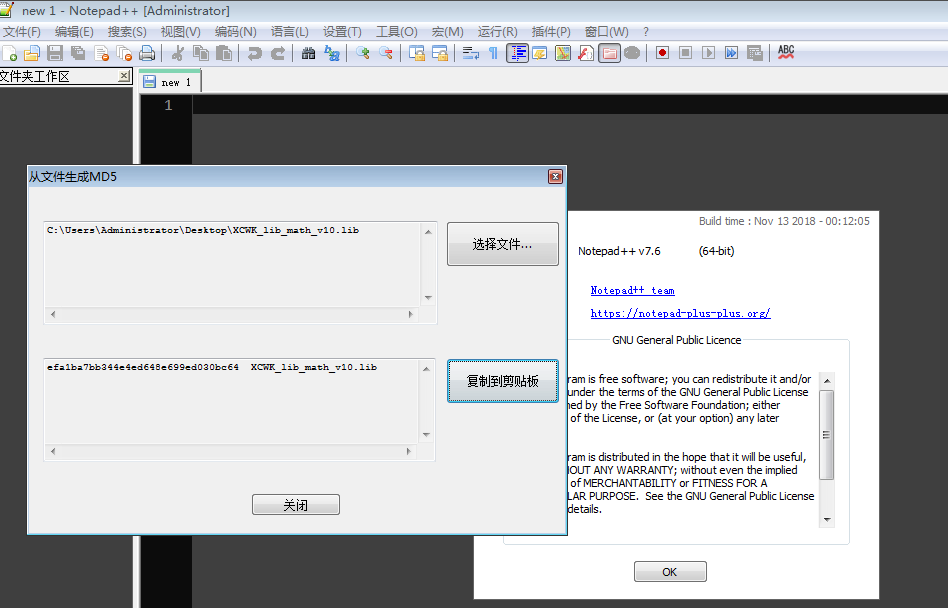
**XCWK\_项目模板\_数学子程序**



* MD5码：

efa1ba7bb344e4ed648e699ed030bc64



* **需要数学子程序及封包原由**

数学计算几乎存在每一个程序中，在秤中我们常用的到有：

多字节的加减法与取差值绝对值，多字节的加减法与取差值绝对值，多字节的乘除法，多字节的BCD转换等

通常这些子程序第一次编写除错调试完成后，后面就再也不会对其进行修改了，如果封成库就会减少源码的文件，同时杜绝了其它修改的可能性，对程序的健状性能大有提升。

* **子程序命名规则及命名定义**

1. 子程序命名规则Fun\_Math\_XXX：

子程序以Fun打头，表示实现某一特定功能的方法；

其后紧跟这个方法的类别，这里是数学计算相关的，暂定为Math；

最后为要实现的功能的缩写，根据实际的方法适当命名。

上面的每个部分用’\_’分隔

**2**. 子程序命名定义：

2字节的加法：Fun\_Math\_Add2\_2

2字节的减法：Fun\_Math\_Sub2\_2

2字节的减法后绝对值：Fun\_Math\_Sub2\_2\_Neg

3字节的加法：Fun\_Math\_Add3\_3

3字节的减法：Fun\_Math\_Sub3\_3

3字节的减法后绝对值：Fun\_Math\_Sub3\_3\_Neg

3字节的乘法：Fun\_Math\_Mul3\_3

6字节/3字节的除法：Fun\_Math\_Div6\_3

6字节/3字节的除法四舍五入：Fun\_Math\_Div6\_3\_Rounded

3字节BCD码：Fun\_Math\_Hex3\_Bcd

注：以上的数据代码的意义是，操作数的字节数，拿Fun\_Math\_Add2\_2来说，它实现了两字节的减法，其中的2\_2表明2字节的变量-2字节的变量

* 使用的RAM数量与RAM命名

1. RAM使用数量

目前实现以上的子程序需要的RAM为16字节

1. RAM的定义与命命名（注：该组RAM定义在BANK0）

Buffer0

Buffer1

Buffer2

Buffer3

Buffer4

Buffer5

Buffer6

TempRam1

TempRam2

TempRam3

TempRam4

TempRam5

TempRam6

TempRam11

TempRam12

TempRam13

* 使用方法与示例

1. Fun\_Math\_Add2\_2

功能描述：两字节长度的变量加法，TempRam3、4 += TempRam5、6

输入：操作数1： TempRam3(高8位)、TempRam4（低8 位）

操作数2： TempRam5(高8位)、TempRam6（低8 位）

输出：操作数1： TempRam3(高8位)、TempRam4（低8 位）

例：1000 = 112 + 888

MVL LOW 112

MOVWF TempRam4

MVL HIGH 112

MOVWF TempRam3

MVL LOW 888

MOVWF TempRam6

MVL HIGH 888

MOVWF TempRam5

CALL Fun\_Math\_Add2\_2

上面的代码段执行完后，TempRam3中值为03H，TempRam4中值为E8H，十六进制03E8H的十进制为1000。

1. Fun\_Math\_Sub2\_2

功能描述：两字节长度的变量减法，TempRam3、4 -= TempRam5、6

输入：操作数1： TempRam3(高8位)、TempRam4（低8 位）

操作数2： TempRam5(高8位)、TempRam6（低8 位）

输出：操作数1： TempRam3(高8位)、TempRam4（低8 位）

例：776 = 888 - 112

MVL LOW 888

MOVWF TempRam4

MVL HIGH 888

MOVWF TempRam3

MVL LOW 122

MOVWF TempRam6

MVL HIGH 122

MOVWF TempRam5

CALL Fun\_Math\_Sub2\_2

上面的代码段执行完后，TempRam3中值为03H，TempRam4中值为08H，十六进制0308H的十进制为776。

1. Fun\_Math\_Sub2\_2\_Neg

功能描述：两字节长度的变量减法后，如果需要取这两个差的绝对值，调用这个子程序，必须在调用Fun\_Math\_Sub2\_2后再调用Fun\_Math\_Sub2\_2\_Neg

