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

1

1.1

1.2

	Plc( )
Mc5000,Mc5100	
MEGreator	MC5000 MU
	MEGreator

2

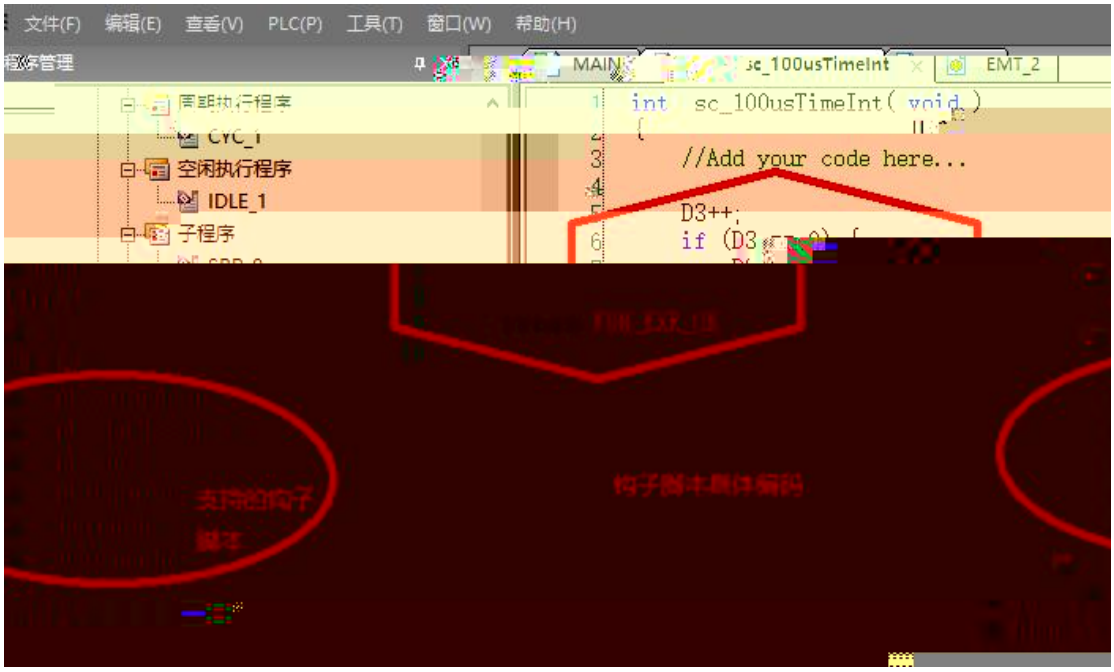
2.1

Mc5x00 Plc

1>. C C ,

2>. Plc , X, Y, M, S, SM, T, C; D, R, SÏ Xp |

D



3

2.3

C

plc

export_module.h	Plc
user_common.c	
user_common.h	

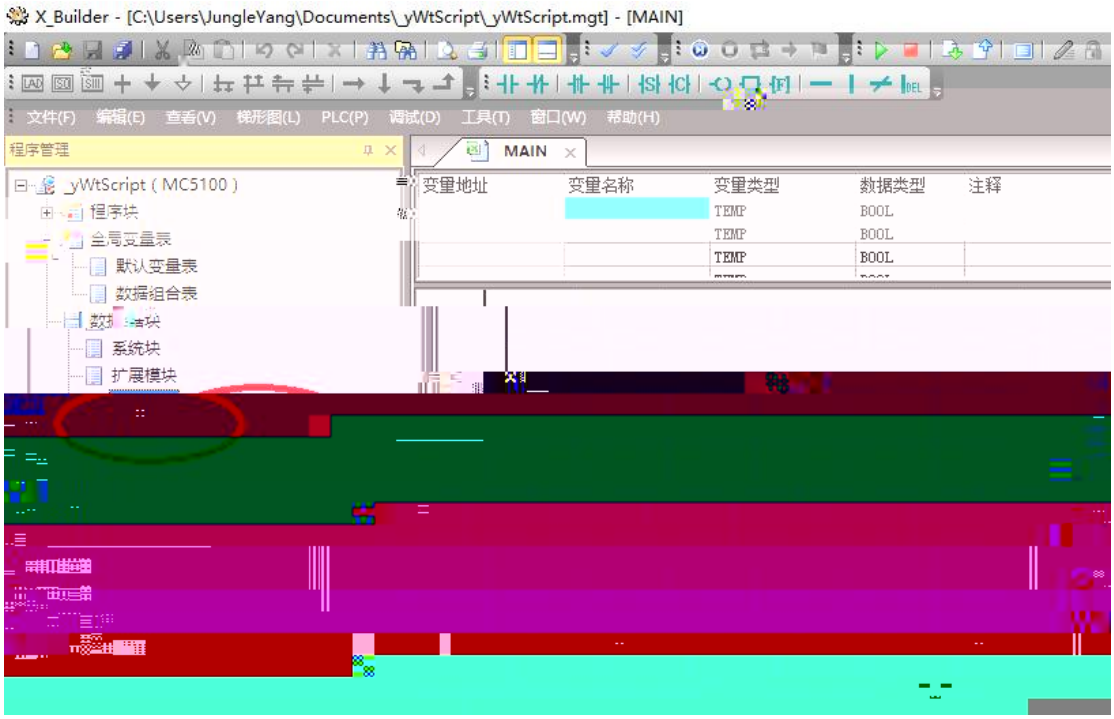
3

xBuild C

D1000 20000 D998( )
D996( )

3.1

“ ” , “ ” “ ”

