

T5L51 工程规范参考

要求

后面所有的 T5L 51 相关的开发都以此为模板

增加或者删除 lib_projects 中的模块可以直接复制修改/lib_projects/template/工程(基本设置已经配置好)

增加或者删除 user_projects 中的模块可以直接复制修改/ user_projects/template/工程(基本设置已经配置好)

各级目录功能简介

名称	修改日期	类型	大小
docs	2019/7/8 16:40	文件夹	
lib	2019/7/9 11:16	文件夹	
lib_projects	2019/7/9 11:28	文件夹	
user_projects	2019/7/9 11:25	文件夹	

docs:相关参考文档保存区域

lib:各个模块的 lib 文件保存区域

lib_projects:各个模块编译 lib 的 keil 工程及其源码

user_projects:各个项目的应用开发 demo keil 工程

各级目录详细介绍:

lib:

默认生成 模块名+.LIB, ALL.LIB 表示包含所有模块的总.LIB 文件

名称	修改日期	类型	大小
dgus_api.lib	2019/7/9 10:53	LIB 文件	6 KB
dwin.lib	2019/7/9 10:50	LIB 文件	121 KB
dwin_51.lib	2019/7/9 11:08	LIB 文件	3 KB
uart.lib	2019/7/9 11:06	LIB 文件	5 KB
usb.lib	2019/7/9 11:10	LIB 文件	107 KB

lib_projects:

/ALL 表示包含所有模块的编译 lib 的 keil 工程

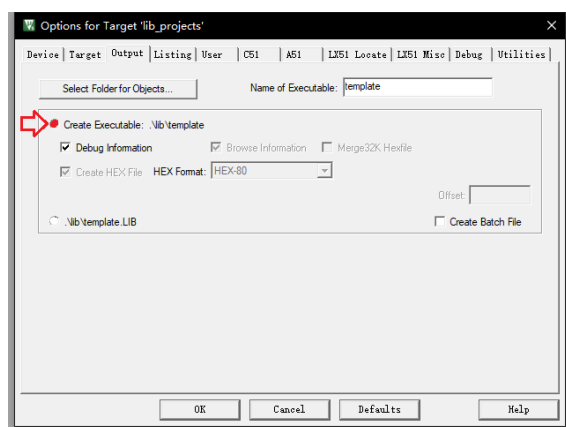
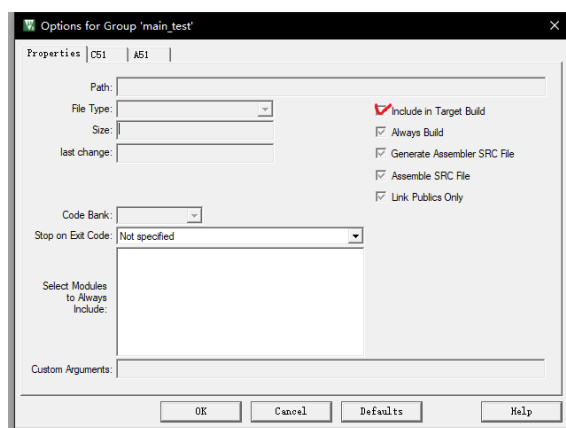
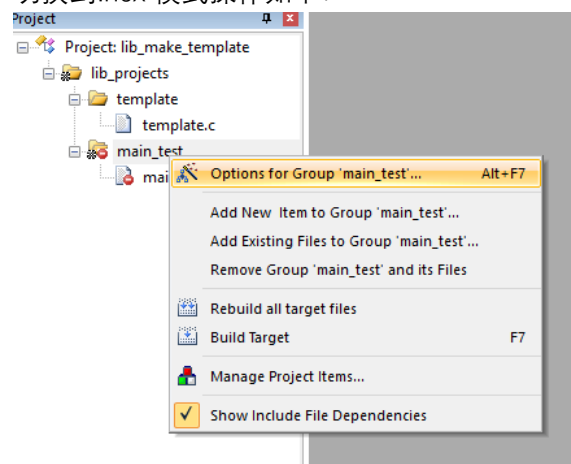
/模块名 表示包含当前模块的编译 lib 的 keil 工程及其源码

