



	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

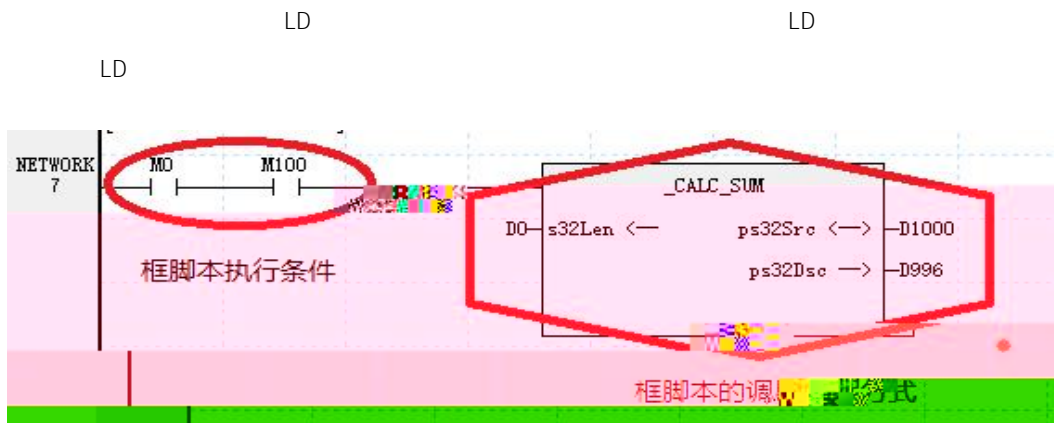
## 2.1

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

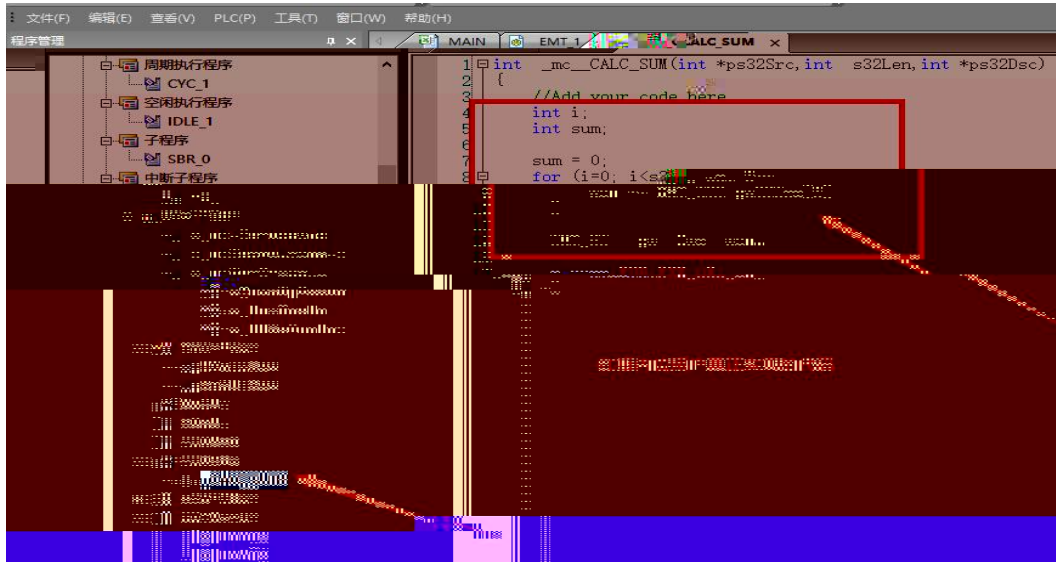
- 3> X , D
- 4> ANSI C , C
- 5> C ;
- 6>

