

KF8 系列

库生成与使用说明

V1.1

2023 年 2 月

目录

库生成与使用说明 V1.1	1
目录	2
1 概述	3
2 库生成	3
2.1 库源码开发	3
2.2 编译输出库文件	3
2.2.1 集成开发环境 ChipON IDE KF8 功能输出库	3
2.2.2 工具命令实现输出库	5
3 库使用	6
3.1 工具层库配置使用	7
3.2 代码层引入声明示例	7
3.3 代码层调用示例	7
3.4 库应用编译结果示例	7
4 附录 A: KF8LIB 库管理程序选项	8

1 概述

本文档介绍 KungFu8 系列工具链的生成自定义库以及使用自定义库。

源码编译默认输出为后缀".o"文件，可作为库输入，也可直接提供其他项目使用；将.o文件添加(包括全新创建)到默认后缀“.lib”的库文件，用于其他项目使用。

注：输出“.o”，“.lib”的文件应为同目标机平台的编译环境产生，这里专指 ChipON KungFu8, 即库是运行平台专有化产物。

2 库生成

2.1 库源码开发

库的源码组成文件可以是汇编文件，也可以是.c文件，其可以包含头文件。库可提供变量和函数方法供其他项目使用。

源码编写可以使用宏定义，但宏展开编译，即不存在2次修改宏控制代码分支或改变参数。即不建议内部宏编写入提供用户的头文件中。

不建议存放与库服务不相关的对象，如变量或函数，若源码存在可以static修饰使能外部不可见。

2.2 编译输出库文件

当库发布时，需要提供库配套的头文件供其他项目包含，该头文件中应包括全局对外变量和函数的声明。推荐库以lib为库文件名前缀的标识。

KungFu8 系列芯片具有完整的工具链，其包括编译器 kf8cc、汇编器 kf8asm、链接器 kf8link、库管理器 kf8lib。可以使用集成开发环境 IDE 执行包含库的项目的编译，通过编译后执行"界面输入命令的伴随编译的完成库制作，也可以通过外部命令行实现库制作。支持可以通过编写命令行脚步调用 KungFu8 工具链执行源码的编译与制作库文件。

2.2.1 集成开发环境 ChipON IDE KF8 功能输出库

从项目资源管理器中选择对应的项目，右键进入"项目属性"页面。

在左侧"C/C++构建"下面选择"设置"。

配置模式具有独立的可配置属性，因此选择你需要的配置，如 Debug（默认带调试信息编译）和 Release（默认不带调试信息编译）。当前页面选择构建步骤，在"构建步骤"的"构建后步骤"的"命令"栏下面输入输出库命令，可以使用“;”或“&&”（推荐）间隔添加多组

