

Table of Contents

Preface.....	1
Macro Editor	2
Interface Description.....	2
a. Script Clearing	4
b. Script Reading	4
c. Script Saving	5
d. Cut	5
e. Copy	5
f. Paste	6
g. Search	6
h. Rung Number	7
i. Syntax Analysis	7
j. Auto Syntax Analysis	8
k. Help	8
Function Window.....	9
a. Toolbox	10
b. Search.....	11
Editing window.....	14
a. ID: 0000	15
b. Switch macro.....	16
c. Comment.....	17
d. Channel	18
e. Sub-funtion.....	19
f. Cycle interval	20
g. Data Mode.....	21
h. Triggering Setup	22
i. Enlarge Script Area	24
j. Device tag.....	25
k. Input Device	26
Error Window.....	29
Introducing the Basic Commands	30
a. Operation Description	30
b. Device Usage	31
c. Control statement	35
d.Function	36
e. Arithmetic Function	38
f. ASCII Communication.....	41

Appendix A General Practice.....	A-1
Interlock Keys.....	A-1
Criteria Selection for Export.....	A-4
Call Sub-function.....	A-7



EU series TM

Preface

Macros are a simple programming language that enables the user to write their own programs for specific functions. The main purposes of macro are for automation or for triggering other functions according to certain criteria. For those who are familiar with programming language can effectively take advantage of human and machine interface register and save time on screen editing.



Macro Editor

Interface Description

Click [Create](#) to open the macro editor from the Macro Library. The editing screen consists of a toolbar, a functional window, an editing window, and an error window. See the figure below.

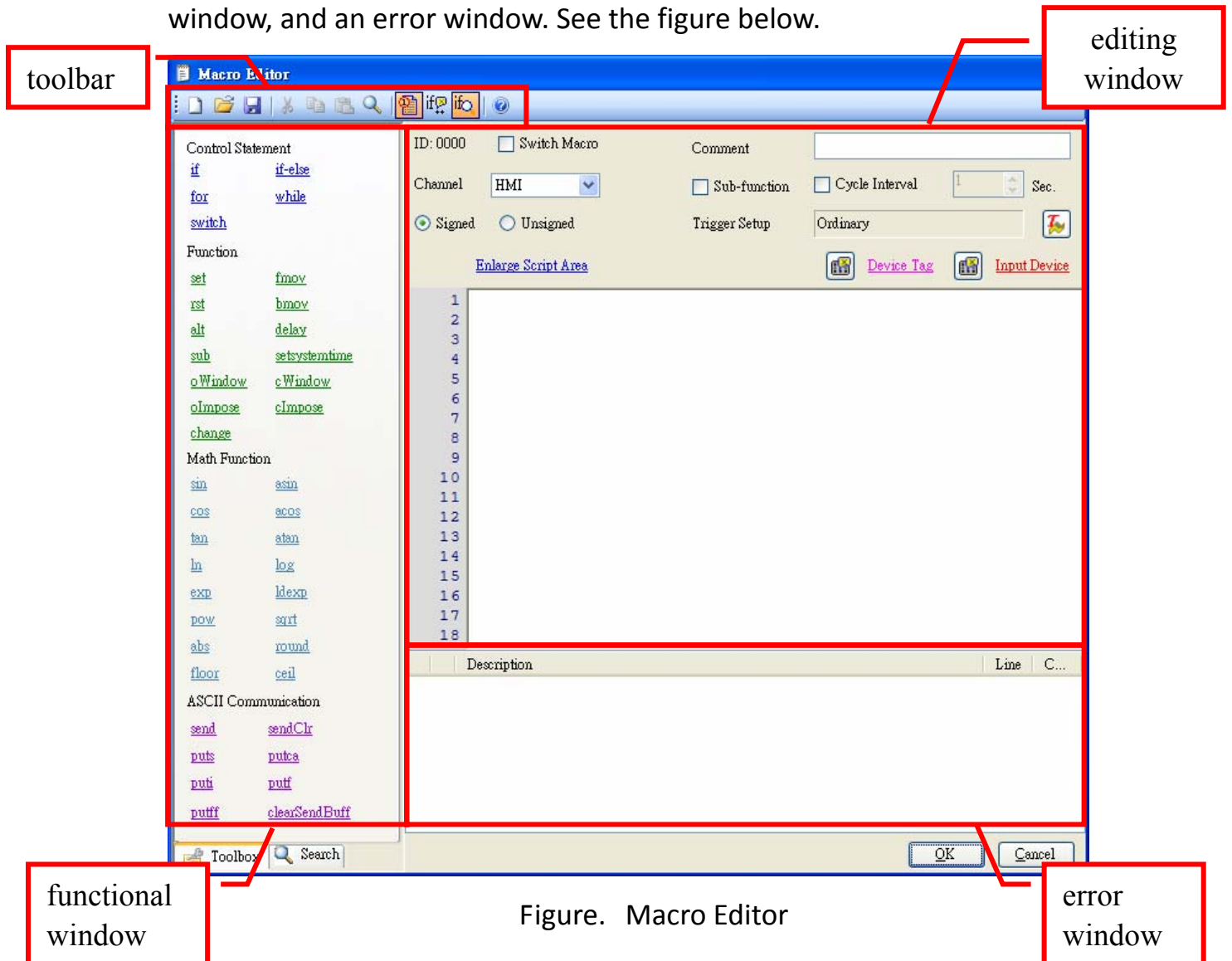













Figure. Macro Editor

Toolbar

All the functions provided by the toolbar are listed in the following table.


Figure. Toolbar Function Description




Name	Function
	Clearing out macro data from the editing window
	Reading the script from the file
	Saving macros into the file
	Cutting the selected macro
	Copying the selected macro
	Pasting the cut or copied macro
	Searching data inside the editing window
	Opening /closing row number display
	Run syntax analysis
	Opening / closing the automatic syntax analysis
	Opening the online PDF Help file

EU series™

a. Script Clearing

Click  from the toolbar to delete all the edited data from the screen to complete the initialization of the macro editor.

b. Script Reading

Click  from the toolbar to open the to be read data file (*.txt).

See the figure below.

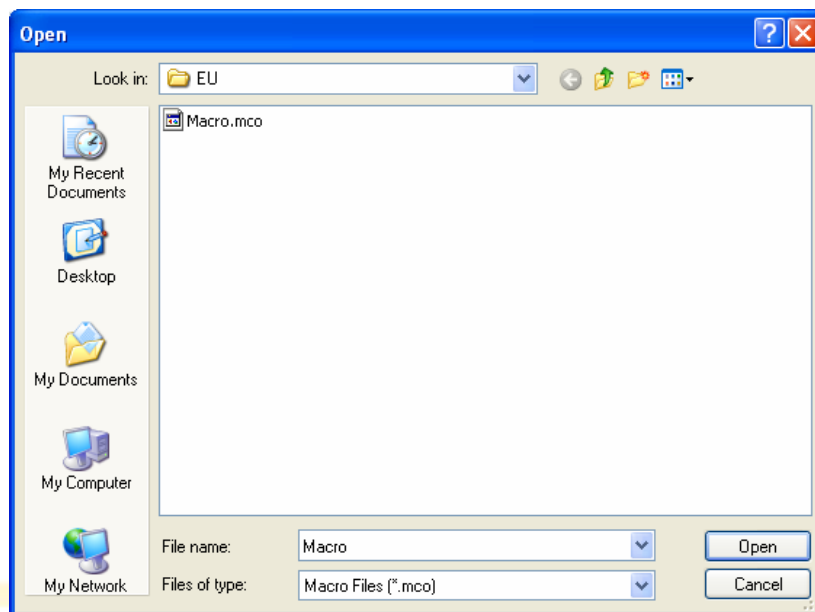



Figure. Script Reading

c. Script Saving

Click  from the toolbar to save the macro data from the editing screen into a file (*.txt). See the figure below.

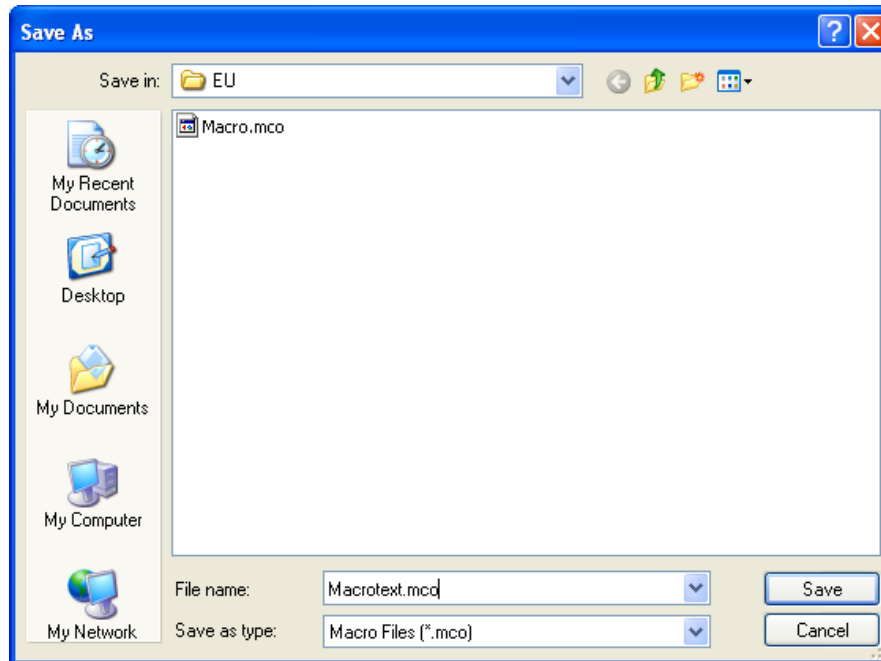


Figure. Script Saving


d. Cut

Select the cutting range, and either click from the toolbar or press the hotkey Ctrl+X to cut the selected macros.



e. Copy

Select the copying range and either click from the toolbar or press the hotkey Ctrl+C to cut the selected macros.

f. Paste

To paste a copied or cut macro, choose  from the toolbar, or press Ctrl+V.

g. Search

To search macro data in the editing screen, click  from the toolbar to change from the toolbox to the search tab or directly click  Search to carry out tab switching. See the figure below.

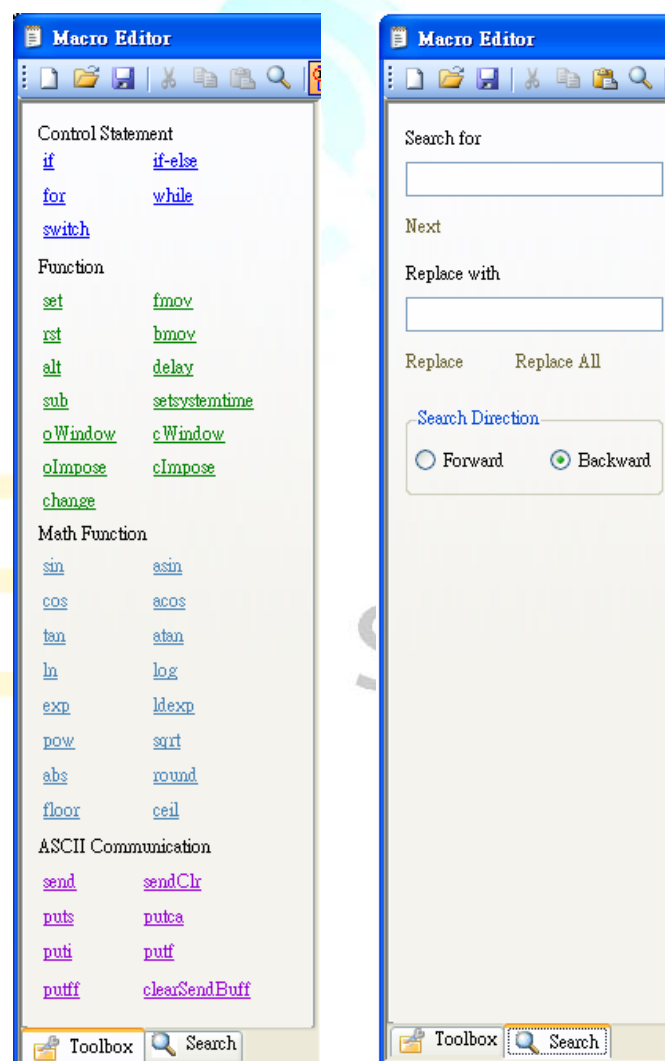



Figure. Functional Window: (a) Toolbox; (b) Search

h. Rung Number

The rung number displayed at the side of the editing window enables the user to juxtapose it with the macro rung number. Click  from the toolbar to open/close the rung number display. See the figure below.

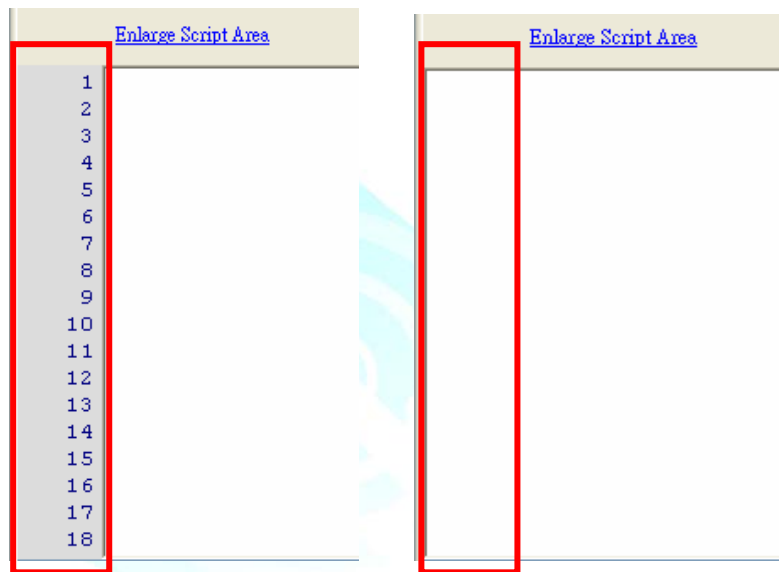



Figure. Rung Number Display: (a) Displaying Rung Number; (b) Hiding Rung Number

i. Syntax Analysis

To check if the macro syntax in the editing screen is correct or not, click  from the toolbar for the analysis function. The result will be displayed at the Error window. See the figure below.