例：776 = | 112 – 888 |

MVL LOW 112

MOVWF TempRam4

MVL HIGH 112

MOVWF TempRam3

MVL LOW 888

MOVWF TempRam6

MVL HIGH 888

MOVWF TempRam5

CALL Fun\_Math\_Sub2\_2

CALL Fun\_Math\_Sub2\_2\_Neg

上面的代码段执行完后，TempRam3中值为03H，TempRam4中值为08H，十六进制0308H的十进制为776

1. Fun\_Math\_Add3\_3

同Fun\_Math\_Add2\_2，只不过是操作数为3字节长度

1. Fun\_Math\_Sub3\_3

同Fun\_Math\_Sub3\_3，只不过是操作数为3字节长度

1. Fun\_Math\_Sub3\_3\_Neg

同Fun\_Math\_Sub3\_3\_Neg，只不过是操作数为3字节长度

1. Fun\_Math\_Mul3\_3

功能描述：三字节长度的变量乘法，

输入：

操作数1： TempRam4(高8位)、TempRam5（中8 位）、TempRam6（低8 位）

操作数2： TempRam11(高8位)、TempRam12（中8 位）、TempRam13（低8 位）

输出：

TempRam1(高)~TempRam6（低 ）

例：99456 = 888 \* 112

MVL LOW 888

MOVWF TempRam6

MVL HIGH 888

MOVWF TempRam5

MVL 00H

MOVWF TempRam4

MVL LOW 122

MOVWF TempRam13

MVL HIGH 122

MOVWF TempRam12

MVL 00H

MOVWF TempRam11

CALL Fun\_Math\_Mul3\_3

上面的代码段执行完后，

TempRam1中值为00H，TempRam2中值为00H, TempRam3中值为00H,TempRam4中值为01H, TempRam5中值为84H, TempRam6中值为80H

十六进制018480H的十进制为99456

1. Fun\_Math\_Div6\_3

功能描述：六字节长度 / 三字节长度变量除法,不带四舍五入

输入：

操作数1： TempRam1~ TempRam6

操作数2： TempRam11 (高8位)、TempRam12（中8 位）、TempRam13（低8 位）

输出：

商 ：TempRam4(高)~TempRam6（低 ）

余数：TempRam1(高)~TempRam3（低 ）

例：1000/256 = 3 ….. 232

MVL LOW 1000

MOVWF TempRam6

MVL HIGH 1000

MOVWF TempRam5

CLRF TempRam4

CLRF TempRam3

CLRF TempRam2

CLRF TempRam1

MVL LOW 256

MOVWF TempRam13

MVL HIGH 256

MOVWF TempRam12

CLRF TempRam11

CALL Fun\_Math\_Mul3\_3

上面的代码段执行完后，

商： TempRam4中值为00H, TempRam5中值为00H, TempRam6中值为03H

十六进制000003H的十进制为3

余数：TempRam1中值为00H，TempRam2中值为00H, TempRam3中值为E8H

十六进制0000E8H的十进制为232

1. Fun\_Math\_Div6\_3\_Rounded

功能描述：调用六字节长度 / 三字节长度变量除法后，如果需要四舍五入时调用这个子程序

例：1000/256 = 3 ….. 232

MVL LOW 1000

MOVWF TempRam6

MVL HIGH 1000

MOVWF TempRam5

CLRF TempRam4

CLRF TempRam3

CLRF TempRam2

CLRF TempRam1

MVL LOW 256

MOVWF TempRam13

MVL HIGH 256

MOVWF TempRam12

CLRF TempRam11

CALL Fun\_Math\_Mul3\_3

CALL Fun\_Math\_Div6\_3\_Rounded

上面的代码段执行完后，

商： TempRam4中值为00H, TempRam5中值为00H, TempRam6中值为04H

十六进制000004H的十进制为4

余数：TempRam1中值为00H，TempRam2中值为00H, TempRam3中值为E8H

十六进制0000E8H的十进制为232

1. Fun\_Math\_Hex3\_Bcd

功能描述：三字节的变量转换成6位的BCD码 ，主要用于显示查表工作

输入：

操作数1： TempRam11 (高8位)、TempRam12（中8 位）、TempRam13（低8 位）

输出：

TempRam1(高)~TempRam6（低 ）

例：01E240H 转换在6位的BCD码为123456

MVL 040H

MOVWF TempRam13

MVL 0E2H

MOVWF TempRam12

MVL 001H

MOVWF TempRam11

CALL Fun\_Math\_Hex3\_Bcd

上面的代码段执行完后，

TempRam1中值为01H, TempRam2中值为02H, TempRam3中值为03H

TempRam4中值为04H, TempRam5中值为05H, TempRam6中值为06H

* 其它用途

多字节减法不仅可以用来做减法用，减法同时具有大小判断功能，例如

当调用Fun\_Math\_Sub2\_2后，我们根据MCU程序状态字STATUS中C标记位来判断大小

MVL LOW 112

MOVWF TempRam4

MVL HIGH 112

MOVWF TempRam3

MVL LOW 888

MOVWF TempRam6

MVL HIGH 888

MOVWF TempRam5

CALL Fun\_Math\_Sub2\_2

上面的代码运行后，STATUS[C]为0,表示TempRam3、4 < TempRam5,6

2018年12月11日星期二

夏龙