(Kit)

设备

环境

文本编辑器

Fake Vim

帮助

C++

Qt Quick

构建和运行

CMake

Qbs

调试器

设计师

设备

Android

QNX

SSH

安卓设置

JDK 位置:

C:\java\jdk

Android SDK的路径:

C:\Kong\NDK

安卓 NDK 列表:

C:\Kong\NDK\ndk\21.3.6528147

C:\Kong\NDK\ndk\25.1.8937393

✓ 安卓设置完成。 (SDK 版本: 9.0, NDK 版本: 21.3.6528147)

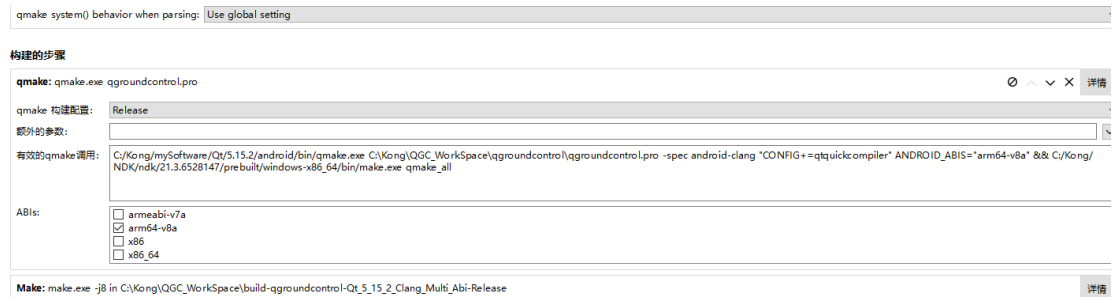
☒ 为安卓工具链自动创建套件

安卓 OpenSSL 设置 (可选)

OpenSSL 二进制位置:

C:\Kong\mySoftware\AndroidSDK\android\_openssl

✓ OpenSSL 设置完成。



5. Gstreamer 版本在源码文件中有说明：/src/VideoReceiver/README.md。Android 端使用的版本 gstreamer-1.0-android-universal-1.18.6。

## 二、代码

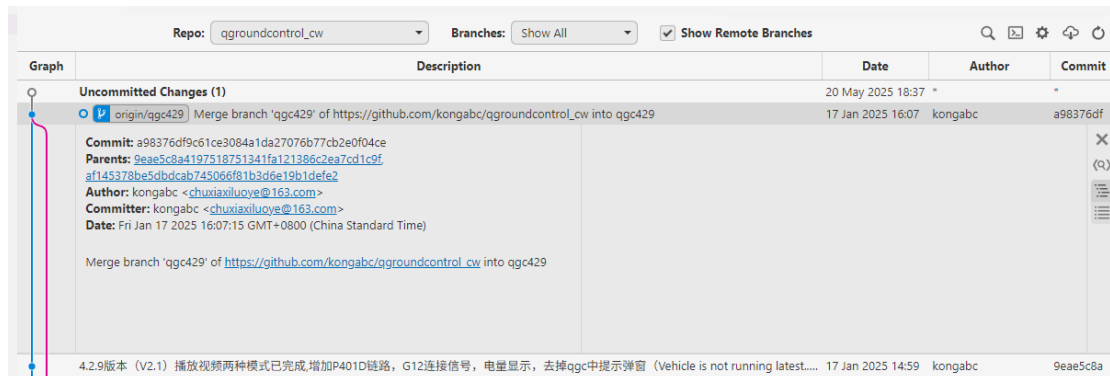
1. 获取代码，优先使用分支代码：

[origin/qgc429](https://github.com/kongabc/qgroundcontrol_cw.git) (Commit: [a98376df9c61ce3084a1da27076b77cb2e0f04ce](https://github.com/kongabc/qgroundcontrol_cw/commit/a98376df9c61ce3084a1da27076b77cb2e0f04ce))。

`git clone --recursive -j8 https://github.com/kongabc/qgroundcontrol\_cw.git`

强制切换远程分支'`git checkout -f origin/qgc429`'获得远程分支 `qgc429`

如果使用 Visual Studio Code 编译器切换分支，会报错，按提示需要把 `libs/shapelib` 文件删除，再 `checkout` 分支即可。