	Description	Line	C...
✖ 1	Parameter error,the first parameter must be 16-bit non-negative integer, the second one i...	2	3
✖ 2	The statement of assigning initial value to device is wrong! (Word device=[integer])	3	6
✖ 3	Logical expression error.	3	7
✖ 4	The counting expression error.	3	9

Figure. Syntax Analysis error window

j. Auto Syntax Analysis

When writing macros in the editing screen, the system will run automatic syntax analysis and display the result at the bottom of the screen. The default setting of this function is ON.

k. Help



Click  from the toolbar to open the instruction manual (PDF file).

Use Adobe Acrobat Reader software to read the manual.



EU series TM

Function Window

The macro editing function window provides a toolbox and a search tab, which can be modified according to the requirement of the user. The user can also directly click  **Toolbox** or  **Search** to change tab. See the figure below.

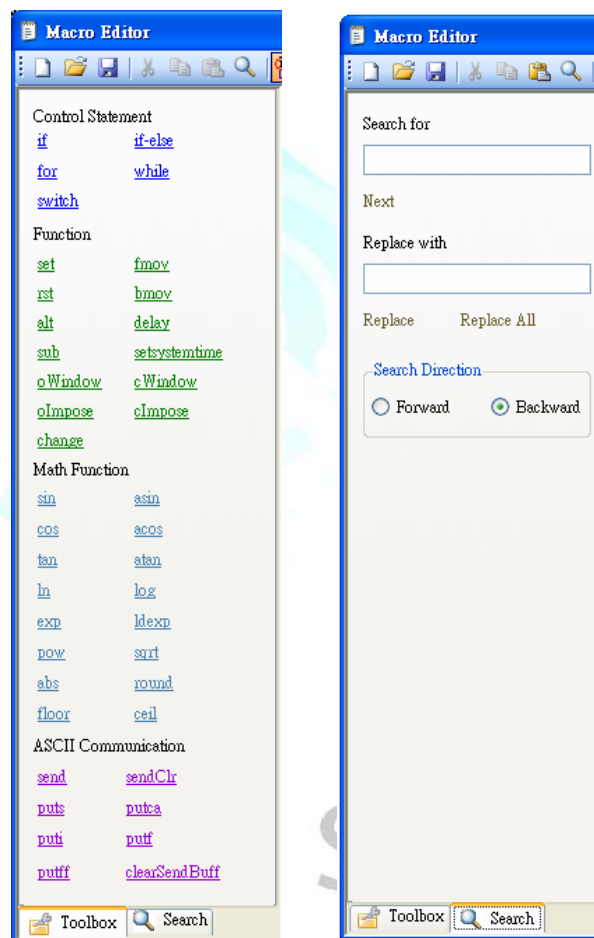


Figure. Functional Window: (a) Toolbox; (b) Search

a. Toolbox

The toolbox provides syntax that can be used by editing macros. The user directly clicks on the required syntax, which will be automatically transferred to the editing window. See the figure below.

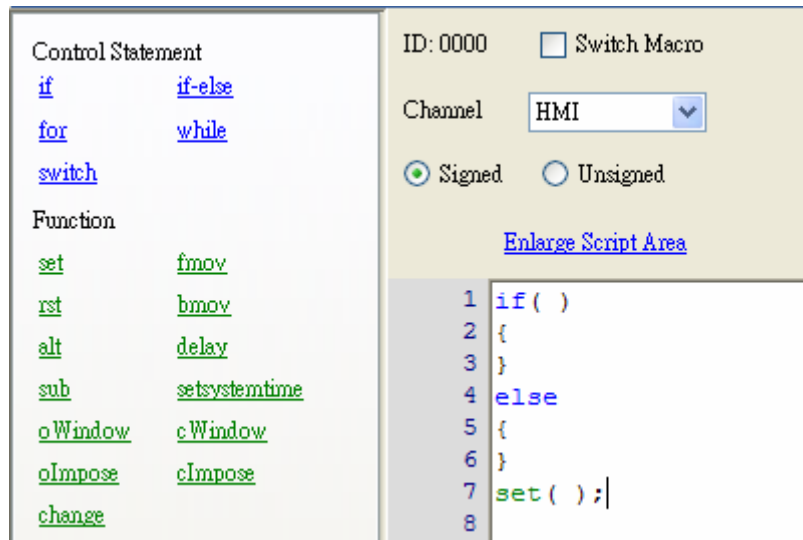




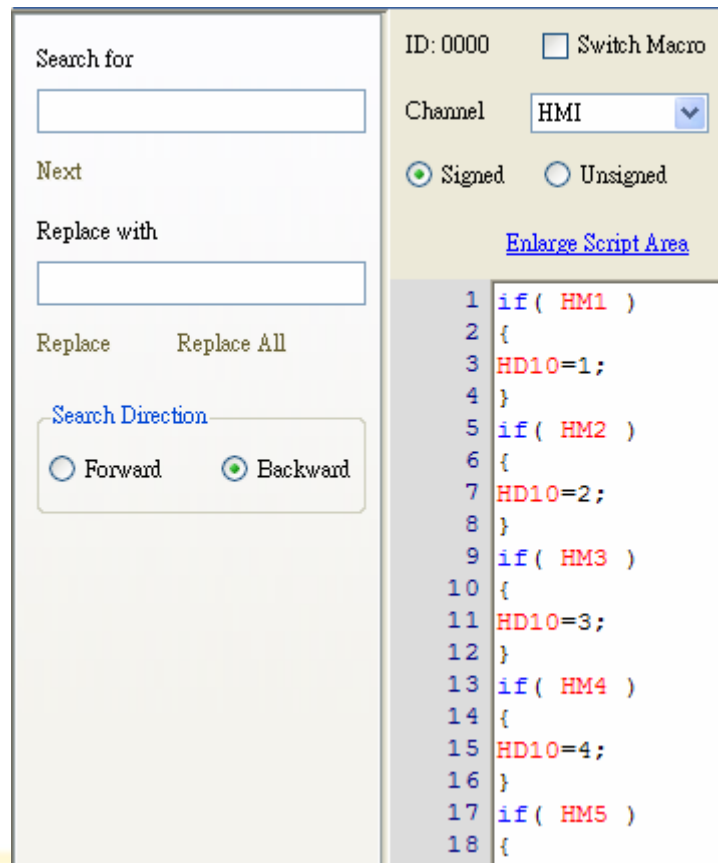
Figure. Set up Syntax Function

EU series™

b. Search

To search macro data in the editing screen, click  from the toolbar to change from the toolbox to the search tab or directly click

 Search to change tab. See the figure below.



Search for

Next

Replace with

Replace Replace All

Search Direction

☐ Forward ☒ Backward

ID: 0000 ☐ Switch Macro

Channel HMI

☒ Signed ☐ Unsigned

[Enlarge Script Area](#)

```
1 if ( HM1 )
2 {
3 HD10=1;
4 }
5 if ( HM2 )
6 {
7 HD10=2;
8 }
9 if ( HM3 )
10 {
11 HD10=3;
12 }
13 if ( HM4 )
14 {
15 HD10=4;
16 }
17 if ( HM5 )
18 {
```

Figure. Search Function

EU series™

The user only needs to enter the to be searched text into the box and click [Next](#) to carry out the search. See the figure below.

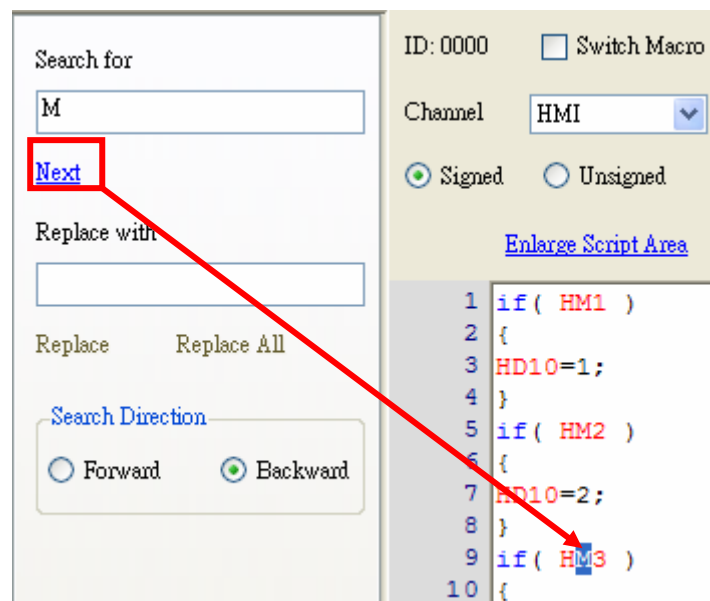


Figure. Search Target

NOTE

Please note that absolute upper or lower cases are required for macro editing, or otherwise a warning window will pop out to remind the user. See the figure below.

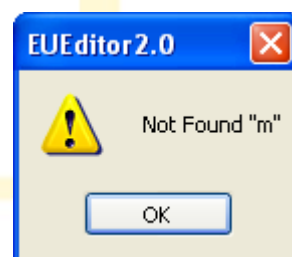
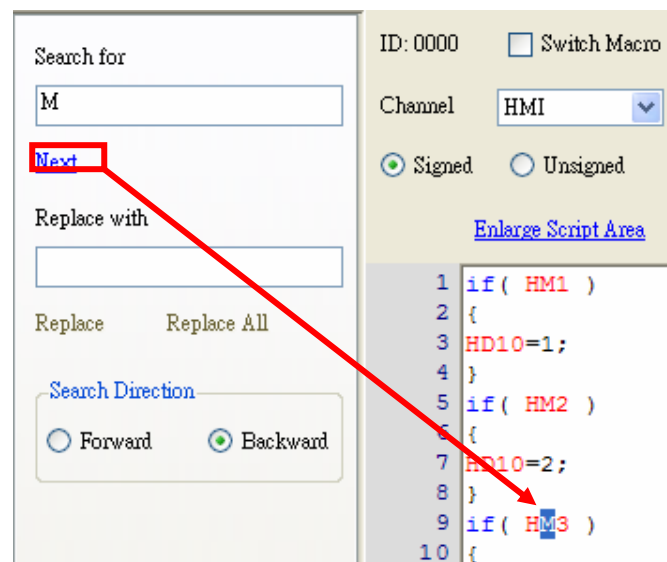
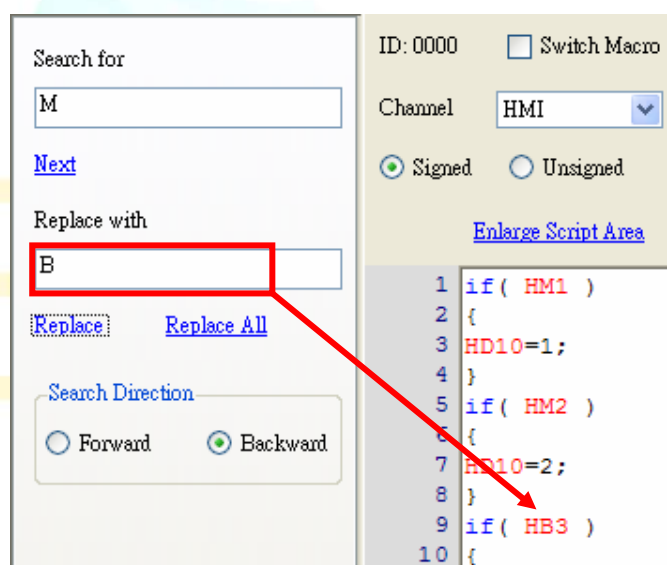


Figure. Warning Window

First, enter the to be replaced text into the box and click [Next](#) to carry out the search. Next, click [Replace](#) or [Replace All](#) for the replacing action. See the figure below.



(a)



(b)

Figure. Search Function: (a) Search; (b) Replace

Editing window

The macro editing window of the macro editor enables the user to make modifications on ID descriptions, to write macro syntax, and to arrange subroutines and the triggering setup. See the figure below.

The screenshot shows the macro editing window with the following elements:

- ID:** 0000
- ☐ Switch Macro
- Comment:** [Empty text box]
- Channel:** HMI (dropdown menu)
- ☐ Sub-function
- ☐ Cycle Interval: 1 Sec. (spinner box)
- ☒ Signed ☐ Unsigned
- Trigger Setup:** Ordinary (dropdown menu)
- [Enlarge Script Area](#) (text link)
- [Device Tag](#) (text link)
- [Input Device](#) (text link)
- Script Area:** A large text area with line numbers 1 through 18 on the left margin.

Name	Function
Comment	Editing macro descriptive name
<input checked="" type="checkbox"/> Switch Macro	Setting as a switch macro
Channel	Changing the control communication port
<input checked="" type="checkbox"/> Sub-funtion	Setting as a subroutine
<input checked="" type="checkbox"/> Cycle Interval	Setting the cycle interval of macro
<input checked="" type="radio"/> Signed <input type="radio"/> Unsigned	Changing data mode
Trigger Setup	Setting up macro triggering criteria
Enlarge Script Area / Display Setting	Enlarging/returning to the editing window
Device Tag	Setting device tag
Input Device	Selecting the input device

Figure. Editing Screen

a. ID: 0000

ID: 0000 is the code of the macro that means the user is currently editing. See the figure below.

The screenshot shows a software configuration window. At the top left, the 'ID: 0000' label is enclosed in a red rectangular box. To its right is a 'Switch Macro' checkbox. Further right is a 'Comment' text field. Below these are a 'Channel' dropdown menu set to 'HMI', a 'Signed' radio button (selected) and an 'Unsigned' radio button, a 'Sub-function' checkbox, a 'Cycle Interval' spinner set to '1' with a 'Sec.' label, and a 'Trigger Setup' dropdown set to 'Ordinary'. At the bottom of the configuration section are two buttons: 'Device Tag' and 'Input Device'. Below the configuration section is a large white area for a script, with a vertical line number column on the left ranging from 1 to 18. A blue link 'Enlarge Script Area' is located above the script area.

