



```
if (QL_APP_FEATURE_ALIPAY_IOT_SDK)
    target_link_libraries(${target} PRIVATE ql_app_alipay_iot)
endif()

if (QL_APP_FEATURE_HELLO)
    target_link_libraries(${target} PRIVATE ql_app_hello)
endif()

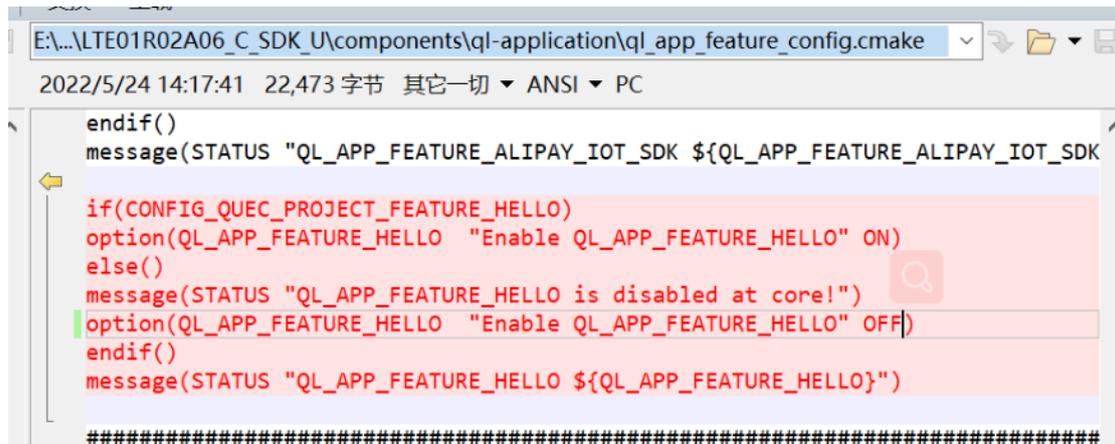
relative_glob(srcs *.h *.c)
```

```
#ifdef QL_APP_FEATURE_FILE
    //ql_fs_demo_init();
#endif

//ql_dev_app_init();
//ql_adc_app_init();

#ifdef QL_APP_FEATURE_HELLO
    ql_hello_app_init();
#endif
```

5. At the same time, add a switch to the ql\_app\_feature\_config.cmake file to control whether QL\_APP\_FEATURE\_HELLO is enabled.

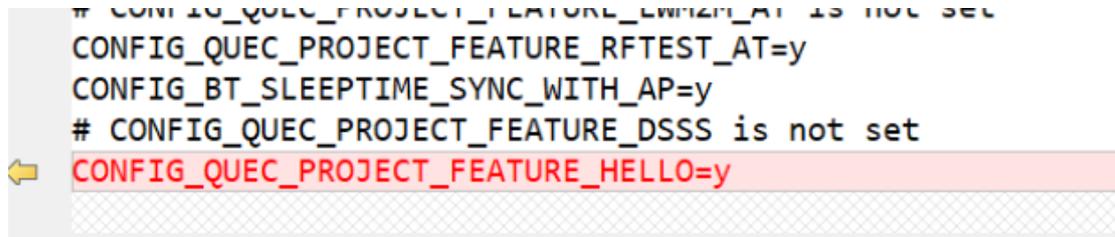


```
endif()
message(STATUS "QL_APP_FEATURE_ALIPAY_IOT_SDK ${QL_APP_FEATURE_ALIPAY_IOT_SDK}")

if(CONFIG_QUEEC_PROJECT_FEATURE_HELLO)
    option(QL_APP_FEATURE_HELLO "Enable QL_APP_FEATURE_HELLO" ON)
else()
    message(STATUS "QL_APP_FEATURE_HELLO is disabled at core!")
    option(QL_APP_FEATURE_HELLO "Enable QL_APP_FEATURE_HELLO" OFF)
endif()
message(STATUS "QL_APP_FEATURE_HELLO ${QL_APP_FEATURE_HELLO}")

#####
```

6. In components\ql-config\build\EC600UCN\_LB\8915DM\_cat1\_open\target.config, change the default value of CONFIG\_QUEEC\_PROJECT\_FEATURE\_HELLO=y to y.



```
# CONFIG_QUEEC_PROJECT_FEATURE_LWIP2_AT IS NOT SET
CONFIG_QUEEC_PROJECT_FEATURE_RFTEST_AT=y
CONFIG_BT_SLEEPTIME_SYNC_WITH_AP=y
# CONFIG_QUEEC_PROJECT_FEATURE_DSSS is not set
CONFIG_QUEEC_PROJECT_FEATURE_HELLO=y
```

7. In ql\_kernel\kconfig, the default value of CONFIG\_QUEEC\_PROJECT\_FEATURE\_HELLO is also set to Y.

```
E:\opencpu\8910\LTE01R02A06_C_SDK_U\components\ql-kernel\Kconfig
2022/5/24 11:42:00 19,610 字节 其它一切 ▾ ANSI ▾ PC

default n

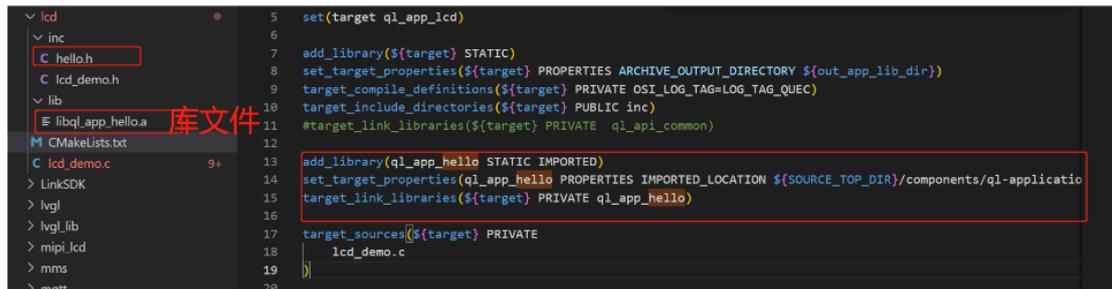
config CONFIG_QUEC_PROJECT_FEATURE_HELLO
    bool "BOOT CONFIG_QUEC_PROJECT_FEATURE_HELLO"
    default y

config QUEC_PROJECT_FEATURE_BOOT_SCATTER
    bool "BOOT SCATTER feature for Quectel"
    depends on QUEC_PROJECT_FEATURE
```

NOTE: If the above steps are performed without errors, but you still do not get the final result, try deleting the out folder and the target folder and then try again.

## 2. Adding a static library for the client:

1. Place the library files in a new "lib" folder within the directory where the library is being used. Place the corresponding ".h" files in the "inc" folder.



```
5 set(target qi_app_lcd)
6
7 add_library(${target} STATIC)
8 set_target_properties(${target} PROPERTIES ARCHIVE_OUTPUT_DIRECTORY ${out_app_lib_dir})
9 target_compile_definitions(${target} PRIVATE OSI_LOG_TAG=LOG_TAG_QUEC)
10 target_include_directories(${target} PUBLIC inc)
11 #target_link_libraries(${target} PRIVATE qi_api_common)
12
13 add_library(qi_app_hello STATIC IMPORTED)
14 set_target_properties(qi_app_hello PROPERTIES IMPORTED_LOCATION ${SOURCE_TOP_DIR}/components/ql-application...
15 target_link_libraries(${target} PRIVATE qi_app_hello)
16
17 target_sources(${target} PRIVATE
18     lcd_demo.c
19 )
20
```

Then, add the path code to the corresponding CMakeLists.txt file to call the files in the library.