Qt Creator 中显示：



2. 更新子模块

`git submodule init`

`git submodule update`

或

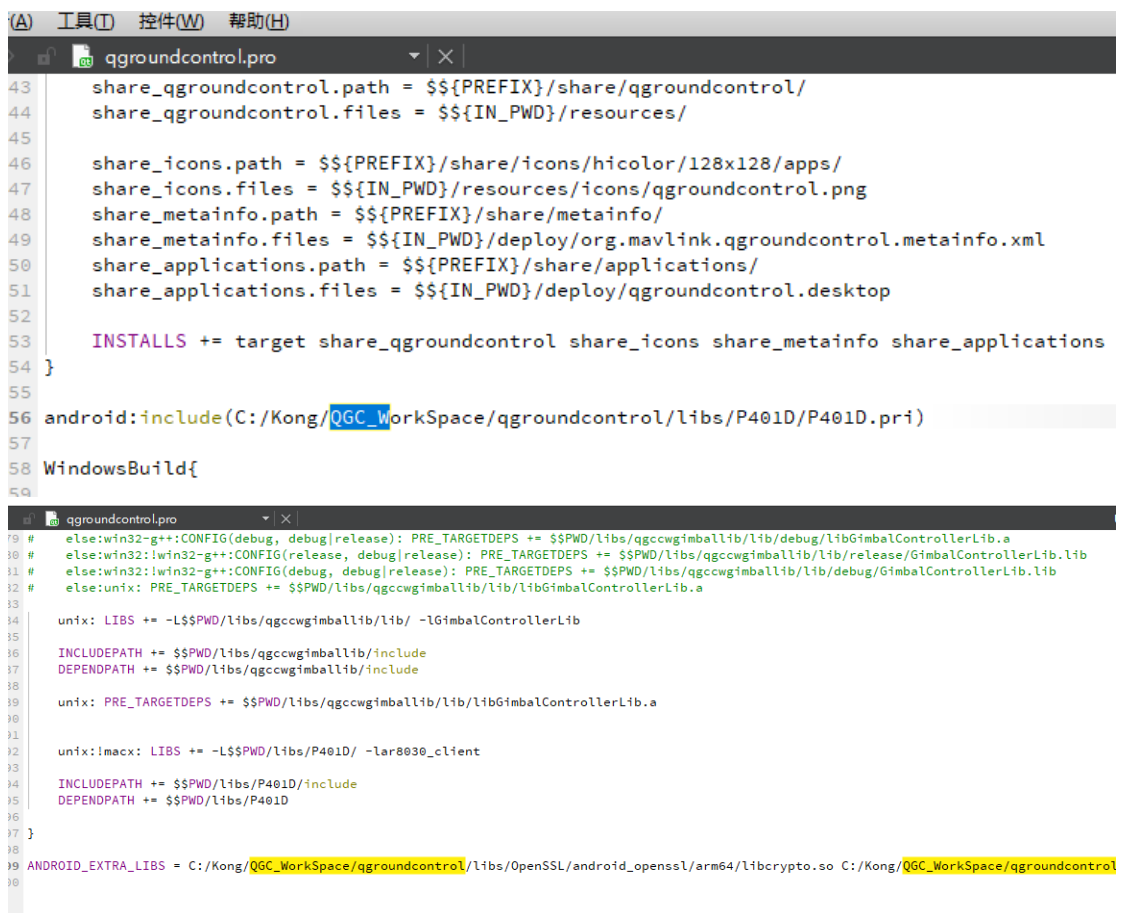
`git submodule update --init --recursive`

编译 android 端遇到的问题：

```
compiling C:\Kong\qgc_test\qgroundcontrol_cw\src\Camera\QGCCameraIO.cc
compiling C:\Kong\qgc_test\qgroundcontrol_cw\src\Camera\QGCCameraManager.cc
compiling C:\Kong\qgc_test\qgroundcontrol_cw\src\CmdLineOptParser.cc
compiling C:\Kong\qgc_test\qgroundcontrol_cw\src\Compression\QGCLZMA.cc
compiling C:\Kong\qgc_test\qgroundcontrol_cw\src\Compression\QGCZLib.cc
In file included from C:\Kong\qgc_test\qgroundcontrol_cw\src\Camera\QGCCameraControl.cc:11:
C:\Kong\qgc_test\qgroundcontrol_cw\src\VideoManager\VideoManager.h:43:10: fatal error: 'libavformat/avformat.h' file not found
#include "libavformat/avformat.h"
~~~~~
In file included from C:\Kong\qgc_test\qgroundcontrol_cw\src\api\QGCCorePlugin.cc:18:
C:\Kong\qgc_test\qgroundcontrol_cw\src\VideoManager\VideoManager.h:43:10: fatal error: 'libavformat/avformat.h' file not found
#include "libavformat/avformat.h"
~~~~~
1 error generated.
make: *** [Makefile:138689: arm64-v8a\QGCCorePlugin.obj] Error 1
make: *** Waiting for unfinished jobs....
1 error generated.
make: *** [Makefile:145814: arm64-v8a\QGCCameraControl.obj] Error 1
18:34:31: 进程"C:\Kong\NDK\ndk\21.3.6528147\prebuilt\windows-x86_64\bin\make.exe"退出，退出代码 2。
Error while building/deploying project qgroundcontrol (kit: 安卓 Qt 5.15.2 Clang Multi-Abi)
When executing step "Make"
18:34:31: Elapsed time: 00:35.
```