Figure. ID Code

EU series™

b. Switch macro

You can check the ☒ Switch Macro to change the macro program to the switch macro. When it is the switch macro, you need to use the switch element to call the switch macro as shown in the following figure.

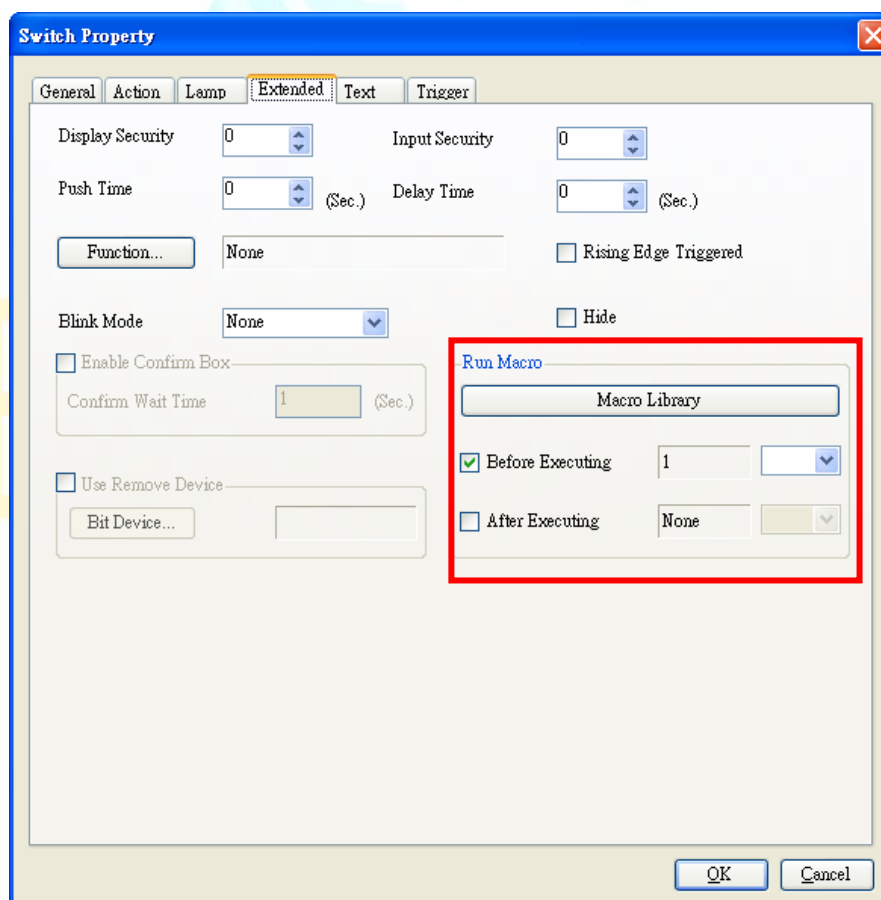
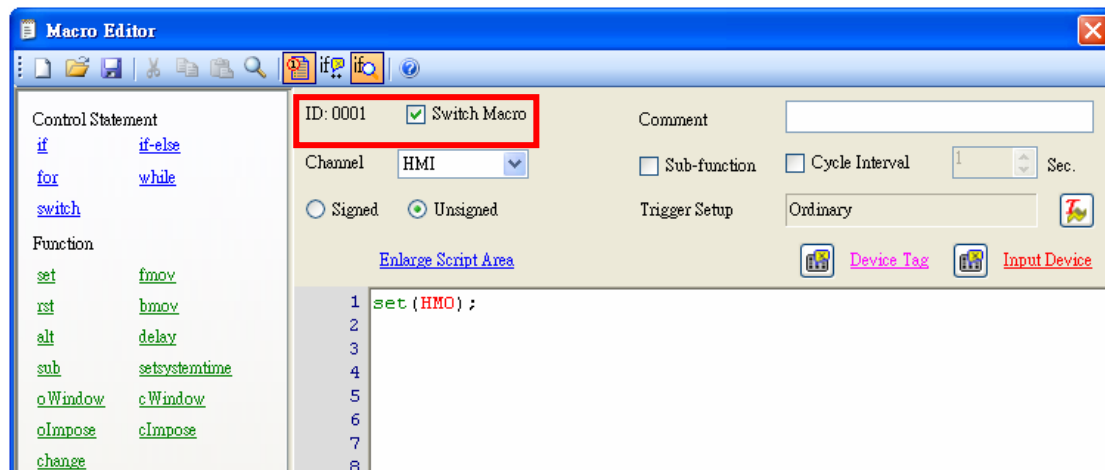


Figure: Setup of switch macro

c. Comment

To make modifications on the content of the description, the user can edit the required text in the description box, which will make later viewing more convenient. See the figure below.

The screenshot shows a configuration window for a macro. At the top, the 'ID' is set to '0000' and there is a 'Switch Macro' checkbox. The 'Channel' is set to 'HMI' via a dropdown menu. Below this are radio buttons for 'Signed' (selected) and 'Unsigned'. To the right, there are checkboxes for 'Sub-function' and 'Cycle Interval' (set to 1 second). The 'Trigger Setup' is set to 'Ordinary'. There are two icons with labels: 'Device Tag' and 'Input Device'. A large text area at the bottom is for the script, with line numbers 1 through 18 on the left. The 'Comment' field at the top right is highlighted with a red rectangular box.

Figure. Description on Modifications

d. Channel

To set up communication port devices, use the pull-down menu to select the required communication port device. See the figure below.

Channel HMI

HMI
COM1
COM2
COM3

ID: 0000 ☐ Switch Macro

Channel HMI

☒ Signed ☐ Unsigned

Comment

☐ Sub-function ☐ Cycle Interval 1 Sec.

Trigger Setup Ordinary

[Enlarge Script Area](#)

[Device Tag](#) [Input Device](#)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

Figure. Changing Communication Port

e. Sub-funtion

Select ☒ Sub-funtion from the macro editor to change macro programs to sub-funtion. The macro library will display these programs as sub-funtion. When a macro is a sub-funtion, the user needs to use sub command to call and to run a sub-funtion. See the figure below.

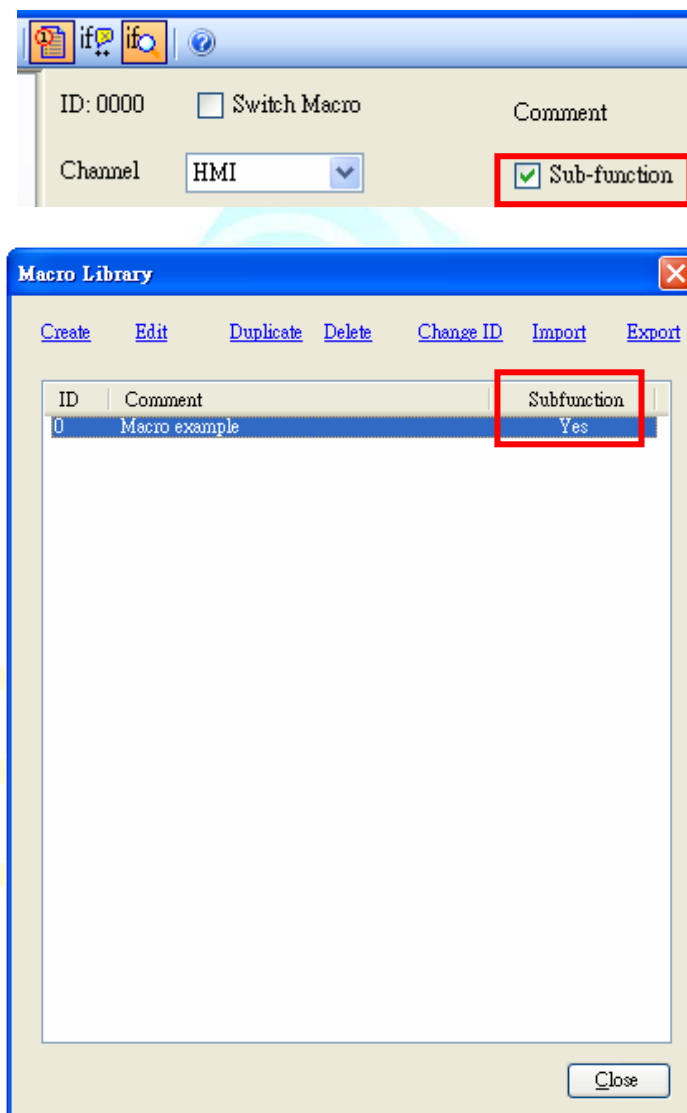


Figure. Setting up a sub-funtion

f. Cycle interval

When checking the ☒ Cycle Interval, the program will execute the macro after the interval set is reached, as shown in figure below.

The screenshot shows a macro configuration window with the following fields and settings:

- ID: 0001
- ☐ Switch Macro
- Comment: [Empty text box]
- Channel: HMI (dropdown menu)
- ☐ Sub-function
- ☒ Cycle Interval (highlighted with a red box) | 10 (spin box) | Sec. (unit)
- ☐ Signed | ☒ Unsigned
- Trigger Setup: Ordinary (dropdown menu)
- Buttons: Device Tag, Input Device
- Link: [Enlarge Script Area](#)

The script area contains the following code:

```
1 set (HMO) ;  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18
```

Figure: Setup of cycle interval

EU series™

g. Data Mode


To change the data mode of a device from with signed to unsigned or vice versa, select ☒ Signed or ☒ Unsigned accordingly. See the figure below.

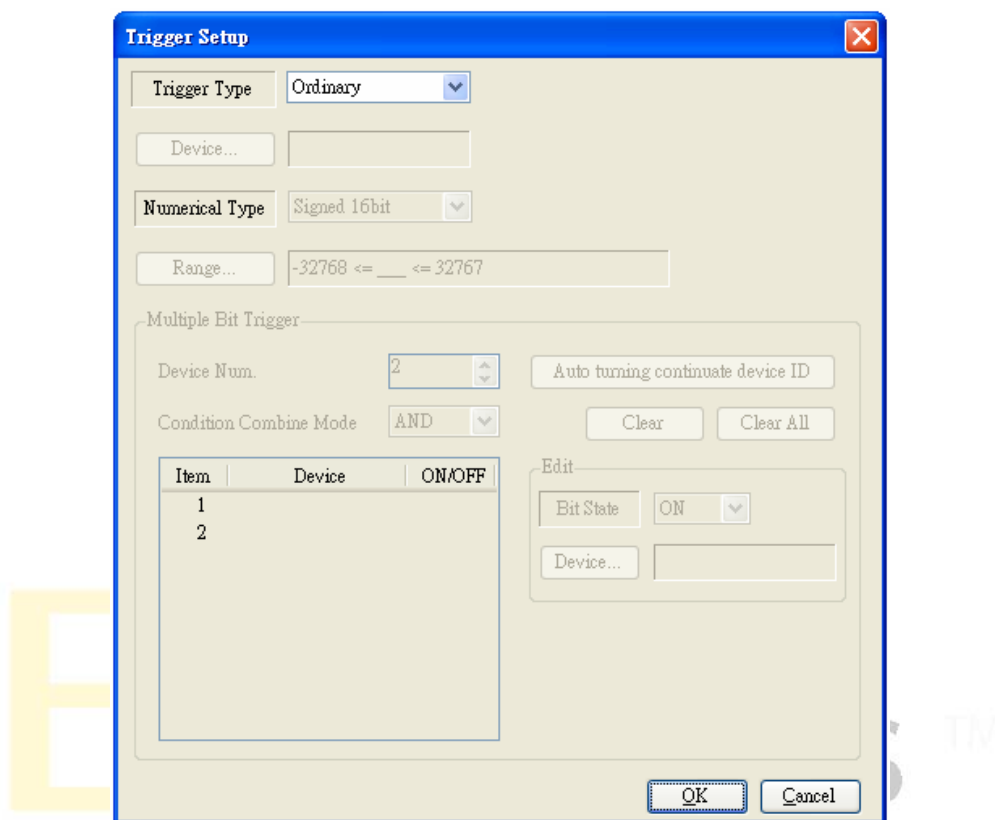
The screenshot shows a configuration window for a device. At the top, there are fields for 'ID: 0000', 'Channel' (set to 'HMI'), 'Comment' (empty), 'Sub-function' (unchecked), 'Cycle Interval' (set to 1 Sec.), and 'Trigger Setup' (set to 'Ordinary'). Below these, there are two radio buttons: 'Signed' (selected) and 'Unsigned' (unselected). These two radio buttons are enclosed in a red rectangular box. To the right of the radio buttons, there are two icons: a device icon and a tag icon, with labels 'Device Tag' and 'Input Device' respectively. Below the configuration fields, there is a section labeled 'Enlarge Script Area' with a list of numbers from 1 to 18 on the left side, indicating a script editor area.

Figure. Changing Data Mode

h. Triggering Setup

When selecting the triggering mode, the program will run the macro according to the triggering criteria. If the user chooses no effect for the triggering mode, the macro will be run continuously.


To set up macro trigger criteria, click  to open the Trigger Setup window. From there, the user can modify the trigger mode and the criteria. See Figure 3-4-28 below.

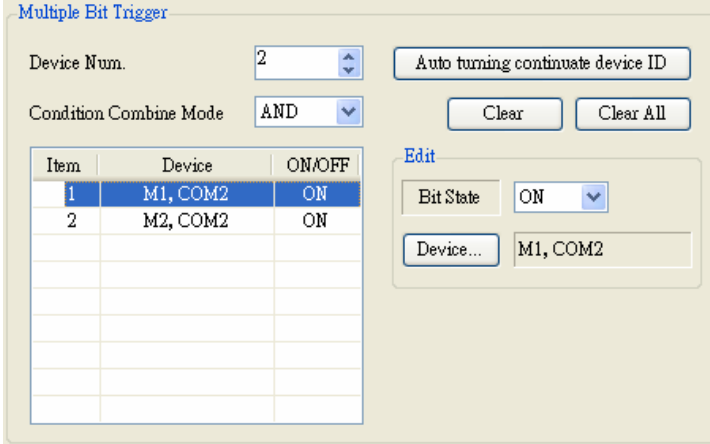


Triggering Mode	Description
Ordinary	No Triggering Mode
ON	Delivering actions only when the device is ON
OFF	Delivering actions only when the device is OFF
Range	Delivering actions only when the value of the device is within this range.
Multiple bit Trigger	Delivering actions only when all the devices (more than two) have met the criteria.

