文档类型	开发文档
保密级别	公开

技 术 报 告

名　称：C 语言软件设计规范

编　号：LY-STD001-2019

版本号：V1.0

编　　制：乐育科技

审　　核：

批　　准：

日　　期：2019 年 05 月 01 日

目　录

版本

序号	修订日期	版本号	修订内容	修订人
1	2019/05/01	1.0.0	初稿	乐育科技

1 文档目的

本文档详细介绍了 C 代码的书写规则以及规范，包括排版、注释、命名规范等，紧接着是“C 文件”模板和“H 文件”模板，并对这两个模板进行了详细的说明。

使用代码书写规则和规范可以使程序更加规范和高效，对代码的理解和维护起到至关重要的作用。

2 排版

（1）程序块要采用缩进风格编写，缩进的空格数为 2 个。对于由开发工具自动生成的代码可以有不一致。

（2）须将 Tab 键设定为转换为 2 个空格，以免用不同的编辑器阅读程序时，因 Tab 键所设置的空格数目不同而造成程序布局不整齐。对于由开发工具自动生成的代码可以有不一致。

（3）相对独立的程序块之间、变量说明之后必须加空行。

例如：

```
int tick;
int hour;
--------------------------------空行隔开--------------------------------
hour = tick / 3600;
--------------------------------空行隔开--------------------------------
if(hour >= 59)
{
    //program code
}
```

（4）不允许把多个短语句写在一行中，即一行只写一条语句。

例如：

```
int recData1 = 0;    int recData2 = 0;
```

应该写为

```
int recData1 = 0;
```

```
int recData2 = 0;
```

（5）if、for、do、while、case、switch、default 等语句自占一行，且 if、for、do、while 等语句的执行语句部分无论多少都要加括号{}。

例如：

```
if(s_iFreqVal > 60)
    return;
```

应该写为

```
if(s_iFreqVal > 60)
{
    return;
}
```

（6）在两个以上的关键字、变量、常量进行对等操作时，它们之间的操作符之前、之后或者前后要加空格；进行非对等操作时，如果是关系密切的立即操作符（如->），后不应加空格。

例如：

```
int a, b, c;
if(a >= b && c > d)
a = b + c;
a *= 2;
a = b ^ 2;
*p = 'a';
flag = !isEmpty;
p = &mem;
p->id = pid;
```

3 注释

注释是源码程序中非常重要的一部分，通常情况下规定有效的注释量不得少于 20%。其原则是有助于对程序的阅读理解，所以注释语言必须准确、简明扼要。注释不宜太多也不宜太少，内容要一目了然，意思表达准确，避免有歧义。总之该加注释的一定要加，不必要的地方就一定别加。

（1）边写代码边注释，修改代码同时修改相应的注释，以保证注释与代码的一致性。不再有用的注释要删除。

（2）注释的内容要清楚、明了，含义准确，防止注释二义性。

（3）避免在注释中使用缩写，特别是非常用缩写。

（4）注释应考虑程序易读及外观排版的因素，使用的语言若是中、英兼有的，建议多使用中文，除非能用非常流利准确的英文表达。

4 命名规范

标识符的命名要清晰、明了，有明确含义，同时使用完整的单词或大家基本可以理解的缩写，避免是人产生误解。

较短的单词可通过去掉“元音”形成缩写，较长的单词可取单词的头几个字母形成缩写；一些单词有大家公认的缩写。

例如：

message 可缩写为 msg；

flag 可缩写为 flg；

increment 可缩写为 inc；

4.1 三种常用命名方式介绍

（1）骆驼命名法（camelCase）

骆驼式命令法，正如它的名称所表示的那样，是指混合使用大小写字母来构成变量和函数的名字。例如，下面是用骆驼式命名法命名的函数： printEmployeePayChecks()。

