



## 目录

### 目录

1.Lib 库文件使用（C 例程） .....	2
1.1 Lib 库文件生成 .....	2
1.1.1 先在 WinScopeIDE 中建立一个工程，如： .....	2
1.1.2 在软件的菜单 项目管理--> 编译生成库文件.....	3
1.2Lib 文件调用 .....	4
1.2.1 建立头文件.....	4
1.2.1 调用例程.....	4
2.Lib 库文件使用（汇编例程） .....	5
1.1Lib 库文件生成 .....	5
1.1.1 先在 WinScopeIDE 中建立一个工程，如： .....	5
1.1.2 在软件的菜单 项目管理--> 编译生成库文件.....	6
1.2Lib 文件调用 .....	7
附录 修改记录.....	7



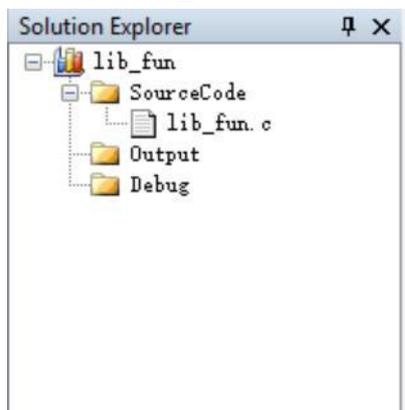
关于用户算法库的生成与应用

## 1.Lib 库文件使用（C 例程）

### 1.1 Lib 库文件生成

#### 1.1.1 先在 WinScopeIDE 中建立一个工程，如：

选择 MC32P7511(一般在库中最好不要涉及到具体型号的外设资源，仅纯算法处理)，而且参数也都是通过参数或指针传递处理工程名称暂定为 lib\_fun,同时添加一个文件 lib\_fun.c 到工程中，显示如下图所示：



建好工程之后，开始进行算法函数的编写。例子中定义了两个函数，fun\_001,fun\_002.具体代码如下：

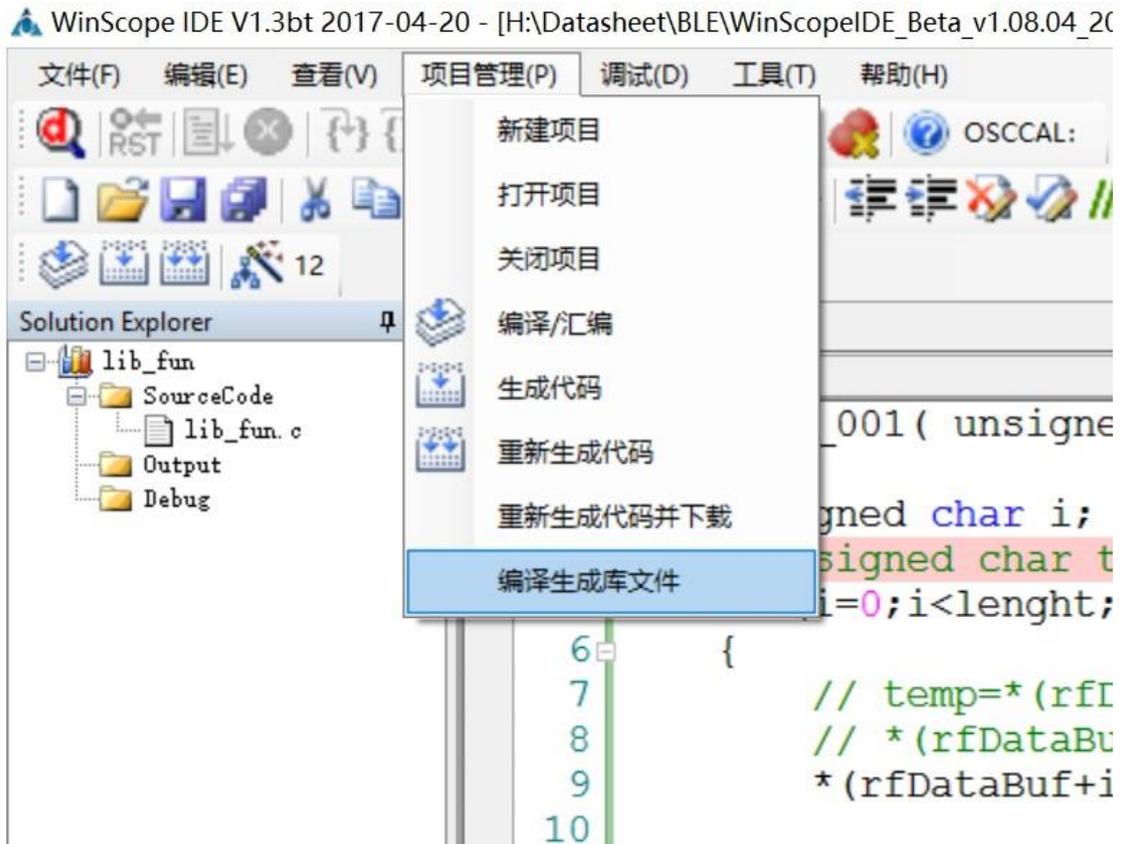
```
void fun_001( unsigned char *rfDataBuf,unsigned char lenght)
{
    unsigned char i;
    //unsigned char temp;
    for(i=0;i<lenght;i++)
    {
        // temp=*(rfDataBuf+i);
        // *(rfDataBuf+i)=temp+1;
        *(rfDataBuf+i)=*(rfDataBuf+i)+1;
    }
}
```

```
unsigned int fun_002(unsigned char x,unsigned char y)
{
    unsigned char temp;
```



```
temp=x+1;  
return temp+y;  
}
```

### 1.1.2 在软件的菜单 项目管理--> 编译生成库文件



点击之后，会对当前 C 文件进行编译，同时在工程目录下会生成一个与 C 文件相同名字的 XXX.lib 文件。编译的过程中，如果语法有错误，软件会给出对应的提示。



## 1.2 Lib 文件调用

注：需将生成的 Lib 文件拷贝至工程目录下。如下图：

名称	修改日期	类型
Debug	2018/12/28 15:17	文件夹
Output	2018/12/28 15:17	文件夹
System	2020/2/24 16:11	文件夹
lib_fun.lib	2018/12/29 16:35	LIB 文件
lib_test.c	2018/12/29 16:50	C 文件
lib_test.h	2018/12/26 13:56	H 文件
lib_test.Proj	2020/2/24 16:11	PROJ 文件

### 1.2.1 建立头文件

头文件主要用来对库文件中的函数进行声明，同时也是提供给用户调用函数时能知道库中函数的名字、参数等内容。

```
#ifndef __LIB_TEST_H  
#define __LIB_TEST_H
```

```
extern void fun_001( unsigned char *rfDataBuf,unsigned char lenght);  
extern unsigned int fun_002(unsigned char x,unsigned char y);
```

```
#endif
```

到此，库的建立就结束了。在后期的使用中，需要把头文件和库文件拷到相应的工程中进行调用。

### 1.2.1 调用例程

```
#include "MC32P7511.h"  
#include "lib_test.h"
```

```
uchar data[10];
```

```
void main()  
{
```

```
    uchar i;  
    uint    temp2;
```



```
    uchar    *pt;

    __asm
;lab_clear_ram:
        movai    0x7f ;first ram addr
        movra    FSR0
;lab_clear_loop:
        clr     INDF0
        djzr    FSR0
        goto    $-2 ;lab_clear_loop
    __endasm;

    for(i=0;i<10;i++)
        data[i]=i;

    pt=data;
    fun_001( pt,10);
    temp2=fun_002(3,2);

    IOP1=data[1];
    FSR0=(uchar)temp2;
    FSR1=temp2>>8;

    while(1);
}
```

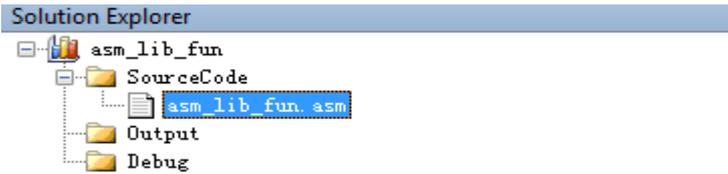
如上面例子中，定义了一个 uchar 类型的数组，数组调用了 fun\_001 进行处理，处理之后原来 0-9 的数据，应变成了 1-10。fun\_002 调用之后，temp2 的值应该变成了 6。

## 2.Lib 库文件使用（汇编例程）

### 1.1Lib 库文件生成

#### 1.1.1 先在 WinScopeIDE 中建立一个工程，如：

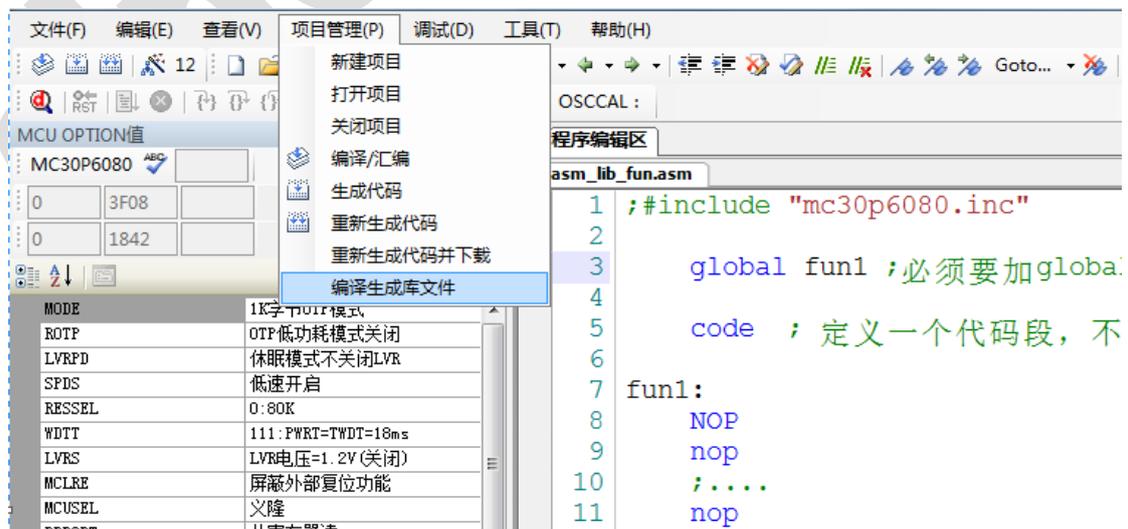
选择 MC30P6080(一般在库中最好不要涉及到具体型号的外设资源，仅纯算法处理)，处理工程名称暂定为 asm\_lib\_fun,同时添加一个文件 asm\_lib\_fun.asm 到工程中，显示如下图所示：



建好工程之后，开始进行算法函数的编写。例子中定义了一个函数，fun1.具体代码如下：

```
asm_lib_fun.asm
;#include "mc30p6080.inc"
    global fun1 ;必须要加 global 修饰
    code ; 定义一个代码段，不指定程序起始地址
fun1:
    NOP
    nop
    ;....
    nop
    return
end
```

### 1.1.2 在软件的菜单 项目管理--> 编译生成库文件

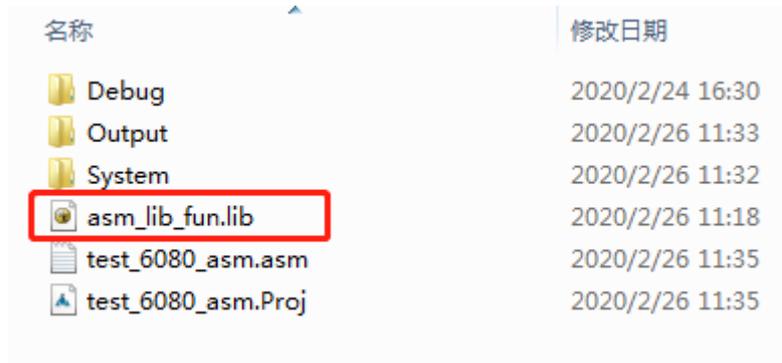




点击之后，会对当前 C 文件进行编译，同时在工程目录下会生成一个与 C 文件相同名字的 XXX.lib 文件。编译的过程中，如果语法有错误，软件会给出对应的提示。

## 1.2 Lib 文件调用

注：需将生成的 Lib 文件拷贝至工程目录下。如下图：



例程：

```
#include "mc30P6080.inc"
```

```
extern fun1 ; 须声明为外部函数
org 0x100
goto start
start:
nop
call fun1
nop

end
```

## 附录 修改记录

版本	时间	修改人	备注
V0.01	2020.02.26	吕亚磊	新建
V0.02	2023.05.08	吕亚磊	修改