Figure 3-6-216 Descriptions on the Triggering Mode Setup



The Triggering Mode Setup has multiple bit trigger and 2 number of devices. Click  to open the Device Setup window, and set the triggering devices as M1 and M2, respectively. Press OK to send the files to HMI. Button pressing would trigger further actions only when M1 and M2 buttons are both ON. See Figure 3-6-294 below.



Multiple Bit Trigger

Device Num. Auto tuning continue device ID

Condition Combine Mode

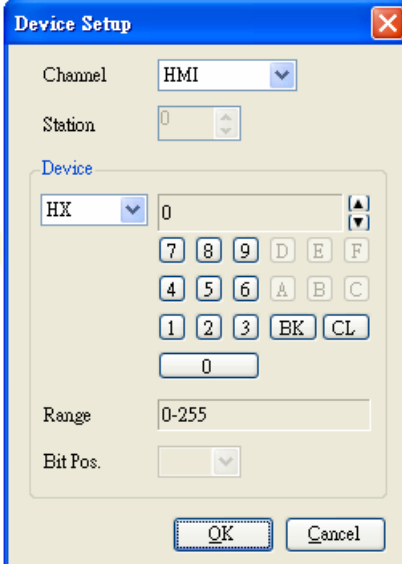
Item	Device	ON/OFF
1	M1, COM2	ON
2	M2, COM2	ON

Edit

Bit State

Device...

(a)



Device Setup

Channel

Station

Device

Range

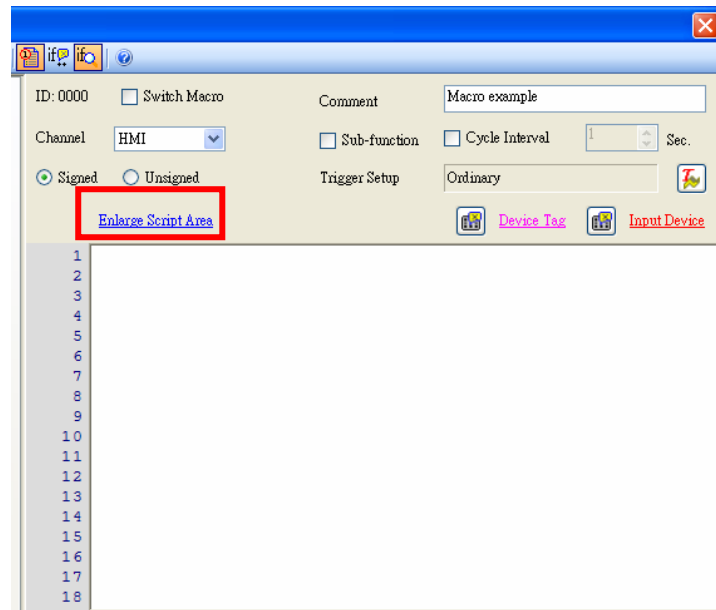
Bit Pos.

(b)

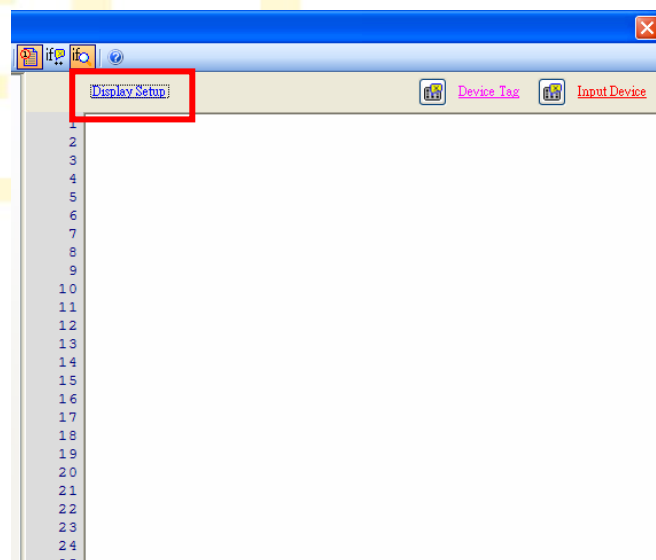
Figure 3-6-237 Multiple bit Trigger: (a) The Multiple bit Triggering Setup Window; (b) Setting up the Devices

i. Enlarge Script Area

To enlarge the script editing screen, click [Enlarge Script Area](#) to enlarge the editing screen and extend the screen toward the top of the screen. Next, click [Display Setting](#) to return to the original state. See the figure below.




(a)



(b)

Figure. Editing Screen: (a) Enlarge Script Area; (b) Display Setting

j. Device tag

If you want to establish the device tag, click the  [Device Tag](#). You can set devices of different COM so that the device of various COM can be used freely when editing the macro, as shown in figure below.

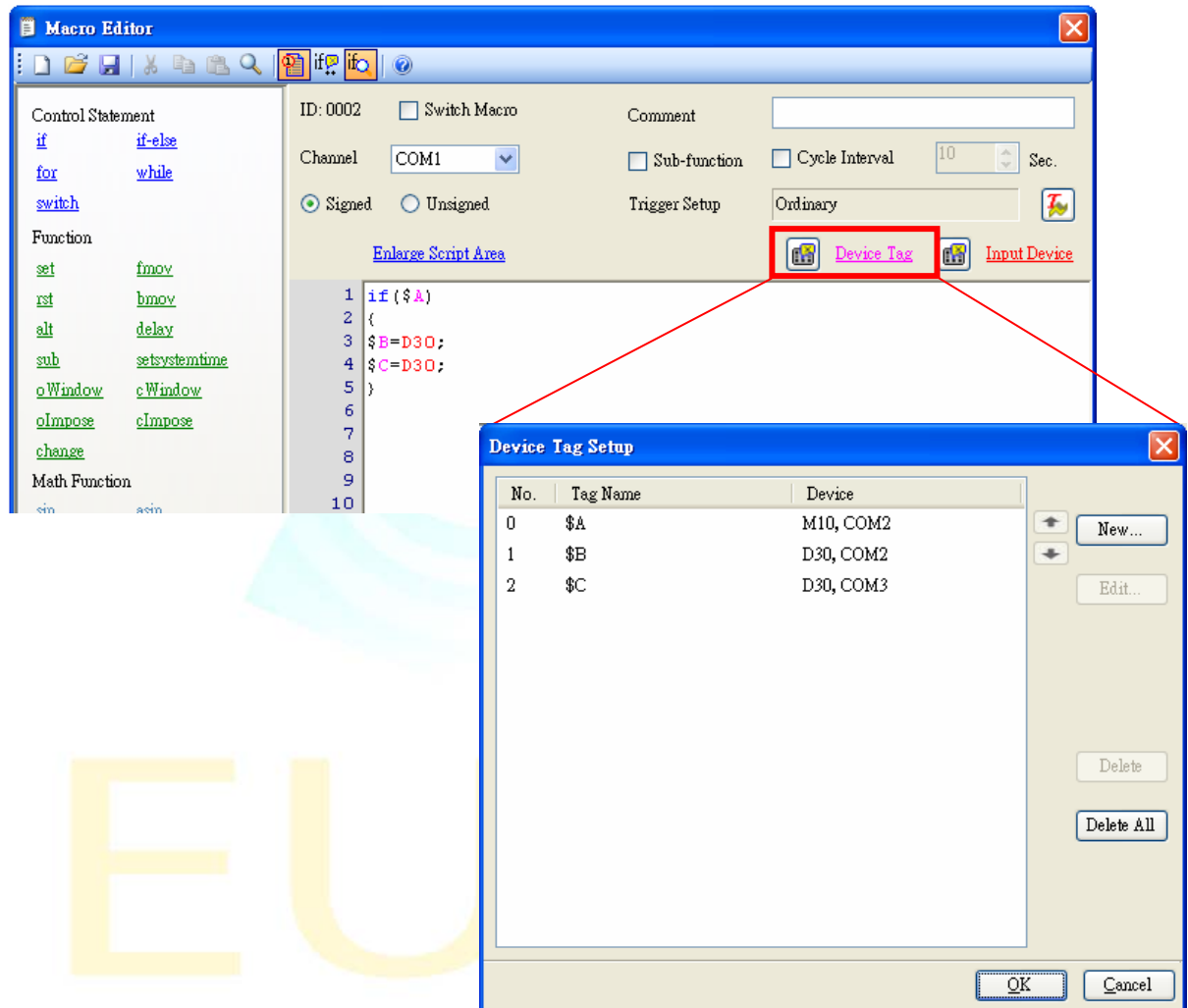


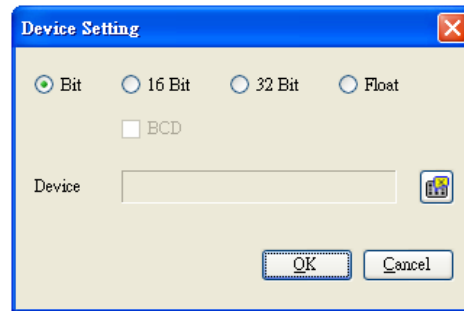


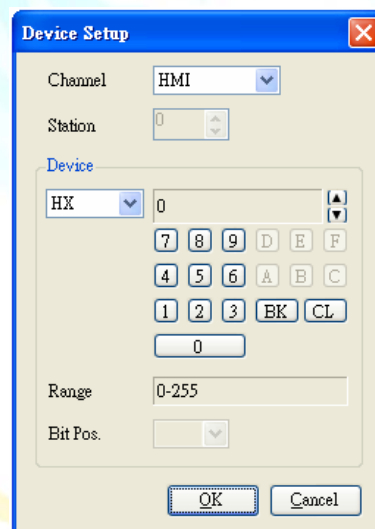
Figure: Setup of register tag

k. Input Device

For device installation, click  to open the Device Setting dialog box. From there, the user can modify the device mode. After selecting the mode, click  again to open the Device Setup dialog box. Press OK to complete the setup. See Figure 3-4-29.



(a)



(b)

Figure 3-4-29 Input Devices : (a) Setting up the Format; (b) Setting up the Device

NOTE

When editing macro using Allen-Bradley PLC special device, Allen-Bradley PLC device coding has been calculated, as shown in the following figure 3-4-5-16.

Special devices include: B、TS、TP、TC、CS、CP、CC、N

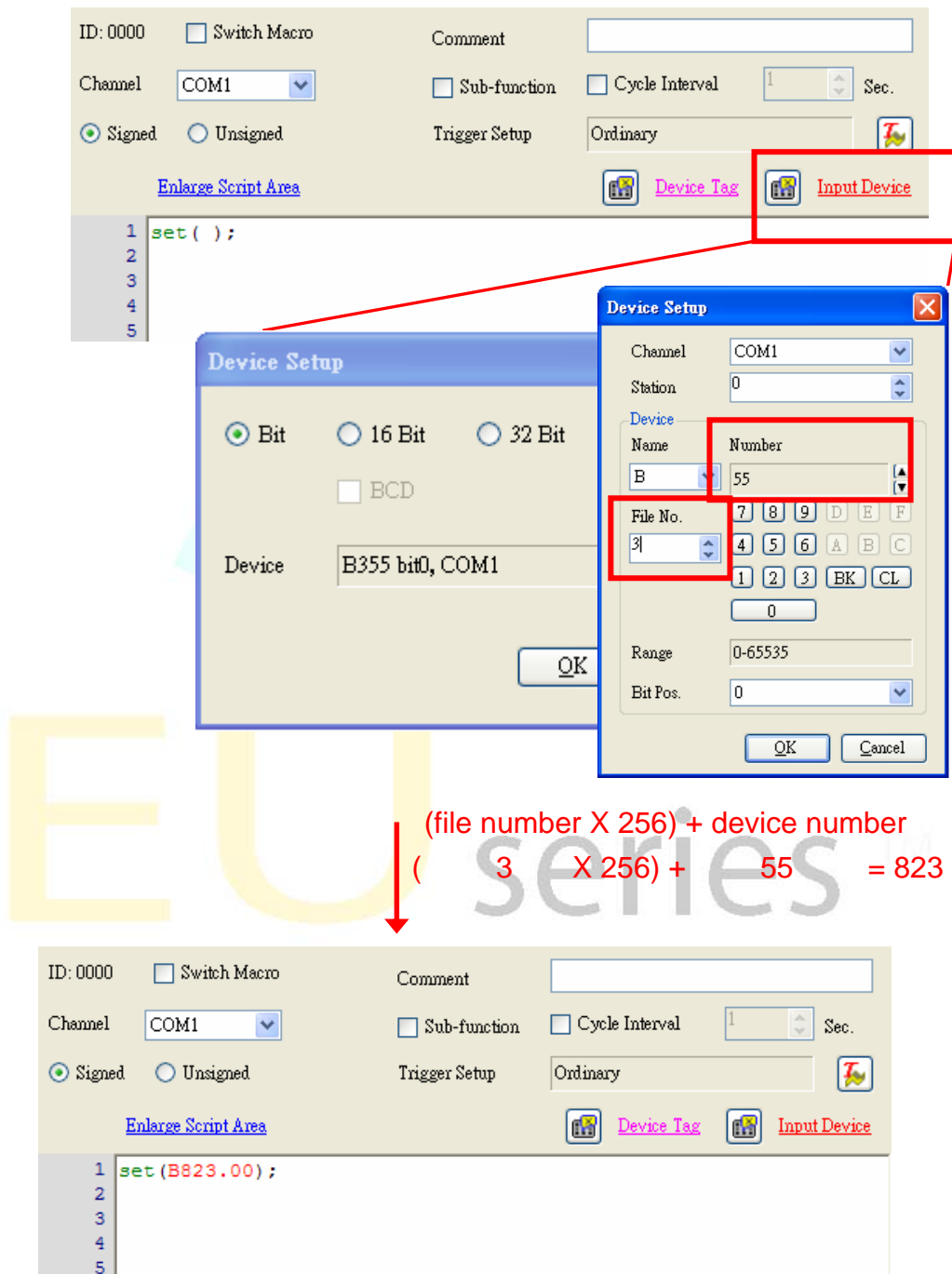


Fig. 3-4-5-16、Device setup

When editing macro using Keyence PLC special device, Keyence PLC device coding has been calculated, as shown in the following figure 3-4-5-17.

Special devices include: RLY 、CR 、MR 、LR

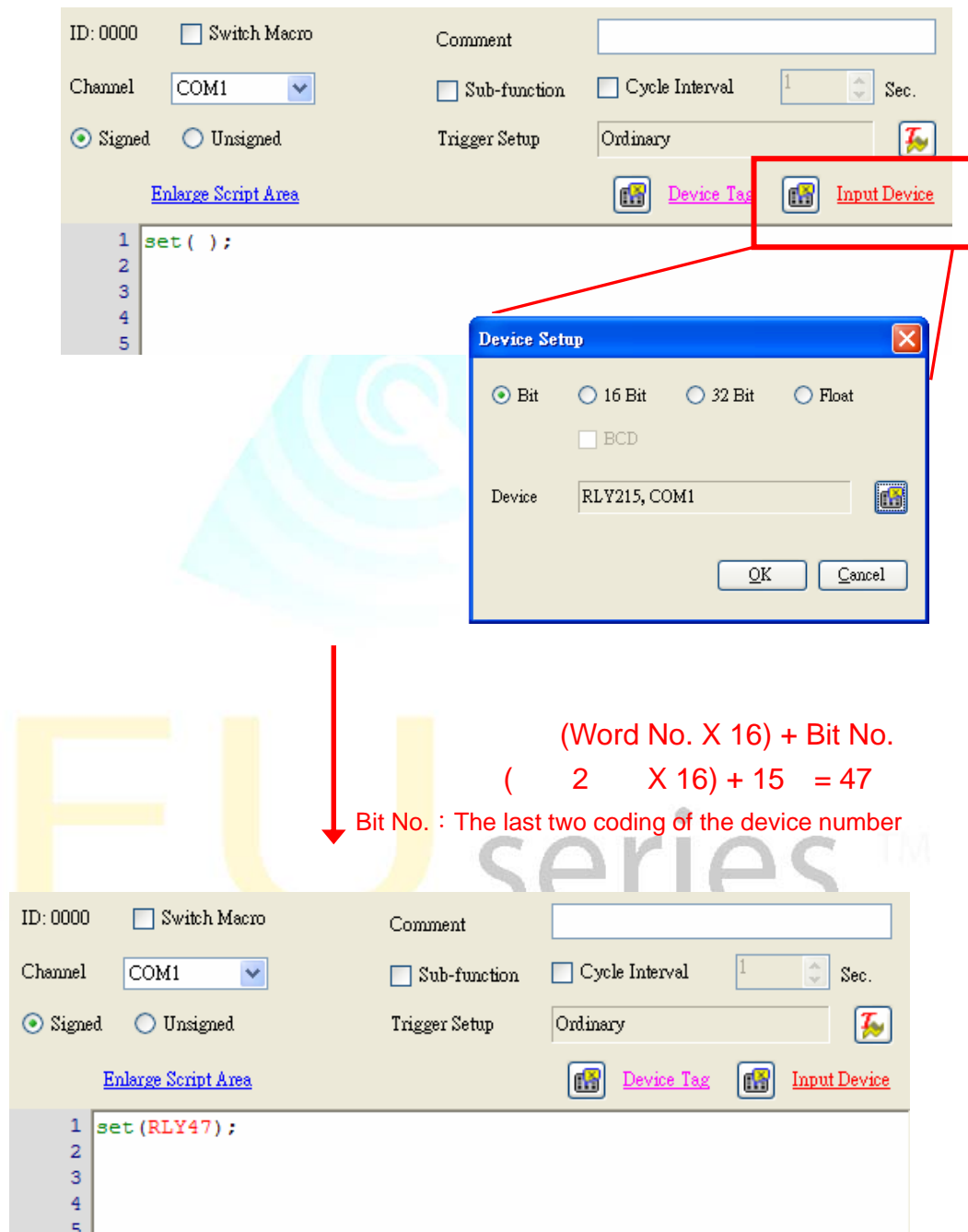




Fig. 3-4-5-17 、 Device setup

Error Window

Error window is available for carrying out syntax analysis when editing macro programs in order to ensure the correctness of the program. The user can make correction according to the message described in the error window. See the figure below.

[Enlarge Script Area](#)

 [Device Tag](#) [Input Device](#)

```
1 HD0=2
2
3 set( HD2 );
4
5 while( )
6 {
7 }
8
9 sin(10)
10
11
12
13
14
15
16
17
18
```

	Description	Line	C...
✖ 1	Parameter error, must be a Bit device.	3	1
✖ 2	Logical expression can not be empty.	5	7
✖ 3	Must be ';	9	8

Figure. Error Window

Introducing the Basic Commands

a. Operation Description

Click Syntax in the function window to have it moved to the editing window directly. Programming language and devices are presented in the corresponding colors. The macro program will be run repeatedly. See the figure below.

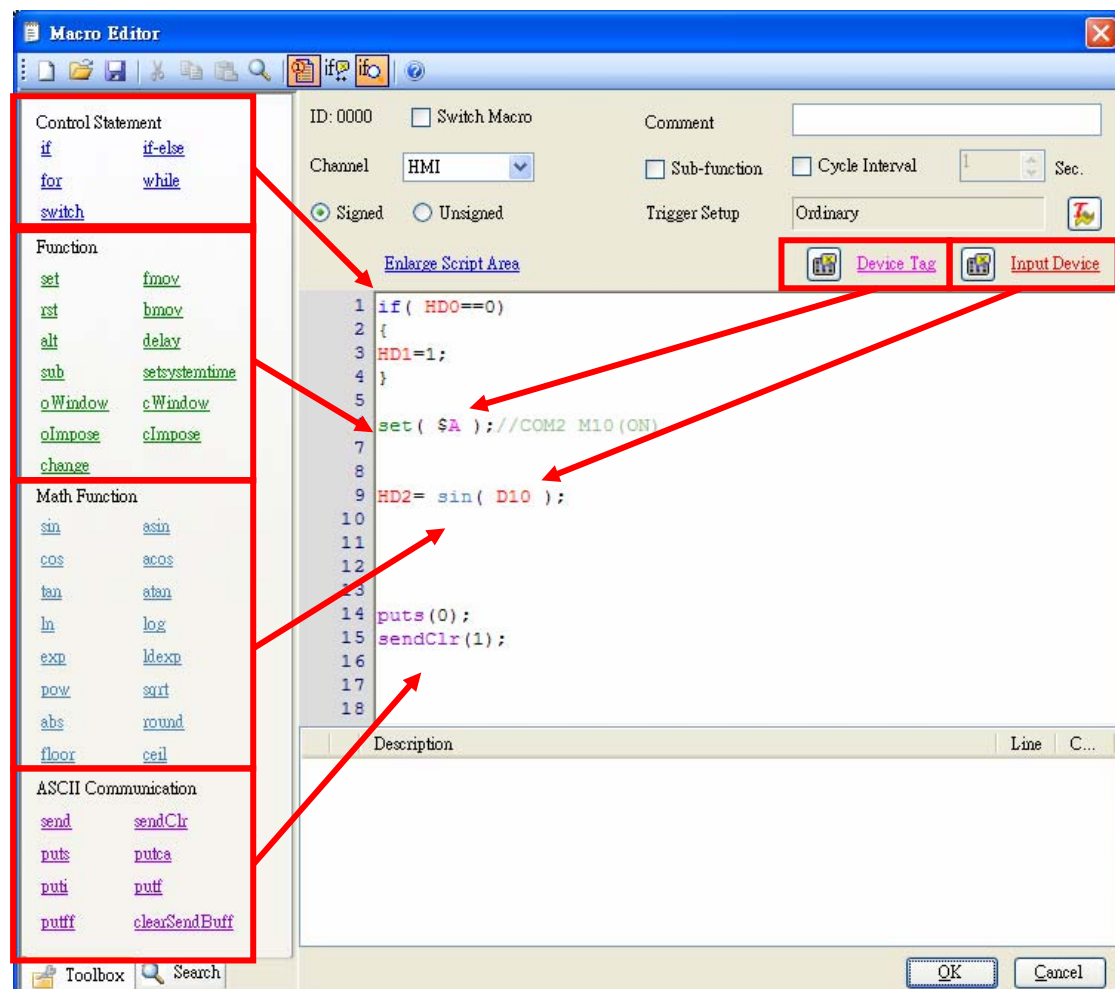


Figure. Operation Description

b. Device Usage

Macro editing follows the general programming language with internal syntax rules. The user can freely change the devices, the operational symbols, and the conditionals. A more detailed description is provided in Table 3-4-24 below Figure. Description on Syntax Rules, Operational Symbols, and Conditionals

Device Syntax: Station Number , 32 bit, Float, Word Device, Bit Device

Device	Example	Description
Bit	M0	Using general bit
	D.M0	Using bits as words
	W.D.M0	Using bits as 32 bit words
	1.M0	Assigning PLC station code using general bit
	1.D.M0	Assigning PLC station code using bits as words
	1.W.D.M0	Assigning PLC station code using bits as 32-bit words
Word	D0	Using general bit
	W.D0	Using bits as 32-bit words
	D0.01	Using words as bits
	1.D0	Assigning PLC station code using general words
	1.W.D0	Assigning PLC station code using 32-bit words
	1.D0.01	Assigning PLC station code using words as bits
	F.D0	Real floating points
Operational Symbol	+ , - , * , / , % , & , , << , >>	
Conditional	== , != , < , > , <= , >= , && ,	

For detailed **Operative Symbols**, please see table 3-4-5-6.

Table 3-4-5-6: **Operative Symbols** description

Operative Symbols	Function	Example	Action
+	Addition	HD2=HD2+5;	HD2 adds 5
		HD2++;	HD2 accumulates 1
		HD3=2; HD4=1; HD2=HD3+HD4;	HD2=2+1=3
-	Subtraction	HD2=HD2-5;	HD2 subtracts 5
		HD2--;	HD2 subtracts 1 gradually
		HD3=2; HD4=1; HD2=HD3-HD4;	HD2=2-1=1
*	Multiplication	HD2=HD2*5;	HD2 multiplies by 5
		HD3=6; HD4=2; HD2=HD3*HD4;	HD2=6x2=12
/	Division	HD2=HD2/5;	HD2 divides by 5
		HD3=6; HD4=2; HD2=HD3/HD4;	HD2=6/2=3
%	Remainder	HD3=6; HD2=HD3%4;	6/4, remainder=2 HD2=2
		HD3=7; HD4=2; HD2=HD3%HD4;	7/2, remainder=1 HD2=1
&	AND: compare each bit of the operand, and if they are all 1, the result will be 1.	HD3=26; HD2=HD3&3;	26(decimal value)=11010(binary value) 3(decimal value)=00011(binary value) $\begin{array}{r} 11010 \\ \& 00011 \\ \hline 00010 \end{array}$ HD2=2(decimal value)=00010(binary value)
		HD3=26; HD4=105; HD2=HD3&HD4;	26(decimal value)=0011010(binary value) 105(decimal value)=1101001(binary value) $\begin{array}{r} 0011010 \\ \& 1101001 \\ \hline 0001000 \end{array}$ HD2=8(decimal value)=0001000(binary value)

	OR: compare each bit of the operand, and if one of them is 1, the result will be 1.	$HD3=26;$ $HD2=HD3 3;$	26(decimal value)=11010(binary value) 3(decimal value)=00011(binary value) 11010 <u>OR 00011</u> 11011 HD2=27(decimal value)=11011(binary value)
		$HD3=26;$ $HD4=105;$ $HD2=HD3 HD4;$	26(decimal value)=0011010(binary value) 105(decimal value)=1101001(binary value) 0011010 <u>OR 1101001</u> 1111011 HD2=123(decimal value)=1111011(binary value)
<<	Left shift: shift a bit to left.	$HD3=26;$ $HD2=HD3<<4;$	26(decimal value)=0011010(binary value) Shift 4 bits to left: 0 0 1 1 0 1 0 0 0 0 HD2=32(decimal value)=0100000(binary value)
		$HD3=26;$ $HD4=3;$ $HD2=HD3<<HD4;$	26(decimal value)=0011010(binary value) Shift 3 bits to left: 0 0 1 1 0 1 0 0 0 0 HD2=80(decimal value)=1010000(binary value)
>>	Right shift: shift a bit to right.	$HD3=26;$ $HD2=HD3>>4;$	26(decimal value)=0011010(binary value) Shift 4 bits to right: 0 0 0 0 0 0 1 1 0 1 0 HD2=1(decimal value)=0000001(binary value)
		$HD3=26;$ $HD4=3;$ $HD2=HD3>>HD4;$	26(decimal value)=0011010(binary value) Shift 3 bits to right: 0 0 0 0 0 1 1 0 1 0 HD2=3(decimal value)=0000011(binary value)

The conditional expressions are listed in table 3-4-5-7.

Table 3-4-5-7: Conditional expressions

Conditional expression	Function	Example	Action description
==	Equal to	HD2==0	When HD2=0
!=	Unequal to	HD2!=0	When HD2≠0
<	Less than	HD2<0	When HD2<0
>	Greater than	HD2>0	When HD2>0
<=	Less than or equal to	HD2<=0	When HD2<=0
>=	Greater than or equal to	HD2>=0	When HD2>=0
&&	AND	HD2==0&&HD3>5	When HD2==0 and HD3>5
	OR	HD2==0 HD3>5	When HD2==0 or HD3>5

NOTE

Apply real float points as the device for calculating the data if mathematical functions are used.

EU series™

c. Control statement

Detailed statement control commands are described in the table below.