## 2.2

1>

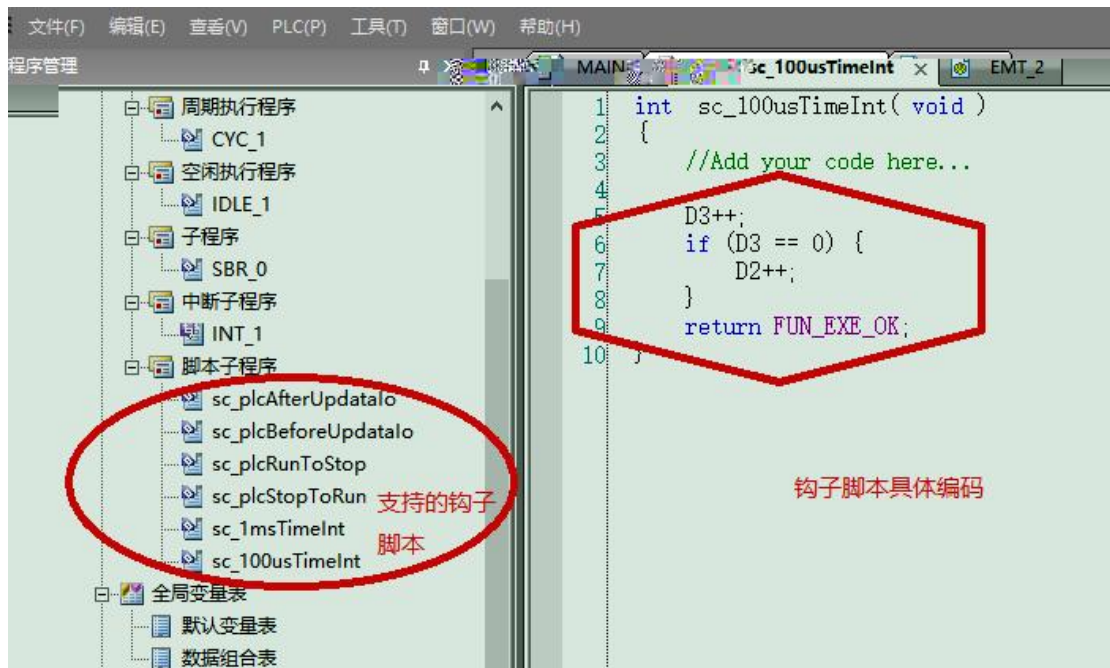


1



2

1>



3

## 2.3

C

pl c

export_nodule.h	Pl c
user_common.c	
user_common.h	

# 3

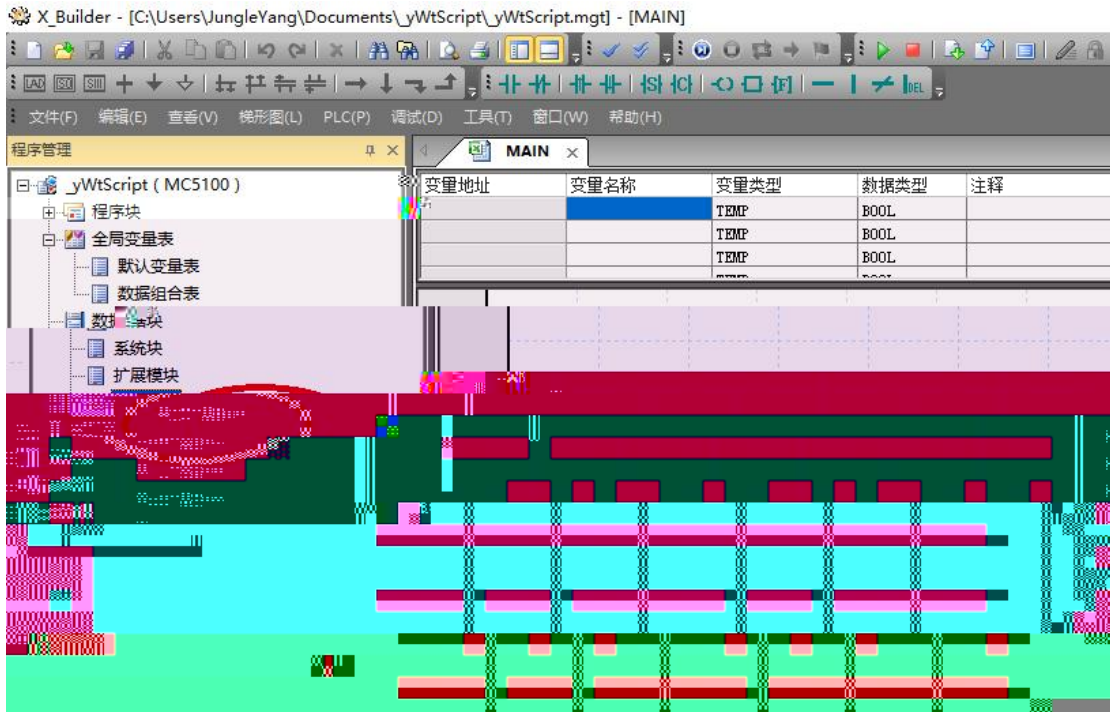
xBuild

C

D996( )      D1000      20000      D998( )

## 3.1

" " , " " " "



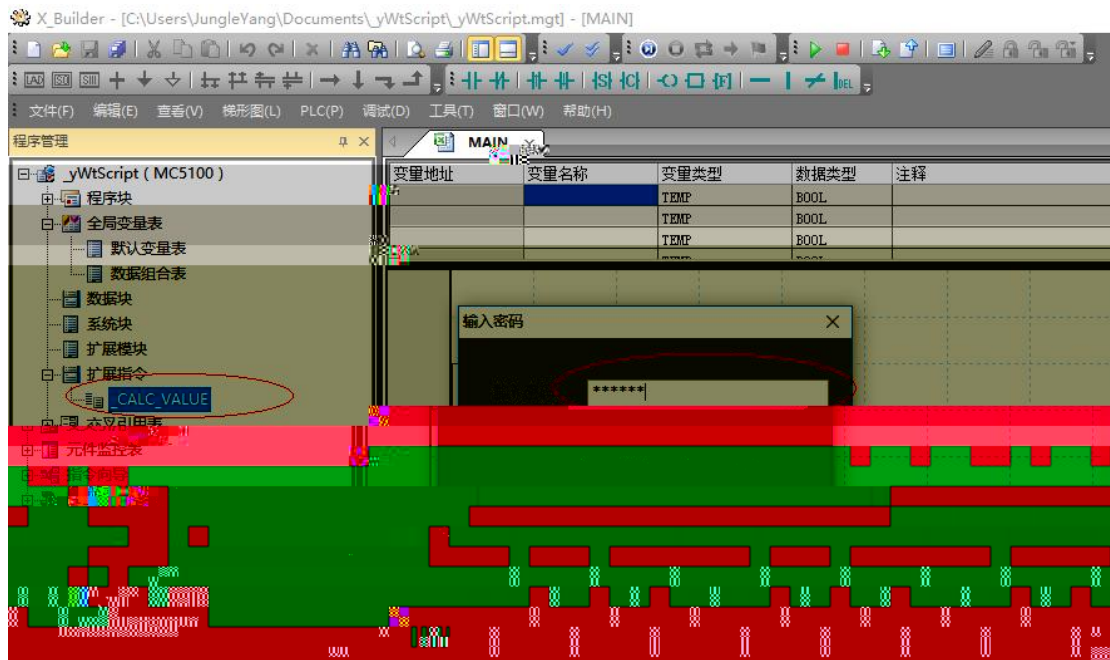
---

3.6

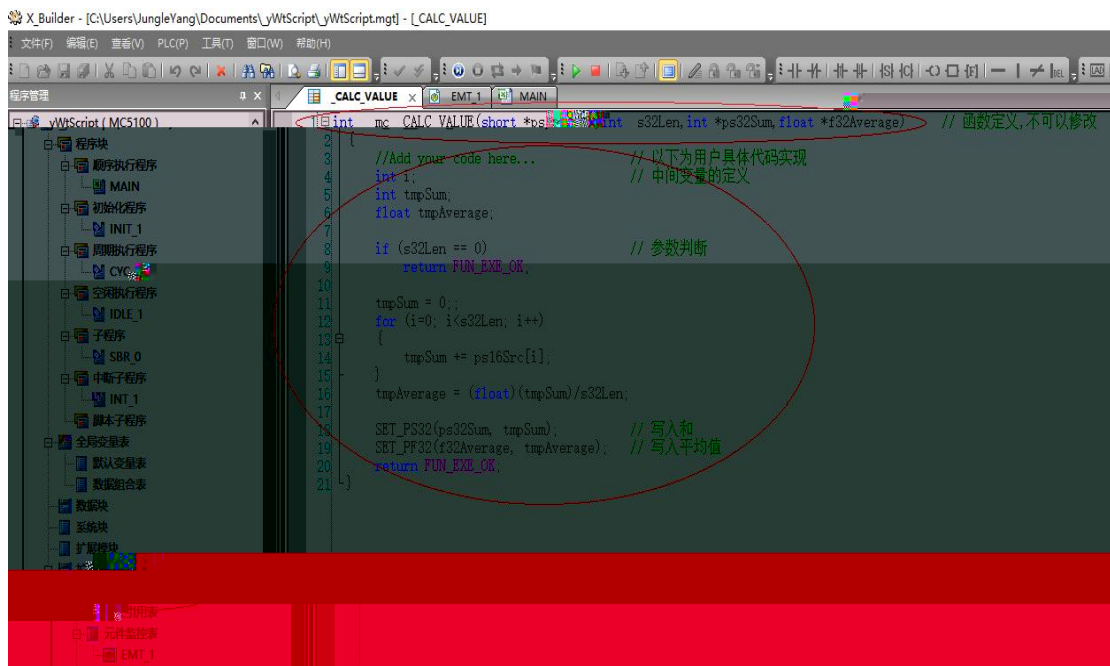
6

### 3.3

”\_CALC\_VALUE”



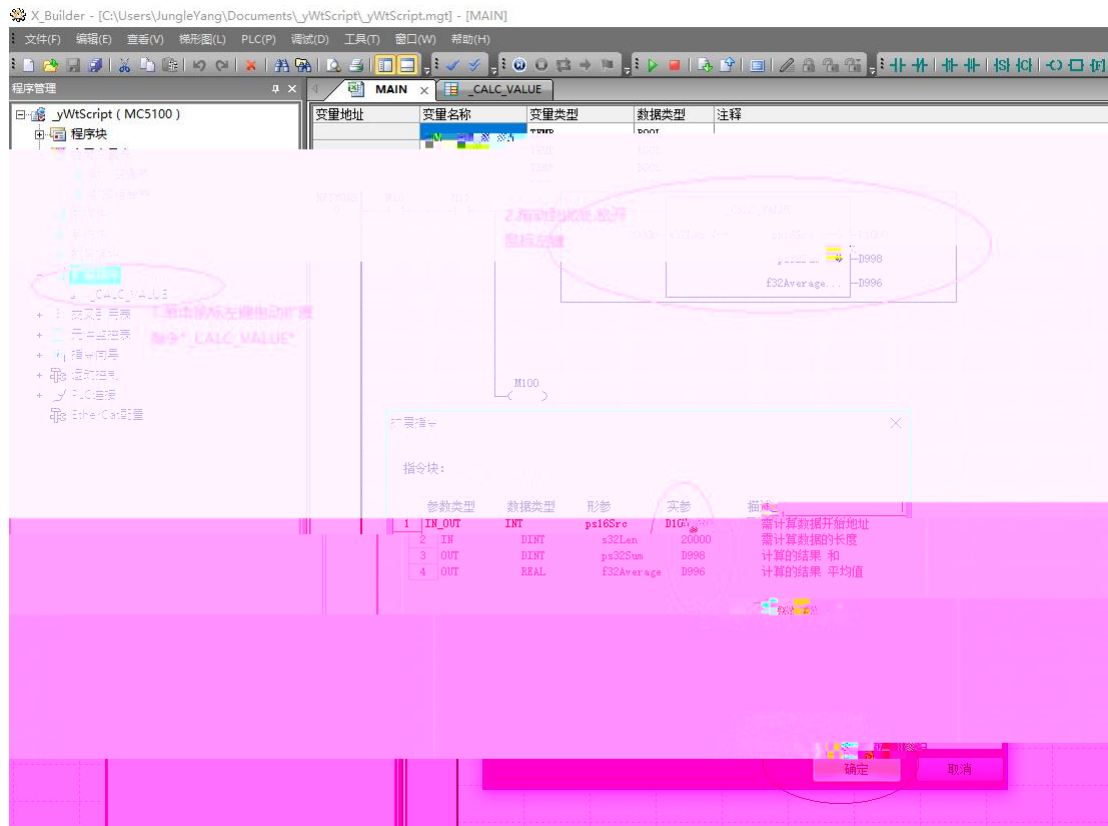
6



7

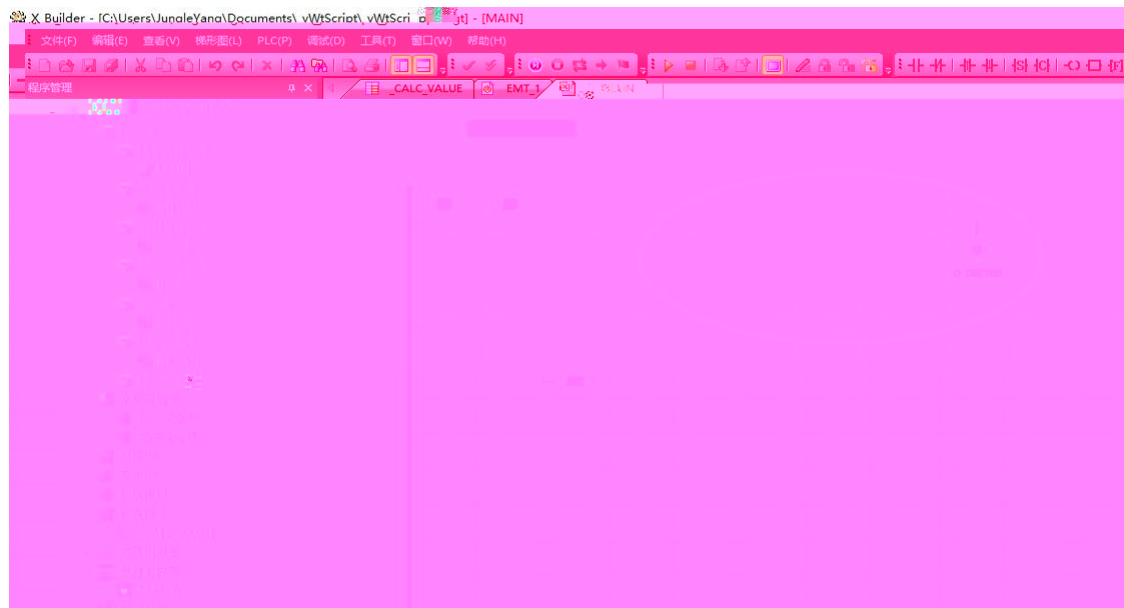
”C ”

