	MGT-MODE-3-003		
	V1.02		
			2024-12-12

1

1.1

1.2

	Plc( )
Mc5000,Mc5100	
MEGreator	MC5000 MU
	MEGreator

2

2.1

Mc5x00 Plc

1>. C C ,

2>. Plc , X, Y, M, S, SM, T, C; D, R, SD ,Z,F

X , D

3>. , 64K

4>. ANSI C , C

5>. C ;

6>.

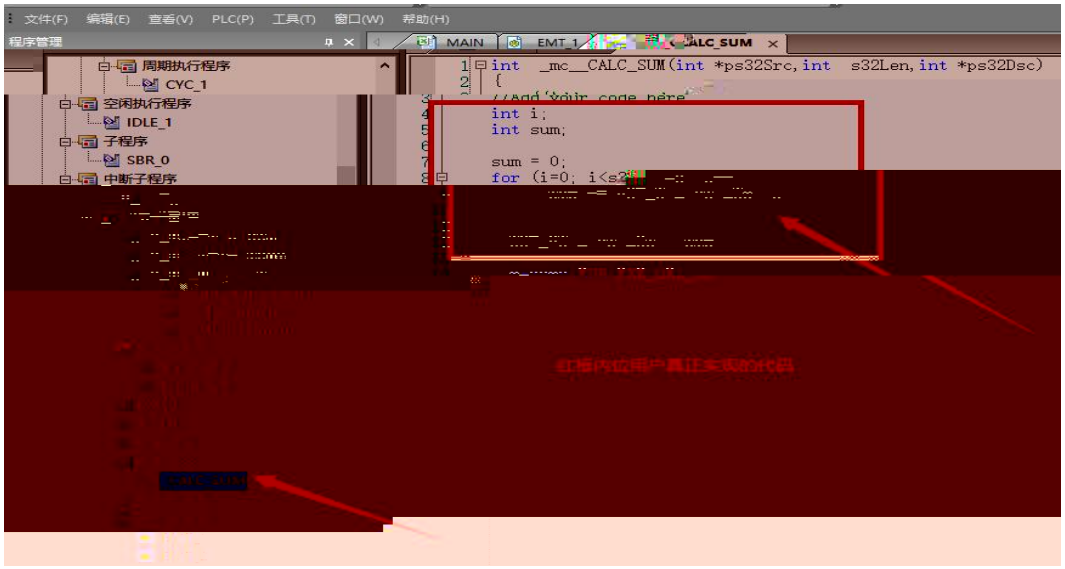
2.2