Figure 3-4-24 Descriptions of the Statement Control Commands

Statement Control	Command Function	Example	Action Description
if	Execute if condition holds; Do not execute if condition does not hold;	<pre>if (HD0==0) { HD1=1; }</pre>	If HD0=0, HD 1=1
if-else	If condition holds, Excute the true statement; If condition does not hold, Excute the false statement;	<pre>if (HD0==0) { HD1=1; } else { HD1=0; }</pre>	If HD0=0, HD 1=1 If HD0≠0, HD1=0
while	Execute if condition holds; Stop if condition does not hold;	<pre>while (HD0==0) { HD1=HD1+1; }</pre>	When HD0=0, HD1=HD1+1 When HD0≠0, stop
for	After determining the condition holds, add value accumulatively to the original value.	<pre>for (HD0=0; HD0<3; HD0++) { HD1=HD0; }</pre>	For HD0<3 , HD0=HD1 The values of HD0 are 0→1→2→3 The values of HD1 are 0→1→2→3
switch	The acquired values, should match the corresponding cases.	<pre>switch (HD0) { case 1: HD1=1; break; case 2: HD1=2; break; }</pre>	If HD0=0, HD 1=1 If HD0=2, HD 1=2

d.Function

Detailed descriptions of functional instructions are listed in Table 3-4-25 and Table 3-4-26 below.

Table 3-4-25 Functional Instructions

Instruction	Function	Example	Action
set(B)	Bit device is normally open	<code>set (HM0) ;</code>	HM0=ON
rst(B)	Bit device is normally closed	<code>rst (HM0) ;</code>	HM0=OFF
alt(B)	Bit device is alternately open/closed	<code>alt (HM0) ;</code>	HM0 continuously alternates.
sub(W/Z)	Call a subroutine	<code>HD0=3;</code> <code>sub (HD0) ;</code>	Call the subroutine with ID 3
		<code>sub (3) ;</code>	
fmov(Z,W,Z)	Multi-cast value	<code>fmov (123,HD0,2) ;</code>	Send the value123 to HD0 and HD1.
bmov(W,W,Z)	Transfer value in batch	<code>HD0=123;</code> <code>HD1=456;</code> <code>bmov (HD0,HD10,2) ;</code>	Set HD0=123 and send it to HD10. Set HD1=456 and send it to HD11.
delay(W/Z)	delay the user setting time	<code>HD0=1000;</code> <code>delay(HD0) ;</code>	Delay 1000 ms ◦ (Remark 2)
		<code>delay(1000) ;</code>	
setsystemtime(W)	System time setting	<code>setsystemtime(HD0) ;</code>	Send those value HD0~HD6 to system time ◦ HD0=year ; HD1=month ; HD2=date ; HD3=week ; HD4=hour ; HD5= minute ; HD6= second (Remark 3) (Remark 4) (Remark 5)
Remark	1. B : Bit device , W : Word device , N : value of 1 or 2 , Z : positive integer 2. After Delay command is executed, HMI will stop the operation. The operation will resume after the delay time has elapsed. 3. Year setting limits from 2000 to 2037, more than this range will not be write to those system time devices . 4. The execution setsystemtime instruction and the date and time display devices HDR ,The two can not use in the same time. 5. Do not for a long time to trigger the system time setting instructions.		

Table 3-4-26 Functional Instructions(continued)

Instruction	Function	Example	Action
Change(W/Z)	Change basic screen	HD0=3; change (HD0) ;	Jump to basic careen 3.
		change (3) ;	
oWindow(N,W/Z)	Open the overlapped Nth window screen numbered Z.	HD0=3; oWindow (1, HD0) ;	Open the overlapped 1 st window screen numbered 3.
		oWindow (1, 3) ;	
cWindow(N,W/Z)	Close the overlapped Nth window screen numbered Z.	HD0=4; cWindow (2, HD0) ;	Close the overlapped 2 nd window screen numbered 4.
		cWindow (2, 4) ;	
oImpose(N,W/Z)	Open the additional overlapped Nth window screen numbered Z.	HD0=3; oImpose (1, HD0) ;	Open the additional overlapped 1 st window screen numbered 3.
		oImpose (1, 3) ;	
cImpose(N,W/Z)	Close the overlapped Nth window screen numbered Z.	HD0=4; cImpose (2, HD0) ;	Close the additional overlapped 2 nd window screen numbered 4.
		cImpose (2, 4) ;	
Remark	1. B : Bit device , W : Word device , N : value of 1 or 2 , Z : positive integer		

EU series™

e. Arithmetic Function

More detailed function commands are described in the table below.

Figure. Description on Arithmetic Function Commands

Math Function	Description	Example	Action
sin(W/R)	To obtain the sine of X	<code>HD0=10;</code> <code>F.HD1=HD0*3.14/180;</code> <code>F.HD2= sin(F.HD1);</code> <code>F.HD2= sin(0.174);</code>	HD2 \doteq 0.173
cos(W/R)	To obtain the cosine of X	<code>HD0=10;</code> <code>F.HD1=HD0*3.14/180;</code> <code>F.HD2= cos(F.HD1);</code> <code>F.HD2= cos(0.174);</code>	HD2 \doteq 0.985
tan(W/R)	To obtain the tangent of X	<code>HD0=10;</code> <code>F.HD1=HD0*3.14/180;</code> <code>F.HD2= tan(F.HD1);</code> <code>F.HD2= tan(0.174);</code>	HD2 \doteq 0.176
asin(W/R)	To obtain the arcsine of X	<code>HD0=10;</code> <code>F.HD1=HD0*3.14/180;</code> <code>F.HD2= asin(F.HD1);</code> <code>F.HD2= asin(0.174);</code>	HD2 \doteq 0.175
acos(W/R)	To obtain the arccosine of X	<code>HD0=10;</code> <code>F.HD1=HD0*3.14/180;</code> <code>F.HD2= acos(F.HD1);</code> <code>F.HD2= acos(0.174);</code>	HD2 \doteq 1.396
atan(W/R)	To obtain the arctangent of X	<code>HD0=10;</code> <code>F.HD1=HD0*3.14/180;</code> <code>F.HD2= atan(F.HD1);</code> <code>F.HD2= atan(0.174);</code>	HD2 \doteq 0.172
Remark	W : Word device R : radian, radian= value *π/180		

Figure. Description on Arithmetic Function Commands

Arithmetic Function	Command Function	Example	Action Description
ln(W/X)	Get ln(X)	HD0=5; HD1=ln(HD0); HD1=ln(5);	HD1=ln5
log(W/X)	Get log(X)	HD0=5; HD1=log(HD0); HD1=log(5);	HD1= log ₁₀ 5
exp(W/X)	Exp(X)	HD0=5; HD1=exp(HD0); HD1=exp(5);	HD1=e ⁵
ldexp(W/X,W/Y)	Get $X * 2^Y$	HD0=5; HD1=10; HD2=ldexp(HD0,HD1); HD2=ldexp(5,10);	HD2=5*2 ¹⁰
pow(W/X,W/Y)	Get X^Y	HD0=5; HD1=10; HD2=pow(HD0,HD1); HD2=pow(5,10);	HD2=5 ¹⁰
sqrt(W/X)	Get the square root of X	HD0=5; HD1= sqrt(HD0); HD1=sqrt(5);	HD1=√5
abs(W/X)	Get the absolute value of x	HD0=-5; HD1= abs(HD0); HD1=abs(-5);	HD1= -5 =5
round(F.W/X)	Round off it after getting the decimal point of X	F.HD0=5.5; F.HD1= round(F.HD0); F.HD1=round(5.5);	HD1=6
floor(F.W/X)	Floor it after getting the decimal point of X	F.HD0=5.5; F.HD1= floor(F.HD0); F.HD1=floor(5.5);	HD1=5
ceil(F.W/X)	Ceil it after getting the decimal point of X	F.HD0=5.4; F.HD1= ceil(F.HD0); F.HD1=ceil(5.4);	HD1=6
Note	W: Word Device; X and Y: Integrals		

Ex

In the case when math functions from the macro editor are used to calculate $\sqrt{3^2}=3$, for which the actual answer from the calculation is 3, but device setup will affect the displayed data. In this case, one should use the real float point device to run the math function computation. The highest accuracy of float point device is five digits after the decimal point. See the table below for more details.

Figure. Description on Device Setup

<pre>Device1=sqrt(Device2); Device2=pow(3,2);</pre>			
State	Device1	Device2	Result
1	HD0	HD2	0
2	HD0	F.HD2	0
3	F.HD0	HD2	3
4	F.HD0	F.HD2	3

f. ASCII Communication

Detailed descriptions of ASCII Communication are listed in Table 3-4-27 below.


Table 3-4-27 ASCII Communication

ASCII Communication	Description	Example	Action
send(W/Z)	Transfer ASCII communication instructions to the printer.	HD0=1; send(HD0); send(1);	Transfer ASCII communication instructions to the COM1 printer.
sendClr(W/Z)	Transfer ASCII communication instructions to the printer, and clear ASCII communication instructions in the buffer.	HD0=1; sendClr(HD0); send(1);	Transfer ASCII communication instructions to the COM1 printer, and clear ASCII communication instructions in the buffer.
puts(W/Z)	Specified ASCII command stored in the buffer.	HD0=5; puts(HD0); puts(5);	NO.5 ASCII command stored in the buffer.
puti(W)	values stored in the buffer.	HD0=5; puts(HD0); W.HD0=5; puts(W.HD0);	The “5” stored in the buffer.
putf(F,W/Z)	Float values stored in the buffer. putf(F.Device,Decimal point)	F.HD0=5.26; putf(F.HD0,2); F.HD0=5.26; HD10=2; putf(F.HD0,HD10);	The “5.26” stored in the buffer.
putff(W,W/Z)	values stored in the buffer. putff(Device,Decimal point)	HD0=5.26; putf(HD0,2); HD0=5.26; HD10=2; putf(HD0,HD10);	The “5.26” stored in the buffer.
putca(W,W/Z,W/Z)	Characters stored in the buffer. putca(Device,Digits,Byte order) No change byte order : 0 Change byte order : 1	HD11=3; HD12=0; putf(HD0,HD11,HD12); putf(HD0,3,0);	When HD0 setting ABC, the ABC stored in the buffer.
Remark	W : Word device , Z : positive integer , F : Float		

NOTE

fmov and bmov of macro function is not supported PLC brands: Siemens, Allen-Bradley and Schneider.

NOTE

When editing comments in Chinese in the macro editor, you need to disable the  auto syntax analysis, as shown in the following figure.

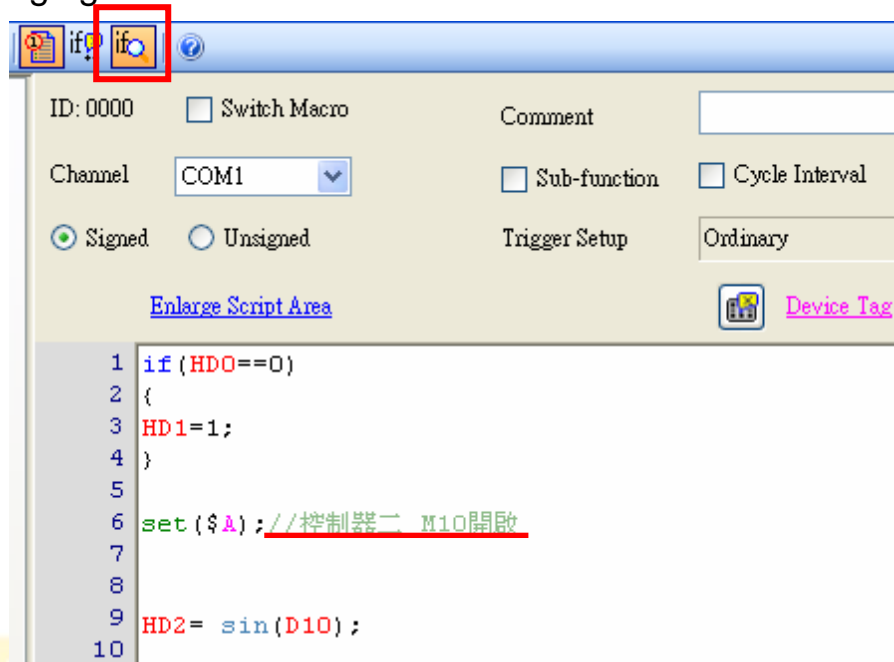


Figure: Edit Chinese notes

EU series™

Appendix A General Practice

Interlock Keys

Open EU Editor 2.0 Software and set up the following components in the editing screen in order to have a key interlock function. Read the following Table for more details.

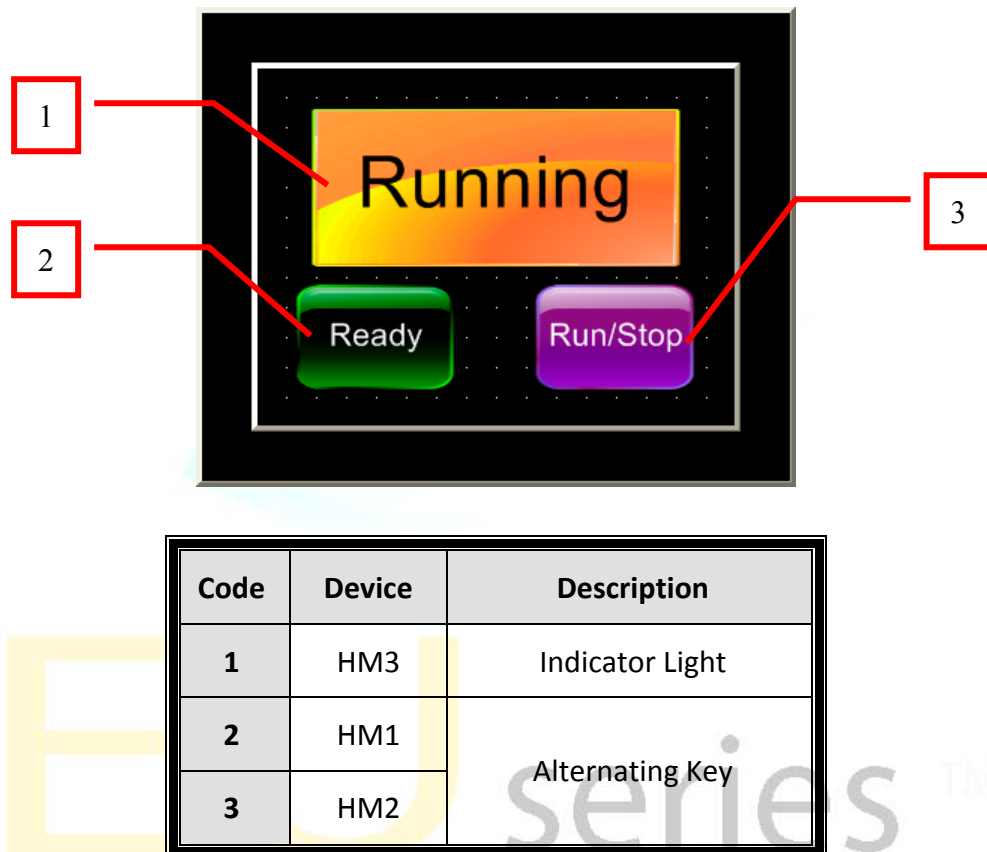





Figure. Component Device Description

Choose  Macro Library... from  of the toolbar to open the Macro Library dialog box. Click  to open the Macro Editor dialog box for editing. Please use the following figure as a reference for macro writing.

