研发部软件编写规范

版本号：Ver1.1.0

编写人：wind

北京核地科技发展中心

1简介

代码编写规则应该在建立在一个工程项目之前。该规则应该贯穿整个项目的始终以保证代

码的一致性。采用标准的代码编写惯例，可大大简化项目的维护负担。在C语言中可以有多种代码的编写方法（当然其它编程序语言亦如此），你可以尽可能采用一种好的风格，以达到以下目的：

可移植 (Portability)

连贯 (Consistency)

整洁 (Neatness)

易于维护 (Easy maintenance)

易于理解 (Easy understanding)

简洁 (Simplicity)

本规约将规范北京核地科技发展中心软件研发团队编码风格。

我所强调的就是：这种风格一定要贯穿你项目的始终。在以后的内容中我还要提到：即使在一个团队合作的大型项目中，这种风格也要贯穿始终。采用通用的代码编写风格可以减轻代码维护的工作量并降低维护费用，通用的代码风格还可以避免重写代码。

2基本原则

制定标准的基本目的就是加强代码的可维护性。也就是说代码必须易于阅读，易于理解，易于测试，易于移植。

2.1所有的代码必须采用ANSI C标准

函数原型必须采用ANSI标准。

2.2保持代码的简单清晰

不要在语言中使用晦涩难懂的表述，直接表明你的思想。

2.3保持一致性

代码编写过程中通篇使用使用同样的规则。

2.4避免使用复杂语句

一个语句若有太多的决策点（Decision points）将会使代码难于理解。尤其是对于测试，不要使用GOTO语句。

2.5更新原有代码

一旦修改已存在的代码，就要随时更新相关文档以遵守本文中所制定的规则，这将确

保原有代码即时更新。

2命名规则

3编码规则

3.1变量，工程规范定义统一的变量类型，无关于CPU，便于代码移植。

typedef unsigned char Bool;

typedef unsigned char Uint8;

typedef unsigned short int Uint16;

typedef unsigned int Uint32;

typedef signed char Int8;

typedef signed short int Int16;

typedef signed int Int32;

typedef float Float32;

3.2

3排版

4代码可读性

4.1注意运算符的优先级，使用括号明确表达式的操作顺序，避免使用默认优先级。

说明 防止阅读时产生误解，防止使用默认优先级而与设计思想不符产生错误

4.2避免使用不易理解的数字，应用有意义的标识来代替。涉及物理状态或者有物理意义的常量，不能直接使用数字，必须用有意义的枚举或者宏来代替。

示例：如下程序可读性差。

if(TaskState == 0)

{

TaskState = 1;

}

应书写为如下形式：

#define TASK\_IDLE 0

#define TASK\_BUSY 1

if(TaskState == TASK\_IDLE)

{

TaskState = TASK\_BUSY;

}

4.4源码中关系较为紧密的代码应尽可能相邻。

说明 便于阅读和查找

示例：以下代码布局不合理

rect.length = 10;

char\_poi = 9;

rect.width = 7;

应调整为如下格式：

rect.length = 10;

rect.width = 7;

char\_poi = 9;

4.5不要使用不易理解的技巧性很高的语句，除非非常有不要时。

说明 切记高手不是装出来的，高技巧不等于高效率，程序的高效在于算法。

示例：如下表达式，将让人难以理解。

\* start\_pi ++ +=1;

\* ++ start\_pi +=1;

应分别改为如下形式：

\* start\_pi +=1;

\* start\_pi ++;

++ start\_pi;

\*start\_pi +=1;

4.6当判断常量和变量是否相等时，常量应当书写于等号左边。

说明 防止因书写错误而导致隐性逻辑错误。

示例：以下程序容易书写出错

if(index == MAX\_NUM)

{

//program code

}

应改为如下形式：

if(MAX\_NUM == index)

{

//program code

}

5变量和结构

5.1尽可能少用公共变量。

随手写:

头文件包含：每个源文件所包含的头文件尽可能是该文件中所用，切忌不可包含大的头文件。如果包含大的头文件，将加长整个工程的编译时间。

函数参数：每个功能函数的形参原则上不可大于4个，如果大于需使用参数指针方式传递。如：

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 名 称：Uint8 LCD\_DrawChart(ChartDat \*Disp)

\* 功 能：在坐标系中绘制图形

\* 入口参数：CoordinateDat \* Disp 绘制图形控制数据

\* 出口参数：成功/失败

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

Uint8 LCD\_DrawChart(ChartDat \*Disp){

//绘制图形

return success;

}

定义数据类型：

typedef struct

{

CoordinateDat \*CDatP; //坐标数据指针

LCD\_ColorDat Color; //谱线颜色

Uint16 Index; //索引

Uint16 DatSize; //谱线数据长度

Uint16 \*DatBuf; //原始谱线数据指针

Uint16 \*DisP; //显示谱线数据指针

}ChartDat;