### 3.4



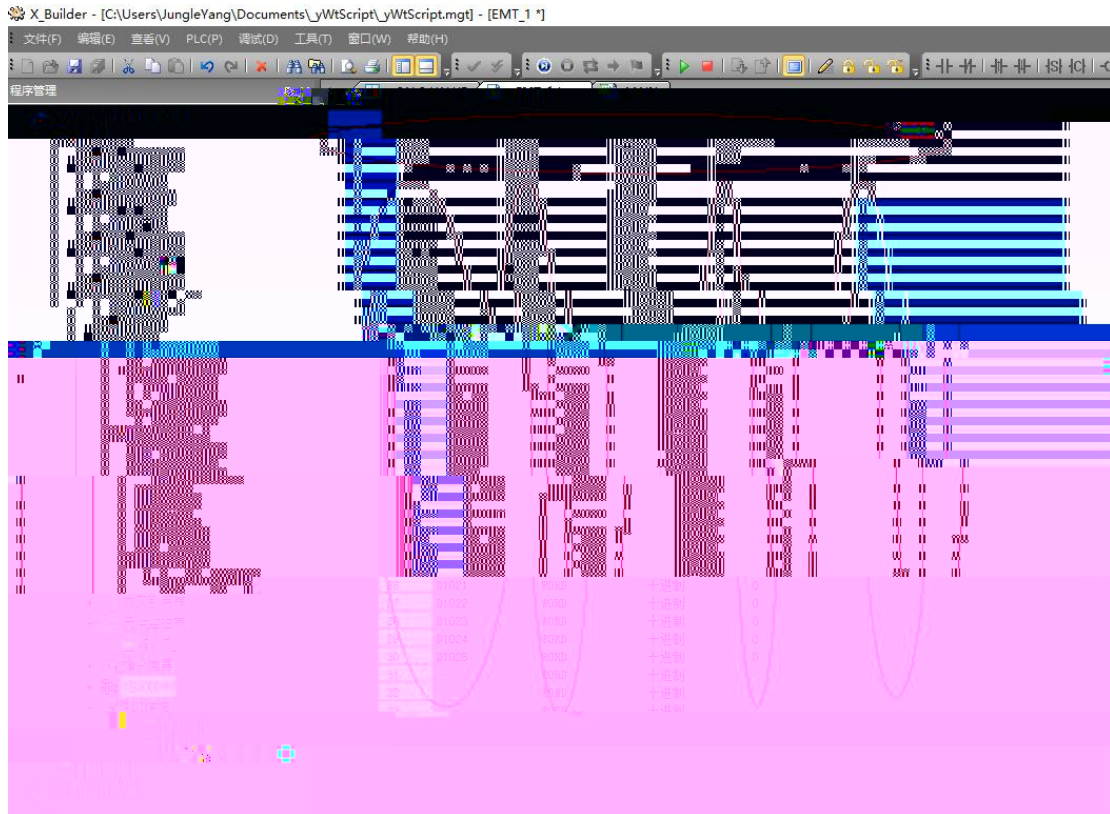
8

M10 M11 On ”\_CALC\_VALUE” \_CALC\_VALUE  
 D1000 20000 D998( ) D996(  
 )



9

## 3.5



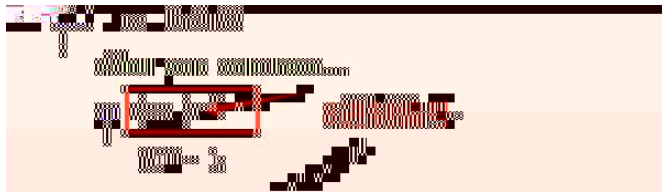
10

## 4

### 4.1

#### 4.1.1 X

	X
	PLC
	X0 ~ X7777
	Bit
	8 , 'X'



X[10]

X[XX]

### 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

Y[15]

Y[XX]

### 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[xx]

SM

## 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Ixx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Ixx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...
```

```
if ( X4 )
```

```
{
```

```
    # www.megmeet.com
```

M位元件支持Mxx及M[xx]读与写操作

## 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]

32 SD

16 SD

32

## 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字元件支持Zxx及Z[xx]读与写操作

## 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 = RD101;
        R[111] = RD [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 D

	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 D

	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);
    }
}
```

### 4.2.3 D

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

4.2.4

### 4.2.4 D

	int SET_Multi DD(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 visible with comments in Chinese. The middle window shows a register list with addresses D300 to D316 and their corresponding decimal values. On the right, a data display window shows a DWORD value of 0 at memory address D400. Red circles highlight the values 300, 400, 500, 600, 700, 800, 900 in the register window and the 0 in the data display window.

```

1          GET_Miti DD(int stNum int len, int *ps32Dsc)
SET_Miti DD(int stNum int len, int *ps32Src)          D
              DWORD    D400          DWORD    D400    10

2          int *ps32Dsc int *ps32Src

```

### 4.2.5 D

	float GET_FD(unsigned short stNum)
	" D"
	stNum D
	float,

#### 4.2.6

## 4.2.6

## D

	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

## D

	int GET_MultiFD(int stNum, int len, float *pf32Dsc)
	" D"
	stNum: D
	Len :
	ps32Dsc:
	0 ,

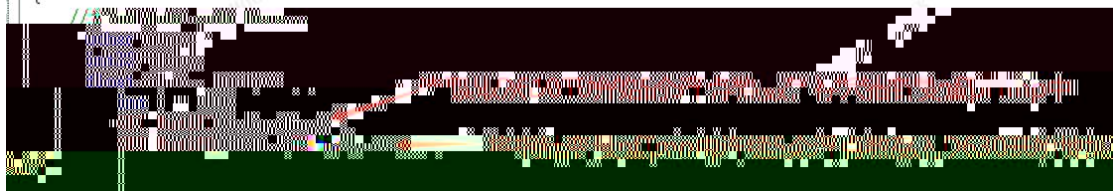
4.2.8

## 4.2.8

## D

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

```
int _mc_DwordData(OUT int32 *dWORD)
```



i 16

#### 4.2.9 R

R D

	int GET_DR(unsigned short stNum)
	" R"
	stNum R
	int,

#### 4.2.10 R

	void SET_DR(unsigned short stNum, int val)
	" R"
	stNum: R
	val :