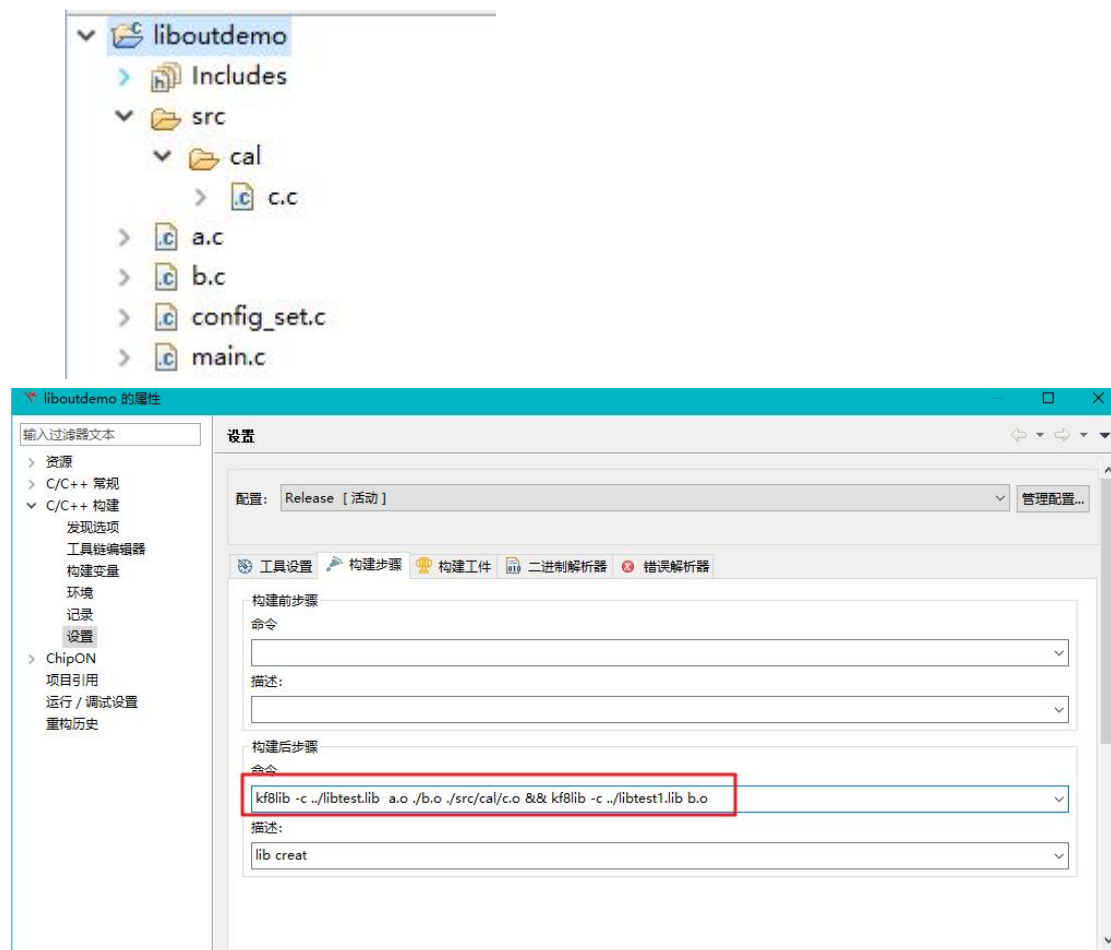
命令。

示例输入如下 `kf8lib -c ../outputlib.a a.o b.o` 完成库输出，其中参数 `-c` 为参数创建，其他参考 `-d` 为删除，`-r` 为替换，`-h` 显示命令行格式参数，其中创建不需要库的存在，删除和替换针对存在库进行文件单位成员维护。更多选项可以输出 `kf8lib --help` 查看。示例中的“../”对应系统的上一级文件路径，即构建项目时设定当前路径为当前的编译模式输出文件夹下，如 Release，此时对于这项目的根目录下。若对应的库的源文件存放在子文件夹中，输入库文件时应使用相对文件路径，如 `./bsp/a.o`，当源文件在项目根目录时可以直接使用文件名或相对文件名，如 `b.o` 或 `./b.o`。

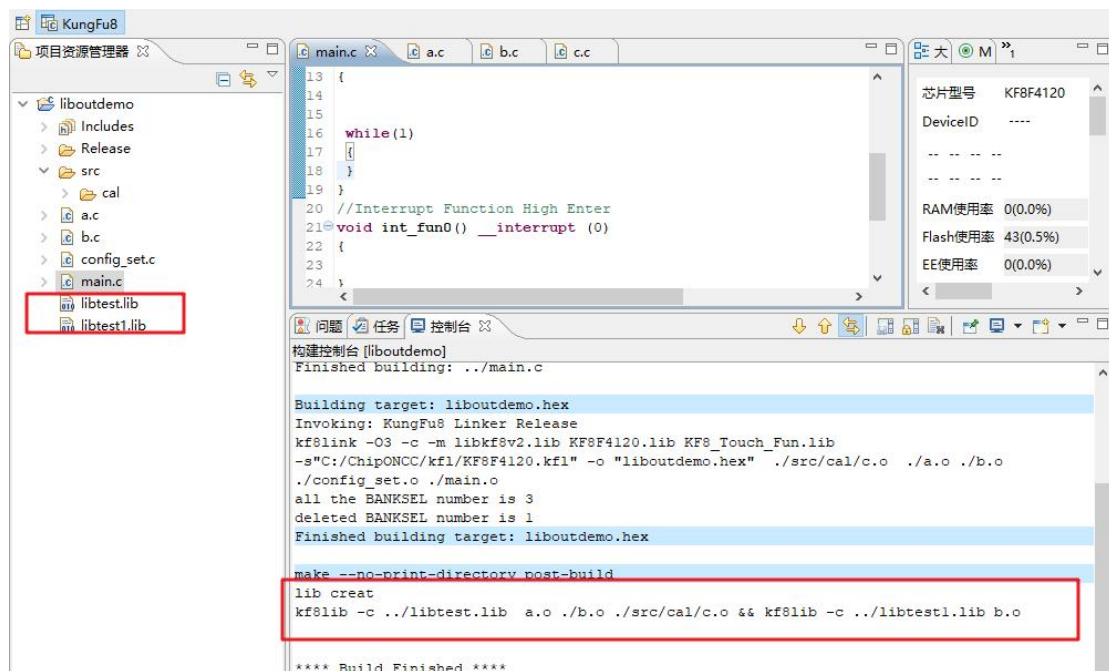
需要注意的是，默认 Debug 打开调试选型 `-debug`，因此可以制作库时关闭，ChipON 未提供匹配与验证的工具针对库文件做调试信息的删除支持【可以根据需要发布支持调试信息的库和正式发行不具有调试信息的库】。示意配置如下，输入样例“`kf8lib`

`-c ../libtest.lib a.o ./b.o ./src/cal/c.o && kf8lib -c ../libtest1.lib b.o`。

若针对库存在并引入项目，仅更新对应单个或多个源码库时，使用参数 `-r` 的替换替代，如：`kf8lib -r ../libtest.lib a.o`。



注：使用集成环境开发库时，只有编译正常通过时才出发构建后步骤，因此，该库源码所在项目应该是完整的，若非原始项目的独立库开发，可优先利用 IDE 创建一个最小样例项目，然后添加库相关文件并编译。构建后步骤的执行会输出在控制台中指示，如下：



2.2.2 工具命令实现出库

这里假定用于制作库文件已加工完成，即根据需保留调试信息。可通过 kf8lib 程序执行库文档创建、更新、移除等操作。可以在工具所在路径运行，或注册工作路径后运行。

常用的命令包括：添加库对象文件到库、删除库中某库对象文件、查看库中库对象文件。

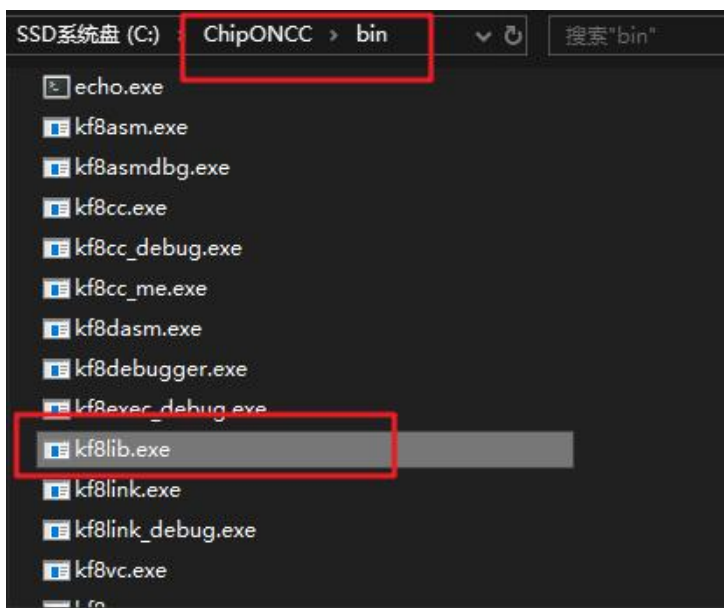
```
kf8lib -c libA.lib libA.o *.o
```

```
kf8lib -d libA.lib libA.o *.o
```

```
kf8lib -t libA.lib
```

```
for %%I in (*.o) do kf8lib -c %%~n1.lib %%~n1.o
```

ChipON IDE KF8 工具的库程序位于编译工具路径下，如图



可在该目录下打开批处理文件后运行库管理命令，也可以自行编写脚本注册 KungFu8 工具路径后操作。



3 库使用

库的使用支持汇编项目和 C 项目，但应提供满足其语言的头文件。如将库与库头文件复制到项目根目录与目录下。

项目属性下配置链接器选项的库路径与库传递。

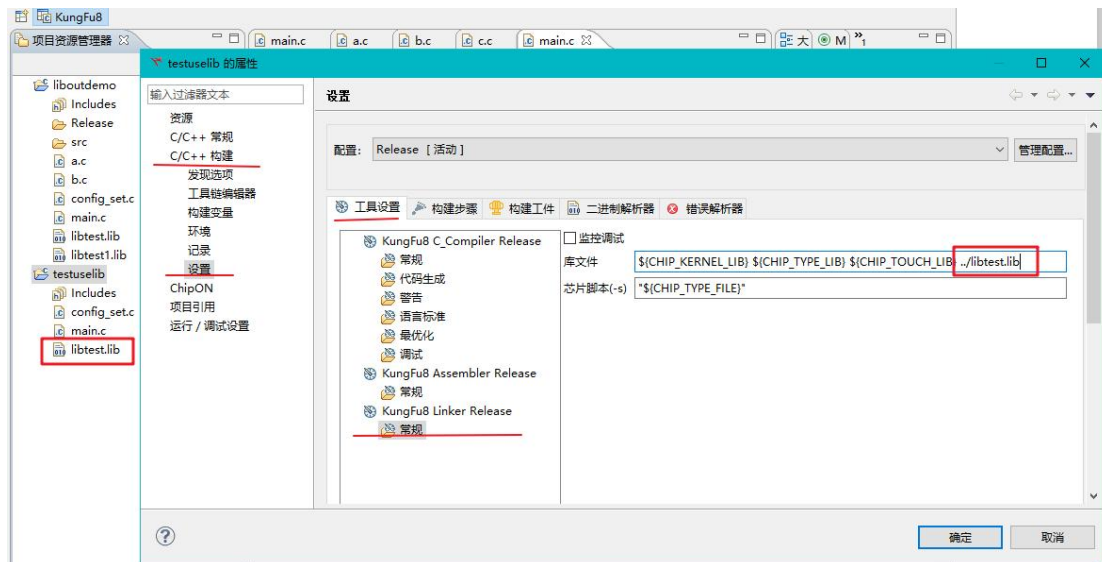
在使用库的项目下开发代码时，其应包括库声明的头文件。

3.1 工具层库配置使用

从"项目资源管理器"中选择对应的项目，右键进入项目属性。

在 C/C++ 构建下面选择设置。

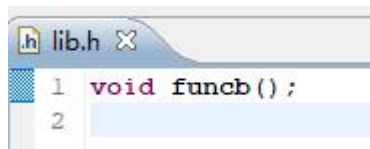
➤ 应分别选定配置模式 Debug 和 Release 进行库引用配置。不推荐选择所有配置，即 Debug 和 Release 模式存在差异化配置选项（否则所有工具选项将同步为一致）。在对应模式下，选项“链接”工具下的通用设定下，引用库选型下进行库文件的添加。



3.2 代码层引入声明示例

代码中包含其头文件：

```
#include "lib.h"
```



3.3 代码层调用示例

```
15 funcb();
```

3.4 库应用编译结果示例

若未配置库，如切换模式并该模式未追加库时的工具编译错误示例如下：

```
./Config_Set.0 ./main.0  
error: missing definition for symbol "_funcb", required by "./main.o"  
make: *** [makefile:27: testuselib.hex] Error 1
```

添加后编译通过

```
kf8link -O3 -c -m libkf8v2.lib KF8F4120.lib KF8_Touch_Fun.lib ../libtest.lib -s"C:/ChipONCC/kf1/KF8F4120.kf1" -o  
"testuselib.hex" ./config_set.o ./main.o  
all the BANKSEL number is 3  
deleted BANKSEL number is 1  
Finished building target: testuselib.hex
```

4 附录 A：kf8lib 库管理程序选项

C:\ChipONCC\bin>kf8lib --help

Usage: kf8lib [options] library [member]

Options: [defaults in brackets after descriptions]

-c, --create	Create a new library.
-d, --delete	Delete member from library.
-h, --help	Show this usage message.
-n, --no-index	Don't add symbol index.
-q, --quiet	Quiet mode.
-r, --replace	Add or replace member from library.
-s, --symbols	List global symbols in library.
-t, --list	List members in library.
-v, --version	Show version.
-x, --extract	Extract member from library.

Report bugs to:

www.chipon-ic.com