



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

1

1.1

1.2

	Pl c()
M:5000, M:5100	
M:EGreator	M:5000 M
	M:EGreator

2

2.1

- M:5x00 Pl c
- 1> C C ,
- 2> Pl c , X, Y, M S, SM T, C D, R SD , Z, F

X , D

3> 64K

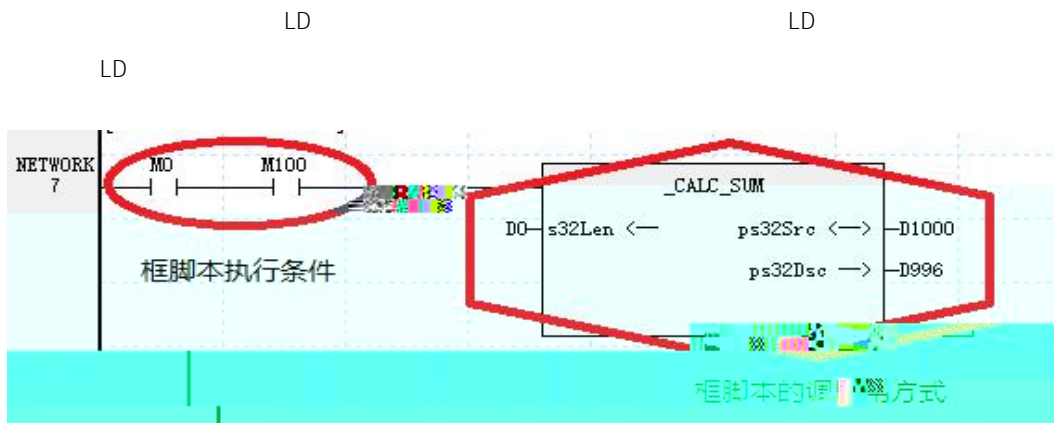
4> ANSI C C

5> C ;