#### 4.2.11 R

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

---

#### 4. 2. 12

**R**

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

#### 4. 2. 13

**R**

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

#### 4. 2. 14

**R**

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

#### 4. 2. 15

**R**

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

	0	,

#### 4.2.16

**R**

	int SET_MitiFR(int stNum, int len, float *ps32Src)	
	" R"	
	stNum: R	
	len :	
	ps32Dsc: R	
	0	,

#### 4.2.17

**F**

**F0 ~ F9**

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

4.2.34 F0~F9

#### 4.2.18

**F**

**F0 ~ F9**

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

4.2.34 F0~F9

#### 4.2.19

**F**

**F0 ~ F9**

	int GET_MitiDF(int stNum, int len, int *ps32Dsc)
	" F"

	stNum: F
	Len :
	ps32Dsc:
	0 ,

4.2.34 F0~F9

4. 2. 20

**F**

**F0 ~ F9**

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

4.2.34 F0~F9

4. 2. 21

**F**

**F0 ~ F9**

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

4.2.34 F0~F9

4. 2. 22

**F**

**F0 ~ F9**

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

4.2.34 F0~F9

4. 2. 23

**F**

**F0 ~ F9**

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

4.2.34 F0~F9

4. 2. 24

**F**

**F0 ~ F9**

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

4.2.34 F0~F9

4. 2. 25

**Fx**

**F0 ... F9**

	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

**Fx**

**F0 ... F9**

	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

**Fx**

**F0 ... F9**

	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

**Fx**

**F0 ... F9**

	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

**Fx**

**F0 ... F9**

	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

**Fx**

**F0 ... F9**

	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)
	" Fx"
	stNum: Fx
	val :

4.2.34 F0~F9

#### 4. 2. 31

**Fx**

**F0 ... F9**

```

int GET_MitiFF0(int stNum, int len, float *pf32Dsc)
int GET_MitiFF1(int stNum, int len, float *pf32Dsc)
int GET_MitiFF2(int stNum, int len, float *pf32Dsc)
int GET_MitiFF3(int stNum, int len, float *pf32Dsc)
int GET_MitiFF4(int stNum, int len, float *pf32Dsc)
int GET_MitiFF5(int stNum, int len, float *pf32Dsc)
int GET_MitiFF6(int stNum, int len, float *pf32Dsc)
int GET_MitiFF7(int stNum, int len, float *pf32Dsc)
int GET_MitiFF8(int stNum, int len, float *pf32Dsc)
int GET_MitiFF9(int stNum, int len, float *pf32Dsc)
" Fx"
v n

```



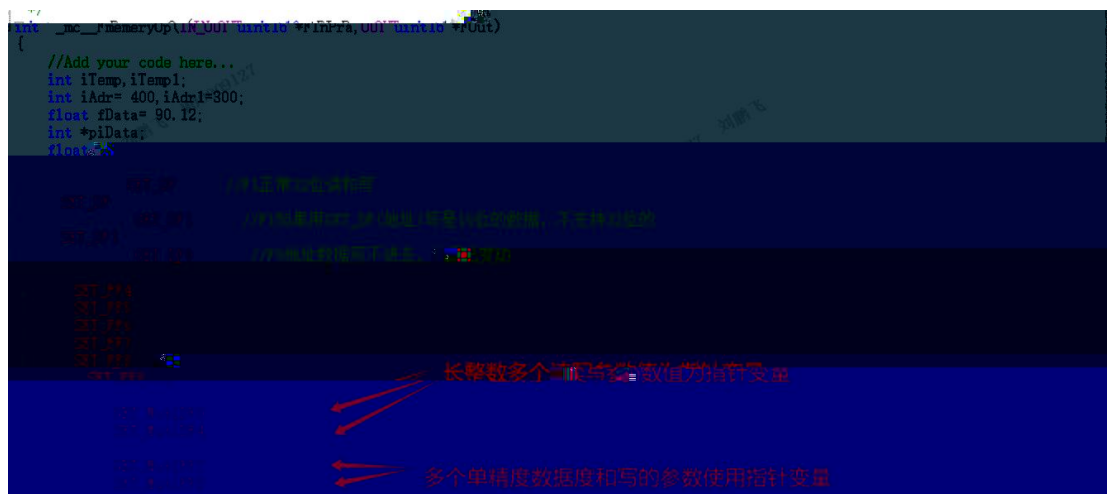
#### 4. 2. 34

#### Fx

#### F0 ... F9

	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



#### 4. 2. 35

#### int

#### ( )

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

```

#define SI *(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;
}

