一、命名规范

文件名：

porting\_xxx.c

名字用小写, xxx 按照项目需求描述功能.

举例该工程使用的文件命令方法：

bsp.c 底层硬件初始化文件

porting\_gloves.c 操做手套相关的底层控制器 + 手套应用逻辑的函数实现文件

soft\_checkout.c 数据校验函数相关的文件，该文件的代码是与硬件无关的

宏：

全部大写字母

函数名：

首字母大写，单词以 \_ 分开

类型名:

结构体 \_T 结尾

枚举 \_E 结尾

联合 \_U 结尾

变量名：

全局变量 g\_ 开头，首字母大写,单词以 \_ 分开。

其它变量根据使用习惯命令，做到见名知意。

常见类型举例:

uint8\_t 以u8 开头, uint16\_t 以u16 开头, uint32\_t 以u32 开头;

int8\_t 以s8 开头, int16\_t 以s16 开头, int32\_t 以s32 开头;

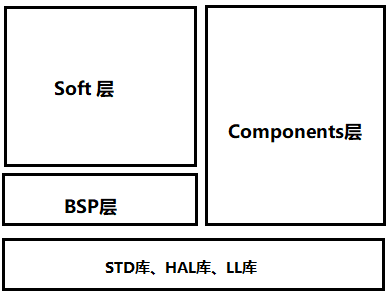
\_T 类型以 st 开头, \_E 类型以e开头, \_U 类型以 u 开头。

指针类型以 p 开头, 结构体指针以 pst 开头

二、代码结构

代码总体按照层次化和模块化的准则编写。

简介：



STD、HAL、LL为 ST 提供的官方库;

BSP层为调用官方库进行实现的底层硬件初始化。

该层在文件结构中会表现出BSP文件夹中的bsp.c，以及Components 层中组件库函数中的底层初始化，如USB 库中的usbd\_conf.c

Components 层可以称为组件块，该模块最大的特点是器件都有属于自己的库，该库由器件的官网进行维护，代码量较多，库函数已经成型且库本身具有层次化的调用关系，因此可以将其需要的底层函数放入该模块中。

Soft 层中实现的代码都与硬件控制器无关，如校验函数、栈、队列、链表函数的实现等等。

1. 文件夹组织结构

Doc

Drivers

EWARM

Users

Doc

文档说明

Drivers

驱动文件的汇总，该文件夹中会有例如底层的BSP文件夹，组件块中的mpu6文件夹，Soft层中Soft文件夹，其中Soft文件中会有soft\_xxx 开头的与硬件无关的代码实现文件也会有porting\_xxx 开头的操作底层控制器的文件，此处没有porting\_xxx 另开一个文件夹只是为了在工程头文件的包含中少些一个文件路径。

EWARM

IAR的配置和编译产生的中间文件

Users

最顶层的函数实现文件，如：main.c 和stm32f4xx\_it.c

1. 其它编码习惯

头文件包含问题

尽量在源文件中包含使用到的头文件。

头文件中如果要包含其它头文件，要有明确的上下层关系，需要上层包含下层。

关于 extern 的使用

如果在源文件中使用时需要在后面缀明外部调用变量或者函数的文件位置