这是因为使用了 ffmpeg 库，参考下面第 3 步关于 Android 端的操作。

并且.pro 文件中要修改路径：QGC\_WorkSpace/qgroundcontrol 都要改成自己的目录



```
(A) 工具(T) 控件(W) 帮助(H)
qgroundcontrol.pro
43 share_qgroundcontrol.path = ${PREFIX}/share/qgroundcontrol/
44 share_qgroundcontrol.files = ${IN_PWD}/resources/
45
46 share_icons.path = ${PREFIX}/share/icons/hicolor/128x128/apps/
47 share_icons.files = ${IN_PWD}/resources/icons/qgroundcontrol.png
48 share_metainfo.path = ${PREFIX}/share/metainfo/
49 share_metainfo.files = ${IN_PWD}/deploy/org.mavlink.qgroundcontrol.metainfo.xml
50 share_applications.path = ${PREFIX}/share/applications/
51 share_applications.files = ${IN_PWD}/deploy/qgroundcontrol.desktop
52
53 INSTALLS += target share_qgroundcontrol share_icons share_metainfo share_applications
54 }
55
56 android:include(C:/Kong/QGC_WorkSpace/qgroundcontrol/libs/P401D/P401D.pri)
57
58 WindowsBuild{
59
60     else:win32-g++:CONFIG(debug, debug|release): PRE_TARGETDEPS += $$PWD/libs/qgccwgmimballib/lib/debug/libGimbalControllerLib.a
61     else:win32:win32-g++:CONFIG(release, debug|release): PRE_TARGETDEPS += $$PWD/libs/qgccwgmimballib/lib/release/GimbalControllerLib.lib
62     else:win32:win32-g++:CONFIG(debug, debug|release): PRE_TARGETDEPS += $$PWD/libs/qgccwgmimballib/lib/debug/GimbalControllerLib.lib
63     else:unix: PRE_TARGETDEPS += $$PWD/libs/qgccwgmimballib/lib/libGimbalControllerLib.a
64
65     unix: LIBS += -L$$PWD/libs/qgccwgmimballib/lib/ -lGimbalControllerLib
66
67     INCLUDEPATH += $$PWD/libs/qgccwgmimballib/include
68     DEPENDPATH += $$PWD/libs/qgccwgmimballib/include
69
70     unix: PRE_TARGETDEPS += $$PWD/libs/qgccwgmimballib/lib/libGimbalControllerLib.a
71
72     unix:imacx: LIBS += -L$$PWD/libs/P401D/ -lar8030_client
73
74     INCLUDEPATH += $$PWD/libs/P401D/include
75     DEPENDPATH += $$PWD/libs/P401D
76
77 }
78
79 ANDROID_EXTRA_LIBS = C:/Kong/QGC_WorkSpace/qgroundcontrol/libs/OpenSSL/android_openssl/arm64/libcrypto.so C:/Kong/QGC_WorkSpace/qgroundcontrol
```

### 3. 特别说明播放视频模块

代码中使用了两种方式，第一种保留 QGC 原来的播放逻辑，第二种加入了 ffmpeg 相关的播放视频逻辑。对于第二种说明如何引入 ffmpeg 库(编译生成的库与构建说明文档同目录：lib/android\_ffmpeg 和 lib/windows\_ffmpeg。 lib/ gstreamer-1.0-android-universal-1.18.5 /arm64/是我正在使用的 gstreamer，只上传了 arm64 文件)：

ffmpeg 库文件地址：[https://github.com/kongabc/build\\_instructions/tree/main/lib](https://github.com/kongabc/build_instructions/tree/main/lib)

**Window:** 把本地安装的 gstreamer 中 lib 和 include 中的文件替换成 windows\_ffmpeg 库中对应的文件（注：lib 目录下或许直接复制进去就行，实际编译试试。有问题的话就把原来文件中相同名字的库替换成 ffmpeg 文件中的 lib 库），把 bin 文件中的 dll 文件也替换。

	组织	新建	打开	选择	
磁盘 (C:) > gstreamer > 1.0 > msvc_x86_64 > lib >					
名称	修改日期	类型	大小		
avcodec.lib	2024/5/26 20:30	Object File Library	134 KB		
avdevice.lib	2024/5/26 20:30	Object File Library	14 KB		
avfilter.lib	2024/5/26 20:30	Object File Library	55 KB		
avformat.lib	2024/5/26 20:30	Object File Library	125 KB		
avutil.lib	2024/5/26 20:30	Object File Library	454 KB		
postproc.lib	2024/5/26 20:30	Object File Library	10 KB		
swresample.lib	2024/5/26 20:30	Object File Library	20 KB		
swscale.lib	2024/5/26 20:30	Object File Library	27 KB		
libavfilter.def	2020/10/29 6:06	Export Definition...	2 KB		
libswresample.def	2020/10/29 6:06	Export Definition...	1 KB		
libavcodec.def	2020/10/29 6:06	Export Definition...	6 KB		
libavcodec.la	2020/10/29 6:06	LA 文件	2 KB		
libavfilter.la	2020/10/29 6:06	LA 文件	2 KB		
libavformat.def	2020/10/29 6:06	Export Definition...	4 KB		
libavformat.la	2020/10/29 6:06	LA 文件	2 KB		
libavutil.def	2020/10/29 6:06	Export Definition...	12 KB		
libavutil.la	2020/10/29 6:06	LA 文件	2 KB		
libswresample.la	2020/10/29 6:06	LA 文件	2 KB		
libavcodec.dll.a	2020/10/29 6:06	A 文件	153 KB		
libavfilter.dll.a	2020/10/29 6:06	A 文件	220 KB		
	组织	新建	打开	选择	
磁盘 (C:) > gstreamer > 1.0 > msvc_x86_64 > bin					
名称	修改日期	类型	大小		
avcodec-58.dll	2020/10/29 6:06	应用程序扩展	91,390 KB		
avcodec-61.dll	2024/5/26 20:30	应用程序扩展	81,954 KB		
avdevice-61.dll	2024/5/26 20:30	应用程序扩展	4,348 KB		
avfilter-7.dll	2020/10/29 6:06	应用程序扩展	1,426 KB		
avfilter-10.dll	2024/5/26 20:30	应用程序扩展	39,958 KB		
avformat-58.dll	2020/10/29 6:06	应用程序扩展	32,866 KB		
avformat-61.dll	2024/5/26 20:30	应用程序扩展	17,853 KB		
avutil-56.dll			4,150 KB		
avutil-59.dll			2,654 KB		
bz2.dll			65 KB		
bz2.pdb			460 KB		
ffi-7.dll	2020/10/29 4:55	应用程序扩展	29 KB		

**Android:** 把 android\_ffmpeg/armv8-a/lib 的 so 文件复制到 gstreamer-1.0-android-universal-1.18.5/arm64/lib 文件中，include 文件中的几个文件替换成 android\_ffmpeg/armv8-a/include 下的几个文件（自测使用 gstreamer 库版本 1.18.6 替换也

能正常使用)。

	组织	新建	打开	选择	
> VideoReceiver > gstreamer-1.0-android-universal-1.18.5 > arm64 > lib >					
	名称 ^	修改日期	类型	大小	
✦	libass.a	2021/9/10 18:17	A 文件	1,500 KB	
✦	libass.la	2021/9/10 18:17	LA 文件	2 KB	
✦	libavcodec.a	2021/9/10 18:38	A 文件	139,275 KB	
✦	libavcodec.la	2021/9/10 18:38	LA 文件	2 KB	
✦	libavcodec.so	2024/8/14 15:38	SO 文件	11,379 KB	
	libavdevice.so	2024/8/14 15:38	SO 文件	51 KB	
	libavfilter.a	2021/9/10 18:38	A 文件	1,417 KB	
	libavfilter.la	2021/9/10 18:38	LA 文件	2 KB	
	libavfilter.so	2024/8/14 15:38	SO 文件	2,760 KB	
	libavformat.a	2021/9/10 18:38	A 文件	48,817 KB	
	libavformat.la	2021/9/10 18:38	LA 文件	2 KB	
	libavformat.so	2024/8/14 15:38	SO 文件	2,097 KB	
	libavutil.a	2021/9/10 18:38	A 文件	3,818 KB	
	libavutil.la	2021/9/10 18:38	LA 文件	1 KB	
	libavutil.so	2024/8/14 15:38	SO 文件	443 KB	

视频相关配置文件在 VideoReceiver.pri。

**注：**如果找不到 libGimbalControllerLib.a，重新导入一下，文件目录在 qgroundcontrol\_cw/libs/qgccwgimballib/lib/。也可自行编译 FFmpeg 库，ffmpeg 源码版本 PC 端版本：7.0.1 ,Android：4.3.8，目前编译的 Android 端只有 armv8-a。