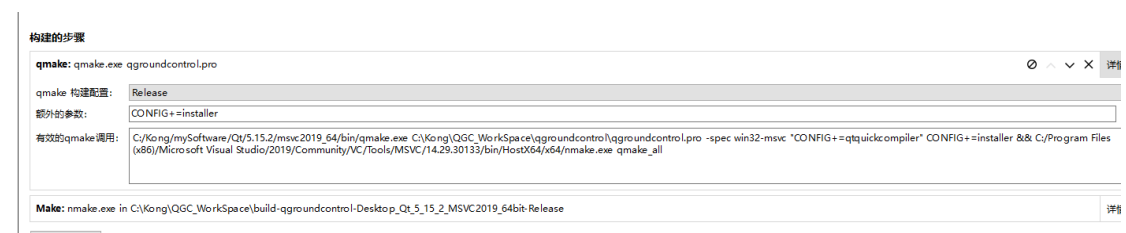
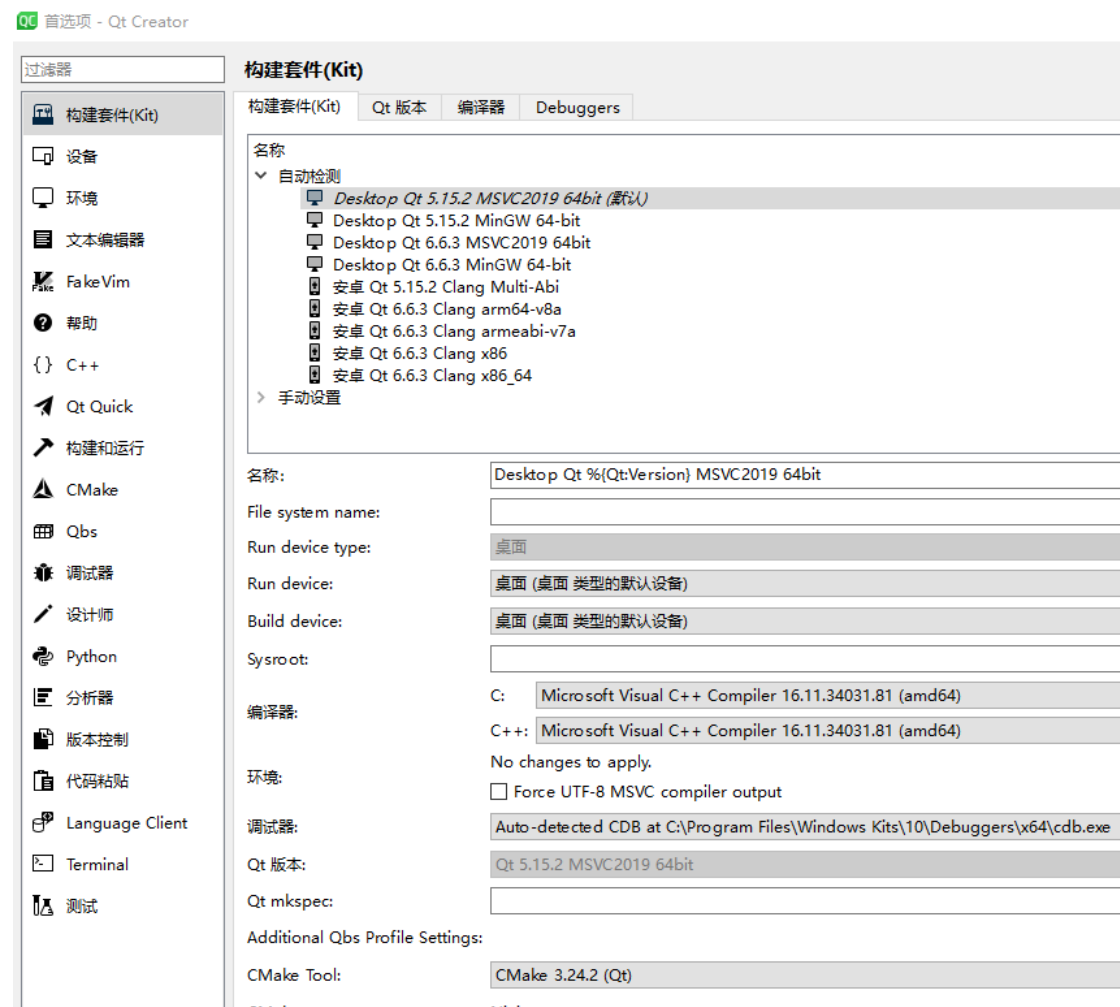


*请先查看 QGC 官网源码构建说明（版本 Stable_V4.4）：
https://docs.qgroundcontrol.com/Stable_V4.4/en/qgc-dev-guide/getting_started/

*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

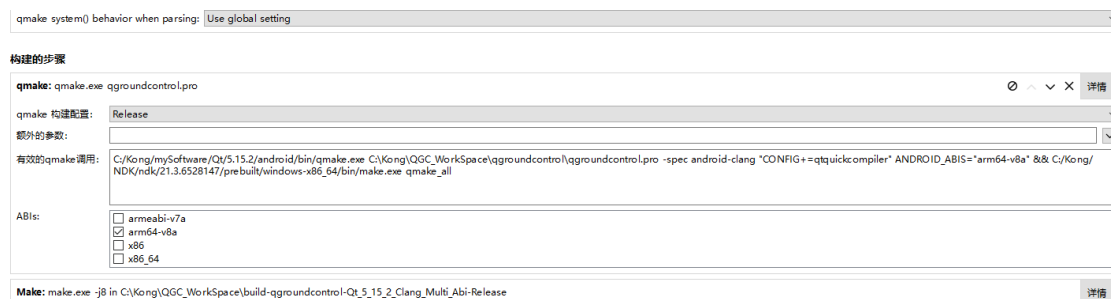
☒

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，优先使用的代码是分支 qgc429。

```
git clone --recursive -j8 https://github.com/kongabc/qgroundcontrol_cw.git
```

2. 更新子模块

```
git submodule update --init --recursive
```

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）：

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，把 gstreamer 的 bin 文件中的 dll 文件替换成 ffmpeg 库 bin 文件中的 dll 文件（gstreamer 包中原来关于 ffmpeg 相关的文件可不删除，直接复制 windows_ffmpeg 中的文件，具体要根据编译是否报错）。

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

	组织	新建	打开	选择	
也磁盘 (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	270 KB		

Android: 把 gstreamer-1.0-android-universal-1.18.5 中 arm64 文件中的 a 库替换成编译的 so 库（报错的话，就直接把 so 文件复制进去）。若有问题把 include 里面的文件也替换掉。

	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：编译生成的库 libGimbalControllerLib.a，环境配置同上 Android 配置。也可自行编译 FFmpeg 库，ffmpeg 源码版本 PC 端版本：7.0.1 ,Android：4.3.8。