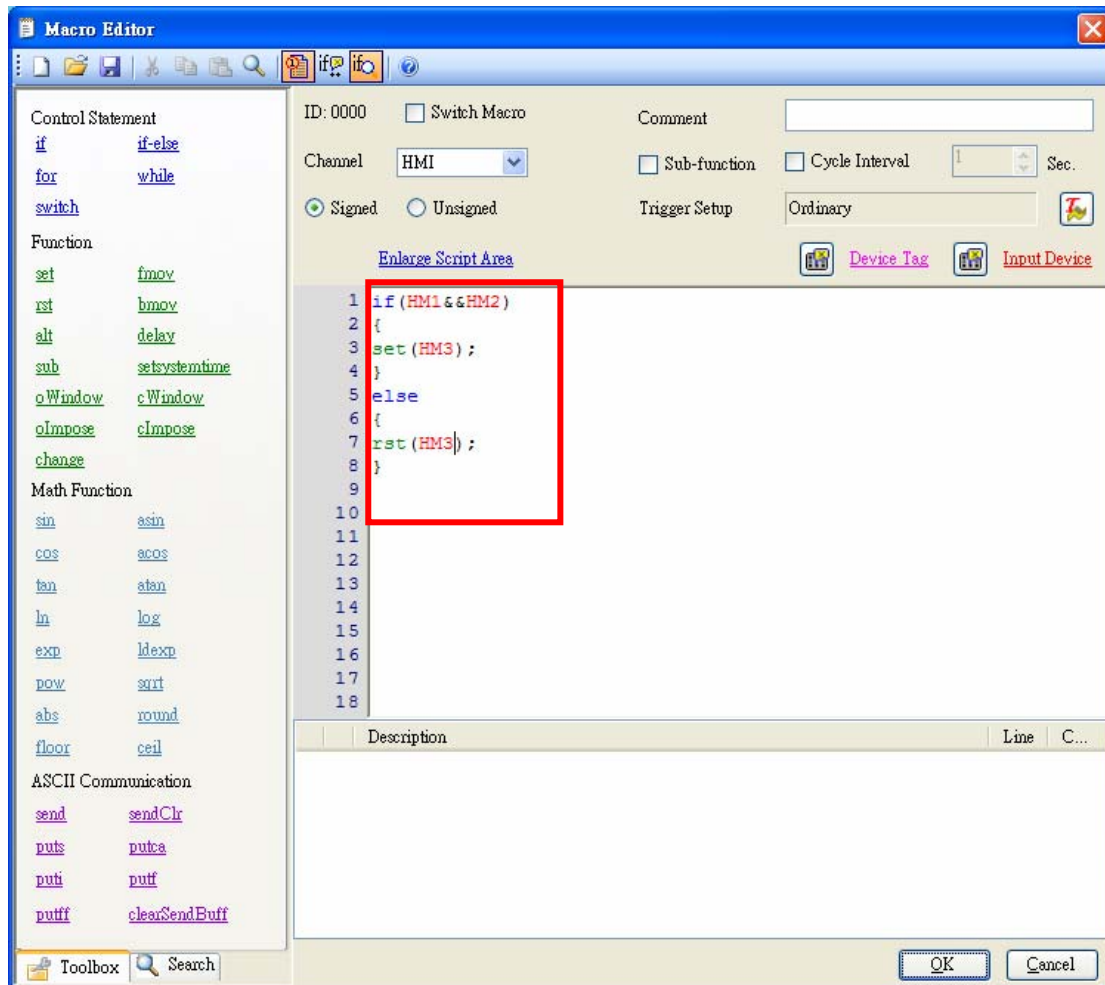


Figure. Macro Editor

When HM1 and HM2 keys are both activated, the indicator light of HM3 will be turned on. Otherwise, the indicator light of HM3 will go off. The procedure flow chart is presented in the figure below.

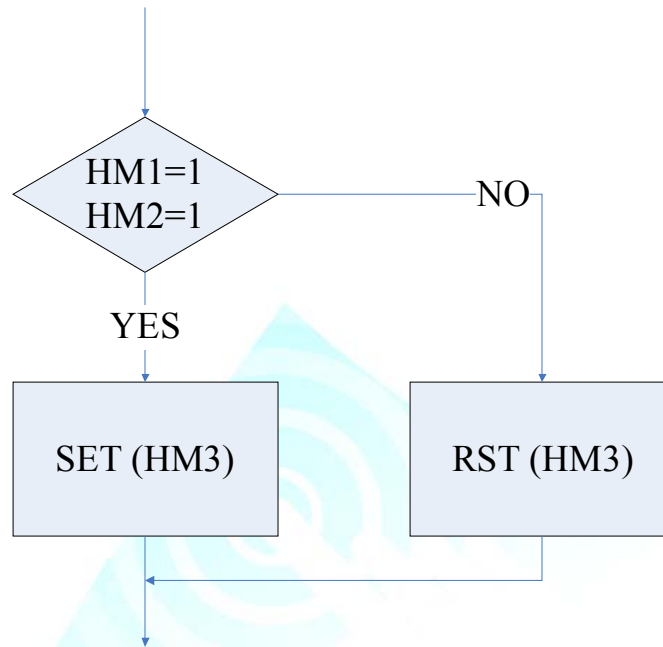
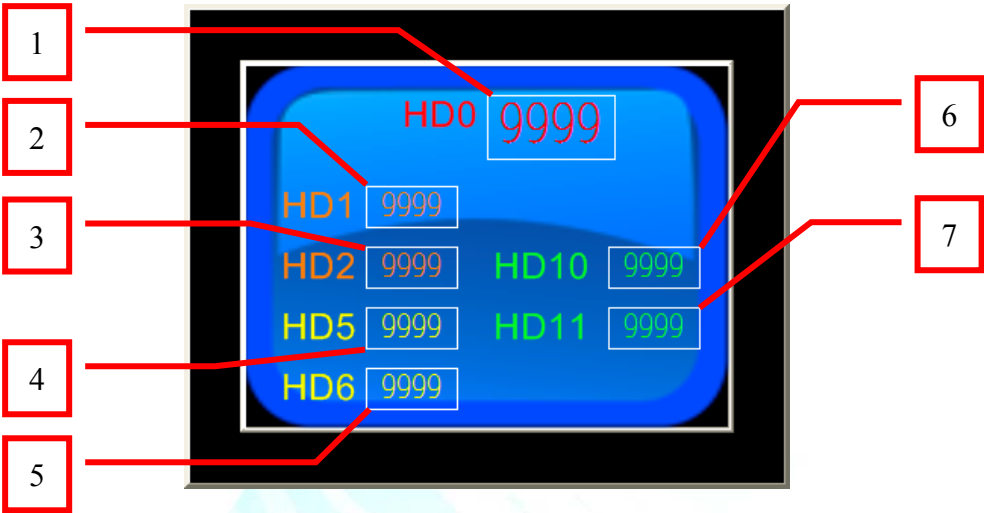


Figure. The Procedure Flow Chart

EU series™


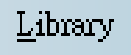
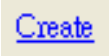
Criteria Selection for Export

Open EU Editor 2.0 Software and create the following components to obtain criteria selection export functions. Read the following Table for more details.



Code	Device	Description
1	HD0	Data Input
2	HD1	Data Output
3	HD2	
4	HD5	
5	HD6	
6	HD10	
7	HD11	

Figure. Component Device Description

Choose  Macro Library... from  of the toolbar to open the Macro Library dialog box. Click  to open the Macro Editor dialog box for editing. Please use the following figure as a reference for macro writing.

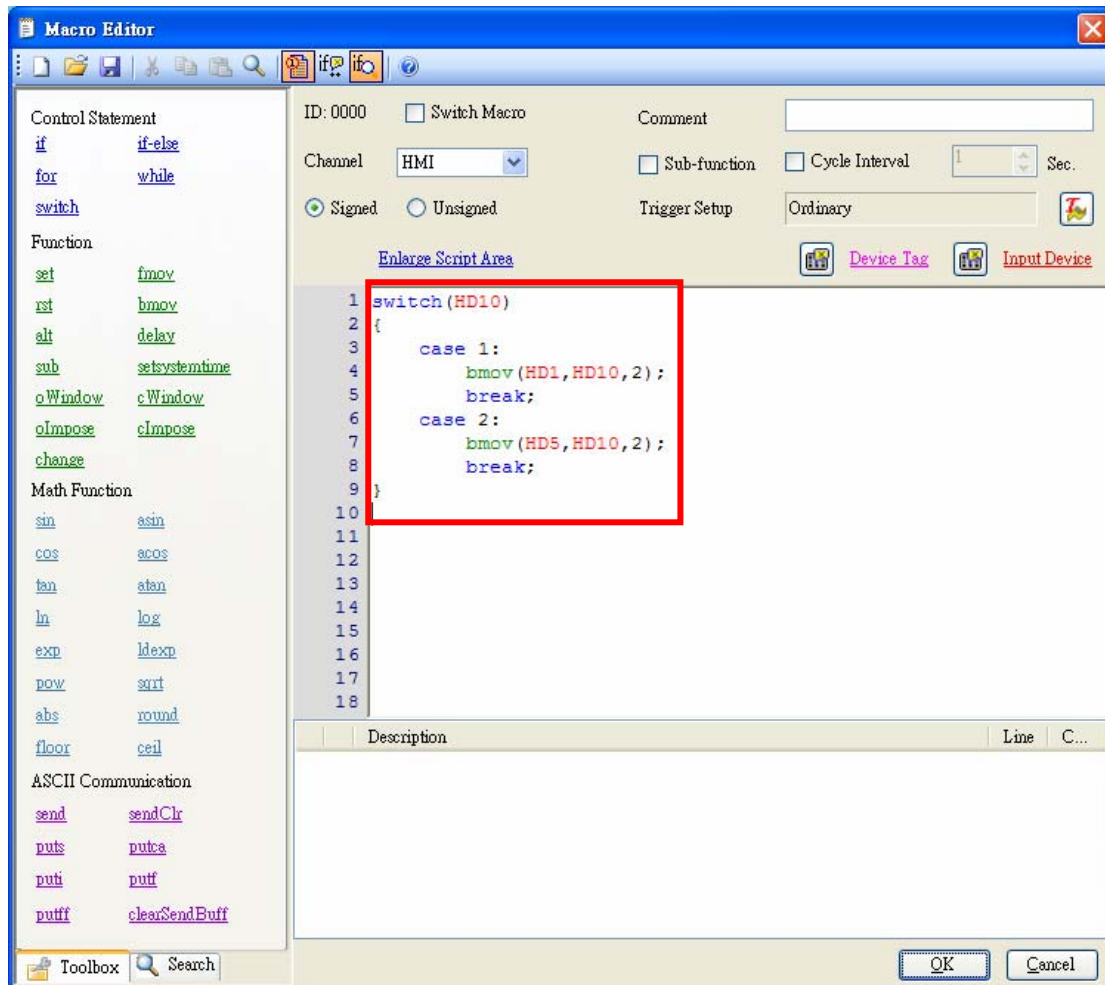


Figure. Macro Editor

When the input value of HD0 is 1, macro `bmov` function would transfer the data in the devices of HD1 and HD2 to HD10 and HD11. When the input value of HD0 is 2, the data in HD5 and HD6 will be transferred to HD10 and HD11. The procedure flow chart is presented as follows.

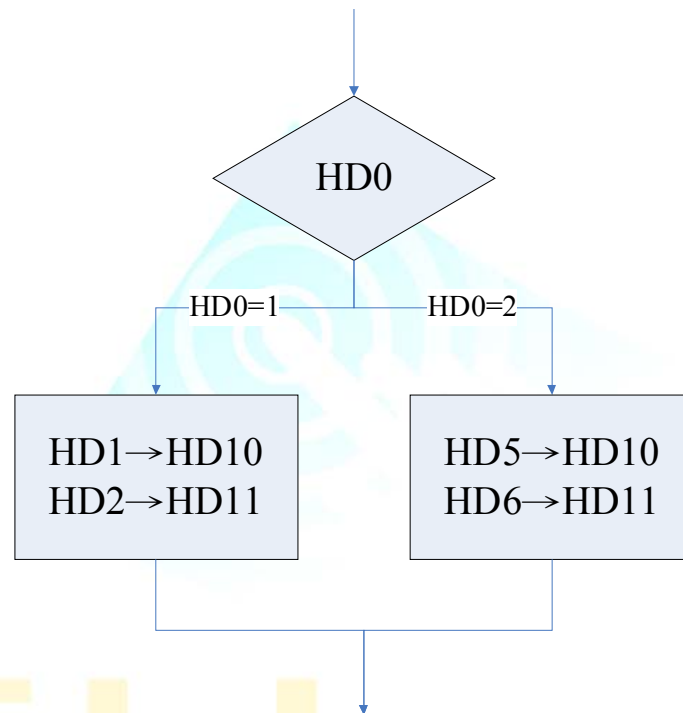
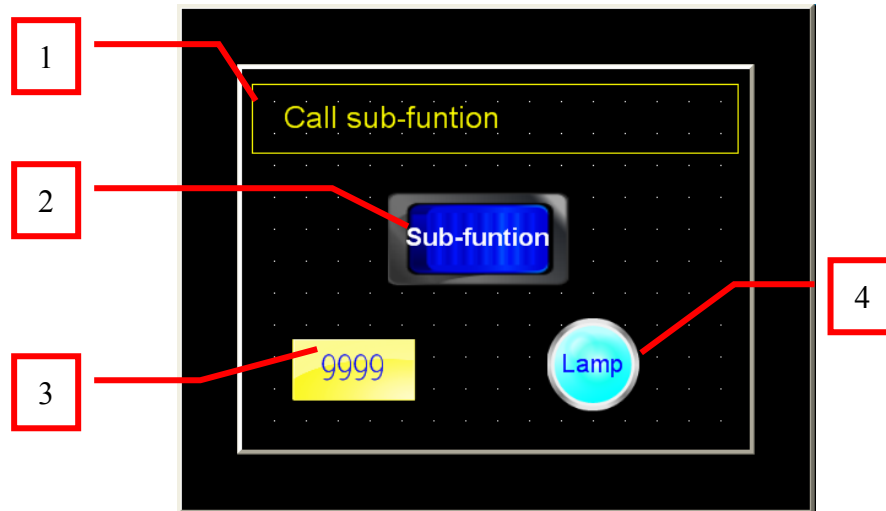


Figure. The Procedure Flow Chart

Call Sub-function

Open EU Editor 2.0 Software and set up the following components in the editing screen to have a subroutine calling function. Read the following table for more details.



Code	Device	Description
1	HM1	Front Message Tickers
2	HM0	Alternating Key
3	HD0	Data Display
4	HM2	Indicator Light

Figure. Component Device Description

Set up 2 macro data in the macro library; one for the main program while the other for the subroutine. See the figure below.

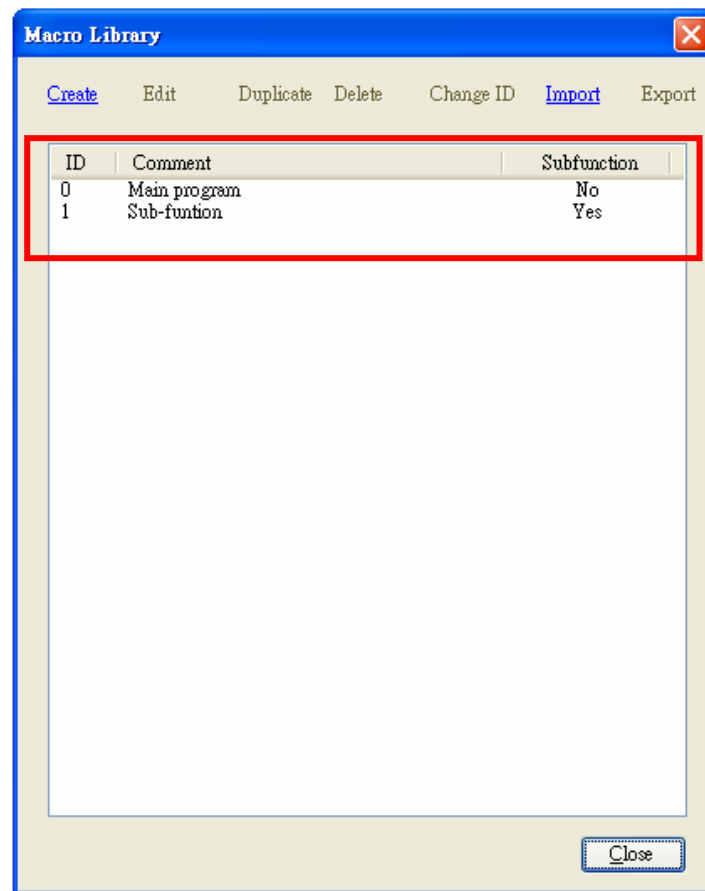





Figure. Macro Library

EU series™

Choose  Macro Library... from  of the toolbar to open the Macro Library dialog box. Click  to open the Macro Editor dialog box for editing. Please use the following figure as a reference for macro writing.

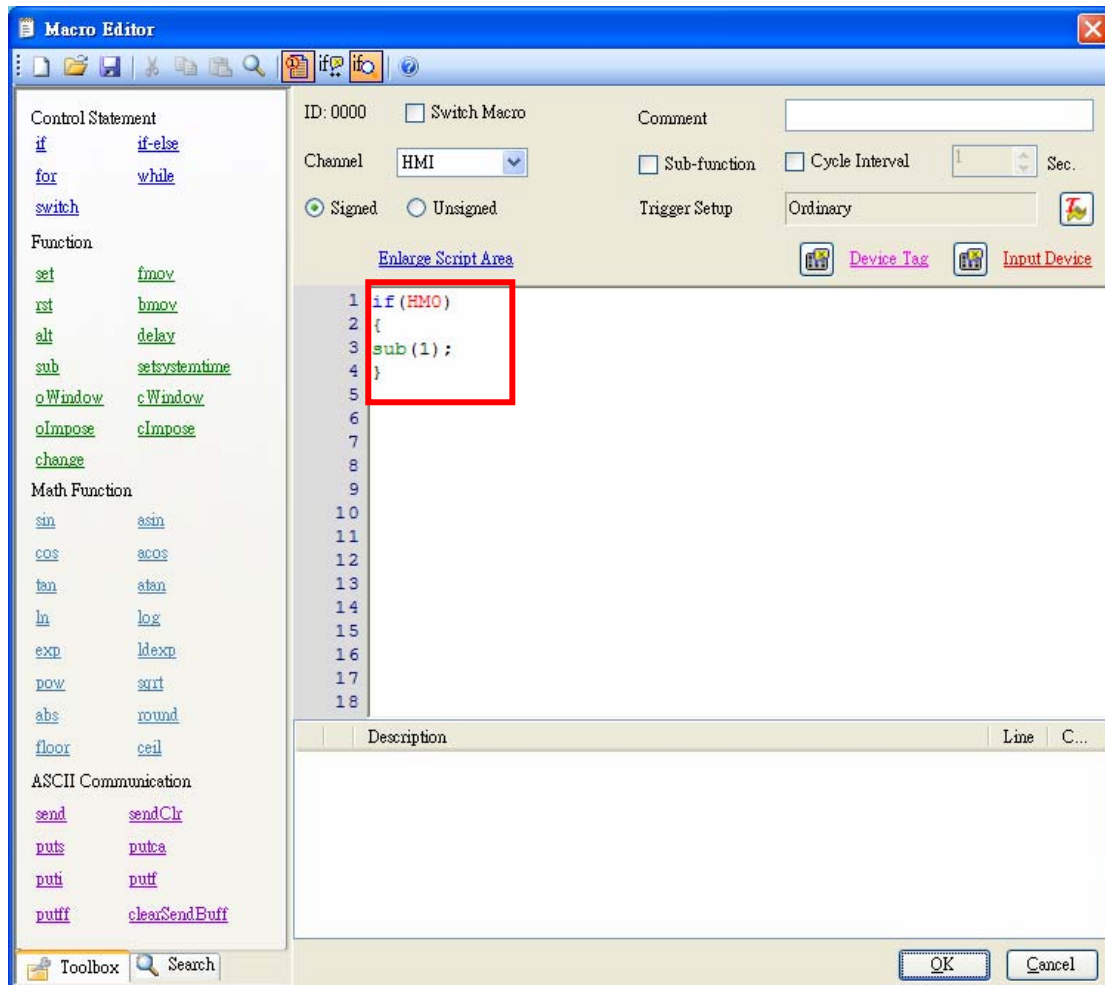


Figure. Main Program

A new set of subroutines is required for run the corresponding action. Select ☒ Sub-function from the Macro Editor to change the macro program to a subroutine. The macro writing is presented below.

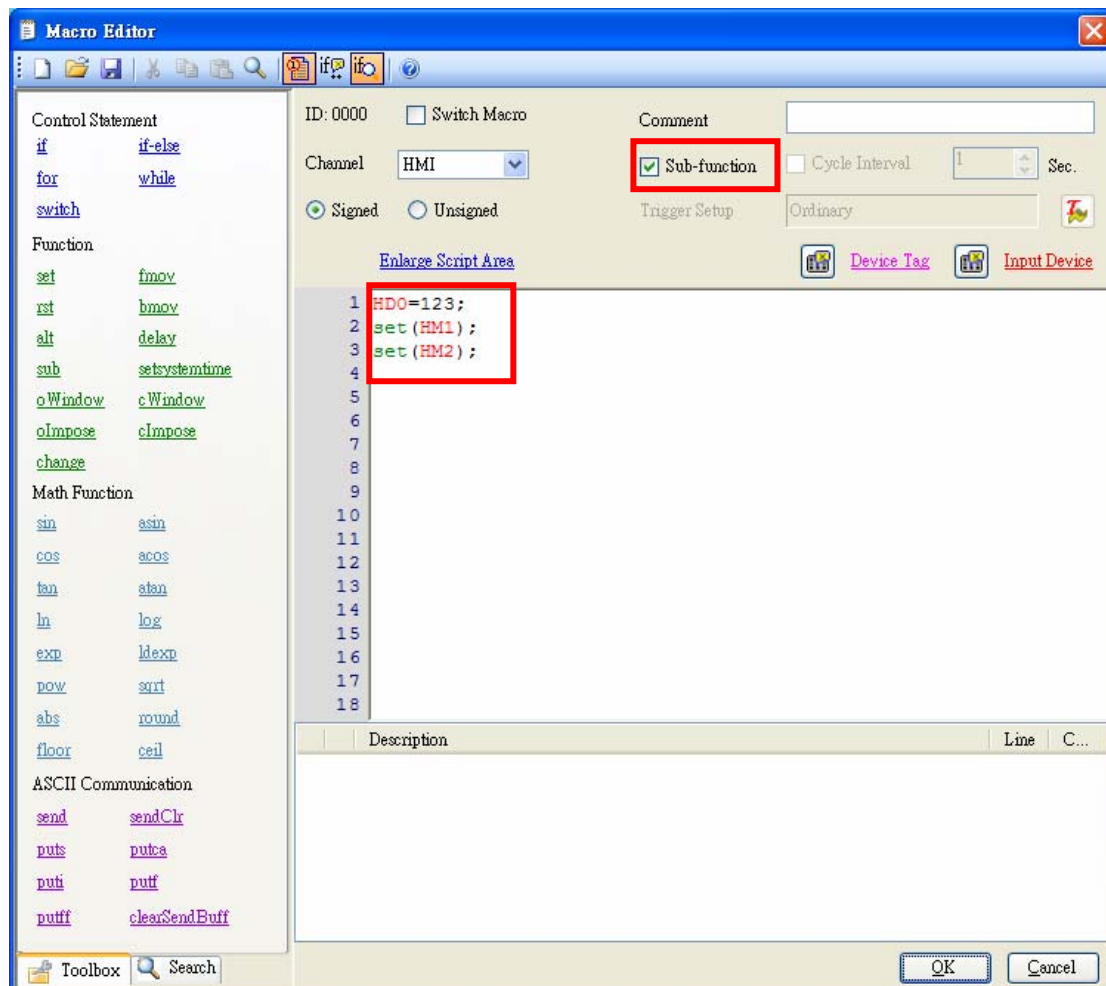


Figure. Subroutine

When pressing the HM0 key, the main program would call for the subroutine. Use `sub` command to call for and to run the subroutine. After run the subroutine, 123 will be written in HD0 device. Also, the front message ticker HM1 and indicator light HM2 will be turned on. The procedure flow chart is presented below.

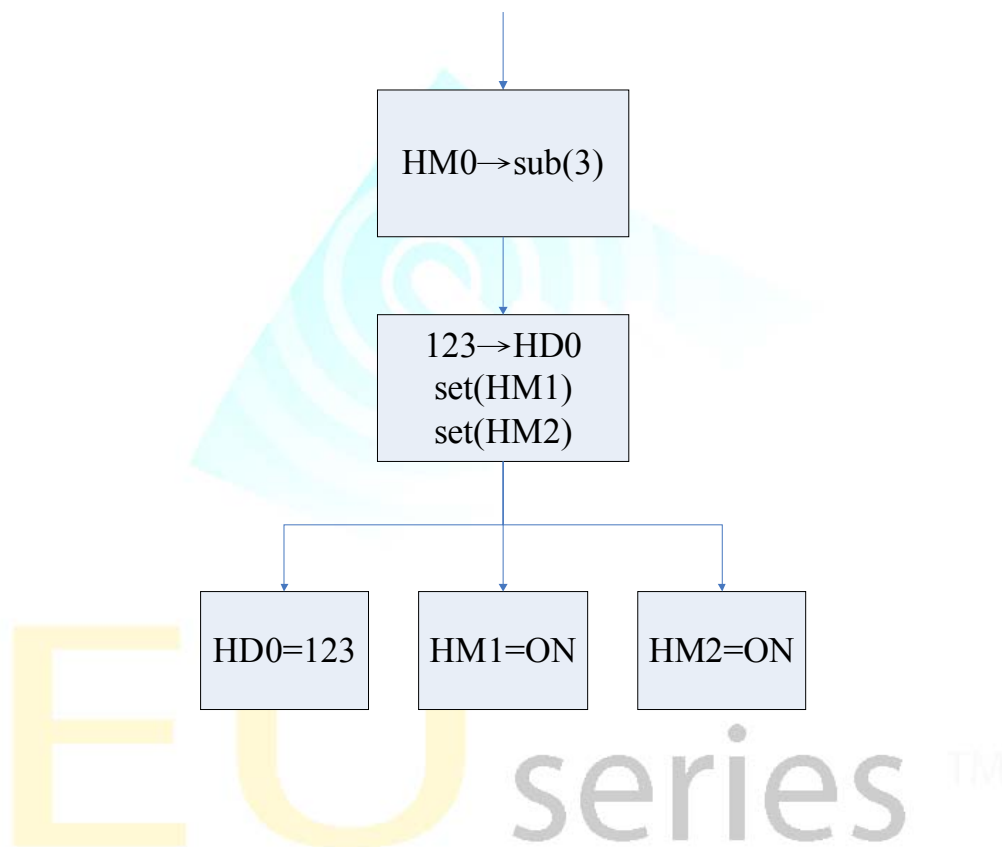


Figure. The Procedure Flow Chart