```

写入寄存器地址寄存器地址  
被写数据为立即数

Adr1 D600 Adr2

R550

### 4.2.36 int ( )

	int GET_PS32(int *ps32Src)
	int
	ps32Src:

```

#define SI *(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;
    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+dInt1); //将dInt1的值大小端调整后赋给指针（Adr1+1）指向的地址
}

```

操作数为指向地址的指针

IOPra

### 4.2.37 int ( )

	int GET_S32(int s32Src)
	int 32
	s32Src:

	PLC

\*1

```

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

int
long

float
float
long... = 101
long...

//Addr1为指针，将长整数Temp的值... +1指向的地址...
dInt1 = GET_U32(D650);
temp = GET_U32(dInt1); //将长整数dInt1的值经大小端调整后赋给变量Temp
D650 = temp; //将经过大小端调整的长整数结果temp的值直接赋给D650
dInt1 *= -2; //将dInt1的值经大小端调整后赋给指针Addr1指向的地址
SET_DR(Addr1, dInt1); //将长整数(1650*100)的值经大小端调整后赋给变量dInt1
dInt1 = GET_U32(1650*100); //将长整数(1650*100)的值经大小端调整后赋给变量dInt1
SET_DR(Addr1, dInt1); //将经过大小端调整的长整数结果dInt1的值直接赋给
R652

```

#### 4. 2. 38 unsigned int ( )

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

4.4.23

#### 4. 2. 39 unsigned int ( )

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

4.4.23

#### 4. 2. 40      **unsigned int**      (      )

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

\*2

```

#define DD *(int32 *)&D
#define FD *(float *)&D
#define GET_U32(Addr) ((uint32_t) (*(uint32_t *) Addr))
#define GET_I32(Addr) ((int32_t) (*(int32_t *) Addr))
#define GET_F32(Addr) (*(float *) Addr)

int
long

float
float
long
long

//Adr1为指针，将长整数Temp的值经大小端调整后赋给dInt1
dInt1 = GET_U32(Addr1); //将长整数Temp的值经大小端调整后赋给变量Temp
Temp = GET_I32(dInt1); //将经过大小端调整的长整数结果Temp的值直接赋给D650
dInt1 *= -2;
SET_DR(Addr3, dInt1*10); //将dInt1*10后的值经大小端调整后赋给指针Addr3指向的地址
dInt1 = GET_U32(650*100); //将长整数(650*100)的值经大小端调整后赋给变量dInt1
dInt1 = GET_I32(dInt1); //将经过大小端调整的长整数结果dInt1的值直接赋给R652

```

#### 4. 2. 41      **float**      (      )

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

4.4.23

#### 4. 2. 42      **float**      (      )

	float GET_PF32(float *pf32Src)
	int
	pf32Src:


4.4.23

## 4.2.43 float ( )

	float GET_F32(float f32Src)
	int
	f32Src:

```

10 #define DD *(int32 *)&D
11 #define FD *(float *)&D
12 #define DR *(int32 *)&R
13 int __mc_pointerOP(IN int32 InPra, IN_OUT uint16 *IOPra, OUT uint16 *OPra)
14 {
15     //Add your code here...
16
17     int *Adr1,*Adr2,Adr3 = 620;
18     long dI=1,&dI=2,...
19
20     float fData1,*fData2;
21     float fTemp = 568.12;

```

1                                    +1                    PLC                                    2  
2 GET\_F32 float Src

PLC

FD[40]=GET\_F32(fTemp );

```

dTemp = GET_PU32(Adr2);
GET_F32(Adr1+20, dTemp)

```

---

# 5

## 5.1.1 **modbus crc**

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

## 5.1.2 **ccit crc**

	unsigned short _ycrcCci tt(unsigned char *ptr, unsigned int len)
	<b>ccitt</b>
	data:            Crc
	Length:          Crc
	Crc

# 6

:

