嵌入式 C 语言代码规范

作者：　陈凯
日期：2021.06.11

第一章 排版格式

1.1 代码缩进

代码缩进要使用制表符，即 TAB 键，不要使用空格键缩进！一般情况下设置 TAB 为 4 个字符。在 switch 语句中，“switch”和“case” 标签应该对齐处于同一列，不要缩进 case，缩进 break，如下所示：

```
switch(buffer){
    case 1:
        ...
        break;
    case 'a':
        ...
        break;
    default:
        ...
        break;
}
```

1.2 代码行规范

1. 一行只写一条语句，不能把多个语句放到一行中。
2. 如果一行代码长度较长，需要分开多行编写，在低优先级操作符处划分新行，操作符放在新行之首，并且进行缩进。if、for、do、while、case、swich、default 等语句单独占用一行。

1.3 括号与空格

1.3.1 括号

在非函数程序控制块中，如 if、switch、for、do、while 等，应该把起始大括号“{”放到行尾，把结束大括号“}”放到行首，如下所示：

```
if (true) {
    /* ... */
    ...
}
else{
    /* ... */
    ...
}
```

在函数中，起始大括号要放置到下一行的开头，如下所示：

```
int function(int a)
{
    /* ... */
    ...
}
```

1.3.2 空格

1. 二元或者三元操作符两侧都要加一个空格，如：= + - < > * / % | & ^ <= >= == != ? : 。
2. 一元操作符后不要加空格，如： & * + - ~ ! sizeof typeof alignof __attribute__ defined。
3. 自加或自减一元操作符前后都不加空格，如：++ --。
4. 结构体成员操作符“.”“->”前后不加空格。
5. 逗号、分号只在后面添加空格，如下所示：

```
int a, b, c;
```

6. 注释符“/*”和“*/”与注释内容之间要添加一个空格。

1.4 注释

1.4.1 注释风格

注释意在让别人看到注释就明白你的代码其中的含义和用法，不要过度注释，注释风格应该使用：

```
/* ... */

尽量不使用下面这种注释
// ...
```

对于多行注释，每一行的开始处都应放置符号“*”，并且所有行的“*”要对齐在一列上，如下所示：

```
/*
 * This is a multi-line comment.
 *
 * Description: Write the description here,
 * A column of asterisks is left on the left,
 * the beginning and ending asterisks are not written.
 *
 */
```

1.4.2 文件信息注释

在文件开始应该对本文件做一个总体的、概括性的注释，如：文件名、作者、日期、版权声明、版本号、文件描述、修改日志等，如下所示：

```
/*********************************************************************************
 * 文 件 名   : main.c
 * 作    者   : Lemon
 * 创建日期   : 2021 年 6 月 11 日
 * 文件描述   : FreeRTOS Demo 芯片：MSP430FR5989
 * 版权声明   : Copyright (C)2021 Lemon. All Rights Reserved
 * 版 本 号   : V1.2
 * 其    他   :
 * 修改日志   : V1.2 20210609 by Lemon, 增加反向检测
*********************************************************************************
/
```

1.4.3 函数的注释

函数需要注释其作用，参数的含义以及返回值的含义，放在函数之前，如下所示：

```c
/*********************************************************************
 * 函 数 名  : main
 * 作    者  : Lemon
 * 创建日期  : 2021 年 6 月 11 日
 * 函数功能  : 用户主函数入口
 * 输入参数  : void
 * 返 回 值  : int
 * 调用关系  :
 * 其    它  :
*********************************************************************/
int main(void)
{
    /* 系统主循环 */
    while(1)
    {
    }
    return 0;
}
```

第二章 命名风格

2.1 命名规则

1.命名一定要清晰！要使用完整的单词或者大家都知道的缩写，让别人一读就懂，避免不必要的误会

2.除了常用的缩写以外，不要使用单词缩写，更不要用汉语拼音！！！

3.不要使用单字节命名变量，但是允许使用 i，j，k 这样的作为局部循环变量。

2.2 文件命名

文件统一采用小写加下划线命名，如 system_config.c。

2.3 变量命名

变量名一定要有意义，并且意义准确，采用大驼峰法定义。比如表示旋转计数的变量，如下命名：

```c
int RotationCounter;
```

全局变量必须在头部加入 g_ -> g_RotationCounter。

静态变量必须在头部加入 s_ -> s_RotationCounter。

在使用 RTOS 编程中，尽量避免使用全局变量。

2.4 函数命名

函数采用单词 + 下划线的方式定义，并且首字母大写。比如表示系统初始化，如下命名：

```
void Initial_System(void)
{
    ...
}
```

2.5 宏定义

对于宏定义命名，建议使用大写，单词之间使用下划线“_”连接在一起，如下命名：

```
#define CONSTANT 12345
```