（2）帕斯卡命名法（PascalCase）

与骆驼命名法类似。只不过骆驼命名法是首字母小写，而帕斯卡命名法是首字母大写，如：public void DisplayInfo()。

（3）匈牙利命名法（Hungarian）

匈牙利命名法通过在变量名前面加上相应的小写字母的符号标识作为前缀， 标识出变量的作用域， 类型等。这些符号可以多个同时使用，顺序是先 m_（成员变量），再简单数据类型，再其他。例如：m_iFreq, 表示整型的成员变量。匈牙利命名法关键是：标识符的名字以一个或者多个小写字母开头作为前缀；前缀之后的是首字母大写的一个单词或多个单词组合，该单词要指明变量的用途。

4.2 函数命名（文件命名与函数命名相同）

函数名应该能体现该函数完成的功能，可采用动词+名词的形式。关键部分应该采用完整

的单词，辅助部分若太常见即可采用缩写，缩写应符合英文的规范。每个单词的第一个字母大写。

例如：

```
AnalyzeSignal();
SendDataToPC();
ReadBuffer();
```

4.3 变量

（1）头文件为防止重编译须使用类似于_SET_CLOCK_H_的格式，其余地方应避免使用以下划线开始和结尾的定义。

如：

```
#ifndef _SET_CLOCK_H_
#define _SET_CLOCK_H_
...
#end if
```

（2）常量使用宏的形式存在，且宏中的所有字母均为大写。

例如：

```
#define       MAX_VALUE     100
```

（3）枚举命名时，枚举类型名应按照 EnumAbcXyz 的格式，且枚举常量均为大写，不同单词之间用下划线隔开。

例如：

```
typedef enum
{
  TIME_VAL_HOUR = 0,
  TIME_VAL_MIN,
  TIME_VAL_SEC,
  TIME_VAL_MAX
}EnumTimeVal;
```

（4）结构体命名时，结构体类型名应按照 StructAbcXyz 的格式，且结构体的成员变量应按照骆驼命名法。

例如：

```
typedef struct
{
  short hour;
  short min;
```

```
    short sec;
}StructTimeVal;
```

（5）在本文档中，静态变量有两类，函数外的定义的静态变量称为文件内部静态变量，函数内定义的静态变量称为函数内部静态变量。注意，文件内部静态变量均定义在“内部变量区”。这两种静态变量命名格式一致，即 s_+变量类型（小写）+变量名（首字母大写）。变量类型包括 i（整型）、f（浮点型）、arr（数组类型）、struct（结构体类型）、b（布尔型）、p（指针类型）。

例如：

s_iHour，s_arrADCConvertedValue[10]，s_pHeartRate

（6）函数内部的非静态变量即为局部变量，其有效区域仅限于函数范围内，局部变量命名采用骆驼命名法，即首字母小写。

例如：

timerStatus，tickVal，restTime

（7）为了最大限度的降低模块之间的耦合，本文档不建议使用全局变量，如若非不得已必须使用，则按照 g_+变量类型（小写）+变量名（首字母大写）进行命名。

5 C 文件模板

每个 C 文件模块由模块描述区、包含头文件区、宏定义区、枚举结构体定义区、内部变量区、内部函数声明区、内部函数实现区以及 API 函数实现区组成。

下面是各个模块的示意：

5.1 模块描述区

```
/*********************************************************************************************************
* 模块名称: SendDataToHost.c
* 摘　　要: 发送数据到 HOST
* 当前版本: 1.0.0
* 作　　者: XXX
* 完成日期: 20XX 年 XX 月 XX 日
* 内　　容:
* 注　　意: none
*********************************************************************************************************
* 取代版本:
* 作　　者:
* 完成日期:
* 修改内容:
* 修改文件:
*********************************************************************************************************/
```

5.2 包含头文件区

```
/*****************************************************************************************************************
*                                                包含头文件
*****************************************************************************************************************/
#include   "SampleSignal.h"
#include   "AnalyzeSignal.h"
#include   "ProcessSignal.h"
```