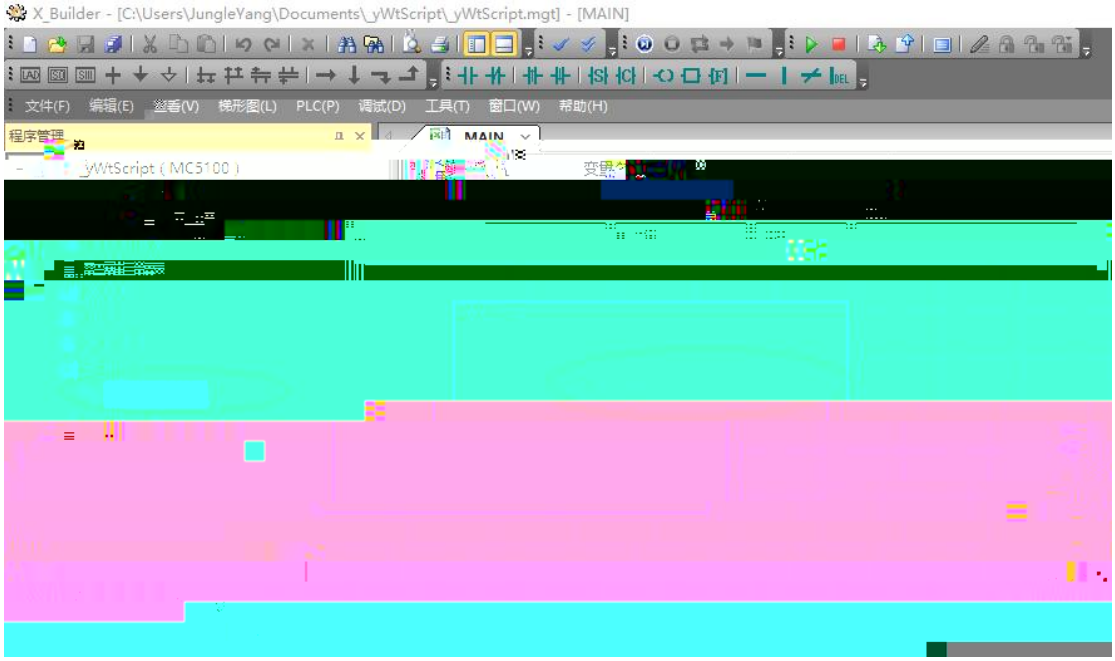


4

3.6 Н Я

3.3

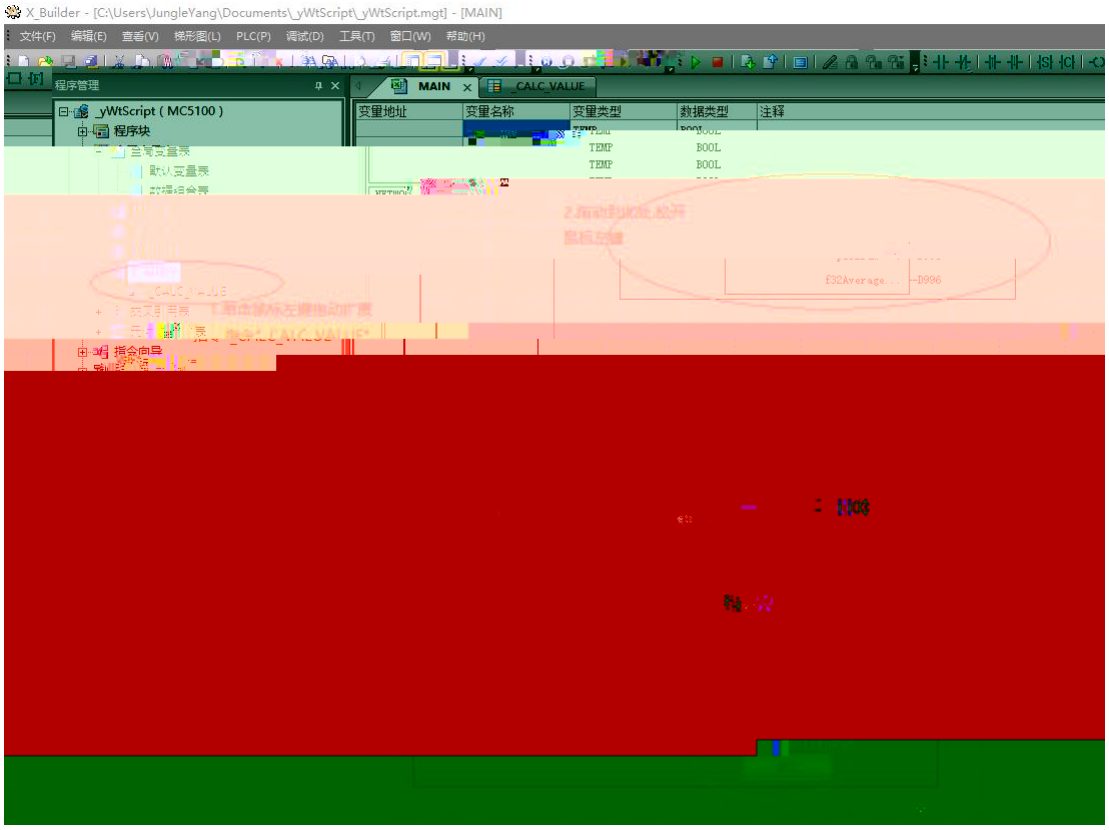
”_CALC_VALUE”



6

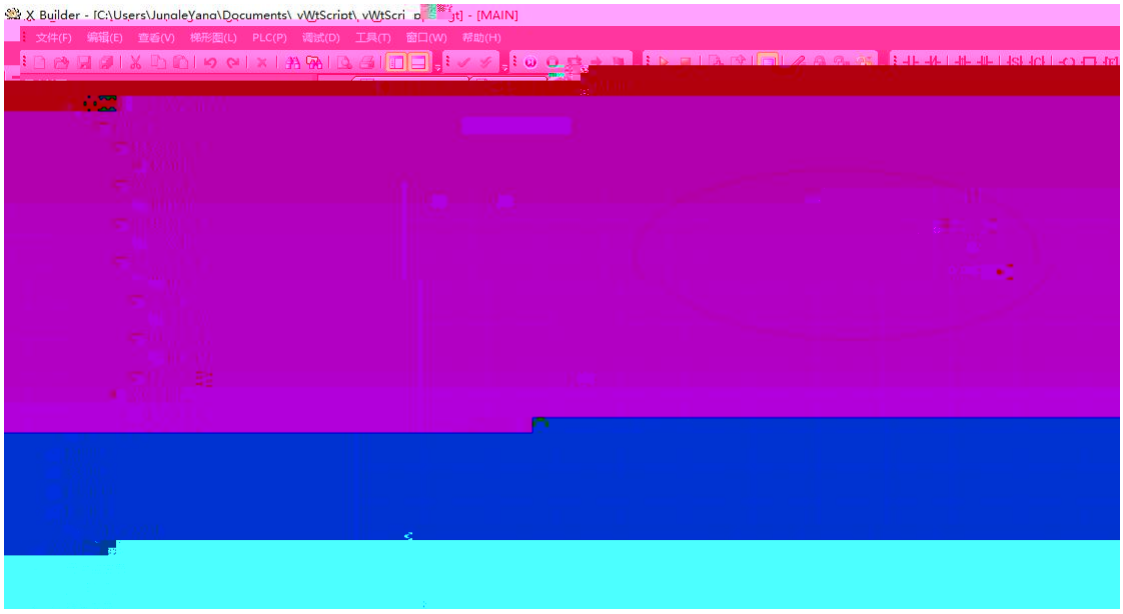
H

3.4



8

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



9

3.5

10

4

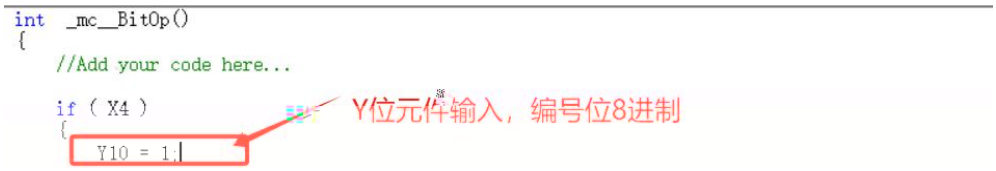
4.1

4.1.1 X

X

4.1.2 Y

	Y
	PLC
	Y0 ~ Y7777
	Bit
	8 , ’Y’

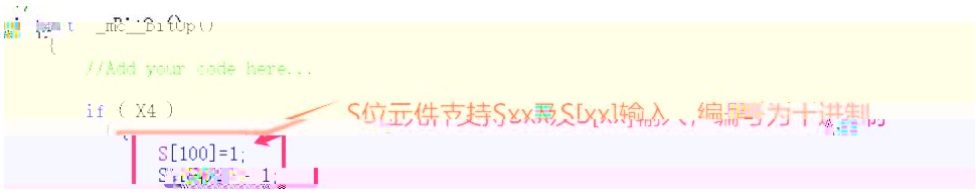


4.1.3 SM

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

4.1.4 S

	S
	PLC S
	S0 ~ S4095
	Bit
	10 , ’S’

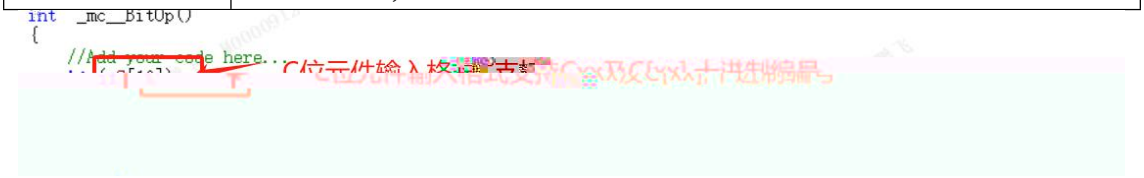


4.1.5 T

	T
	PLC T
	T0 ~ T4095
	Bit
	10 , ’T’

4.1.6 C

	C
	PLC C
	C0 ~ C4095
	Bit
	10 , ’C’



4.1.7 M

	C
	PLC M
	M0 ~ M65535
	Bit
	10 , ’M’

```
int _mc__BitOp()
{
    //Add your code here...
    if ( X4 )
    {
```

M位元件支持Mxx及M[xx]输入，上进制编号

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字元件支持Zxx及Z[xx]读写操作

4.1.10 D

	D
	PLC D
	D0 ~ D65535
	signed short
	10 , ’D’

```
int _mc__BitOp()
{
```

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 )
  {
```

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,

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);
```

将tmp值写入到长整数D150中

4.2.3

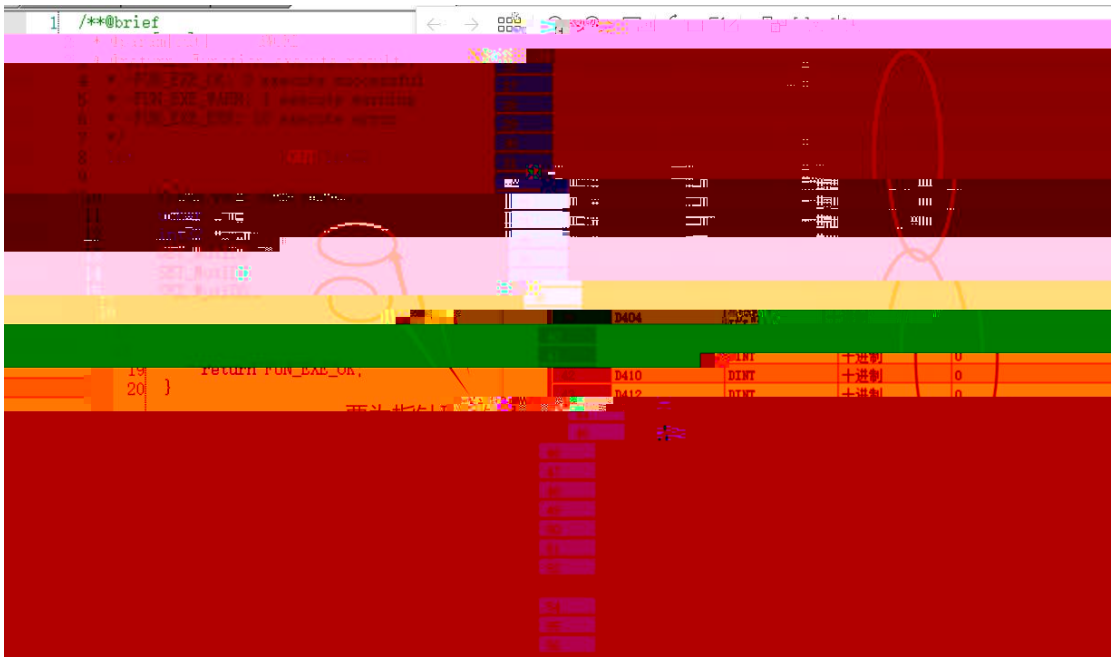
	int GET_MutiDD(int stNum, int len, int *ps32Dsc)
	“ D ”
	stNum: D
	Len :
	ps32Dsc:
	0 ,

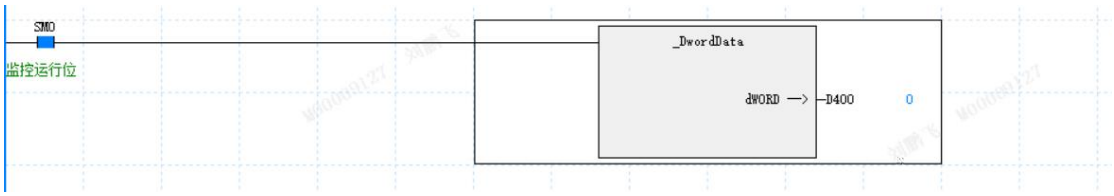
4.2.4

4.2.4

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

	ps32Dsc: D
	0 ,





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)
	“ 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_MutiFD(int stNum, int Len, float *pf32Dsc)
	“ D ”
	stNum: D
	Len :
	ps32Dsc:
	0 ,

4.2.8

4.2.8

```
int  _mc__DwordData(OUT int32 *dWORD)
{
    //Add your code here...
      uchar utmp;
      int32 *dtmp;
      float fTmp, *pfTmp;
        int i = 610;
        GET_MutiFD(i,5,pfTmp);
        SET_MutiFD(i+20,5,pfTmp);
```

读取从D610开始的5个浮点数，存入指针变量pfTmp中

将指针变量pfTmp中连续5个浮点数存入D630开始的地址

i 16

4.2.9

R D

D

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

4.2.10

void SET_DR(unsigned short stNum, int iVal) i

4.2.12

	int SET_MutiDR(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_MutiFR(int stNum, int len, float *pf32

	0 ,

4.2.16

	int SET_MutiFR(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,

	stNum: F
	Len :
	ps32Dsc:
	0 ,

4.2.34 F0~F9

4.2.20

	int SET_MutiDF(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

	GET_MutiFF(int stNum, int len, float *pf32Dsc)
	stNum: F
	Len :
	ps32Dsc:
	0 ,
4.2.34	F0~F9

4.2.24

	int SET_MutiFF(int stNum, int len, float *pf32Src)
	“F” 4
	stNum: F ”
	Len :
	ps32Dsc: F
	0 ,
4.2.34	F0~F9

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
```

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 " يم

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)
	“ 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

	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

[redacted]

4. [redacted]

```
int GE[redacted] 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)
```

“ Fx ”

[redacted]

```
#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;
```

写入目标位指针指向的地址

被写数据为立即数

Adr1 D600 ADr2 R550

4.2.36

	int GET_PS32(int *ps32Src)
	int
	ps32Src:

```
#define 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）
```

操作数为指向地址的指针

IOPra

4.2.37

	int GET_S32(int s32Src)
	int 32
	s32Src:

	PLC

*1

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

4.2.41

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

4.4.23

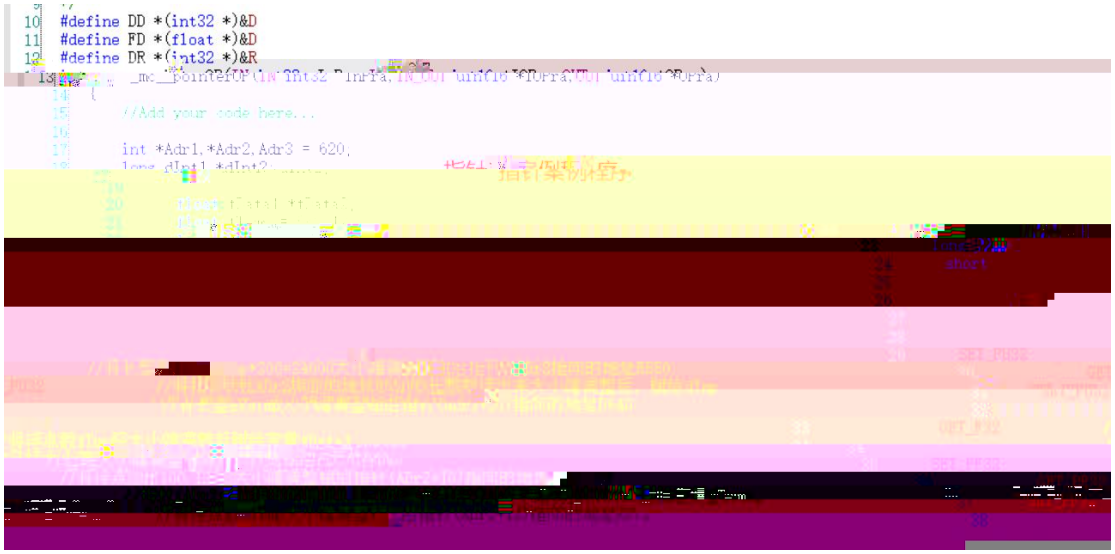
4.2.42

	float GET_PF32(float *pf32Src)
	int
	pf32Src:

4.4.23

4.2.43

	float GET_F32(float f32Src)
	int
	f32Src:



1 +1 PLC 2

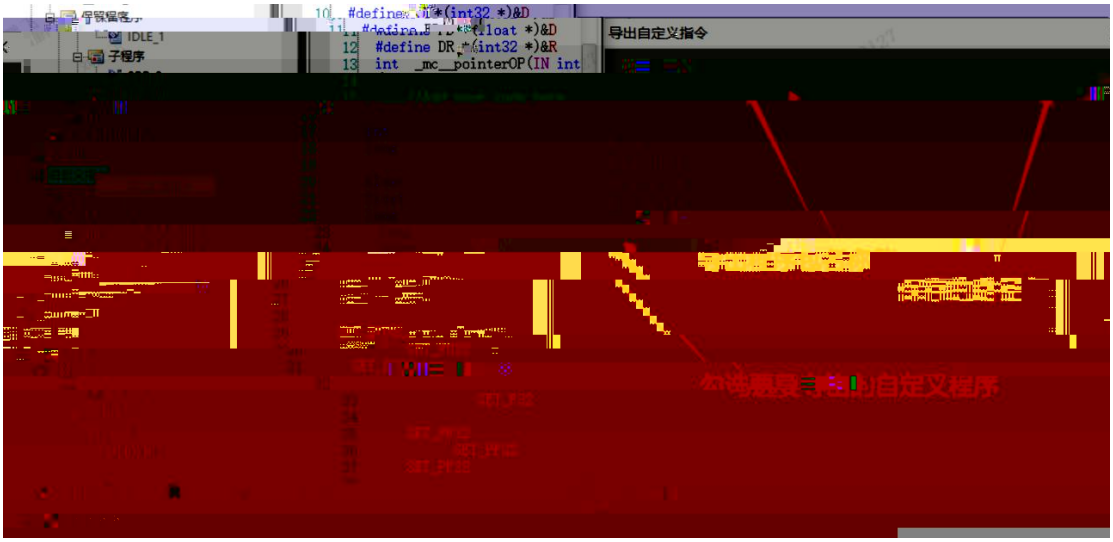
2 GET_F32 float Src

PLC

FD[40]=GET_F32(fTemp);

5

5.1.1


IDLE_1
子程序
导出自定义指令