|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 作者 | 修改内容 |
| V1.0 | 2019-7-24 | 黄庆庆 | 试行版 |
| V1.1 | 2019-7-24 | 黄庆庆 | 增加限制“不能跨层调用”  变量名不能随意起 |
| V1.2 | 2019-12-4 | 黄庆庆 | 初级要求增加 11，12，13，3的描述中“尽量”改为“禁止”。 |
| V1.3 | 2020-1-8 | 黄庆庆 | 增加代码注释风格的要求 |

初级要求：

1. 不用u8 u32 u16 ，unsigned char,unsigned short,这些类型， 使用uint8\_t ，uint16\_t ，uint32\_t 。
2. 非外部使用的接口函数，都使用static声明和定义。
3. 全局变量都定义成static，限制作用域，禁止从外部直接访问。
4. 模块化编程
   1. 每个模块的文件命名都加上模块所属层级的前缀。
   2. 模块内的给外部调用的接口函数都加上模块层级的前缀。
5. 变量和函数都使用下划线命名法。
6. 只有枚举值和宏定义使用全大写。
7. 枚举类型和结构体类型，函数类型 都是用小写加\_t后缀。
8. 模块的接口函数声明放在.h文件中，非接口的内部函数声明放在.c文件中。
9. =，==，>,<, &,|| &&,| 等运算符号的左右各加一个空格。
10. 变量命名不要使用 a,b,c,d这种没有实际意义的名称。
11. 注释要求：
    1. 注释风格要符合doxygen风格，具体参照<https://www.jianshu.com/p/9464eca6aefe>。
    2. 每个函数都要有注释。函数被调用次数在2次以上的要写函数描述，输入参数描述，输出参数描述。
    3. 每个全局变量都要有注释。
    4. 每个结构体，枚举类型，宏定义都要有注释。
    5. 关键语句增加注释。
12. 判断条件中不要出现数字，全部使用宏定义和枚举值。
13. 函数定义和调用时，逗号后＋空格。
14. 左右大括号“{”,”}”,都另起一行写。左大括号“{”后，下一行代码前+4个空格（一个制表符）和左大括号“{”对齐。

高级要求：

模块化分层，一般建议自底向上分为板级驱动层，外设驱动层，基础服务层（外设驱动的封装），应用功能模块层，应用层，系统层

* 板级驱动层：芯片库文件
* 外设驱动层：按键，灯，加速度计，外部flash，字库，摄像头，灯
* 基础服务层：对外设驱动的封装，上层只掉寄出服务层对驱动的封装，这样切换驱动时只需要添加外设驱动模块和修改相应的基础服务模块即可。
* 应用功能模块层：对整个产品的需求进行细分，每一个需求都是一个功能模块。
* 应用层：应用层应该按照产品使用场景和方式定义状态机，不同情况启动和关闭相应的功能。
* 系统层：一般包含，系统调度模块，系统定时器模块，事件队列模块。
* 全局层：定义一些通用的算法，函数，配置，比如CRC校验，内部flash存储配置信息

板级驱动层

外设驱动层

基础服务驱动层

功能模块层

应用层

系

统

层

全

局

层

1. 一般一个模块一个文件，相对复杂的模块可以分多个文件，外部接口都放在一个文件中。
2. 只可以上层调用下层的接口，下层不能直接调用上层接口
3. 下层向上层反馈只可以采用回调函数的方式或者使用系统事件队列。
4. 尽量不要跨层调用接口