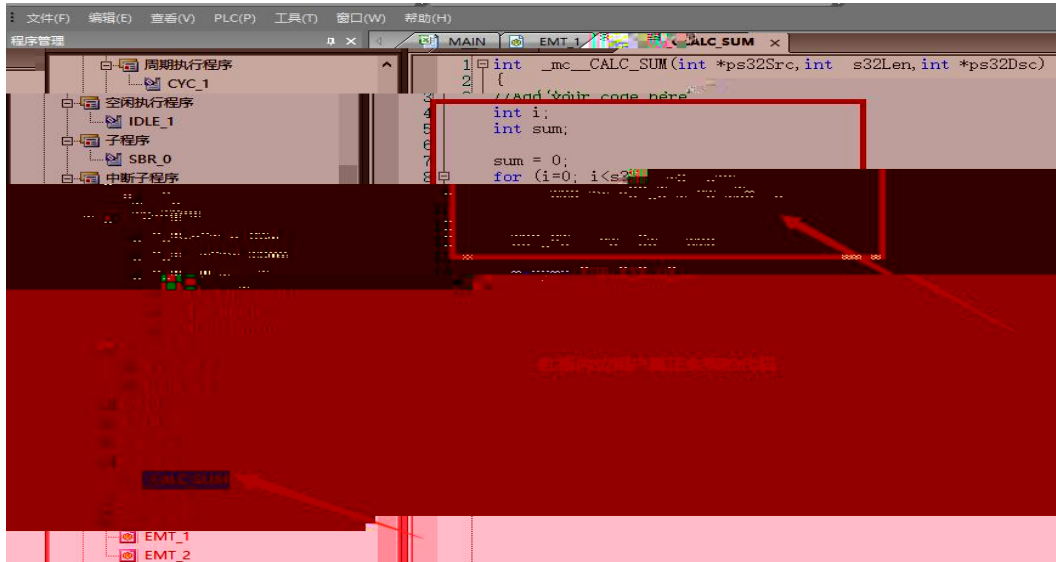
6>

2.2

1>

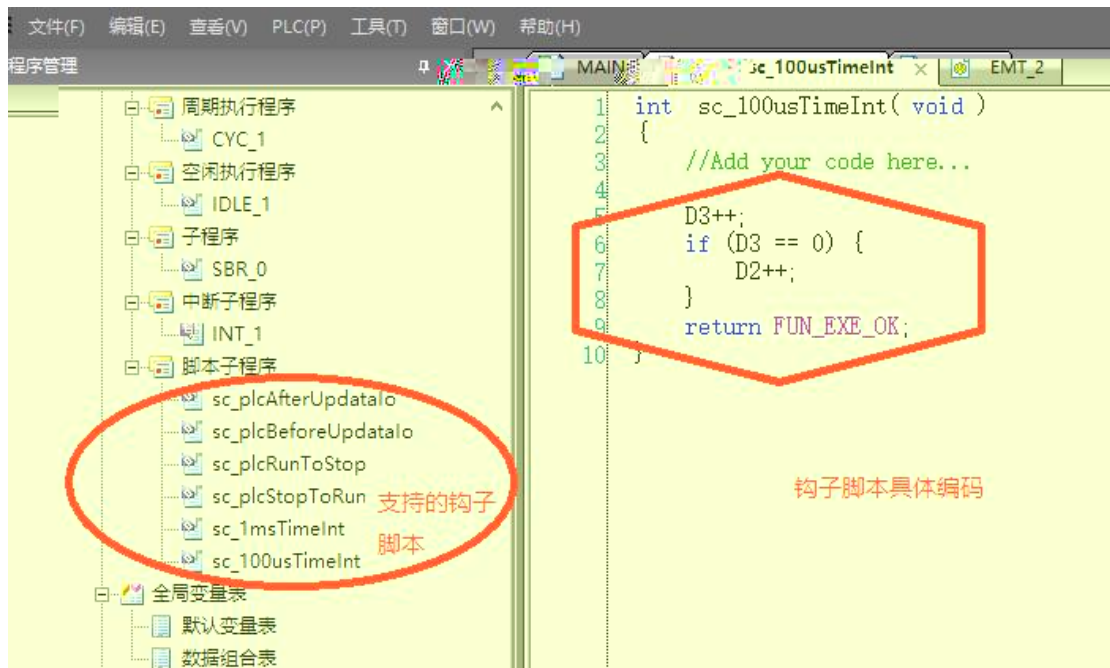


1



2

1>



3

2.3

C

pl c

<code>export_module.h</code>	Pl c
<code>user_common.c</code>	
<code>user_common.h</code>	

3

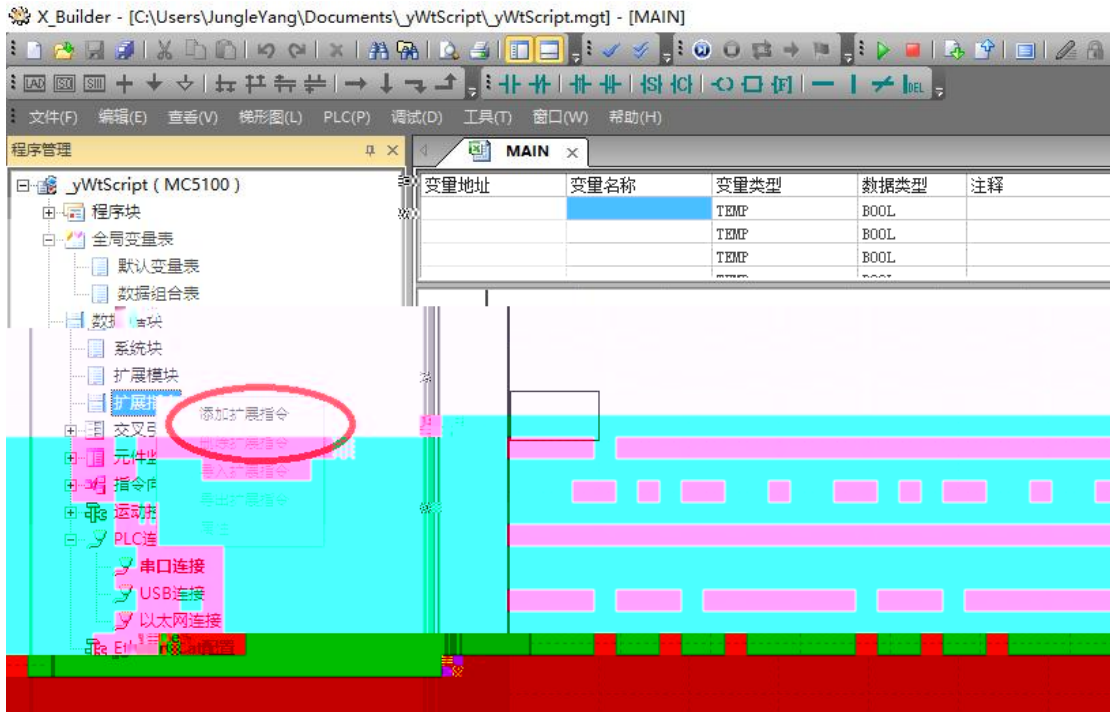
xBuild

C

D996() D1000 20000 D998()

3.1

" " , " " " "



1 13.6

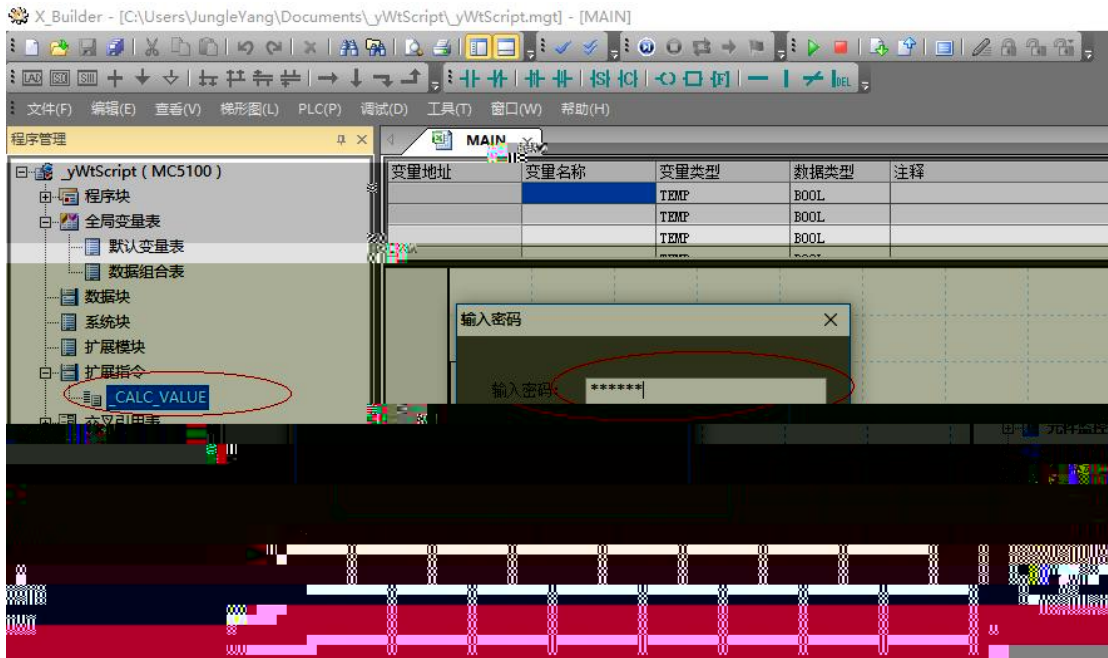
#

p p

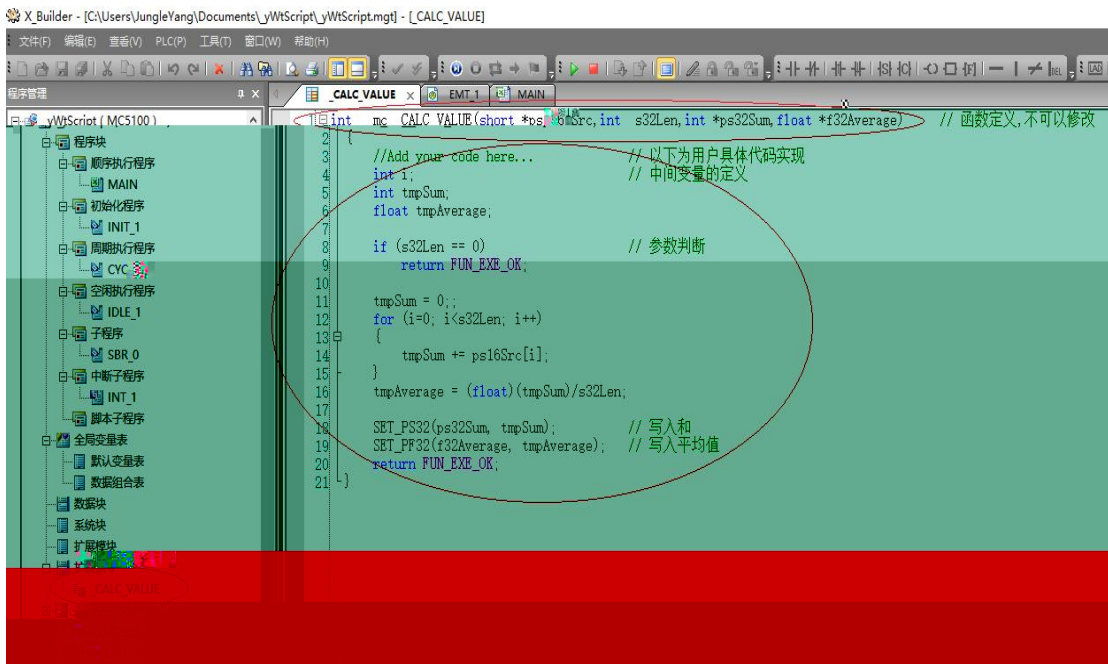
~

3.3

”_CALC_VALUE”