/template 编译 lib 的 keil 工程模板

(所有工程默认是处于编译.LIB 模式, 生成的.LIB 文件会自动复制到 根/LIB 目录供 user_projects 调用, 切换到生成.hex 模式并打开 main 文件包含, 则自动生成 T5L51.BIN)

名称	修改日期	类型	大小
all	2019/7/9 11:08	文件夹	
dgus_api	2019/7/9 11:08	文件夹	
dwin_51	2019/7/9 11:08	文件夹	
template	2019/7/9 13:17	文件夹	
uart	2019/7/9 11:08	文件夹	
usb	2019/7/9 11:10	文件夹	

切换到.hex 模式操作如下:



user_projects:

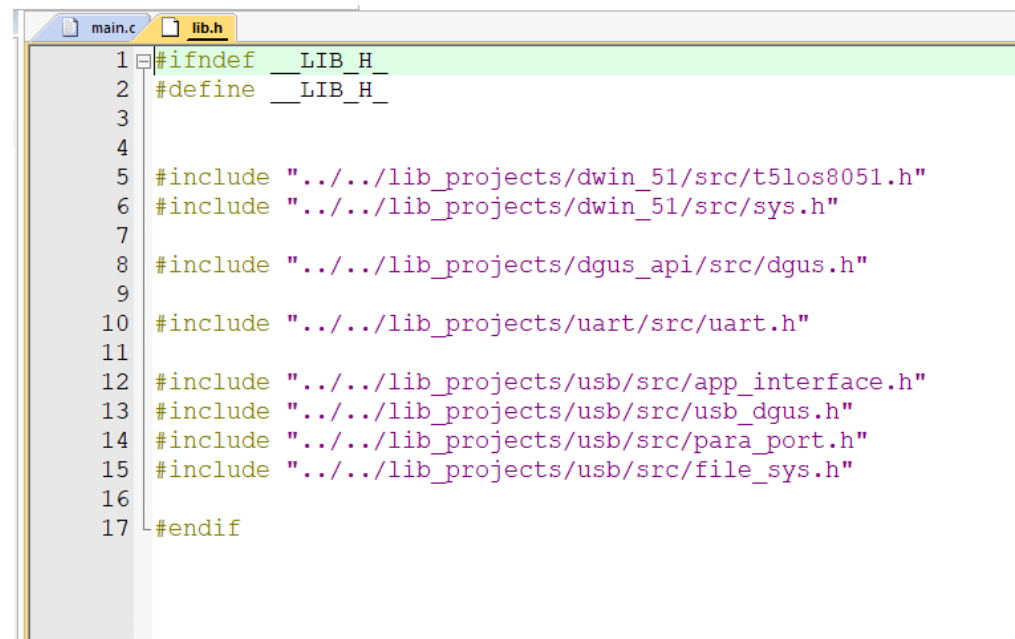
/xxx_demo xxx 项目的应用开发 demo keil 工程

/template 项目的应用开发 demo keil 工程模板

(默认 KEIL 工程会加入所有.LIB 文件到工程中，只需要修改 lib.h 的头文件内容即可实现不同的.LIB 编译入工程)

名称	修改日期	类型	大小
template	2019/7/9 13:20	文件夹	
usb_demo	2019/7/9 11:25	文件夹	

修改 lib.h 内容参考：



```
1 #ifndef __LIB_H
2 #define __LIB_H
3
4
5 #include "../../lib_projects/dwin_51/src/t5los8051.h"
6 #include "../../lib_projects/dwin_51/src/sys.h"
7
8 #include "../../lib_projects/dgus_api/src/dgus.h"
9
10 #include "../../lib_projects/uart/src/uart.h"
11
12 #include "../../lib_projects/usb/src/app_interface.h"
13 #include "../../lib_projects/usb/src/usb_dgus.h"
14 #include "../../lib_projects/usb/src/para_port.h"
15 #include "../../lib_projects/usb/src/file_sys.h"
16
17 #endif
```

只需要删除或者增加不同/lib_projects/xxx/src/xxxx.h 文件即可，工程中的 xxx.lib 文件无需删除。