5.3 宏定义区

```
/*****************************************************************************************************************
*                                                宏定义
*****************************************************************************************************************/
#define   ALPHA   2048        //宏定义必须是全部大写，格式为 ABC_XYZ
```

5.4 枚举结构体定义区

```
/*****************************************************************************************************************
*                                                枚举结构体定义
*****************************************************************************************************************/
//定义枚举
//枚举类型为 EnumTimeVal，枚举类型的命名格式为 EnumXxYy
typedef enum
{
  TIME_VAL_HOUR = 0,
  TIME_VAL_MIN,
  TIME_VAL_SEC,
  TIME_VAL_MAX
}EnumTimeVal;

//定义一个时间值结构体，包括三个成员变量，分别是 hour，min 和 sec
//结构体类型为 StructTimeVal，结构体类型的命名格式为 StructXxYy
typedef struct
{
  short hour;
  short min;
  short sec;
}StructTimeVal;
```

5.5 内部变量区

```
/*****************************************************************************************************************
*                                                内部变量
*****************************************************************************************************************/
static i16 s_iSignalSample = 0;      //信号采样值
```

5.6 内部函数声明区

```
/*****************************************************************************************************************
*                                                内部函数声明
*****************************************************************************************************************/
static void SampleSignalPerSec(void* pBuf);       //每隔 2ms 采样一次信号
```

5.7 内部函数实现区

```
/*****************************************************************************************************************
*                                                内部函数实现
*****************************************************************************************************************/
/*****************************************************************************************************************
* 函数名称: SampleSignal
* 函数功能: 采样信号
* 输入参数: void
* 输出参数: void
* 返 回 值: void
* 创建日期: 20XX 年 XX 月 XX 日
* 注       意:
*****************************************************************************************************************/
static   void   SampleSignal(void)
{
}
```

5.8 API 函数实现区

```
/*****************************************************************************************************************
*                                                API 函数实现
*****************************************************************************************************************/
/*****************************************************************************************************************
* 函数名称: Task
* 函数功能: 任务
* 输入参数: void
* 输出参数: void
* 返 回 值: void
* 创建日期: 20XX 年 XX 月 XX 日
* 注       意:
*****************************************************************************************************************/
void   Task(void)
{
}
```

6 H 文件模板

每个 h 文件模块由模块描述区、包含头文件区、宏定义区、枚举结构体区定义、以及 API 函数定义区组成。

下面是各个模块的示意：

6.1 模块描述区

```
/*********************************************************************************************************
* 模块名称: SendDataToPc.h
* 摘　　要:
* 当前版本: 1.0
* 作　　者:
* 完成日期:
* 内　　容:
* 注　　意: none
*********************************************************************************************************
* 取代版本:
* 作　　者:
* 完成日期:
* 修改内容:
* 修改文件:
*********************************************************************************************************/
#ifndef _SEND_DATA_TO_PC   //注意：这个是必须的，防止重编译
#define _SEND_DATA_TO_PC   //注意：这个是必须的
```

6.2 包含头文件区

```
/*********************************************************************************************************
*                                                包含头文件
*********************************************************************************************************/
#include "DataType.h"
#include "Version.h"
```

6.3 宏定义区

```
/*********************************************************************************************************
*                                                宏定义
*********************************************************************************************************/
//参照“模块（C 文件）描述”中的“宏定义区”
```

6.4 枚举结构体定义区

```
/*********************************************************************************************************
*                                                枚举结构体定义
```

```
*****************************************************************************************************************/
//参照“模块（C 文件）描述”中的“枚举结构体定义区”
//但是“C 文件”中定义的只能用于所在的 C 文件区
//“H 文件定义的”既能用在所在的 H 文件、对应的 C 文件，还能用于其他被应用的 H 文件和 C 文件
```

6.5 API 函数声明区：

```
/*****************************************************************************************************************
*                                                API 函数声明
*****************************************************************************************************************/
void    InitSignal(void);
#endif          //注意：这个是必须的，与#ifndef 对应
```