1>

LD LD

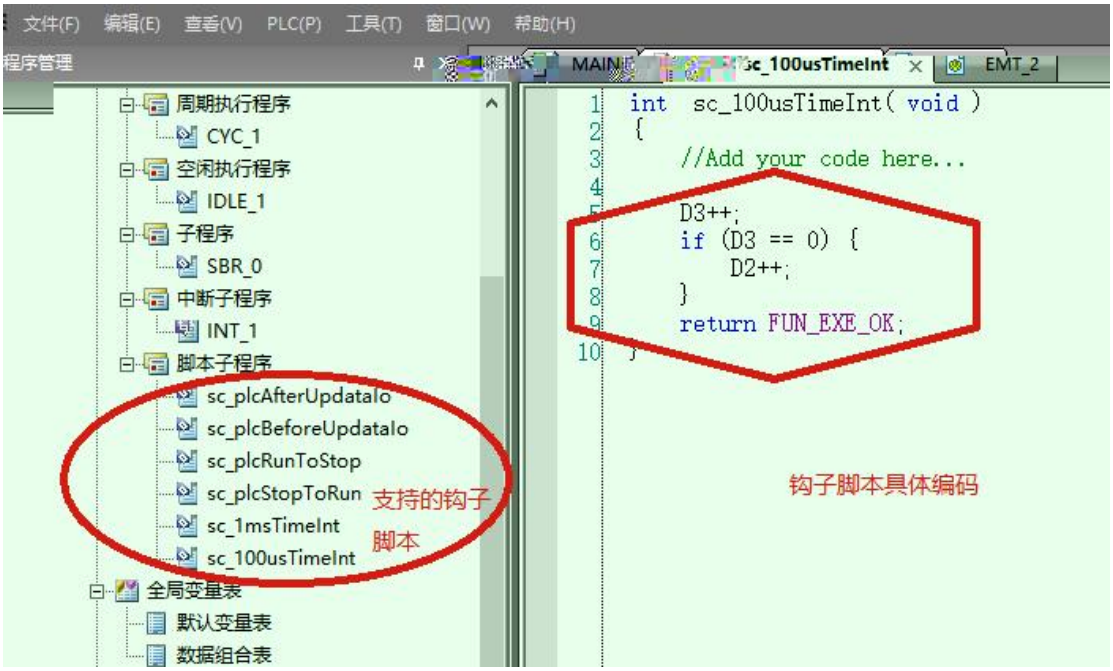
LD

1



2

1>



3

2.3

C

plc

export_module.h	Plc
user_common.c	
user_common.h	

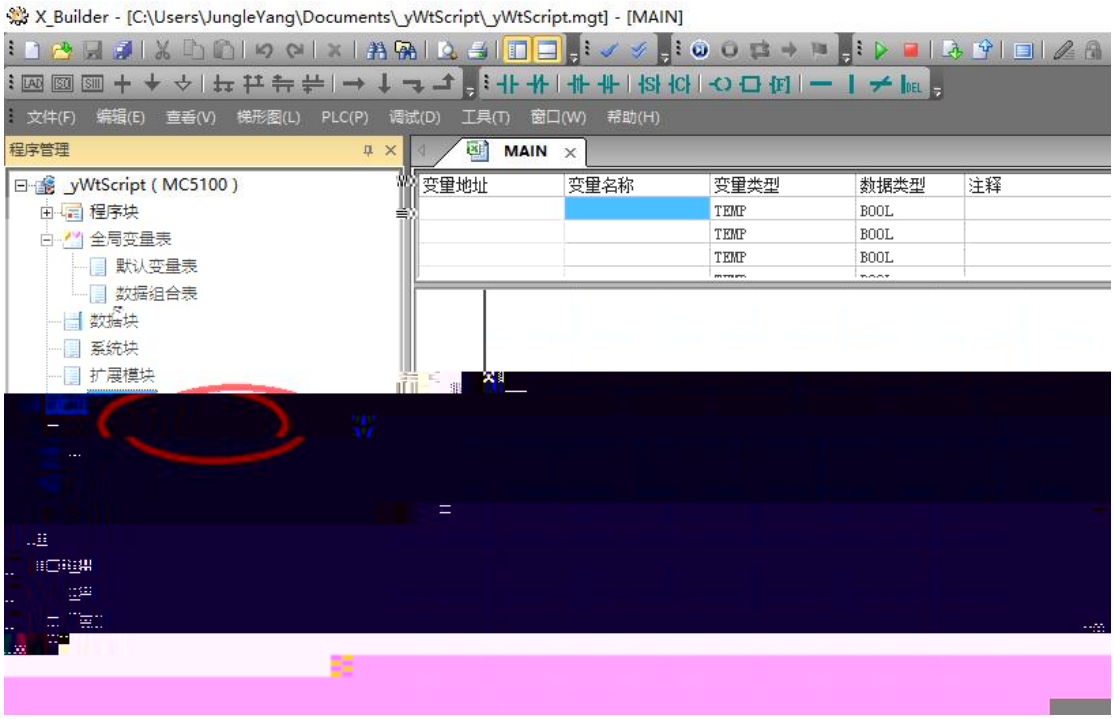
3

xBuild C

D1000 20000 D998( )
D996( )

3.1

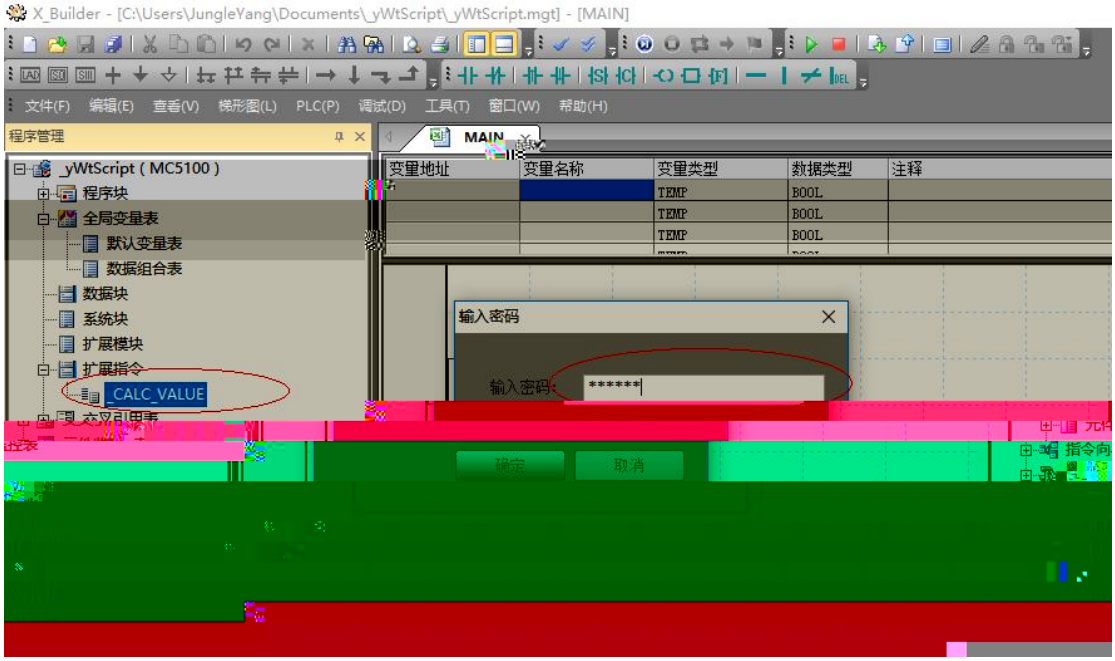
“ ” “ ” “ ”



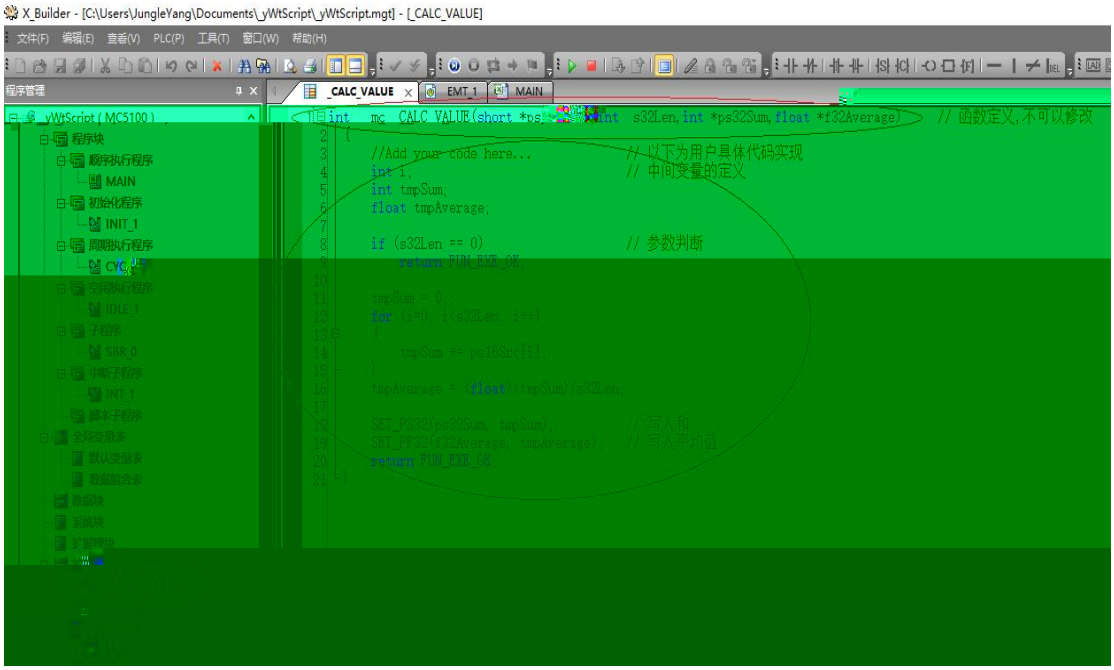
3

3.6 н н 6

3.3

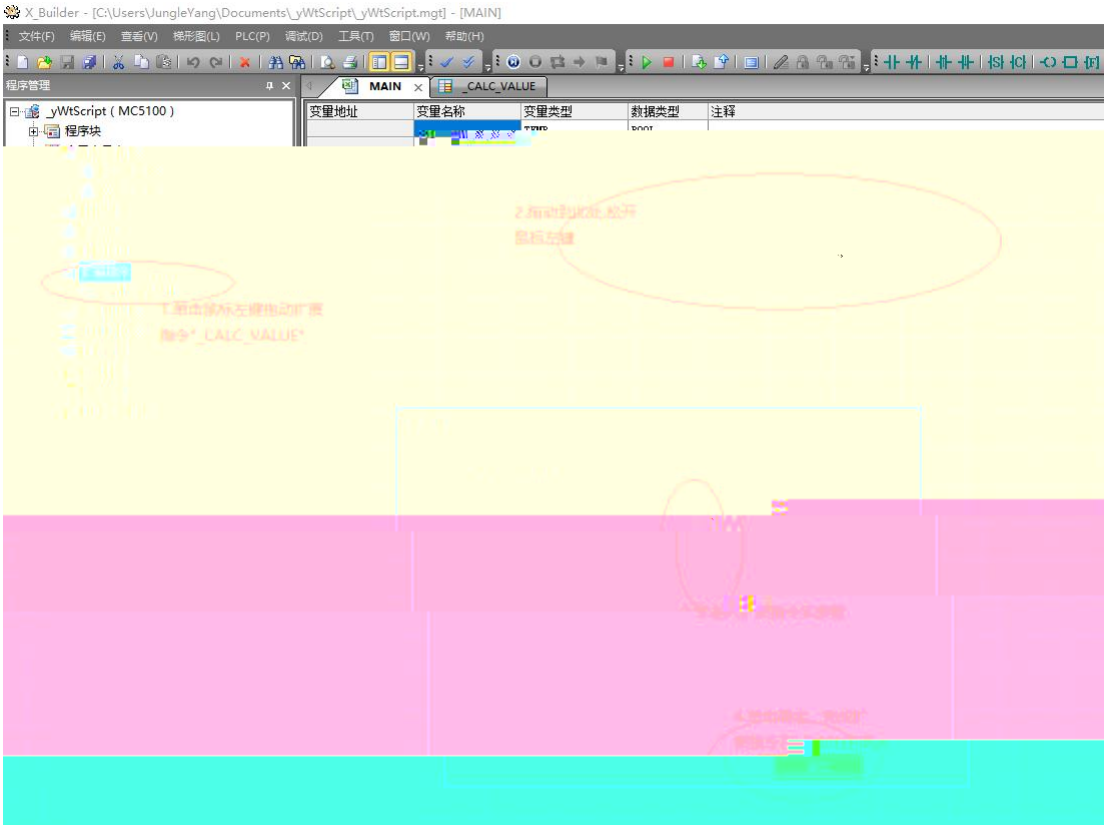

X_Builder - [C:\Users\JungleYang\Documents_yWtScript_yWtScript.mgt] - [MAIN]
文件(F) 编辑(E) 查看(V) 梯形图(L) PLC(P) 调试(D) 工具(T) 窗口(W) 帮助(H)
程序管理
_yWtScript (MC5100)
程序块
全局变量表
默认变量表
数据组合表
数据块
系统块
扩展模块
扩展指令
_CALC_VALUE
MAIN
变量地址
变量名称
变量类型
数据类型
注释
TEMP
BOOL
输入密码
输入密码:
******
确定
取消

6

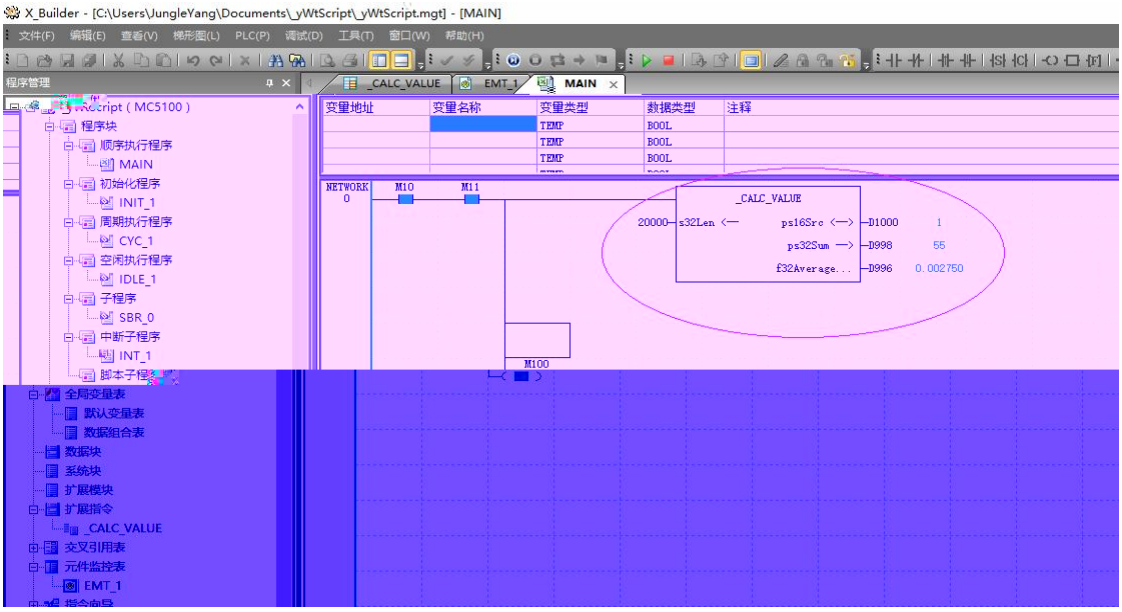

X_Builder - [C:\Users\JungleYang\Documents_yWtScript_yWtScript.mgt] - [_CALC_VALUE]
文件(F) 编辑(E) 查看(V) PLC(P) 工具(T) 窗口(W) 帮助(H)
程序管理
_CALC_VALUE
EMT_1
MAIN
int mc_CALC_VALUE(short *ps16Src, int s32Len, int *ps32Sum, float *f32Average) // 函数定义,不可以修改
{
//Add your code here... // 以下为用户具体代码实现
int i;
int tmpSum;
float tmpAverage;
if (s32Len == 0) // 参数判断
return FUN_EXE_OK;
tmpSum = 0;
for (i=0; i<s32Len; i++)
{
tmpSum += ps16Src[i];
}
tmpAverage = (float)(tmpSum)/s32Len;
SET_PV32(ps32Sum, tmpSum); // 写入和
SET_PV32(f32Average, tmpAverage); // 写入平均值
return FUN_EXE_OK;
}

7

3.4

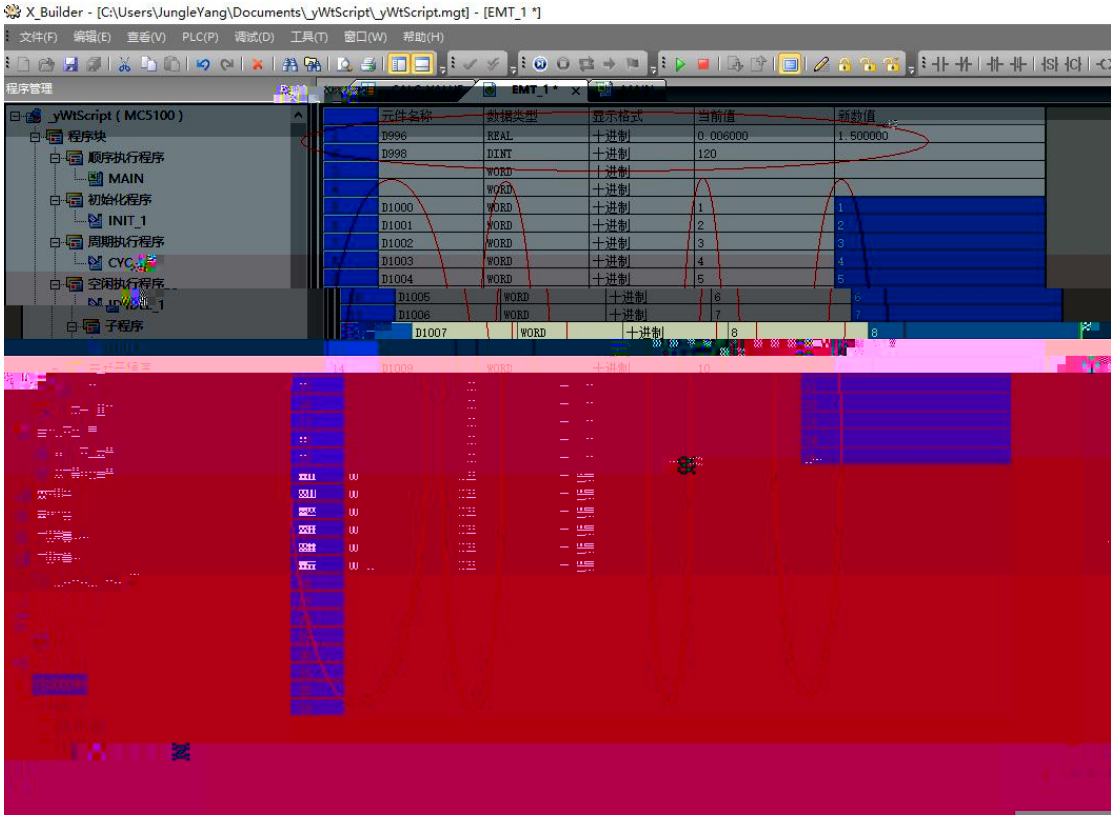

X_Builder - [C:\Users\JungleYang\Documents_yWtScript_yWtScript.mgt] - [MAIN]
文件(F) 编辑(E) 查看(V) 梯形图(L) PLC(P) 调试(D) 工具(T) 窗口(W) 帮助(H)
程序管理
_yWtScript (MC5100)
程序块
MAIN
_CALC_VALUE
变量地址
变量名称
变量类型
数据类型
注释

8


X_Builder - [C:\Users\JungleYang\Documents_yWtScript_yWtScript.mgt] - [MAIN]
文件(F) 编辑(E) 查看(V) 梯形图(L) PLC(P) 调试(D) 工具(T) 窗口(W) 帮助(H)
程序管理
_CALC_VALUE
EMT_1
MAIN
变量地址
变量名称
变量类型
数据类型
注释
TEMP
BOOL
NETWORK
0
M10
M11
M100
_CALC_VALUE
20000
s32Len
ps16Src
D1000
1
ps32Sum
D998
55
f32Average...
D996
0.002750
程序块
顺序执行程序
MAIN
初始化程序
INIT_1
周期执行程序
CYC_1
空闲执行程序
IDLE_1
子程序
SBR_0
中断子程序
INT_1
全局变量表
默认变量表
数据组合表
数据块
系统块
扩展模块
扩展指令
_CALC_VALUE
交叉引用表
元件监控表
EMT_1

9

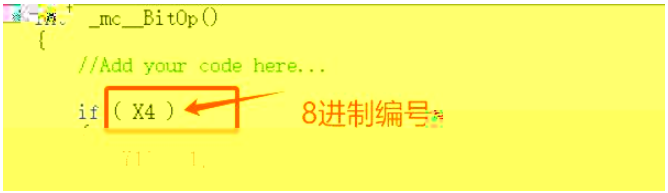
3.5


X_Builder - [C:\Users\JungleYang\Documents_yWtScript_yWtScript.mgt] - [EMT_1 *]
文件(F) 编辑(E) 查看(V) PLC(P) 调试(D) 工具(T) 窗口(W) 帮助(H)
程序管理
_yWtScript (MC5100)
程序块
顺序执行程序
MAIN
初始化程序
INIT_1
周期执行程序
CYC_1
空闲执行程序
IDLE_1
子程序
EMT_1
元件名称 数据类型 显示格式 当前值 新数值
D996 REAL 十进制 0.006000 1.500000
D998 DINT 十进制 120
WORD 十进制
D1000 WORD 十进制 1
D1001 WORD 十进制 2
D1002 WORD 十进制 3
D1003 WORD 十进制 4
D1004 WORD 十进制 5
D1005 WORD 十进制 6
D1006 WORD 十进制 7
D1007 WORD 十进制 8

4

4.1

4.1.1 X


_mc__BitOp()
{
//Add your code here...
if (X4)
8进制编号

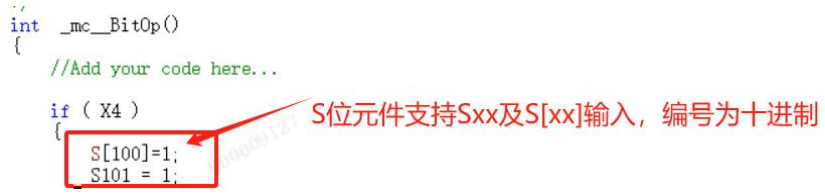
4.1.2 Y


int _mc__BitOp()
{
//Add your code here...
if (X4)
{
Y10 = 1;
Y位元件输入，编号位8进制

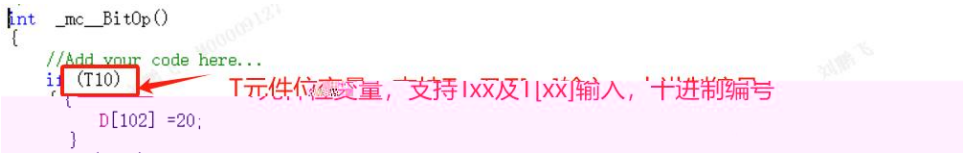
4.1.3 SM


int _mc__BitOp()
{
//Add your code here...
if (X4)
{
SM[40]=1;
SM41=1;
SM位元件支持SMxx及SM[xx]

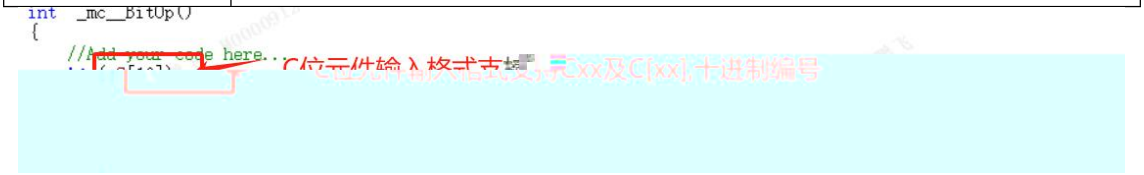
4.1.4 S



4.1.5 T

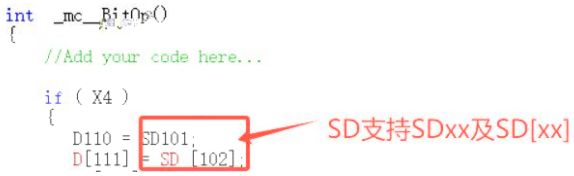


4.1.6 C



4.1.7 M

4.1.8 SD



4.1.9 Z

4.1.10 D


```
int  _mc__BitOp()
{
    //Add your code here...
    if ( X4 )
    {
      D110 = SD101;
      D[111] = SD [102];
```

D字单元直接读写支持Dxx及D[xx]

4.1.11 R


```
int  _mc__BitOp()
{
  //Add your code here...
  if ( X4 )
  {
    R[4095] = 100;
```

R字变量支持Rxx及R[xx]直接读写操作

4.2

4.2.1

	int GET_DD(unsigned short stNum)
	“ ”
	stNum:


```
int  _mc__BitOp()
{
    //Add your code here...

    if ( X4 )
    {
        long tmp;
        tmp = GET_DD(1000);
```

读取D1000长整型数据到tmp

4.2.2

	void SET_DD(unsigned short stNum, int val)
	“ ”

```
int  _mc__BitOp()
{
    //Add your code here...

    if ( X4 )
    {
        long tmp;
        tmp = GET_DD(1000);
        SET_DD(150, tmp);
```

将tmp的值写入到长整数D150中

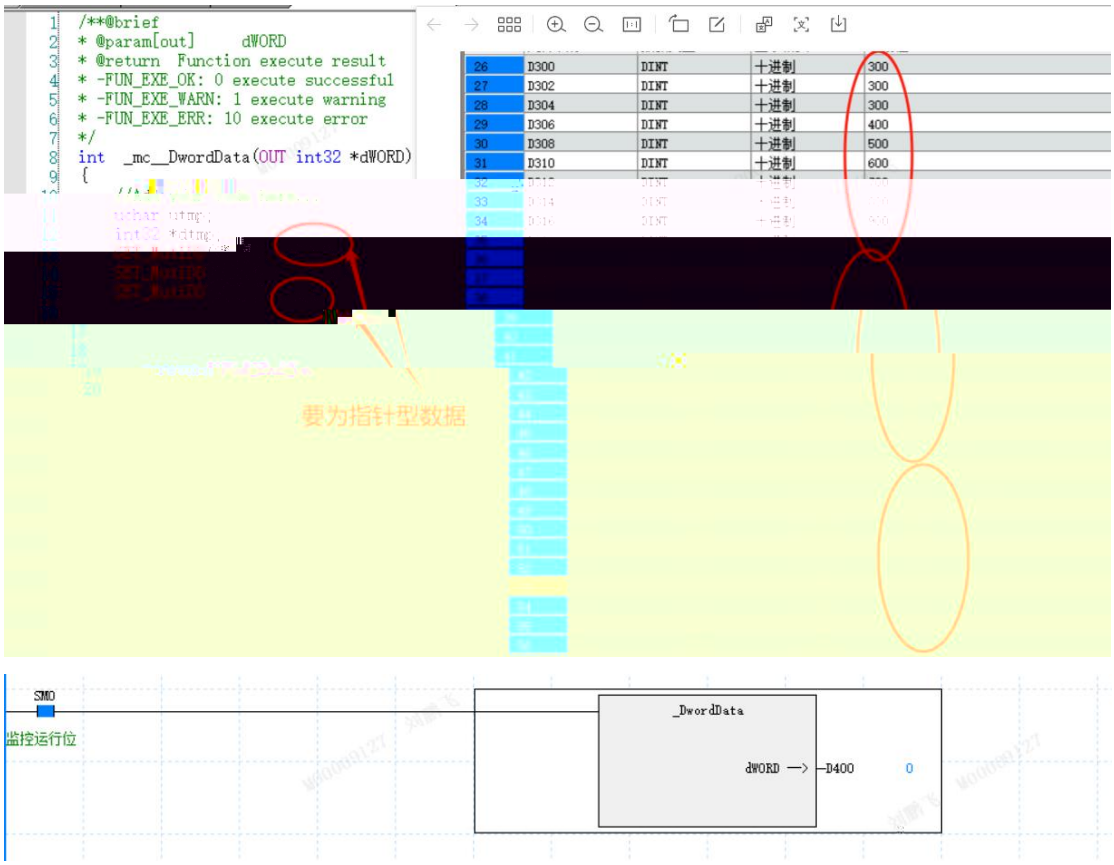
4.2.3

	int GET_MutiDD(int stNum, int len, int *ps32Dsc)
	“ ”
	stNum
	Len :
	ps32Dsc:

4.2.4

	int SET_MutiDD(int stNum, int len, int *ps32Src)
	“ ”
	stNum
	Len :

	ps32Dsc: D



1 GET_MutiDD(int stNum, int len, int *ps32Dsc)
SET_MutiDD(int stNum, int len, int *ps32Src) D
dWORD D400 D400 10

2 int *ps32Dsc int *ps32Src

4.2.5

	float GET_FD(unsigned short stNum)
	“ ”
	stNum:

4.2.6

	void SET_FD(unsigned short stNum, float val)
	“ ”

```
int  _mc__DwordData(OUT int32 *dWORD)
{
    //Add your code here...
    uchar utmp;
    int32 *dtmp;
    float fTmp , *pfTmp;
    fTmp = GET_FD(600);
    SET_FD(700,fTmp);
```

读取d600内浮点数存入变量fTmp中

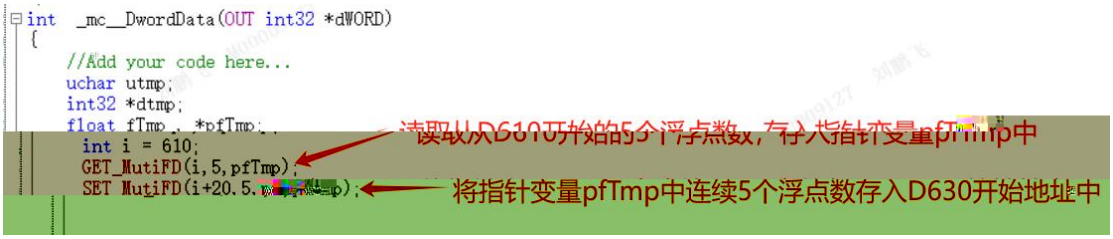
将浮点数fTmp的值写入D700中

4.2.7

	int GET_MutiFD(int stNum, int len, float *pf32Dsc)
	“ ”
	stNum
	Len :
	ps32Dsc:

4.2.8

	int SET_MutiFD(int stNum, int len, float *pf32Src)
	“ ”
	stNum
	Len :
	ps32Dsc: D



4.2.9

	int GET_DR(unsigned short stNum)
	“ ”
	stNum: R

4.2.10

	void SET_DR(unsigned short stNum, int val)
	“ ”

4.2.11

	int GET_MutiDR(int stNum, int len, int *ps32Dsc)
	“ ”
	stNum
	Len :
	ps32Dsc:

4.2.12

	int SET_MutiDR(int stNum, int len, int *ps32Src)
	“ ”
	stNum
	Len :
	ps32Dsc: R

4.2.13

	float GET_FR(unsigned short stNum)
	“ ”
	stNum: R

4.2.14

	void SET_FR(unsigned short stNum, float val)
	“ ”

4.2.15

	int GET_MutiFR(int stNum, int len, float *pf32Dsc)
	“ ”
	stNum
	Len :
	ps32Dsc:

4.2.16

	int SET_MutiFR(int stNum, int len, float *pf32Src)
	“ ”
	stNum
	Len :
	ps32Dsc: R

4.2.17

	int GET_DF(int stNum)
	“ ”
	stNum: F

4.2.18

	void SET_DF(int stNum, int val)
	“ ”

4.2.19

	int GET_MutiDF(int stNum, int len, int *ps32Dsc)
	“ ”

	stNum
	Len :
	ps32Dsc:

4.2.20

	int SET_MutiDF(int stNum, int len, int *ps32Src)
	“ ”
	stNum
	Len :
	ps32Dsc: F

4.2.21

	float GET_FF(unsigned short stNum)
	“ ”
	stNum: F

4.2.22

	void SET_FF(unsigned short stNum, float val)
	“ ”

4.2.23

	int GET_MutiFF(int stNum, int len, float *pf32Dsc)
	“ ”
	stNum
	Len :
	ps32Dsc:

4.2.24

	int SET_MutiFF(int stNum, int len, float *pf32Src)
	“ ”
	stNum
	Len :
	ps32Dsc: F

4.2.25

	int GET_DF0(int stNum)
	int GET_DF1(int stNum)
	int GET_DF2(int stNum)
	int GET_DF3(int stNum)
	int GET_DF4(int stNum)
	int GET_DF5(int stNum)
	int GET_DF6(int stNum)
	int GET_DF7(int stNum)
	int GET_DF8(int stNum)
	int GET_DF9(int stNum)
	“ ”
	stNum: Fx

4.2.26

	void SET_DF0(int stNum, int val)
	void SET_DF1(int stNum, int val)
	void SET_DF2(int stNum, int val)
	void SET_DF3(int stNum, int val)
	void SET_DF4(int stNum, int val)
	void SET_DF5(int stNum, int val)
	void SET_DF6(int stNum, int val)
	void SET_DF7(int stNum, int val)
	void SET_DF8(int stNum, int val)
	void SET_DF9(int stNum, int val)
	“ ”

4.2.27

	int GET_MutiDF0(int stNum, int len, int *ps32Dsc)
	int GET_MutiDF1(int stNum, int len, int *ps32Dsc)
	int GET_MutiDF2(int stNum, int len, int *ps32Dsc)
	int GET_MutiDF3(int stNum, int len, int *ps32Dsc)
	int GET_MutiDF4(int stNum, int len, int *ps32Dsc)
	int GET_MutiDF5(int stNum, int len, int *ps32Dsc)
	int GET_MutiDF6(int stNum, int len, int *ps32Dsc)
	int GET_MutiDF7(int stNum, int len, int *ps32Dsc)
	int GET_MutiDF8(int stNum, int len, int *ps32Dsc)
	int GET_MutiDF9(int stNum, int len, int *ps32Dsc)
	“ ”
	stNum
	Len :
	ps32Dsc:

4.2.28

	int SET_MutiDF0(int stNum, int len, int *ps32Src)
	int SET_MutiDF1(int stNum, int len, int *ps32Src)
	int SET_MutiDF2(int stNum, int len, int *ps32Src)
	int SET_MutiDF3(int stNum, int len, int *ps32Src)
	int SET_MutiDF4(int stNum, int len, int *ps32Src)
	int SET_MutiDF5(int stNum, int len, int *ps32Src)
	int SET_MutiDF6(int stNum, int len, int *ps32Src)
	int SET_MutiDF7(int stNum, int len, int *ps32Src)
	int SET_MutiDF8(int stNum, int len, int *ps32Src)
	int SET_MutiDF9(int stNum, int len, int *ps32Src)
	“ ”
	stNum
	Len :
	ps32Dsc: F

4.2.29

	float GET_FF0(unsigned short stNum)
	float GET_FF1(unsigned short stNum)
	float GET_FF2(unsigned short stNum)
	float GET_FF3(unsigned short stNum)
	float GET_FF4(unsigned short stNum)
	float GET_FF5(unsigned short stNum)
	float GET_FF6(unsigned short stNum)
	float GET_FF7(unsigned short stNum)
	float GET_FF8(unsigned short stNum)
	float GET_FF9(unsigned short stNum)
	“ ”
	stNum: Fx

4.2.30

	void SET_FF0(unsigned short stNum, float val)
	void SET_FF1(unsigned short stNum, float val)
	void SET_FF2(unsigned short stNum, float val)
	void SET_FF3(unsigned short stNum, float val)
	void SET_FF4(unsigned short stNum, float val)
	void SET_FF5(unsigned short stNum, float val)
	void SET_FF6(unsigned short stNum, float val)
	void SET_FF7(unsigned short stNum, float val)
	void SET_FF8(unsigned short stNum, float val)
	void SET_FF9(unsigned short stNum, float val)
	“ ”

4.2.31

```
int GET_MutiFF0(int stNum, int len, float *pf32Dsc)
int GET_MutiFF1(int stNum, int len, float *pf32Dsc)
int GET_MutiFF2(int stNum, int len, float *pf32Dsc)
int GET_MutiFF3(int stNum, int len, float *pf32Dsc)
int GET_MutiFF4(int stNum, int len, float *pf32Dsc)
int GET_MutiFF5(int stNum, int len, float *pf32Dsc)
int GET_MutiFF6(int stNum, int len, float *pf32Dsc)
int GET_MutiFF7(int stNum, int len, float *pf32Dsc)
int GET_MutiFF8(int stNum, int len, float *pf32Dsc)
int GET_MutiFF9(int stNum, int len, float *pf32Dsc)
```

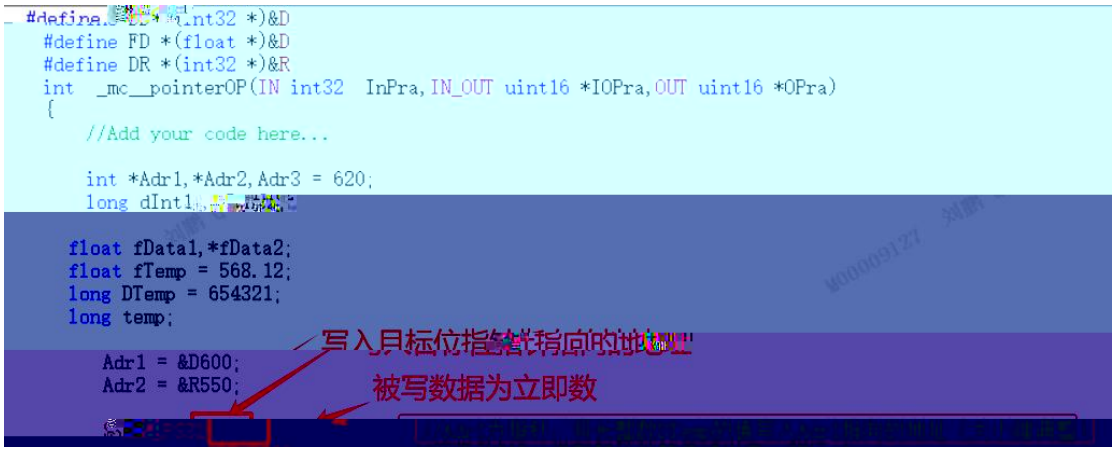
“ ”

4.2.34

	void SET_F0(int stNum, signed short val)
	void SET_F1(int stNum, signed short val)
	void SET_F2(int stNum, signed short val)
	void SET_F3(int stNum, signed short val)
	void SET_F4(int stNum, signed short val)
	void SET_F5(int stNum, signed short val)
	void SET_F6(int stNum, signed short val)
	void SET_F7(int stNum, signed short val)
	void SET_F8(int stNum, signed short val)
	void SET_F9(int stNum, signed short val)
	“ ”

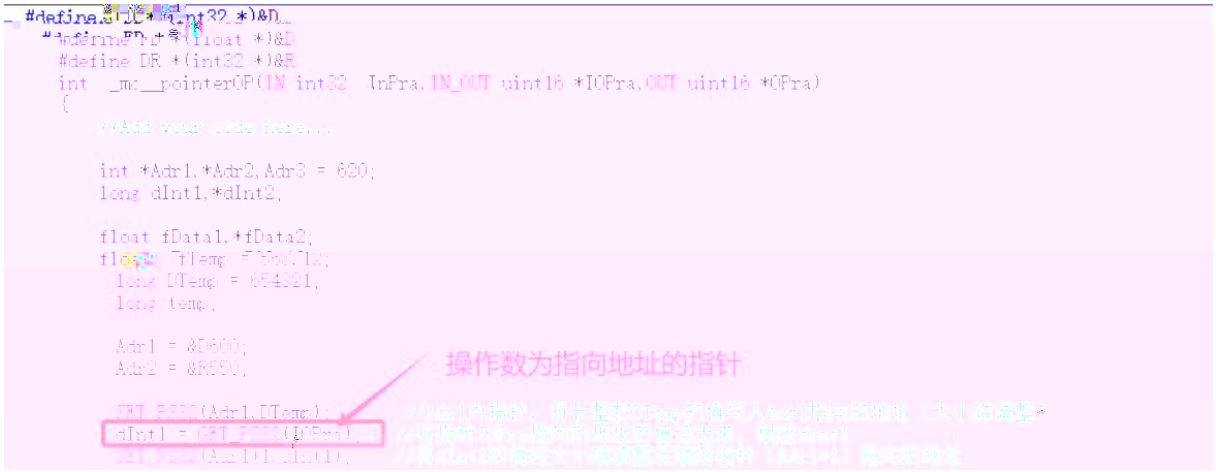
4.2.35

	void SET_PS32(int *ps32Dsc, int s32Src)
	s32Src ps32Dsc
	s32Src:
	ps32Dsc:



4.2.36

	int GET_PS32(int *ps32Src)
	ps32Src:



4.2.37

	int GET_S32(int s32Src)
	s32Src:

4.2.38

	void SET_PU32(unsigned int *pu32Dsc, unsigned int u32Src)
	u32Src pu32Dsc
	u32Src:
	pu32Dsc:

4.2.39

	unsigned int GET_PU32(unsigned int *pu32Src)
	pu32Src:

4.2.40

	unsigned int GET_U32(unsigned int u32Src)
	u32Src:

4.2.41

	void SET_PF32(float *pf32Dsc, float f32Src)
	f32Src pf32Dsc
	f32Src:
	pf32Dsc:

4.2.42

	float GET_PF32(float *pf32Src)
	pf32Src:

4.2.43

	float GET_F32(float f32Src)
	f32Src:

```
#define DD *(int32 *)&D
#define FD *(float *)&D
#define DR *(int32 *)&R
int  _mc__pointerOP(IN int32  InPra,IN_OUT uint16 *IOPra,OUT uint16 *OPra)
{
    //Add your code here...

    int *Adr1,*Adr2,Adr3 = 620;
    long dInt1,*dInt2;

    float fData1,*fData2;
    float fTemp = 568.12;
```

指针案例程序

```
dTmp = GET_PU32(Adr2);
SET_PU32(Adr1+20,dTmp)
```

5

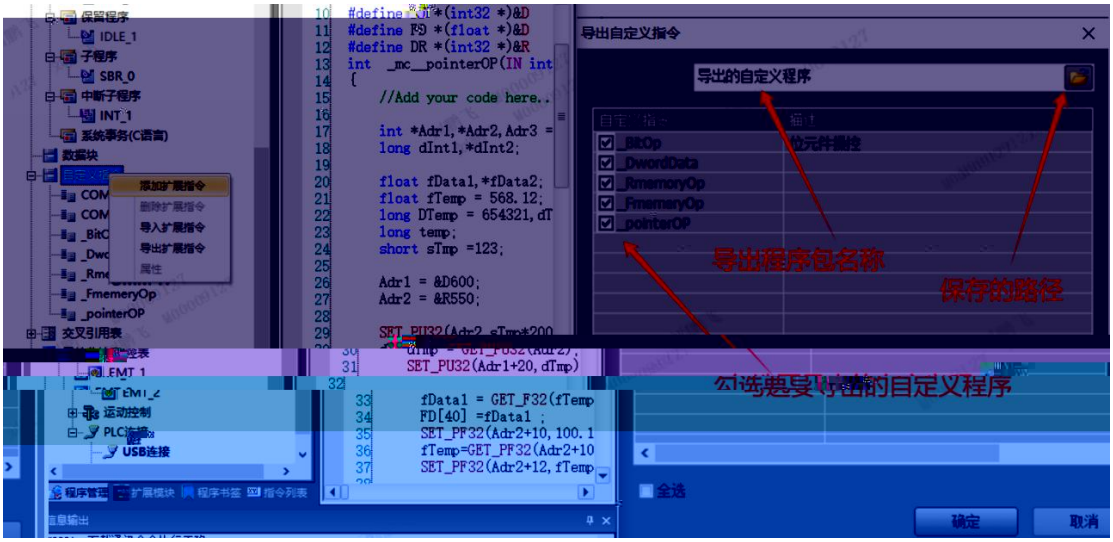
5.1.1

	unsigned short _ycrcModbus(unsigned char *data,unsigned int length)
	data: Crc
	Length: Crc

5.1.2

	unsigned short _ycrcCcitt(unsigned char *ptr, unsigned int len)
	data: Crc
	Length: Crc

6


导出自定义指令
导出的自定义程序
_BitOp
_DwordData
_RmemoryOp
_FmemoryOp
_pointerOP
导出程序包名称
保存的路径
勾选要导出的自定义程序
全选
确定
取消