

# EasyBuilder 500 Macro 使用说明

## 一:文档说明

作 者：程志刚

时 间：2003 年 1 月

版权所有：台湾威纶科技有限公司

概 要：此文档为宏指令功能模块的使用文档,说明宏语言的语法,宏语言的使用,宏指令源程序编写的操作方法与操作顺序,

使用文档包含以下几个部分:

[宏语言文本说明:](#)

[宏语言使用说明](#)

[宏指令程序与 PLC 的通信\(包括本地地址 LocalBit,LocalWord\):](#)

[宏指令操作说明](#)

[有关宏指令使用的若干说明](#)

[编译错误提示](#)

[程序示例源程序](#)

## 二:宏语言文本说明:

### 1: 常数与变量

#### a:常量

- (1) 十进制常量
- (2) 十六进常数
- (3) 单引号内的 ASCII 常量(字符常量)
- (4) 逻辑常量: True (非零),False(零)

#### b:变量

##### (a)变量命名

必须以字每开头,其它字符可以是字每或数字,不得超过 32 个字符的长度

##### (b) 数据类型

- (char) 字符类型
- (int) 整数类型 32-bit
- (short) 短整数类型 16-bit
- (float) 浮点数类型
- (bool) 逻辑类型

### 2: 运算符

#### (1) 赋值运算

赋值运算: =

#### (2) 算术运算:

- 加 : +
- 减 : -
- 乘 : \*

除 : /  
余 : mod ,

## (3) 关系运算:

小于 : <  
小于等于 : <=  
大于 : >  
大于等于 : >=  
等于 : ==  
不等于 : <>

## (4) 逻辑运算:

逻辑与 : And  
逻辑或 : Or  
逻辑异或 : Xor  
逻辑非 : Not

## (5) 位运算:

## (a) 移位运算

左移: <<  
右移: >>

## (b) 逻辑位运算

位与运算: &  
位或运算: |  
位异或运算: ^  
位取反运算: ~

## 3: 运算符的优先级:

表达式中执行多个操作时,每一部分都按规定的顺序计算,这种运算符间的计算次序,叫运算符的优先级

## (1) 同一种类运算符的优先顺序(按左到右,依次从高到低)

算术运算符 : ^ → ( \* , / ) → ( mod ) → ( + , - )

移位运算 : 其出现的优先顺序,按在表达式中出现的次序从左到右进行处理

关系运算符 : 其出现的优先顺序,按在表达式中出现的次序从左到右进行处理

逻辑运算符 : Not → And → Or → Xor,

## (2)

算术运算 优先于 位运算

位运算 优先于 关系运算

逻辑运算 优先于 赋值运算

## 4: 数组

## (1) 数组 :

只支持静态一维数组 ,形式为:

一维数组 : 数组名[数组的大小]

数组的下标为整数其可取值区间为 0 --- 4294967295

下标最小值 :零

下标最小值 :数组的大小 - 1  
 如 : Array[MAX] MAX = 100  
 下标最小值 :0  
 下标最小值 :99 ( 100 - 1)

## 5: 表达式

### (a) 运算对象

- (1) 常量
- (2) 变量
- (3) 数组元素
- (4) 函数调用

### (b) 表达式组成

表达式由运算对象和运算符按一定规则组合而成

## 6: 语句

### (1):说明语句

- (a) type name 说明 name 的类型
- (b) type name[constant] 说明 name 类型的一维数组

### (2): 赋值语句

形式为 : 变量 = 表达式

### (3) 逻辑判断与分支

#### (a) 单行形式

If Condition Then [ Statements ] [ Else elseStatements ]

End If

#### (b) 块形式

If Condition Then  
 [ Statements ]  
 [ Else [If Condition - n Then  
 [elseifstatements] ....  
 [ Else  
 [ elsestatements ] ]

]]

End If

语法说明 :

Condition	必要,为一条件表达式 ,条件表达式值为零,则视为假 FALSE,条件表达式值为非零,则视为真 TRUE
Statements	在块形式中是选参数,在单行形式中,且没有 Else 子句时,为必要参数 ,该语句在 Condition 为真是时执行
Condition-n	可选 ,解释同 Condition
elseifstatements	可选 ,一条或多条语句,在相对应的 Condition - n 为 TRUE 时执行
Elsestatements	可选 ,在上述 Condition 和 Condition—n 都不为 TRUE 时执行

## (C) 多重分支选择 Select Case 语句

```

Select Case TestExpression
  [Case expression -n
    [statements-n] ].....
[Case Else
  [elsestatements ] ]
End Select

```

语法说明:

TestExpression	必要 ,任何数值或字符串表达式
Expression—n	如果有 Case 出现 则为必要参数,为字符串或整形数值常量
Statements—n	可选 ,为一条或多条语句,当 TestExpression 和对应的 expression—n 相等时,则执行
Elsestatements	可选 , 为一条或多条语句,当 TestExpression 不和任一 expression—n 相等时,则执行

## (4)循环控制

## a) For -Next 语句:

常用于次数已确定的情况 ,当为 To 时,此时按步长递增,当为 Down 时,此时按步长递减

```

For counter = start To(Down) end [Step step ]
  [Statements ]
Next [counter ]

```

语法说明:

Counter	必要,循环计数器的数值变量,该变量只能是整形或字符形变量
Start	必要,Counter 的初值
End	必要, Counter 的终值
Step	可选 ,Counter 的步长,如果没有指定,则缺省为 1,只能是数值
Statements	可选 ,For 和 Next 之间的语句块,该语句块将执行指定的次数

## b) While – Wend 语句

由条件控制的循环,条件放置于循环顶部逻辑表达式值为 TRUE 或 FALSE,当指定的条件为 TRUE 时,执行循环体中的语句块

```

While condition
  [statements]

```

Wend

语法说明

Condition	必要,逻辑表达式,其计算结果为 TRUE 或 FALSE,condition 为 TRUE 时执行循环体
Statements	可选 ,为一语句块,当 condition 的值为 TRUE 时

## (5) break ;

用于循环和条件选择语句中,执行到此语句时,退出循环或条件语句

## (6) continue

用于循环,执行到此语句时,中断这一次循环的执行,到下一次循环执行

## (7) return

函数返回语句

## 7: 关键字集:

宏语言规定的,用作特定的用途,不能作为函数名,数组名,变量名等的记号

+ , - , \* , / , ^ , mod , >= , > , < , <= , <> , == , And , Or , Xor , Not , << , >> , = , & , | , ^ , ~

If , Then , Else , EndIf , Select , Case , For , To , Down Step , Next , while , wend

break , continue , return

## 三:宏语言使用说明

## 1: 局部变量及全局变量

(1:)局部变量: 在一个函数范围内有效,

(2:)全局变量: 从定义或声明的地方开始到宏指令程序结束有效

在一个函数内当有相同名称的全局变量与局部变量时,是局部变量有效

## 2: 变量赋初值及常量

## (1:)变量赋初值

(a)在声明语句中直接进行赋值值 e.g: int h = 9

(b) 在声明或定义后,用赋值语句进行变量的赋初值

## (2:)数组变量的赋初值

形式如: int g[10] = { 1,2,3, , 3 }

初值用花括号括起来, 各数组分量用分号隔开,从左到右赋按数组下标从零开始的顺序赋给各数组分量,

## (3:)常量 :

宏语言支持:

十进制整形常量

十六进制整形常量 以 0x 开头

字符形常量,

逻辑常量 : True 代表真, False 代表假 ,

## 3: 逻辑变量及表达式

## (1:)逻辑变量:

它有真(True),假(False)两种取值的情况 ,所有非零的值认为是真(True),零值认为是假(False)

(2:)表达式:

条件表达式的值为零值即为假,代表条件不成立,  
条件表达式的值为非零值即为真,代表条件成立

#### 4: 声明语句

(1:)声明语句在函数外声明或定义全局变量,

(2:)声明语句在函数内声明或定义局部变量,同时局部变量声明语句在函数内的最前面,其间不能有其它的语句,否则会给出编译错的提示

**For example :**

```
Macro_Command main( )
char i ,
i = 9 //在声明语句间的非声明语句,将给出编译错的提示
int g[10]
For g[2] = 0 To 2
    g[3] = 4
Next g[2]
End Macro_Command
```

#### 5: 函数调用及参数传递方式

(1:)函数调用:

被调用的函数须在调用函数的前面定义,否则将给出函数没有定义的编译错误提示

**For example :**

```
Macro_Command main( )
int i ,
i = Func( i ) //将给出函数Func没有定义的编译错误提示
End Macro_Command

Sub int Func( int i )
int h = 9
i = 9 * h
Return i
End Sub
```

(2:)参数传递方式:

- (a) 通过局部变量传值的方式传递参数
- (b) 通过函数间共享全局变量的方式传递参数

#### 6: 主函数

宏指令源程序有且只能有一个主函数,它是宏指令程序开始执行的入口,其形式为:

Macro\_Command 函数名()

End Macro\_Command

#### 四: 宏指令程序与PLC的通信(包括本地地址LocalBit,LocalWord):

功能: 使用库函数实现与 Plc 的沟通

在宏指令程序的计算与控制中当 Plc 地址类型的变量参与时,就通过库函数 GetData( ... )从 EasyView 连接的 Plc 中取数据,通过库函数 SetData( ... )把结果数据送到 EasyView 连接的 Plc 中, 这样就把处理 plc 信息的繁杂工作交给了 GetData( ... )与 SetData( ... )等库函数

1: GetData(各种支持的数据类型 DestData, CString strAddr\_Type, int iAddr\_Off, int iDataCount )

说明:实现从 Plc 中取数据,通过对话框填表的形式输入

参数说明:

DestData	得数据的首址
CString strAddr_Type	Plc 地址类型和编码方式
int iAddr_Off	Plc 地址偏移
int iDataCount	数据项数

返回值: 没有

2: SetData(各种支持的数据类型 DestData ,CString strAddr\_Type , int iAddr\_Off , int iDataCount )

参数说明:

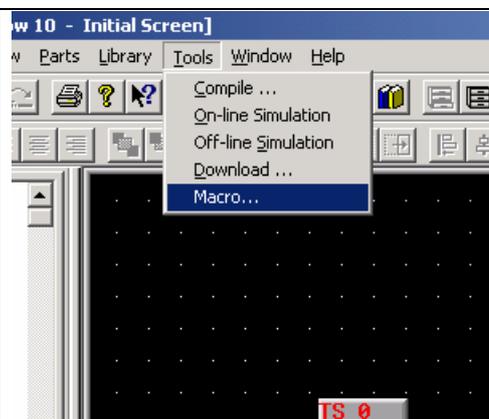
DestData	送出去数据的首址
CString strAddr_Type	Plc 地址类型和编码方式
int iAddr_Off	Plc 地址偏移
int iDataCount	数据项数

返回值: 没有

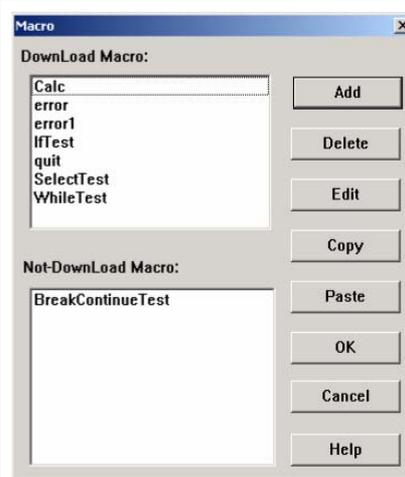
## 五:宏指令操作说明

1: 宏指令程序的编写分三步进行,

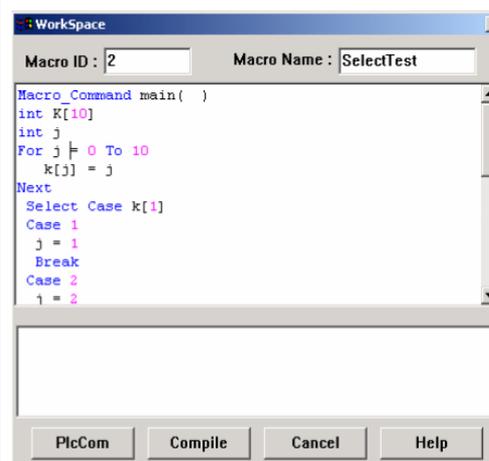
第一步:在 EB500 主界面中启动宏指令功能模块的第一个对话框(MacroControlDlg)



第二步:在 MacroControlDlg 对话框中进行宏指令间的拷贝,删除,编辑,调用 MacroWorkspaceDlg 对话框进行宏指令源程序的编辑,



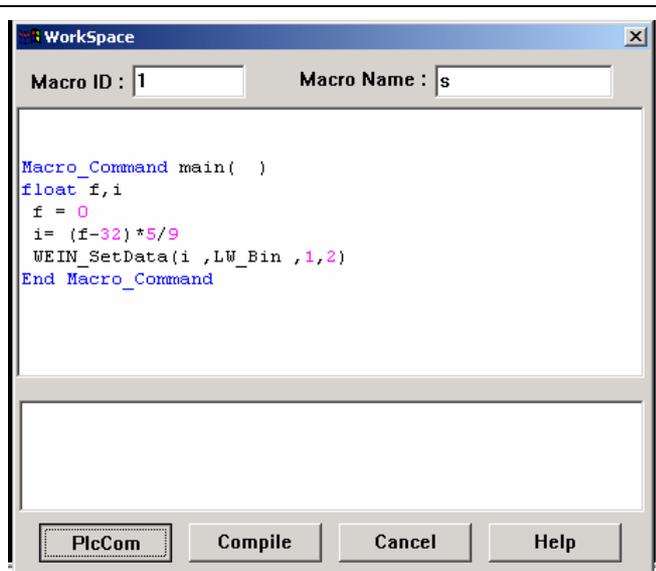
第三步:宏指令源程序的编辑,确定一条宏指令的名称,编号,编译,及排除编译错误



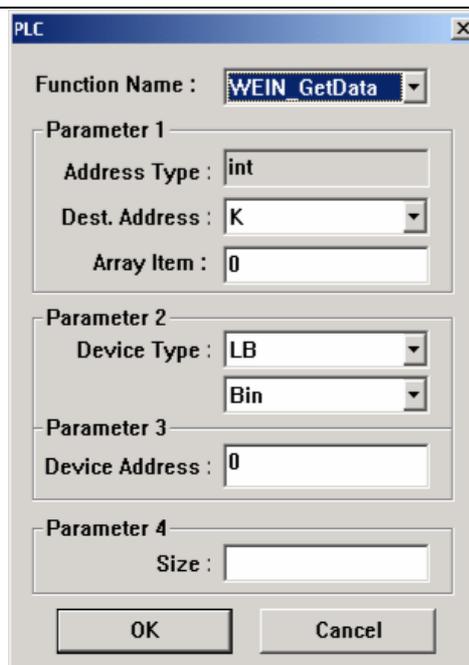
## 2: 宏指令源程序通信函数的编辑方法

(1):输入:

(第一步:) 在宏指令源程序要输入库函数的地方,输入关键字: "Insert"  
{ 或者把光标置于要输入库函数的地方,后呼叫 PlcCom 按钮,}



(第二步:) 在库函数编辑对话框,进行库函数的选择和库函数参数的设置,按 Ok 按钮完成此次库函数的输入,按 Cancel 按钮取消此次输入操作

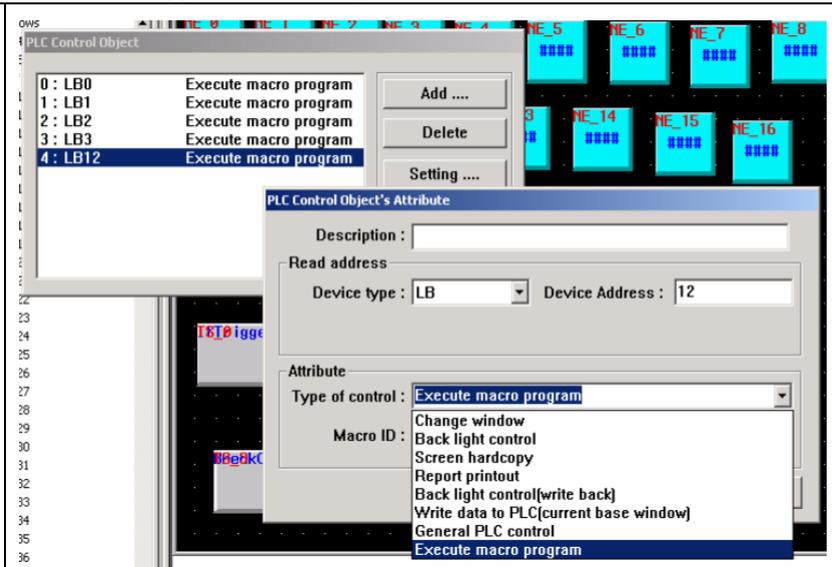


(2):修改:将光标置于要修改的库函数上对其进行修改,按(1:)的第一步,第二步操作即可

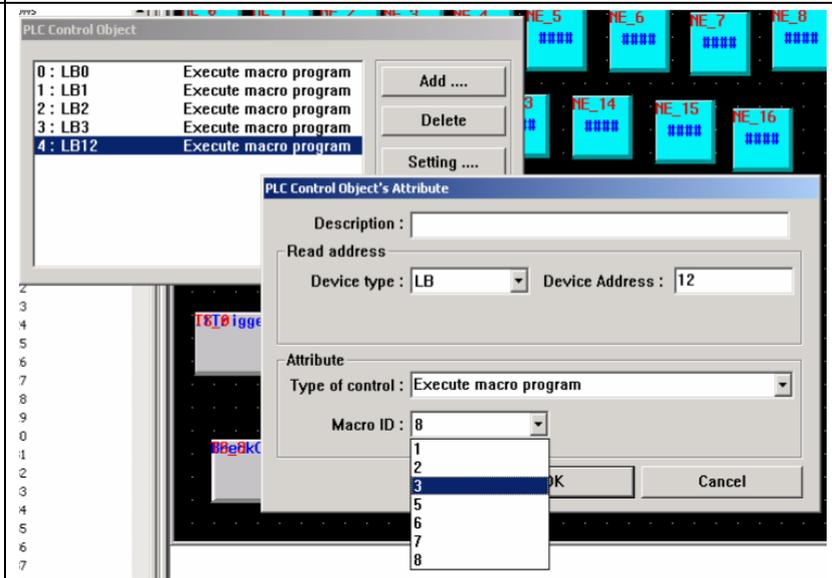
(3):删除:把要删除的库函数选定,在键盘上按 Delete 键即可将其删除

## 3:宏指令触发条件的确定.见图”Trigger(Macro).bmp”

( 第 一 步 : ) 在 PlcControl 物件属性对话框中选择控制类型为 Execute Macro Program,



( 第 二 步 : ) 在 PlcControl 物件属性对话框中选择一个 MacroID 并确定一个触发位



## 六:有关宏指令使用的若干说明

### (1):宏指令程序存储空间的限制

一个eob档中所有宏指令存储空间受公共窗口0空间 大小的限制,同时一个宏指令程序中的局部变量存储空间最大为4k,

### (2):最多可能执行的宏指令条数的限制

一个eob档中最多可以使用256条宏指令程序.

### (3):宏指令可能出现的死机情况

宏指令程序中有死循环,且死循环中没有通信时.

宏指令程序中数组数据使用,存储空间越界时.

### (4):宏指令程序通信时的速度.

宏指令程序在有通信时,它的执行可能会有点慢.这是由于通信占用太多的时间所造成的.

## 七 :编译错误提示

### 1:错误提示形式:

宏指令名称( 编译错误编号 ) 错误提示

当编译没有通过时,可以通过编译错误编号, 查找对错误的说明

### 2:错误描述:

#### (1: ) "语法错:" "'identifier'

很多种情况会给出这条编译错误的提示,

#### **For example :**

```
Macro_Command main( )
```

```
char i ,程 //这是一个不支持的记号 ,”错误提示为:“语法错: 程”
```

```
int g[i] //定义错
```

```
For g[2] = 0 To 2
```

```
    g[3] = 4
```

```
Next g[2]
```

```
End Macro_Command
```

出现这条错误提示时,一般都是多了,或少了一个符号,很容易改正错误

#### (2:)'identifier' : " 数组没有被指定大小";

只支持静态数组,定义数组变量时,必须指定数组的大小

#### **For example :**

```
Macro_Command main( )
```

```
char i
```

```
int g[i] //定义错
```

```
For g[2] = 0 To 2
```

```
    g[3] = 4
```

```
Next g[2]
```

```
End Macro_Command
```

- (3:) strError = "重定义错: " **'identifier'**  
函数,变量名字在其作用域内必须是唯一的,

**For example :**

```
Macro_Command main( )
int g[10] , g
For g[2] = 0 To 2
    g[3] = 4
Next g[2]
End Macro_Command
```

- (4:) "函数名错:" **'identifier'**  
关键字,常量等不能用作函数名

**For example :**

```
Macro_Command If( )
int g[10] , g
For g[2] = 0 To 2
    g[3] = 4
Next g[2]
End Macro_Command
```

- (5:) "括号不配对";  
左括号,右括号不成对出现

**For example :**

```
Macro_Command main ) //少左括号
int g[10] , g
For g[2] = 0 To 2
    g[3] = 4
Next g[2]
End Macro_Command
```

- (6:) "If 后没有表达式";  
(7:) "If 语句没有 Then";  
(8:) "没有 EndIf 语句";  
(9:) End If 前面没有出现配对的 If";  
(10:) "非法的 Else 语句";

If 语法结构为:

    If 条件表达式 Then

    [ Else [If 条件表达式 Then ] ]

EndIf

不服合这种表达形式的 If 选择语句都是不合法的,将给出对应的编译错

- (11:) "Case 后面应为常量" ;  
(12:) "Select 后少 Case 关键字" ;  
(13:) "Select Case 后少表达式" ;  
(14:) "没有 End Select 语句"  
(15:) "非法的 Case 语句" ;  
(16:) "End Select 前面没有出现配对的 Select" ;

Select Case 语法结构为:

Select Case 表达式  
Case 数值常量

Case 数值常量

Case 数值常量

Case Else

End Select

不服合这种表达形式的 Select Case 选择语句都是不合法的,将给出对应的编译错

- (17:) "For 循环不配对,Next 前应有 For 关键字" ;  
(18:) "此处应为整型或字符型变量" ;  
(19:) "此处应为赋值号" ;  
(20:) "此处应为关键字 To" ;  
(21:) "没有 Next 语句" ;

For 循环语法结构为:

For 变量 = 初值 To 终值 [Step 步长]

Next [变量]

不服合这种表达形式的 For 循环语句都是不合法的,将给出对应的编译错

- (22:) "While 循环不配对,Wend 前应有 While 关键字" ;  
(23:) "没有 Wend 语句" ;

While 循环语法结构为:

While 条件表达式

Wend

不服合这种表达形式的 While 循环语句都是不合法的,将给出对应的编译错

- (24:) "break 语句非法" ;

Break 语句只能在 For 循环,While 循环 ,Select Case 选择结构中使用  
且 Break 单独成一语句行

- (25:) "Continue 语句非法" ;

Continue 语句只能在 For 循环, While 循环中使用  
且 Continue 单独成一语句行

(26:) "表达式不正确";

(27:) 运算对象非法";

在表达式中出现与运算符不匹配的运算对象时,将给出此编译错

**For example :**

```
Macro_Command main( )
int g[10] , g
For g[2] = 0 To 2
    g[3] = 4 + 程
Next g[2]
End Macro_Command
```

(28:) "此处应为 Sub";

(29:) "此处应为 Macro\_Command ";

函数的定义形式为:

Sub(Macro\_Command) 数据类型 函数名( ... )

End Sub(Macro\_Command)

不服合这种形式的函数定义,将给出对应的编译错误

(30:) "参数个数不对";

(31:) "实参数据类型不匹配" ;

(32:) "实参不对";

函数调用时,实参与形参必须在数据类型,参数个数上一一对应才能编译通过  
否则将给出对应的编译错

(33:) "没有定义的函数:";

(34:) 数组分量表达形式非法

(35:) 数组定义非法";

(36) 数组下标非法

的表示,将给出对应的编译错

(37:) "没有定义,或声名的变量";

只能使用已经定义了了的或声名的了的变量和函数,否则将给出对应的编译错

**(38:) "不支持的 plc 地址类型" ;**

库函数 GetData( ... ), SetData( ... ) 的参数中有 plc 地址类型信息,当 plc 地址类型不是此种 plc 支持的地址类型时,将给出对应的编译错

(39:) 应为整型,字符形变量或常量";

数组的表达形式为:

定义时: 数组名[常量] (常量描述数组的大小)

使用时: 数组名[整形,字符形变量或常量]

不服合这种表达形式

(40): "变量定义或声名语句的前面不能有执行语句"

**For example :**

```
Macro_Command main( )
int g[10] , g
For g[2] = 0 To 2
  g[3] = 4 + g[9]
int h , k
Next g[2]
End Macro_Command
```

声名语句在此处是错误的,在一个函数内它的前面不能有非定义或声名语句

(41): "移位运算中,操作数不能为浮点数"

(42): "函数应返回一个值" ;

(43): "函数不应返回一个值"

(44): "运算中不能有 Float 型数据" ;

(45): "plc 地址错" ;

(46): "堆栈不能超过 4k" ;

(47): "宏指令程序入口只能有一个" ;

(48): "宏指令入口函数不唯一: " 'identifier' "

宏指令的入口函数只能有一个,形式为:

**Macro\_Command 函数名()**

**End Macro\_Command**

## 八 :程序示例源程序

1: For 循环,各种表达式(算术,移位,逻辑,关系表达式)

```
Macro_Command main( )
int a[10]
int b[10]
int i

b[0]= (400 + 400 << 2) / 401
b[1]= 22 *2 - 30 % 7
b[2]= 111 >> 2
b[3]= 403 > 9 + 3 >= 9 + 3 < 4 + 3 <= 8 + 8 == 8
b[4]= Not 8 + 1 And 2 + 1 Or 0 + 1 Xor 2
b[5]= 405 And 3 And Not 0
b[6]= 8 & 4 + 4 & 4 + 8 | 4 + 8 ^ 4
b[7]= 6 - ~ 4
b[8]= 0x11
b[9]= 409
For i = 0 To 4 Step 1
  If( a[0] == 400 ) Then
    GetData(a[0] ,3x_Bin ,0,9)
    SetData(b[0] ,3x_Bin ,11,10)
  End If
Next
End Macro_Command
```

## 2: while ,if , break , continue

```

Macro_Command main( )
  int a[10]
  int b[10]
  int i

  b[0]= (400 + 400 << 2) - 401 + 99
  b[1]= 100
  b[2]= 111
  b[3]= 403
  b[4]= 404
  b[5]= 405
  b[6]= 406
  b[7]= 407
  b[8]= 400 % 20 + 21
  b[9]= 409
  i = 5
  While i == 5 - 20 % 3
    SetData(b[1] ,3x_Bin ,11,1)
    If b[1] == 100 Then
      Break
    SetData(b[0] ,3x_Bin ,12,1)
    i = 4
    Continue
    SetData(b[0] ,3x_Bin ,13,1)
  End If
Wend

End Macro_Command

```

## 3: 全局变量,函数调用

```

char g
Sub int sin(int j ,int k)
  int y
  SetData(j ,3x_Bin ,14,1)
  GetData(y ,3x_Bin ,15,1)
  g = y
  Return y
End Sub
Macro_Command main( )
  int a[10]
  int b[10]
  int i

  b[0]= (400 + 400 << 2) - 401 + 99
  b[1]= 100
  b[2]= 111
  b[3]= 403
  b[4]= 404
  b[5]= 405
  b[6]= 406
  b[7]= 407
  b[8]= 400 % 20 + 21
  b[9]= 409
  i = 5
  While i == 5 - 20 % 3
    SetData(b[1] ,3x_Bin ,11,1)

```

```

    If b[1] == 100 Then
        SetData(b[0] , 3x_Bin , 12, 1)
        i = sin(b[2], b[3] )
        SetData(g , 3x_Bin , 17, 1)
        SetData(i , 3x_Bin , 16, 1)
    End If
Wend

End Macro_Command

```

#### 4: 控制自己

```

char i
int j
Sub ClientProgram(int flag)
End Sub
Macro_Command main( )
int i[10]
int j[2]
int w
char x

For w = 0 To 9
    i[w] = 0
Next

GetData(i[0] , 3x_Bin , 1, 4)
w = i[0] + i[1] + i[2] + i[3]
If w > 150 Then
    For w = 0 To 3
        i[w] = 0
    Next
    SetData(i[0] , 3x_Bin , 1, 4)

    x = 0
    SetData(x , 1x_Bin , 3, 1)
    GetData(i[0] , 3x_Bin , 212, 2)
    x = 0
    SetData(x , 0x_Bin , 4, 1)
Else
    For w = 0 To 3
        i[w] = i[w] + 2*w
    Next

    SetData(i[0] , 3x_Bin , 1, 4)

End If
End Macro_Command

```

#### 5: If 结构

```

Macro_Command main( )
int K[10]
int j
For j = 0 To 10
    k[j] = j
Next

```

```

If k[0] == 0 Then
  SetData(k[4], 3x_Bin , 0, 1)
End If

If k[0] == 0 Then
  SetData(k[4], 3x_Bin , 0, 1)
Else
  SetData(k[5], 3x_Bin , 0, 1)
End If

If k[1] == 0 Then
  SetData(k[3], 3x_Bin , 1, 1)
Else If k[1] == 1 Then
  SetData(k[1], 3x_Bin , 2, 1)
End If

If k[1] == 0 Then
  SetData(20, 3x_Bin , 3, 1)
Else If k[1] == 2 Then
  SetData(k[1], 3x_Bin , 4, 1)
Else
  SetData(k[2], 3x_Bin , 5, 1)
End If
End Macro_Command

```

## 6: Select 结构

```

Macro_Command main( )
int K[10]
int j
For j = 0 To 10
  k[j] = j
Next
Select Case k[1]
Case 1
  j = 1
  Break
Case 2
  j = 2
  Break
End Select

SetData(j, 3x_Bin , 0, 1)
End Macro_Command

```

```

Macro_Command main( )
int K[10]
int j
j = 90 * 1
k[4] = 8
Select Case k[4]
Case 1
  j = 1
  Break
Case 3
  j = 2

```

```

End Select

SetData(j,3x_Bin ,0,1)
End Macro_Command

```

## 7: while 结构

```

Macro_Command main( )
char Toggleflag
int a[13]
int b[14]
int c = 4848
char i = 0

b[0] = 13
While b[0]
a[i] = 20 + i * 10
If a[i] == 120 Then
c =200
Break
End If
i = i + 1
Wend
SetData(c ,3x_Bin ,2,1)

End Macro_Command

```

```

Macro_Command main( )
int a[250]
int b[250]
int i = 1
a[0] = 1
b[0] = 2
While a[0]
While b[0]
b[i ] = 10 + i * 10
If b[i] == 210 Then
SetData(i ,3x_Bin ,0,2)
b[i] = 200
Break
End If
i = i + 1
Wend
If b[i] == 200 Then
SetData(b[0] ,3x_Bin ,0,4)
Break
End If
SetData(b[0] ,3x_Bin ,0,3)

Wend
End Macro_Command

```

```

Macro_Command main( )
int K[10]
int j ,i
j = 0

```

```

i = 0
While k[i]
  k[j] = j + 1
  j = j + 1
  If j == 5 Then
    Break
  End If
SetData(k[0], 3x_Bin , 0, 5)
End Macro_Command

```

## 8: Break ,Continue 结构

(1)

```

Macro_Command main( )
int K[10]
int j
For j = 0 To 10
  If j % 2 == 0 Then
    k[j] = 5
  Else
    Break
  End If
Next
SetData(k[0], 3x_Bin , 0, 10)
End Macro_Command

```

(2)

```

Macro_Command main( )
char Toggleflag
int a[13]
int b[14]
int c = 4848
char i = 0

b[0] = 13
While b[0]
  a[i] = 20 + i * 10
  If a[i] == 120 Then
    c = 200
    i = i + 1
    Continue
  End If
  i = i + 1
  If c == 200 Then
    SetData(c , 3x_Bin , 2, 1)
    Break
  End If

Wend

End Macro_Command

```

## 9: 数组结构

```

Macro_Command main( )
int a[25]
int b[25]

```

```

int i
b[0] = 13
For i = 0 To b[0] Step 1
  a[i] = 20 + i * 10
Next
SetData(a[0] , 3x_Bin , 0, 13)
End Macro_Command

```

## 10: 其它

```

int x = 0
int y
float f
Sub char func1()
  char Toggleflag
  int a[14]
  int b[14]
  int c = 4848
  char i
  b[0] = 13
  Toggleflag = ( b[0] Xor 0x89 + (100 /2) << 2 - 123 >> 2 )
  If (b[0] Xor 0x89 + (100 /2) << 2 - 123 >> 2) == 0 Then
    SetData(Toggleflag , 3x_Bin , 0, 1)
  End If
  Toggleflag = 1

  b[0] = 13
  For i = 0 To b[0] Step 1
    a[i] = 20 + i * 10
  Next
  If i < b[0] - 1 Then
    GetData(a[0] , 0x_Bin , 0, 1)
    If a[0] <> 1 Then
      End If
    Else
      SetData(c , 3x_Bin , 2, 1)
    End If

  While 1
    If Toggleflag == 1 Then
      Break
    End If
  Wend
  Return Toggleflag
End Sub

Macro_Command main( )
int a[250]
int b[250]
int i

  GetData(a[0] , 3x_Bin , 2, 2)
  a[0] = a[0] + 5
  a[1] = a[1] + 5
  If a[0] >= 150 Then
    a[0] = 5
  End If
  If b[0] >= 100 Then

```

```
    b[0] = 5
End If

func1() == 1

End Macro_Command
```

## 11: 死循环

(1)

```
Macro_Command main( )
int a[250]
int b[250]
int i = 1
a[0] = 1
b[0] = 2
While a[0]
  While b[0]
    b[i ] = 10 + i * 10
    If b[i] == 210 Then
      SetData(i ,3x_Bin ,0,2)
      b[i] = 10
      Break
    End If
    i = i + 1
  Wend
  If b[i] == 200 Then
    SetData(b[0] ,3x_Bin ,0,4)
    Break
  End If
  SetData(b[0] ,3x_Bin ,0,3)

Wend

End Macro_Command
```

(2)

```
Macro_Command main( )
char Toggleflag
int a[13]
int b[14]
int c = 4848
char i = 0

b[0] = 13
While b[0]
  a[i] = 20 + i * 10
  If a[i] == 120 Then
    c =200
    Continue
  End If
  i = i + 1
  If c == 200 Then
    SetData(c ,3x_Bin ,2,1)
    Break
  End If
```

```
Wend

End Macro_Command
```

## 12: 死机

```
int x = 0
int y
float f

Sub func0()

  If y == 1 Then
    y = 2
  Else If y == 3 Then
    Return
  End If
End Sub

Sub char func1()
  char Toggleflag
  int a[10]
  int b[10]
  int c = 4848
  char i

  b[0] = 13
  For i = 0 To b[0] Step 1
    a[i] = 20 + i * 10
  Next
  If i < b[0] - 1 Then
    GetData(a[0] , 0x_Bin , 0,1)
    If a[0] <> 1 Then
      End If
    Else
      SetData(c , 3x_Bin , 2,1)
    End If

  Toggleflag = 0
  If (b[0] Xor 0x89 + (100 /2) << 2 - 123 >> 2) > 0 Then
    GetData(Toggleflag , 0x_Bin , 0,1)
  End If

  While 1
    If Toggleflag == 1 Then
      Break
    End If
  Wend
  Return Toggleflag
End Sub

Macro_Command main( )
  int a[250]
  int b[250]
  int i
```

```
GetData(a[0] ,3x_Bin ,2,2)
a[0] = a[0] + 5
a[1] = a[1] + 5
If a[0] >= 150 Then
  a[0] = 5
End If
If b[0] >= 100 Then
  b[0] = 5
End If

If func1() == 1 Then
  SetData(a[0] ,3x_Bin ,2,2)
End If
End Macro_Command
```