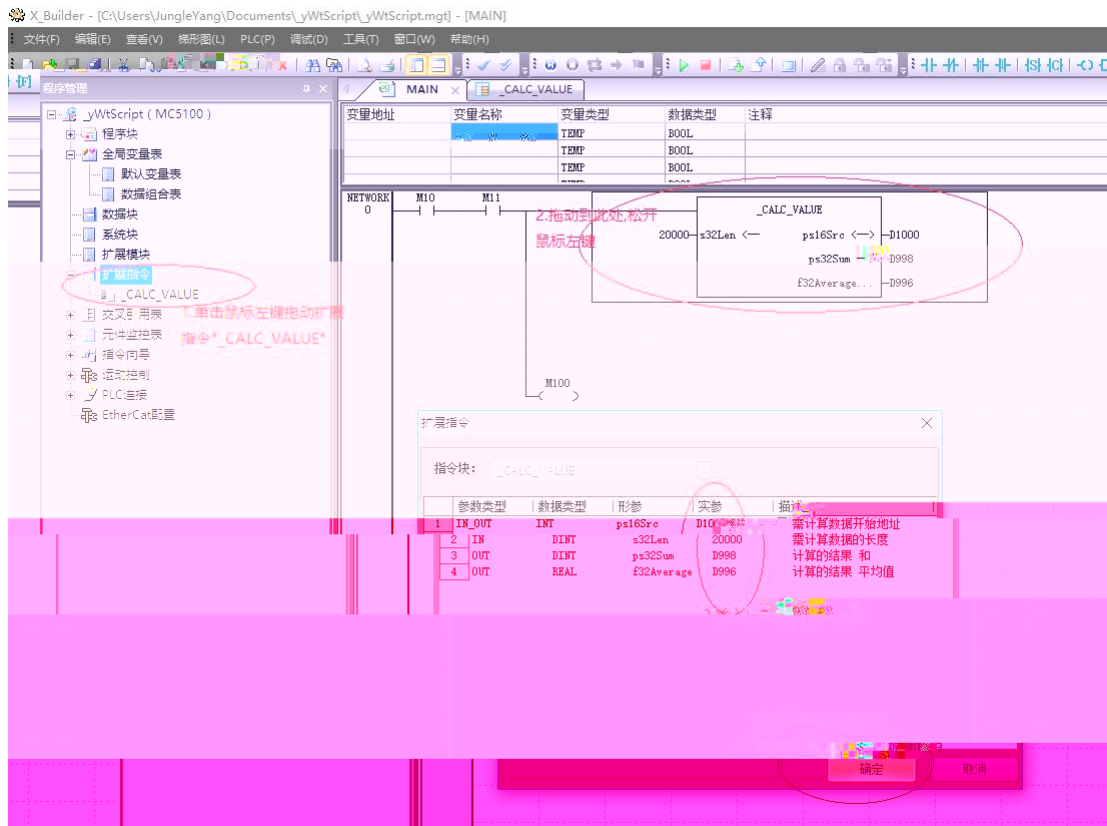
6



7

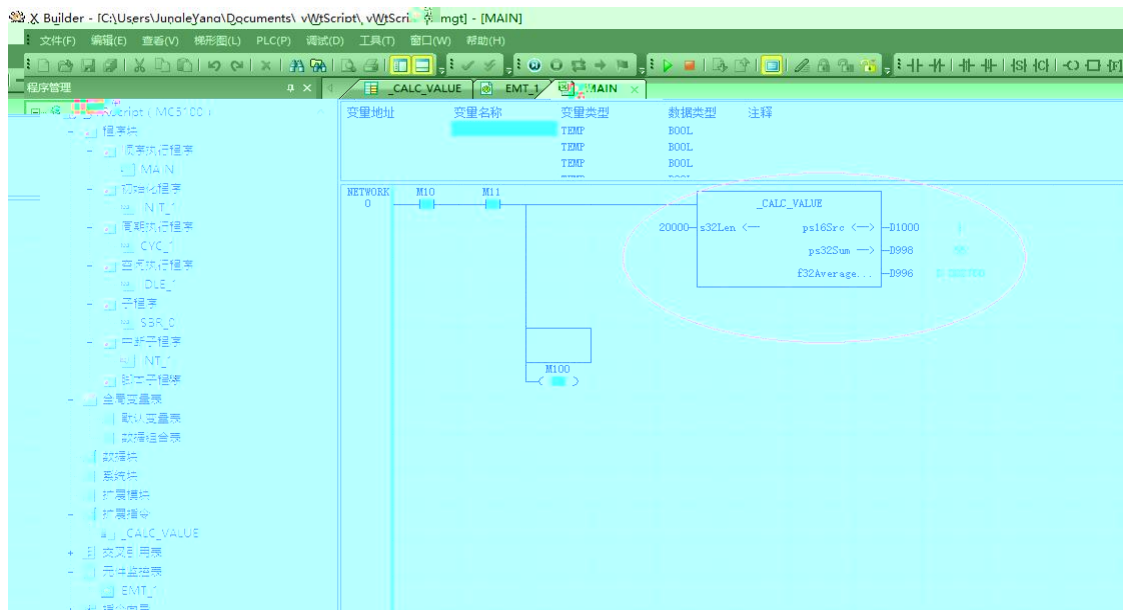
”C ”

3.4



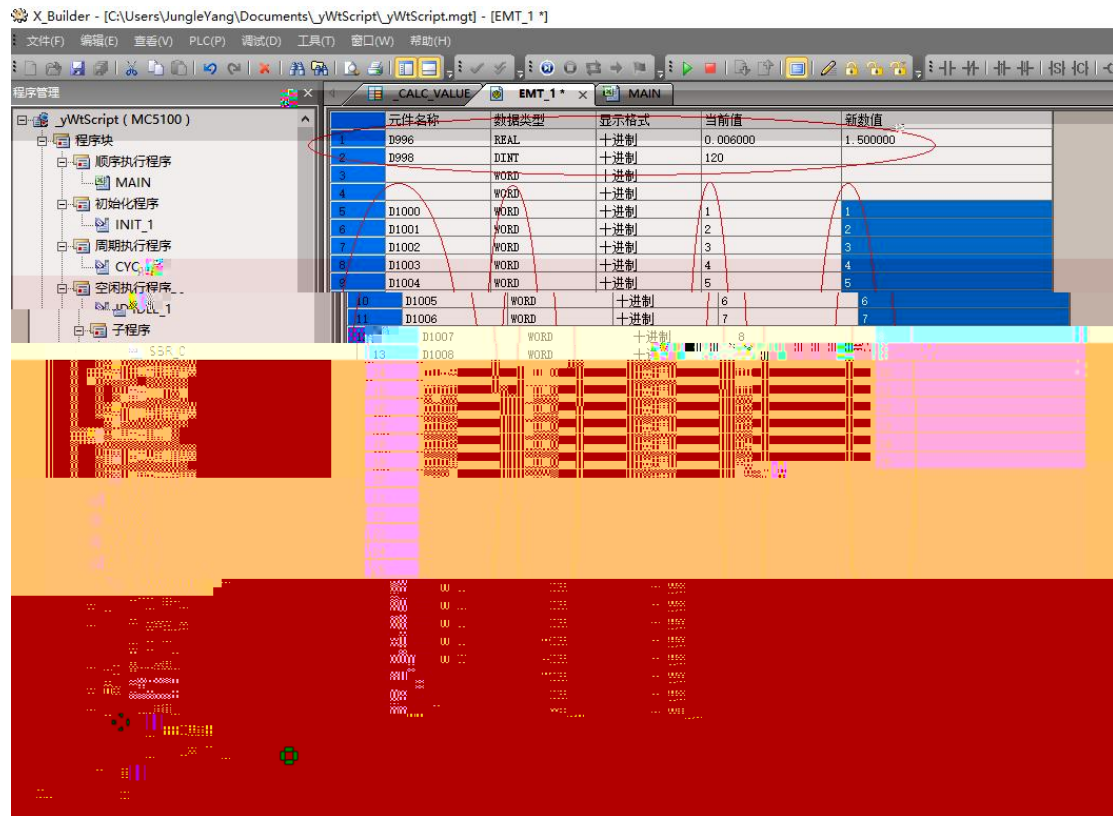
8

M10 M11 On ”_CALC_VALUE” _CALC_VALUE
 D1000 20000 D998() D996()



9

3.5



10

4

4.1

4.1.1 X

	X
	PLC
	X0 ~ X7777
	Bit
	8 , 'X'



4.1.2 Y

	Y
	PLC
	Y0 ~ Y7777
	Bit
	8 , '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

	SM
	PLC SM
	SM0 ~ SM4095
	Bit
	10 , 'SM'

```
int _mc_BitOp()
{
    //Add your code here...
    if ( X4 )
    {
        SM[40]=1;
        SM[41]=1;
    }
}
```

SM位元件支持SMxx及SM[xy]

4.1.4 S

	S
	PLC S
	S0 ~ S4095
	Bit
	10 , 'S'

```

int _mc_BitOp()
{
    //Add your code here...
    if ( X4 )
    {
        S[100]=1;
        S101 = 1;
    }
}

```

S位元件支持Sxx及S[xx]输入，编号为十进制

4.1.5 T

	T
	PLC T
	T0 ~ T4095
	Bit
	10 , 'T'

```

int _mc_BitOp()
{
    //Add your code here...
    if (T10)
    {
        D[102] =20;
    }
}

```

T元件位变量，支持Ix及I[xx]输入，十进制编号

4.1.6 C

	C
	PLC C
	C0 ~ C4095
	Bit
	10 , 'C'

```

int _mc_BitOp()
{
    //Add your code here...
    if (C10)
    {
        D[102] =20;
    }
}

```

C位元件输入格式支持Cx及C[xx]，十进制编号

4.1.7 M

	C
	PLC M
	M0 ~ M65535
	Bit
	10 , 'M'

```
int _mc_BitOp()
```

```
{
  //Add your code here...
}
```



4.1.8 SD

	SD
	PLC SD
	SD0 ~ SD4095
	signed short
	10 , 'SD'

```
int _mc_BitOp()
```

```
{
  //Add your code here...
}
```

```
if ( X4 )
```

```
{
```

```
  D110 = SD101,
```

```
  D[111] = SD[102];
```

SD支持SDxx及SD[xx]

4.1.9 Z

	Z
	PLC Z
	Z0 ~ Z4095
	signed short
	10 , 'Z'

```
int _mc_BitOp()
```

```
{
  //Add your code here...
}
```

```
if ( X4 )
```

```
{
```

```
  Z110 = 10;
```

```
  Z[4095] = 100;
```

Z元件支持Zxx及Z[xx]读写操作

4. 1. 10 D

	D
	PLC D
	D0 ~ D65535
	signed short
	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

	R
	PLC R
	R0 ~ R65535
	signed short
	10 , 'R'

```
int _mc_BitOp()
{
    //Add your code here...
    if ( X4 )
    {
        R110 = R101;
        R[111] = R [102];
    }
}
```

R字变量支持Rxx及R[xx]直接读写操作

4. 2

MC5000

PLC

32

4. 2. 1

	int GET_DD(unsigned short stNum)
	" D"
	stNum D
	int,

--	--

```
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)
	" D"
	stNum: D
	val :

```
int _mc_BitUp()
{
    //Add your code here...
    if ( X4 )
    {
        long tmp;
        tmp = GET_DD(1000);
        SET_DD(tmp);
    }
}
```

将tmp值写入到长整数D150中

4.2.3

	int GET_MultiDD(int stNum int len, int *ps32Dsc)
	" D"
	stNum: D
	Len :
	ps32Dsc:
	0 ,

4.2.4

4.2.4

	int SET_MultiDD(int stNum int len, int *ps32Src)
	" D"
	stNum: D
	Len :

	ps32Dsc:	D
	0	,

The screenshot shows a debugger interface. On the left, assembly code is displayed with comments:


```

1  /**@brief
2  * @param[out]  dWORD
3  * @return Function execute result
4  * -FUN_EXE_OK: 0 execute successful
5  * -FUN_EXE_WARN: 1 execute warning
6  * -FUN_EXE_ERR: 10 execute error
7  */
8  int _mc_DwordData(OUT int32 *dWORD)
9  {
10 //Add your code here...
11  uchar utmp;
12  int32 *dtmp;
  
```

 On the right, a register window shows registers D300 through D316. The values are:

D300	DINT	十进制	300
D302	DINT	十进制	300
D304	DINT	十进制	300
D306	DINT	十进制	400
D308	DINT	十进制	500
D310	DINT	十进制	600
D312	DINT	十进制	700
D314	DINT	十进制	800
D316	DINT	十进制	900

 Below the registers, a diagram shows a box labeled '_DwordData' with a value of 0. The text '监控运行位' is visible on the left side of the diagram area.

1 GET_Miti DD(int stNum int len, int *ps32Dsc) D
 SET_Miti DD(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)
	" D"
	stNum D
	float,

4.2.6

4.2.6

	void SET_FD(unsigned short stNum, float val)
	" D"
	stNum: D
	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_MultiFD(int stNum, int len, float *pf32Dsc)
	" D"
	stNum: D
	Len :
	ps32Dsc:
	0 ,

4.2.8

4.2.8

	int SET_MultiFD(int stNum, int len, float *pf32Src)
	" D"
	stNum: D
	Len :
	ps32Dsc: D
	0 ,

4. 2. 12

	int SET_MultiDR(int stNum, int len, int *ps32Src)
	" R"
	stNum: R
	Len :
	ps32Dsc: R
	0 ,

4. 2. 13

	float GET_FR(unsigned short stNum)
	" R"
	stNum R
	float,

4. 2. 14

	void SET_FR(unsigned short stNum, float val)
	" R"
	stNum: R
	val :

4. 2. 15

	int GET_MultiFR(int stNum, int len, float *pf32Dsc)
	" R"
	stNum: R
	Len :
	ps32Dsc:

	0	,

4. 2. 16

	int SET_Miti FR(int stNum, int len, float *pf32Src)	
	" R"	
	stNum: R	
	len :	
	ps32Dsc: R	
	0	,

4. 2. 17

	int GET_DF(int stNum)
	" F"
	stNum F
	int,

4.2.34 F0~F9

4. 2. 18

	void SET_DF(int stNum, int val)
	" F"
	stNum: F
	val :

4.2.34 F0~F9

4. 2. 19

	int GET_Miti DF(int stNum, int len, int *ps32Dsc)
	" F"

	stNum: F
	Len :
	ps32Dsc:
	0 ,

4.2.34 F0~F9

4. 2. 20

	int SET_MitIDF(int stNum, int len, int *ps32Src)
	" F"
	stNum: F
	Len :
	ps32Dsc: F
	0 ,

4.2.34 F0~F9

4. 2. 21

	float GET_FF(unsigned short stNum)
	" F"
	stNum F
	float,

4.2.34 F0~F9

4. 2. 22

	void SET_FF(unsigned short stNum, float val)
	" F"
	stNum: F
	val :

4.2.34 F0~F9

4. 2. 23

	int GET_Miti FF(int stNum, int len, float *pf32Dsc)
	" F"
	stNum: F
	Len :
	ps32Dsc:
	0 ,

4.2.34 F0~F9

4. 2. 24

	int SET_Miti FF(int stNum, int len, float *pf32Src)
	" F"
	stNum: F
	Len :
	ps32Dsc: F
	0 ,

4.2.34 F0~F9

4. 2. 25

	int GET_DFO(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)
	" Fx"
	stNum Fx
	int,

4.2.34 F0~F9

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)
	" Fx"
	stNum: Fx
	val :

4.2.34 F0~F9

4. 2. 27

	int GET_MitiDF0(int stNum, int len, int *ps32Dsc)
	int GET_MitiDF1(int stNum, int len, int *ps32Dsc)
	int GET_MitiDF2(int stNum, int len, int *ps32Dsc)
	int GET_MitiDF3(int stNum, int len, int *ps32Dsc)
	int GET_MitiDF4(int stNum, int len, int *ps32Dsc)
	int GET_MitiDF5(int stNum, int len, int *ps32Dsc)
	int GET_MitiDF6(int stNum, int len, int *ps32Dsc)
	int GET_MitiDF7(int stNum, int len, int *ps32Dsc)
	int GET_MitiDF8(int stNum, int len, int *ps32Dsc)
	int GET_MitiDF9(int stNum, int len, int *ps32Dsc)
	" Fx"
	stNum: Fx
	Len :
	ps32Dsc:
	0 ,

4.2.34 F0~F9

4. 2. 28

	int SET_MultiDF0(int stNum, int len, int *ps32Src)
	int SET_MultiDF1(int stNum, int len, int *ps32Src)
	int SET_MultiDF2(int stNum, int len, int *ps32Src)
	int SET_MultiDF3(int stNum, int len, int *ps32Src)
	int SET_MultiDF4(int stNum, int len, int *ps32Src)
	int SET_MultiDF5(int stNum, int len, int *ps32Src)
	int SET_MultiDF6(int stNum, int len, int *ps32Src)
	int SET_MultiDF7(int stNum, int len, int *ps32Src)
	int SET_MultiDF8(int stNum, int len, int *ps32Src)
	int SET_MultiDF9(int stNum, int len, int *ps32Src)
	" F"
	stNum: F
	Len :
	ps32Dsc: F
	0 ,

4.2.34 F0~F9

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)
	" Fx"
	stNum Fx
	float,

4.2.34 F0~F9

4. 2. 30

	voi d SET_FF0(unsig ned short stNum, fl oat val)
	voi d SET_FF1(unsig ned short stNum, fl oat val)
	voi d SET_FF2(unsig ned short stNum, fl oat val)
	voi d SET_FF3(unsig ned short stNum, fl oat val)
	voi d SET_FF4(unsig ned short stNum, fl oat val)
	voi d SET_FF5(unsig ned short stNum, fl oat val)
	voi d SET_FF6(unsig ned short stNum, fl oat val)
	voi d SET_FF7(unsig ned short stNum, fl oat val)
	voi d SET_FF8(unsig ned short stNum, fl oat val)
	voi d SET_FF9(unsig ned short stNum, fl oat val)
	" Fx"
	stNum: Fx
	val :

4.2.34 F0~F9

4. 2. 31

```

i nt GET_Miti FF0(i nt stNum, i nt len, fl oat *pf 32Dsc)
i nt GET_Miti FF1(i nt stNum, i nt len, fl oat *pf 32Dsc)
i nt GET_Miti FF2(i nt stNum, i nt len, fl oat *pf 32Dsc)
i nt GET_Miti FF3(i nt stNum, i nt len, fl oat *pf 32Dsc)
i nt GET_Miti FF4(i nt stNum, i nt len, fl oat *pf 32Dsc)
i nt GET_Miti FF5(i nt stNum, i nt len, fl oat *pf 32Dsc)
i nt GET_Miti FF6(i nt stNum, i nt len, fl oat *pf 32Dsc)
i nt GET_Miti FF7(i nt stNum, i nt len, fl oat *pf 32Dsc)
i nt GET_Miti FF8(i nt stNum, i nt len, fl oat *pf 32Dsc)
i nt GET_Miti FF9(i nt stNum, i nt len, fl oat *pf 32Dsc)
" Fx"

```

o



4. 2. 34

	voi d SET_F0(i nt stNum si gned short val)
	voi d SET_F1(i nt stNum si gned short val)
	voi d SET_F2(i nt stNum si gned short val)
	voi d SET_F3(i nt stNum si gned short val)
	voi d SET_F4(i nt stNum si gned short val)
	voi d SET_F5(i nt stNum si gned short val)
	voi d SET_F6(i nt stNum si gned short val)
	voi d SET_F7(i nt stNum si gned short val)
	voi d SET_F8(i nt stNum si gned short val)
	voi d SET_F9(i nt stNum si gned short val)
	" Fx"
	stNum: Fx
	val :

F0~F9

```

//Add your code here...
int iTemp, iTemp1;
int iAdr= 400, iAdr1=800;
float fData= 90.12;
int *piData;
float *s;

SET_FF0(iAdr, 40.12);
SET_FF1(iAdr, 50.12);
SET_FF2(iAdr, 60.12);
SET_FF3(iAdr, 70.12);
SET_FF4(iAdr, 40.12);
SET_FF5(iAdr, 50.12);
SET_FF6(iAdr, 60.12);
SET_FF7(iAdr, 70.12);
SET_FF8(iAdr, fData*10);
SET_FF9(iAdr, fData);

SET_Bu112F0;
SET_Bu112F4;

SET_Bu112F5;
SET_Bu112F9;

```

长整数多个连续参数故用指针变量

多个单精度数相加和的参数使用指针变量

4. 2. 35

	voi d SET_PS32(i nt *ps32Dsc, i nt s32Src)
	s32Src ps32Dsc
	s32Src:
	ps32Dsc:

```

#define ID *(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;

    float fData1,*fData2;
    float fTemp = 568.12;
    long DTemp = 654321;
    long temp;

    Adr1 = &D600;
    Adr2 = &R550;
    SET_PS32(Adr1, DTemp); //Adr1为指针，将长整数DTemp的值写入Adr1指向的地址（大小端调整）
}

```

写入目标地址指向的地址
被写数据为立即数

Adr1 D600 Adr2

R550

4.2.36

	int GET_PS32(int *ps32Src)
	int
	ps32Src:

```

#define ID *(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;

    float fData1,*fData2;
    float fTemp = 568.12;
    long DTemp = 654321;
    long temp;

    Adr1 = &D600;
    Adr2 = &R550;
    SET_PS32(Adr1, DTemp); //Adr1为指针，将长整数DTemp的值写入Adr1指向的地址（大小端调整）
    dInt1 = GET_PS32(IOPra); //将指针IOPra指向的地址的值读出来，赋给dInt1
    SET_PS32(Adr1+1, dInt1); //将dInt1的值经大小端调整后赋给指针（Adr1+1）指向的地址
}

```

将数据为指向地址的指针

IOPra

4.2.37

	int GET_S32(int s32Src)
	int 32
	s32Src:

	PLC

*1

```
#define DD *(int32 *)&D
#define FD *(float *)&D
#define ...

int
long

float
float
long...
long...
```

4. 2. 38

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

4.4.23

4. 2. 39

	unsigned int GET_PU32(unsigned int *pu32Src)
	int
	pu32Src:

4.4.23

4. 2. 40

	unsigned int GET_U32(unsigned int u32Src)
	Unsigned int 32
	u32Src:
	PLC

*2

```
#define DD *(int32 *)&D
#define FD *(float *)&D
// ...
int
long
float
float
long
long
// ...
GET_P32
// ...
// ...
```

4. 2. 41

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

4.4.23

4. 2. 42

	float GET_PF32(float *pf32Src)
	int
	pf32Src:

5

5.1.1

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

5.1.2

	unsigned short _ycrcCrc16(unsigned char *ptr, unsigned int len)
	ccitt
	data: Crc
	Length: Crc
	Crc

6

:

