PLC Programmer for Linux Controller

-Ⅲ series

User Manual

2019/05 Ver：V00.00.006

Leading Numerical Controller

LNC Technology Co., Ltd.

Contents

Version	Date	Editor	Revision
00.00.001		Yu	Draft
00.00.002	2017/01	Yu	Added description for elements of RTimer and Larger than or Equal to/Less than or Equal to/Not Equal to. Fixed the incorrect description for Larger than/Less than elements.
00.00.003	2017/04	Yu	Added I/RmapN and O/RmapN.
00.00.004	2018/05	Yu	Change corporation name.
00.00.005	2019/04	Yu	Added MOD/RET.
00.00.006	2019/07	Yu	Corrected MOD example figure.

1. PLC Programmer for Linux Controller Introduction

This document is to instruct users to familiarize Windows-based LNC PLC Programmer for Linux Controller which can develop customized ladders used in all controller product lines.

PLC module provides several means to interact with other modules and environment within the controller and the outer environment like I/O, timer, counter, functional elements, etc. With those data types, the user can develop sequential control to meet the needs.

The following table briefs the data types that PLC module can access:

Data Type	Name	Size	Description
bool	I	4096	Digital Input----------I0 ~ I4095
	O	4096	Digital Output----------O0 ~ O4095
	A	4096	Internal Address----------A0 ~ A4095
	S	4096	Internal Address (Status)----------S0 ~ S4095
	C	4096	Internal Address(Command)------C0 ~C4095
long	TIMER (TIM)	256	Timer-------TM0 ~ TM255 Types----------1ms，10ms，0.1s，1s
	Counter (CNT)	256	Counter---------CT0 ~ CT255 Types: Up, Down, Ring Up, Ring Down, Reset
	Register	Refer to product line	Retain Type Range:------- Refers to product line

Table 1 PLC Data Types

2. Introduction of the Functionalities

PLC Programmer may have different user interface by the evolving of versions but the following functionalities are supported at the least：

2.1. Ladder in LMLC Format Edit

- **File Operation**
  - Open existing file：3.4.2
  - New file：3.4.3
  - Save(Check and Compiling would be carried out at the same time)：3.6.7
  - Save as：3.6.9
  - Close：3.6.8
  - Encryption/Decryption：3.6.16，3.6.6
  - Print：3.6.10
  - Close Application：3.3
- Edit
  - Elements insertion and set：3.7
  - Copy：3.6.17
  - Cut：3.6.19
  - Paste：3.6.18
  - Delete：3.6.20
  - Delete Vertical Connection：3.6.21
  - Undo：3.6.22
  - Redo：3.6.23
  - Search/Replace：3.9
  - Insert One Line：3.6.24
  - Insert One Space Line：3.6.25
  - Insert Comment Element：3.6.26
  - Insert Label Element：3.6.27
  - Insert Return From Subroutine Element：3.6.28
  - Insert END Element：3.6.29
  - Edit Symbol/Comment for addresses：3.6.11
  - Annotation：3.6.12
- Others
  - Switch to Monitor Mode：3.6.13
  - Switch to Simulation Mode：3.6.14

- Application Setting：3.6.15
- Switch Application User Interface Language：3.6.5
- Application Information：3.6.4

3. PLC Programmer for Linux Controller Operations

3.1. Installation

Execute PLCProgrammerForLinuxController_00.03.xx.xx.xx_Release_Setup.exe, then select the intended installation directory as Fig. 1. There will be a short cut on desktop like  after installation。

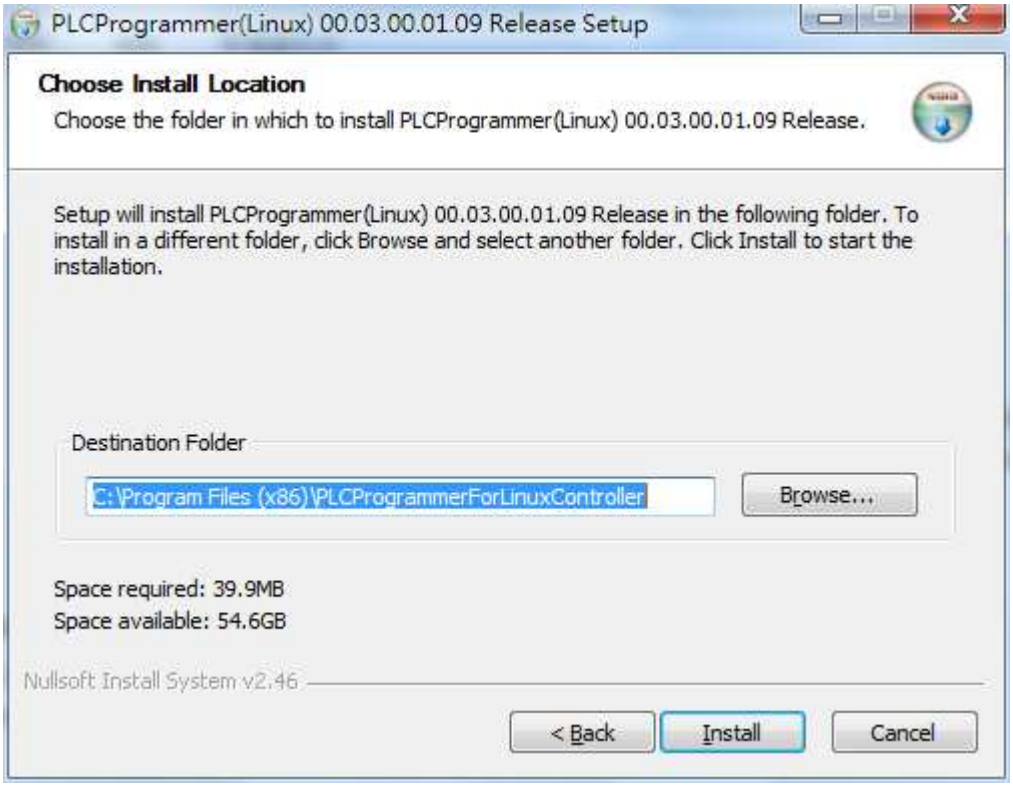


Fig. 1 Installation

3.2. Run Application

There are two ways to execute PLC Programmer for Linux Controller

- Method I

Run QPLCProgrammer.exe from the installation directory, for example C:\Program Files (x86)\ (Installation folder).

- Method II

Use short cut  on desktop。

PLC Programmer for Linux Controller starts as Fig. 2， the application would

adjust the size according to the resolution of the screen. The maximum width of the application is 1366 pixel， and the height would expand to the task bar to provide the maximum working area.

Fig. 2 PLC Programmer Initialization

3.3. Start Layout

Fig. 3 Introduction of the Start Layout

The application only showed the functionalities that are suit for the current stage and will display corresponding functions after opening file. Move the cursor to the function icon would show tip string about its functionality.

① Functions needed for all occasions：

 ：Open an existing file and start edit session.

 ：Create a new file and start edit session.

 ：Provide application information on versions.

 ：Switch the application user interface language.

 ：Decrypt an encrypted file.

② Applicatiion Operation

 ：Close the application.

 ：Minimize the application.

③ Working Area：Show the set homepage at the starting.

Use open or new icons to start an edit session.

3.4. Edit for Ladder in LMLC Format

Programmer supports there mode on ladder in LMLC format, editing, monitoring and simulation. Edit Mode is the default mode to start and exit the application.

3.4.1.Getting Started

Use the following procedure to start the editing。

3.4.2.  Open Existing File

Hot Key：Ctrl+Alt+O

- Click  or use the hot key.
- Select the file format the file to open as Fig. 4.
- Select the language for address symbol/comment editing in the session as Fig. 5. There are already address symbol/comment string files for Traditional Chinese, Simplified Chinese and English.in this example.
- Opening finishes as Fig. 6.

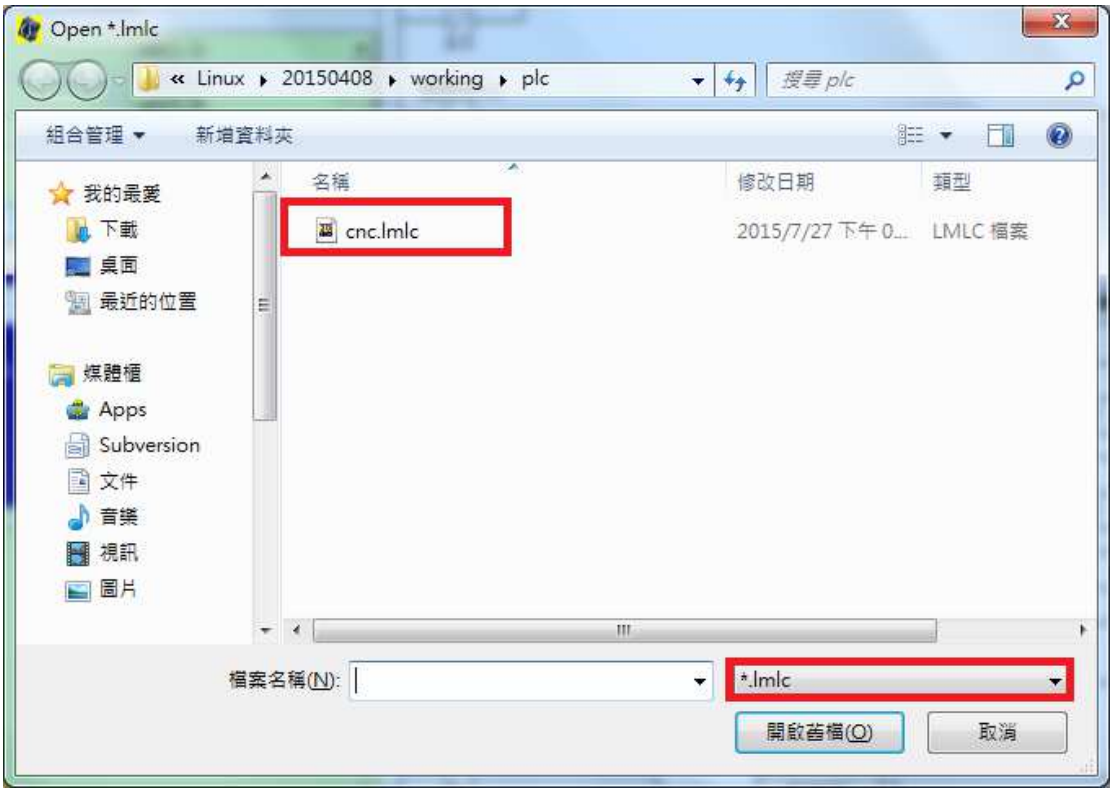


Fig. 4 Open an Existing File

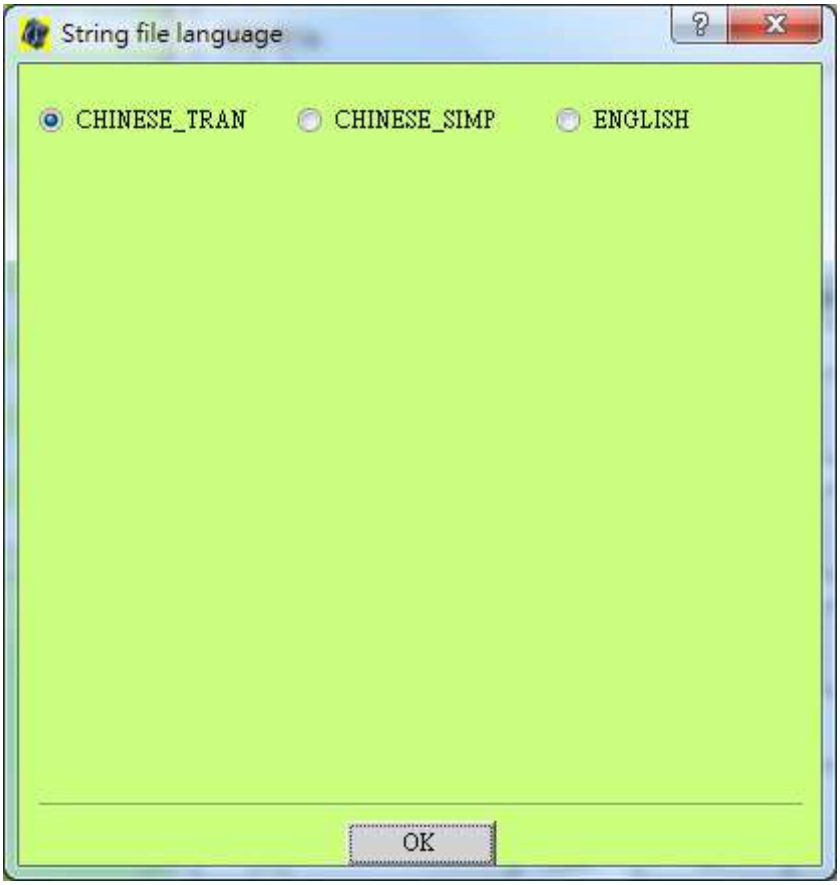


Fig.　5 Select the Language of Address Symbol/Comment

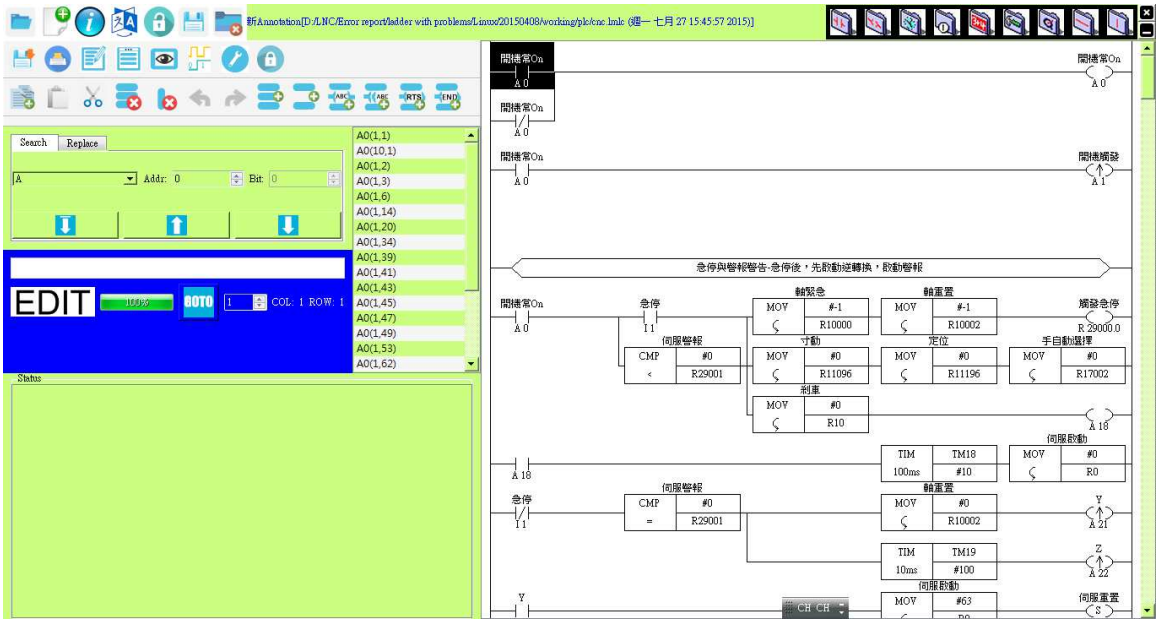


Fig.　6 Finish Opening

3.4.3. New a file

Hot Key：Ctrl+Alt+N

- Click or the hot key。
- Select the language for the address symbol/comment editing as Fig.　7.
- Click OK to new an LMLC file.
- Finish as Fig.　8, user can continue to develop above the END element.

Fig. 7 Select The Language of Address Symbol/Comment

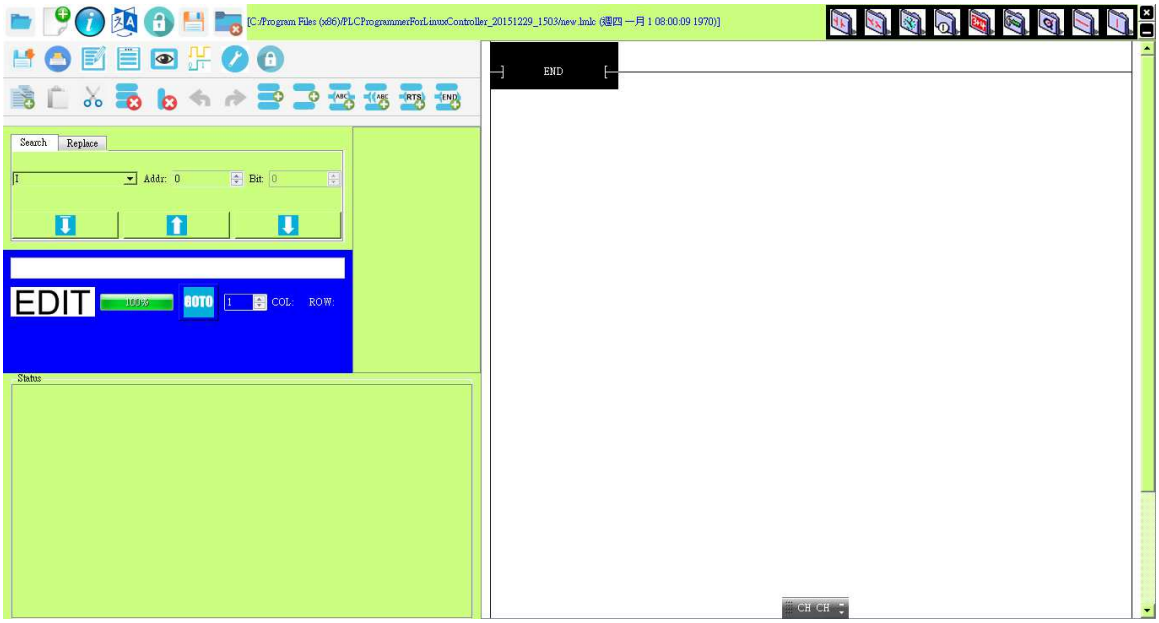


Fig. 8 Finish creating a new LMLC file

3.5. Introduction of Edit Mode for Ladder in LMLC Format

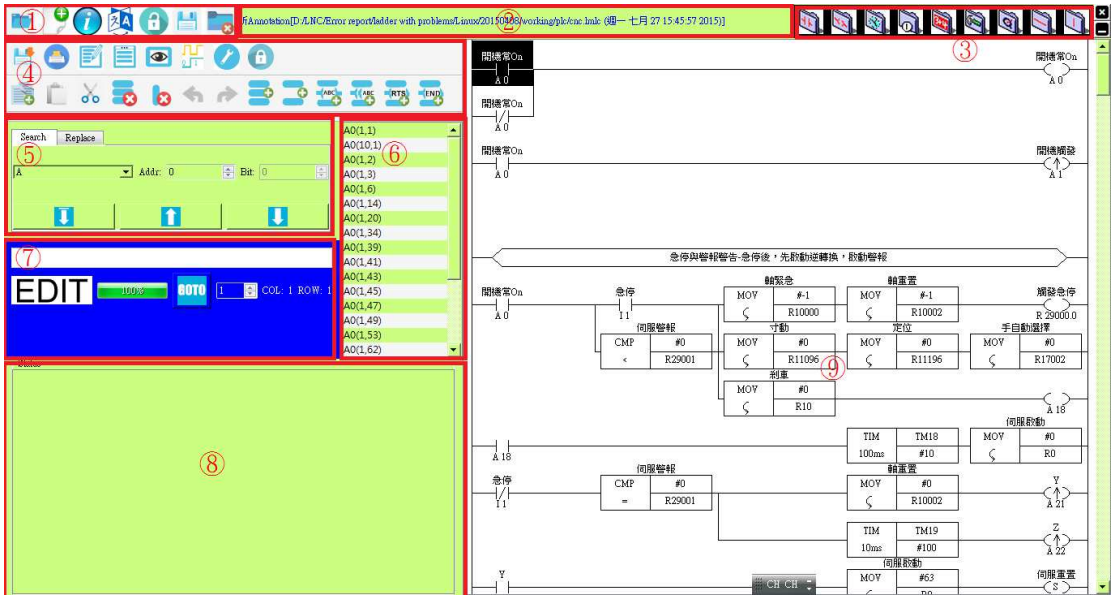


Fig. 9 Edit Mode for Ladder in LMLC Format

The layout of the edit mode is as Fig. 9 to achieve

- Instant access to full functionalities by toolbar icons.
- Larger application output area to let users easily notice the messages from application like compiling errors or warnings.
- Maximize working area.

Introduction to each area as following:

① Common tool bar, two additional toolbar icons in edit mode are as following:

 ：Save file, checking and compiling would be carried out with saving. The icon would change to  after current file being modified.

 : Close file, application would return to the start layout. The user would be reminded by a dialog if the file has been modified.

② The area of displaying the annotation and directory of the opened file.

③ PLC Element Toolbar

 ：Contact class.

 ：Coil class

 ：Timer class

 ：Counter class

 ：Arithmetic class

 ：Tool class

 ：Sequence class

 ：Horizontal connection

 ：Add vertical connection on left

④ Edit Toolbar

 ：Saveas

 ：Print

 ：Pop the dialog of editing address symbol/comment

 ：Pop the dialog of editing annotation

 ：From edit mode to monitor mode

 ：From edit mode to simulation mode

 ：Pop the dialog of application setting

 ：Encrypt the current file

 ：Copy the element on focus or marked area

 ：Paste to the current position with focus

 ：Cut the element on focus or marked area

 ：Delete the element on focus or marked area

 ：Delete the vertical connection on the left

 ：Undo, enable only it is doable

 ：Redo, enable only if it is doable

 ：Insert one horizontal connection line above focus

 ：Insert one empty line above focus

：Insert comment element above focus

：Insert label element above focus

：Insert RTS element above focus

：Insert END element above focus

⑤ Search and replace

⑥ List of the occurrence in the file of the addresses used by the element with focus. Click one item to go the occurrence position.

⑦ Edit status area to show current mode, the comment of the element with focus, current row, column, etc.

⑧ Application output to feedback messages to the users like compiling result.

⑨ Working Area to display the opened ladder.

3.6. Edit Functionalities of Ladder in LMLC Format

3.6.1.Mouse and Keyboard

Mouse left key：

Mark area：Press mouse left key to mark area.

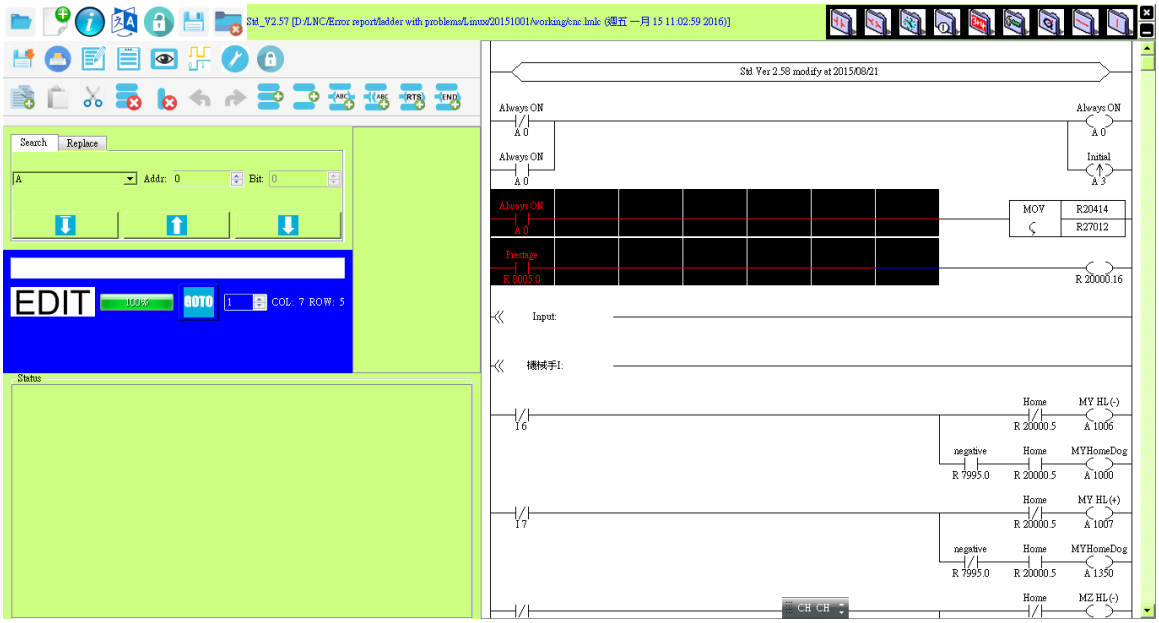


Fig. 10 Marking Area

Mouse right key：

Right key menu：Click the right key to pop up right key menu.

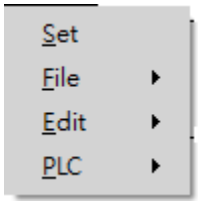


Fig. 11 Right key menu

Mouse wheel：Move focus up and down.

Moving the cursor to a toolbar icon can show the tip for the functionality of the icon.

Fig. 12 Tip for toolbar icon

Up, Down, Left, Right：Move focus up, down, left and right.

Page Up/Down：Move focus up or down one page.

Shift+Up, Down, Left, Right /Page Up/Down：Mark area.
Return/Enter：Pop up mouse right key menu.

3.6.2.Open an Existing File

Refer to 3.4.2.

3.6.3.New

Refer to 3.4.3.

3.6.4.  ：Application Information

Show module version information of the application.

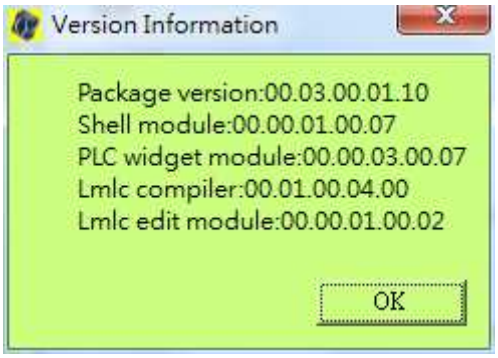


Fig. 13 Application Information

3.6.5.  ：Application Language Selection

Select the language used by the application user interface.

Fig. 14 Select Language of Application User Interface

：This function is not to select the language used by ladder editing like address symbol/comment, string for label/comment elements, etc.

3.6.6. ：Decrypt

Decrypt an encrypted file to the current working directory. Select File as Fig. 15, ask user to input password as Fig. 16 and finish decryption as Fig. 17.

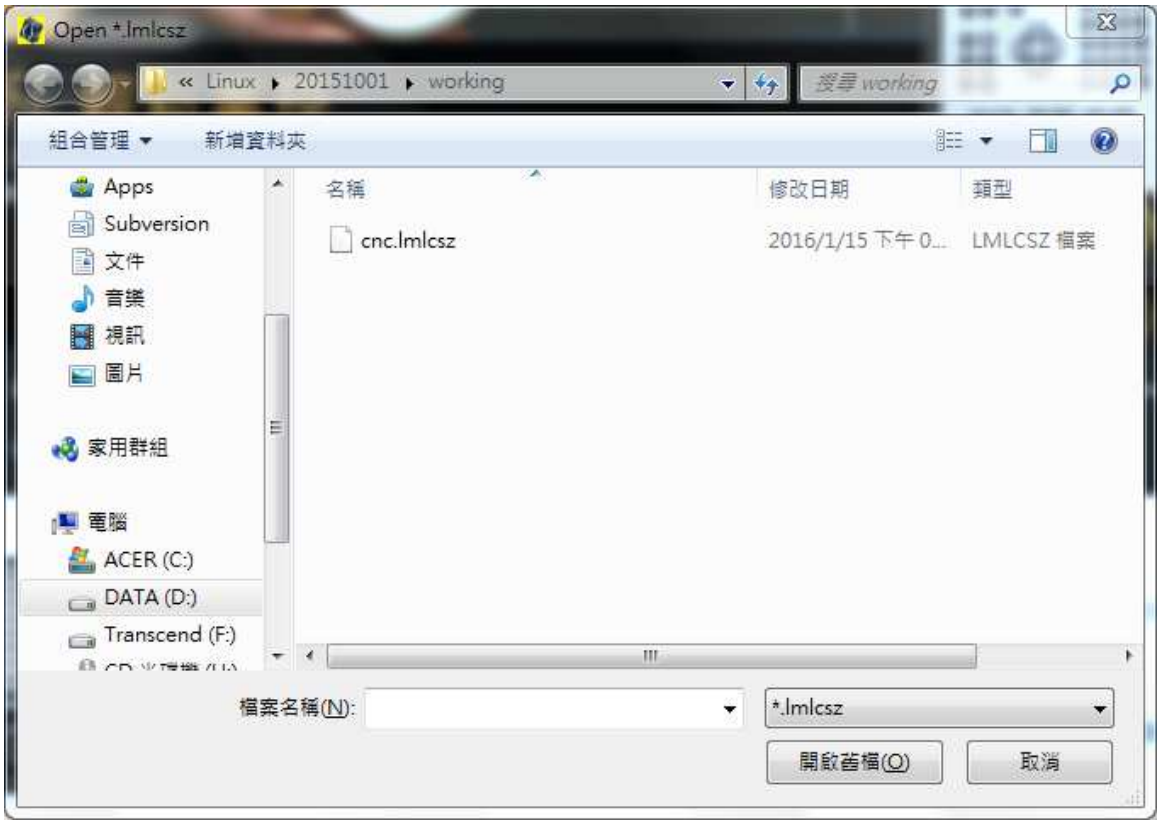


Fig. 15 Select the Encrypted File

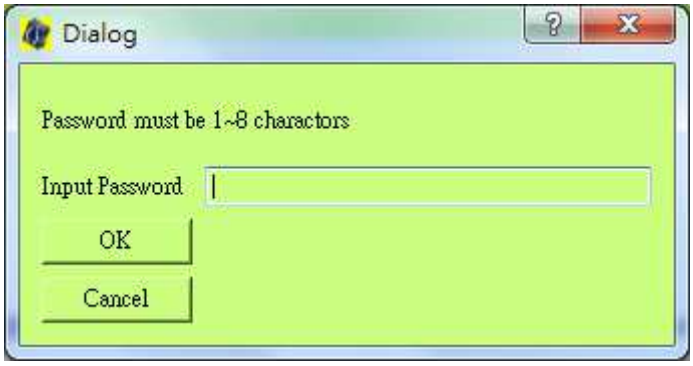


Fig. 16 Ask User to Input Password

Fig. 17 Finish Decryption

: The file with the same name of the one in the encrypted file in the working folder will be overwritten.

3.6.7. Save

Hot Key：Ctrl+Alt+S

Save the currently opened file and do check and compiling. The result message would be showed on application output area as Fig. 18. Click an message item would lead to the position which is related to the message.

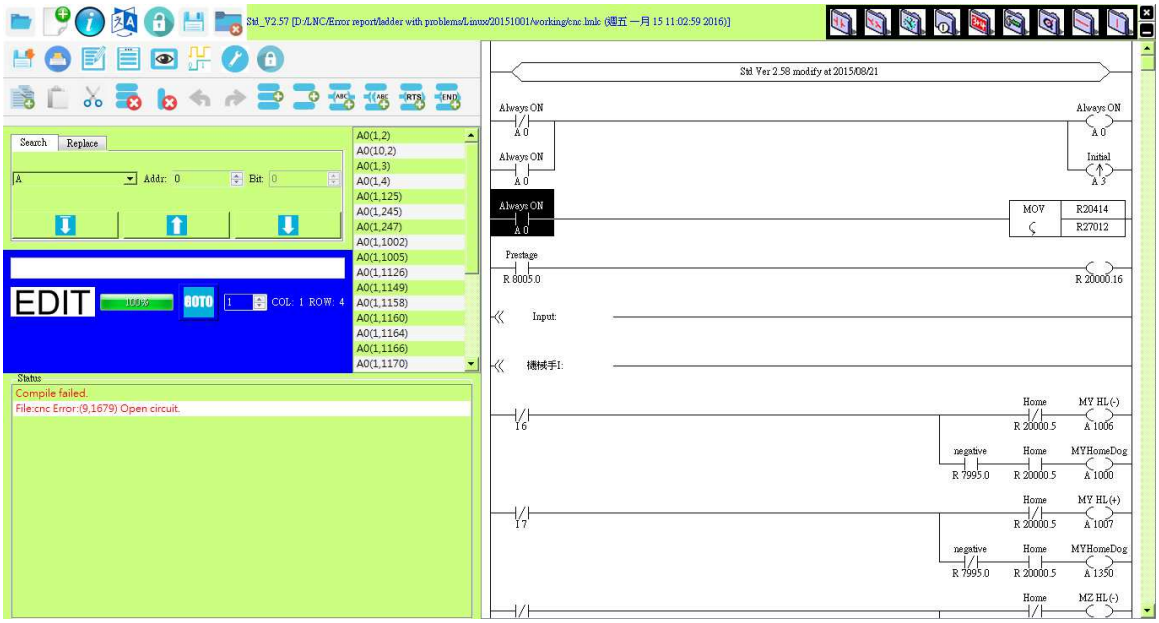


Fig. 18 The result message after saving

The following files would be generated after a successful compiling.

cnc.lmlc：Ladder file.

cnc.lcod：Binary executable file.

cnc.lpar：Parameter file come with the binary executable file.

cnc_plc_xxxx.str and cnc_plc_xxxx_utf8.str：Address symbol/comment string file, xxxx are four digits and represent the language index used by the string file.

0000=Traditional Chinese	0011=Australia	0022=German	0033=Greek
0001=Simplified Chinese	0012=Turkey	0023=French	0034=Icelandic
0002=English	0013=Afghanistan	0024=Belgium	0035=Hungary
0003=Japanese	0014=Argentina	0025=Switzerland	0036=Norwegian
0004=Korean	0015=Peru	0026=Sweden	0037=Czech
0005=Thai	0016=Chile	0027=Italian	0038=Ukrainian
0006=Indonesian	0017=Mexico	0028=Russian	0039=Romanian
0007=Vietnamese	0018=Costa Rica	0029=Holland	0040=Hebrew
	0019=Panama	0030=Polish	0041=Denmark
	0020=Spain	0031=Finland	

0008=Indian 0009=Malaysian 0010=Philippine	0021=Canada	0032=Portuguese	

Table 2 Language Index

3.6.8. Close File

Hot Key：Ctrl+Alt+C

Close the currently opened file and remind the user if the file has been modified.

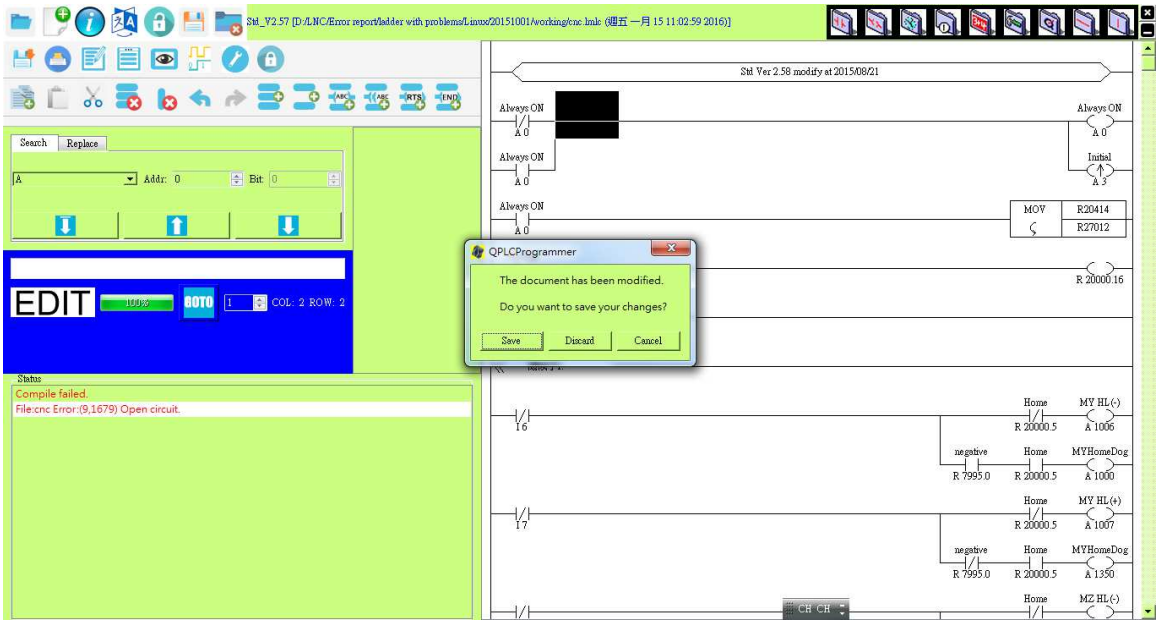


Fig. 19 Reminding of the Modified File

3.6.9. Save as

Hot Key：Ctrl+Alt+A

Save the currently opened file to another file and open the newly saved file.

：The address symbol/comment string files would not be automatically saved to the new file's name.

3.6.10. Print

Multi-select the print options (address symbol/comment and ladder.)

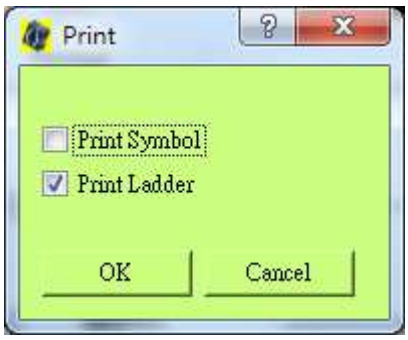
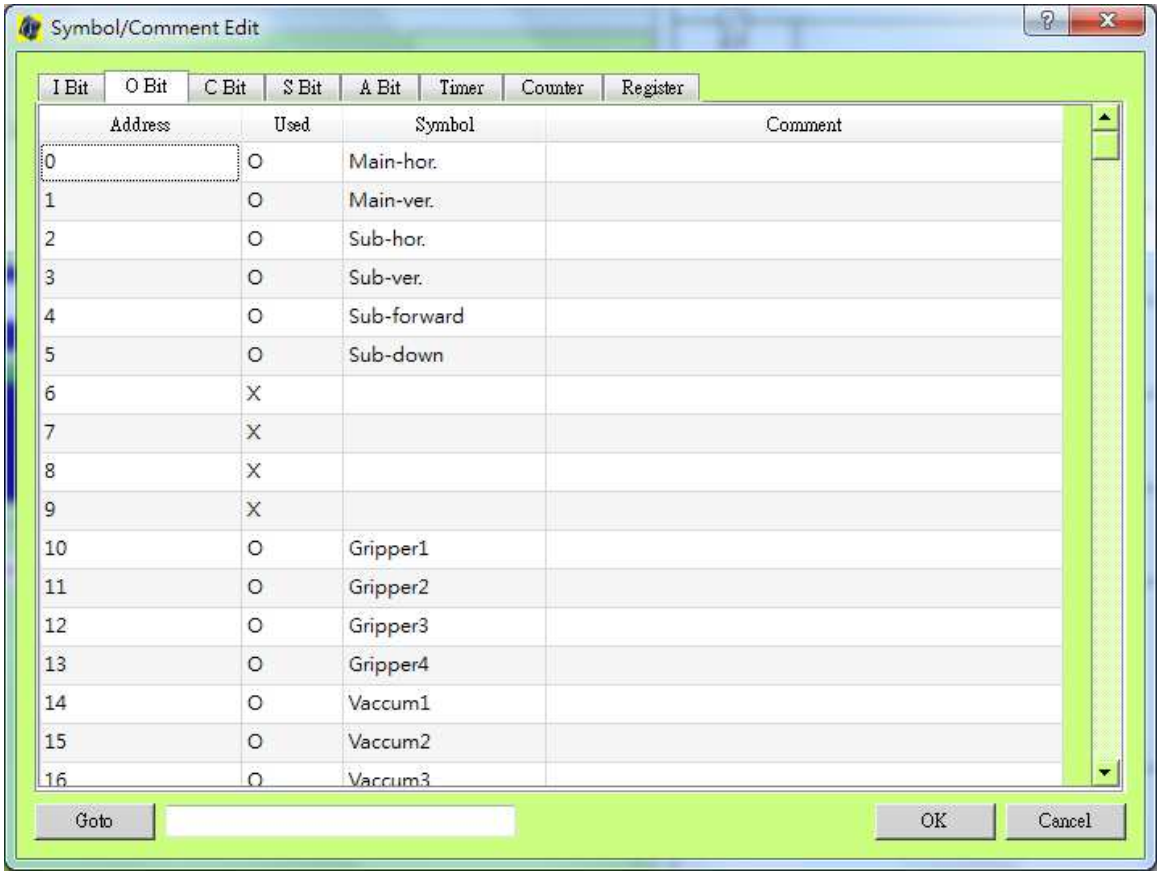


Fig. 20 Select Print Option

3.6.11. ：Pop Address Symbol/Comment Editing Dialog

Click to pop up address symbol/comment editing dialog as Fig. 21. The address that has been used in the ladder would be marked with ◯. The user can write symbol and comment on I, O, C, S, A, Timer, Counter, Register, Register Bit to note the purpose of the usage. The symbol would be shown on the top of the ladder element when the element is able to do so. The comment which set by the element with focus would be shown to the edit status area as Fig. 22.



Address	Used	Symbol	Comment
0	O	Main-hor.	
1	O	Main-ver.	
2	O	Sub-hor.	
3	O	Sub-ver.	
4	O	Sub-forward	
5	O	Sub-down	
6	X		
7	X		
8	X		
9	X		
10	O	Gripper1	
11	O	Gripper2	
12	O	Gripper3	
13	O	Gripper4	
14	O	Vaccum1	
15	O	Vaccum2	
16	O	Vaccum3	

Fig. 21 Address Symbol/Comment Editing Dialog

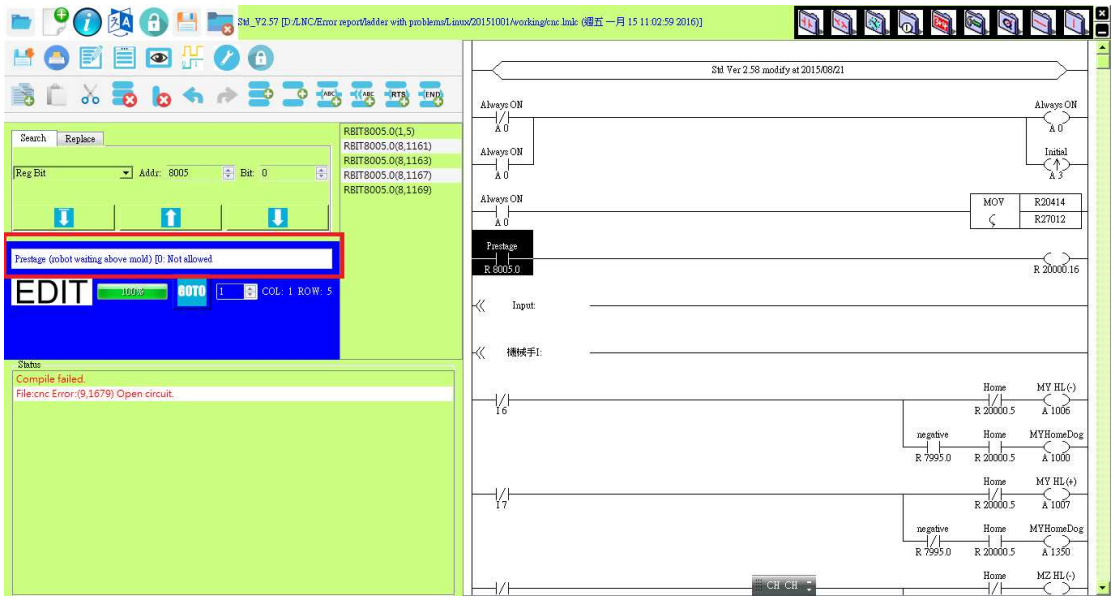


Fig. 22 The Area of Displaying Comment

: The modified string would be written to the string file only OK button is clicked.

: PLC Alarm/Warning String Editing

The user can assign PLC alarm string by writing comment for register bit ranging from R29000 to R29049 and PLC warning string by writing comment for register bit ranging from R29050 to R29099. The controller would show the corresponding PLC alarm/warning string after the string files being uploaded. Firstly select the register editing page then expand the edit field for register bit symbol/comment editing as Fig. 23. The register bit edit mode is as Fig. 24.

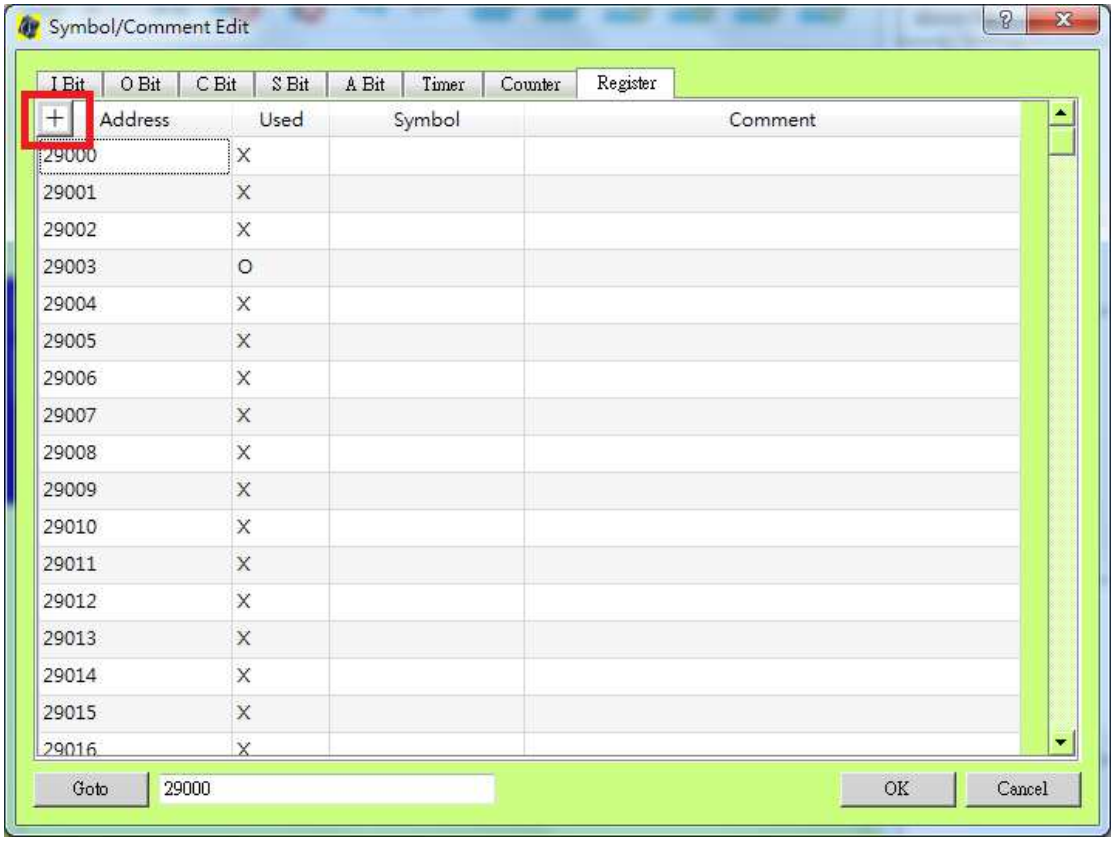


Fig. 23 Expand Register Bit

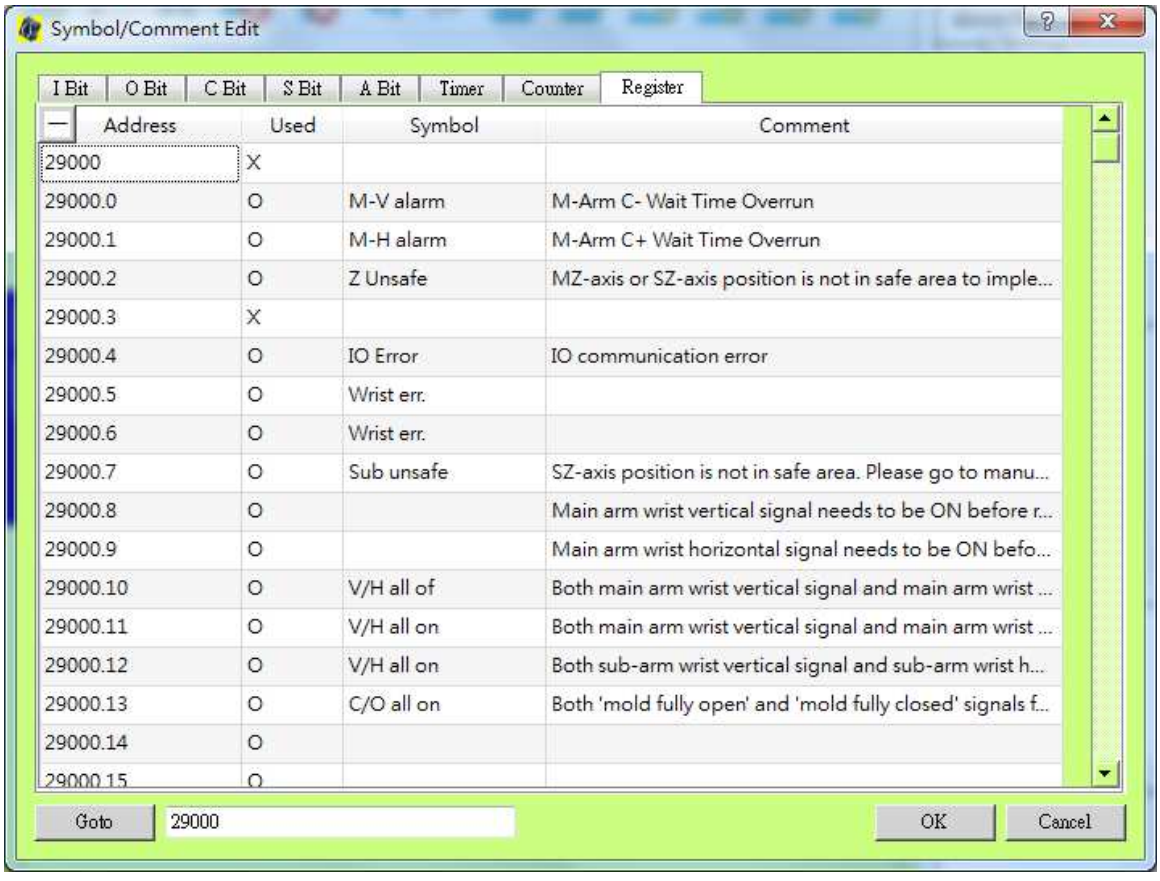


Fig.　24 Register Bit Symbol/Comment Editing

3.6.12.  ：Pop Annotation Dialog

Annotation dialog is as Fig.　25. The maximum allowed string is to 80 bytes.

Fig.　25 Annotation Dialog

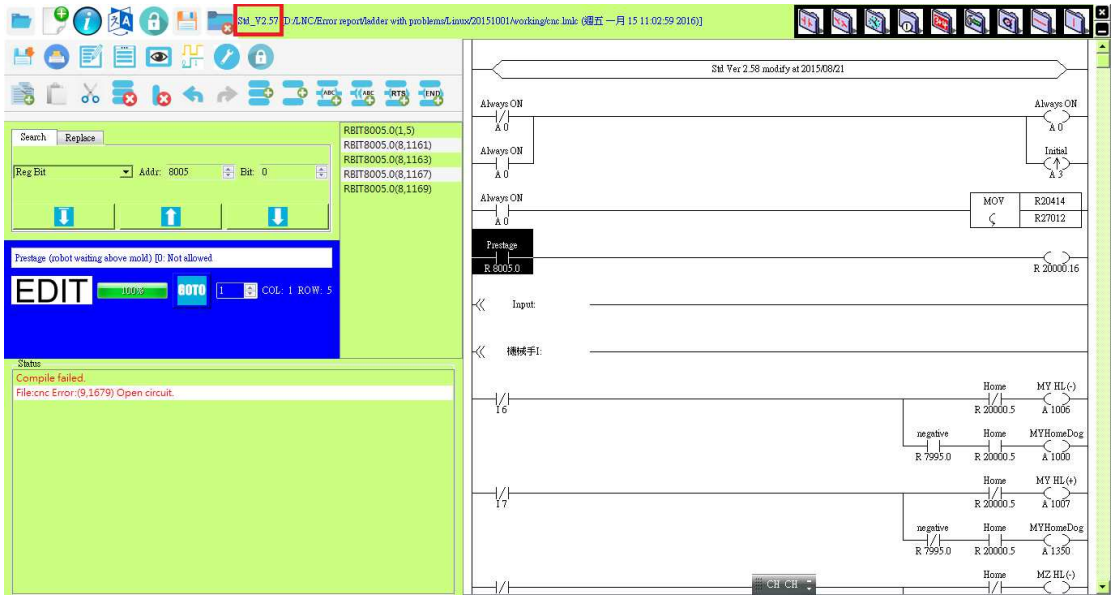


Fig.　26 Annotation Displaying

3.6.13.  ：From Edit Mode to Monitor Mode

Click to pop up the connection dialog, use Detect to list the controllers on which connection is possible as Fig.　27. Select the controller in the list and click connect to build the connection. If the connection can be built then the application would change to monitor mode as Fig.　28, otherwise the application remains in edit mode. The functionalities of the monitor mode refer to chapter 6 **錯誤! 找不到參照來源。**.

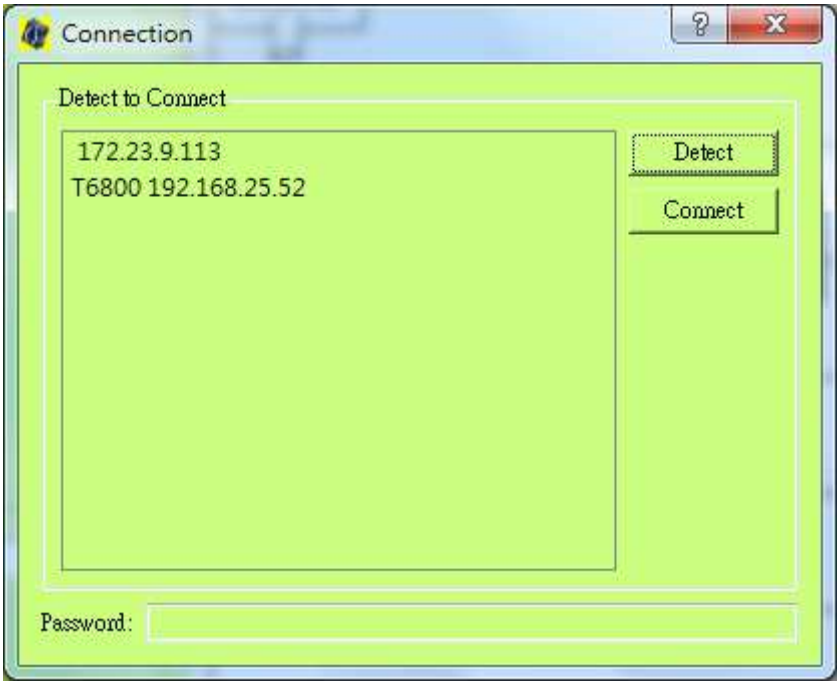


Fig. 27 Connection Dialog

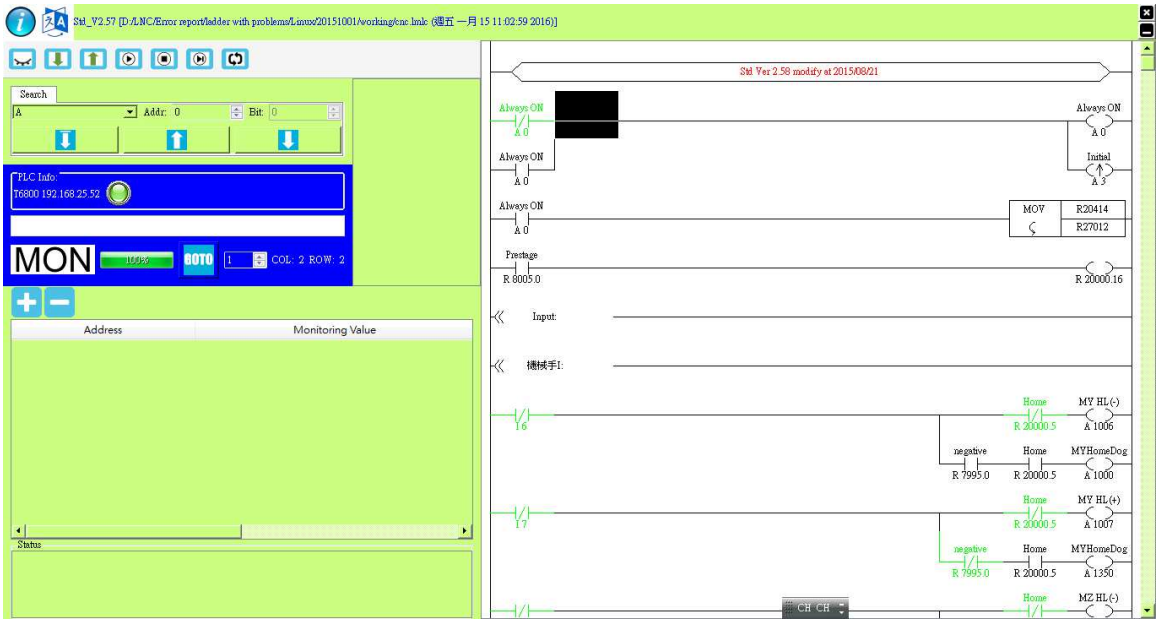


Fig. 28 Monitor Mode

：The opened file must be the same version with the one running in the controller to reflect correct monitor result.

3.6.14. ：From Edit Mode to Simulation Mode

Before entering simulation mode, the application does saving, checking and

compiling. The application would enter simulation mode as Fig. 29 if the compiling is successful or remain in edit mode. The functionalities of the simulation mode refer to chapter 5.

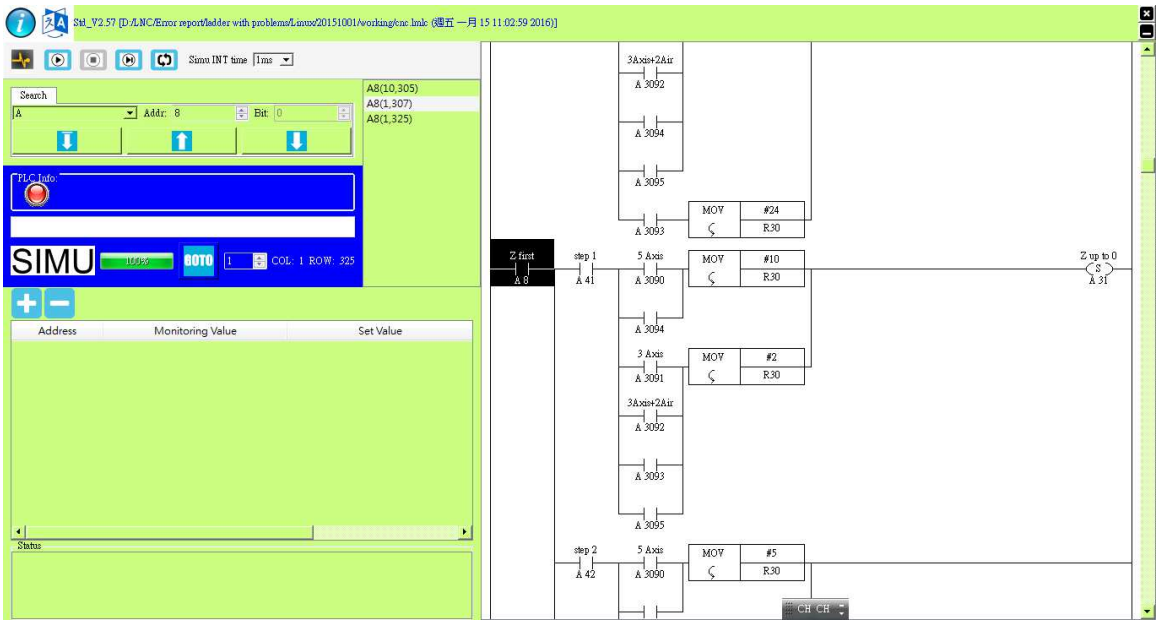


Fig. 29 Simulation Mode

3.6.15. ：Application Setting

Application setting dialog is as Fig. 30. The user can use it to customize the application environment.

Fig. 30 Application Setting Dialog

Color Setting is to set:

- Ladder background color, the background color for the element with focus is the inverted color.
- Element color without focus, the element color with focus is the inverted color.
- Element color when is active in monitor/simulation mode. The inactive color is the inverted color.
- Element color for being marked in edit mode.

Example: The following figures show the effect of setting ladder color black, element color without focus white, the element active color green and the marked color red.

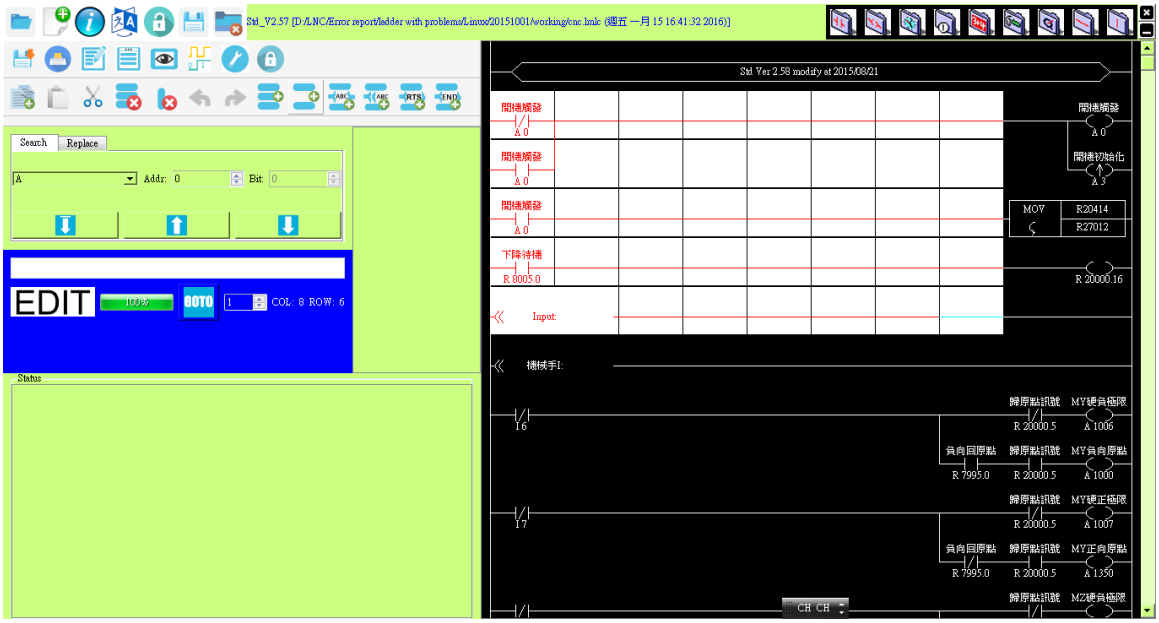


Fig. 31 Edit Mode

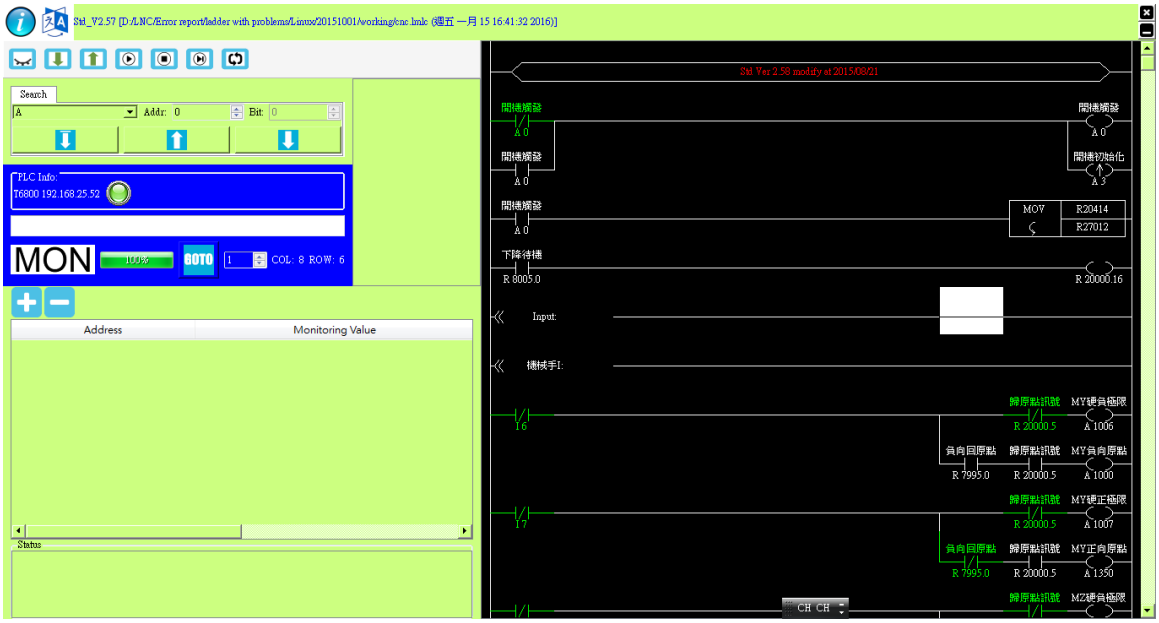


Fig. 32 Monitor Mode

3.6.16.  : Encrypt the Currently Opened File

This function is to encrypt the currently opened LMLC file. The password input dialog is as Fig. 33. The password must be 1 to 8 characters long and the characters must be in the set of a to z, A to Z and 0 to 9. Choosing the directory into which the encrypted files would be put as Fig. 34. The application would do save, check and compiling after the user confirm the directory. The encryption would take place if the compiling is successful and

the encrypted files, which are *.lmlcsz and *.lcodsz, would be put into the chosen directory as Fig. 35.

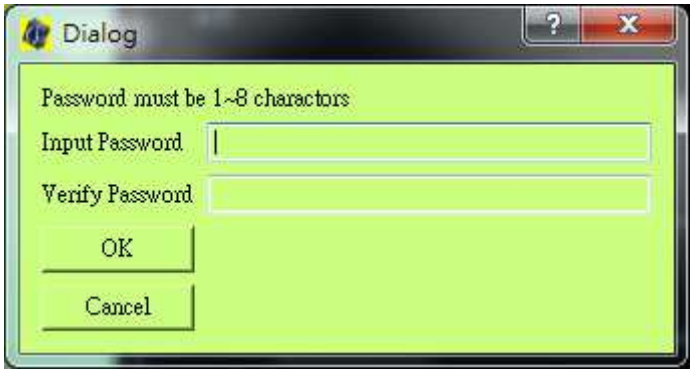


Fig. 33 Password Input

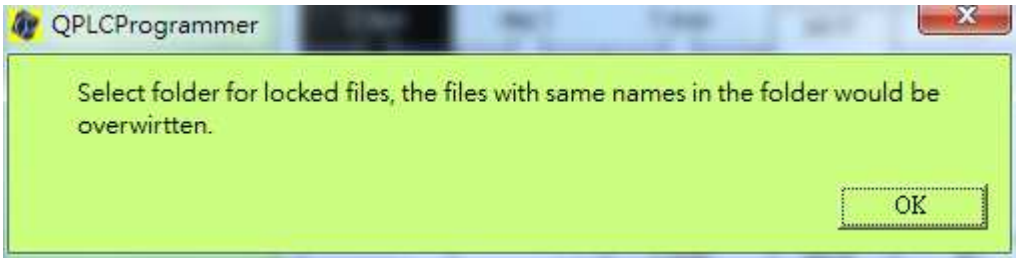


Fig. 34 Selecting Directory

Fig. 35 Encrypted Successfully

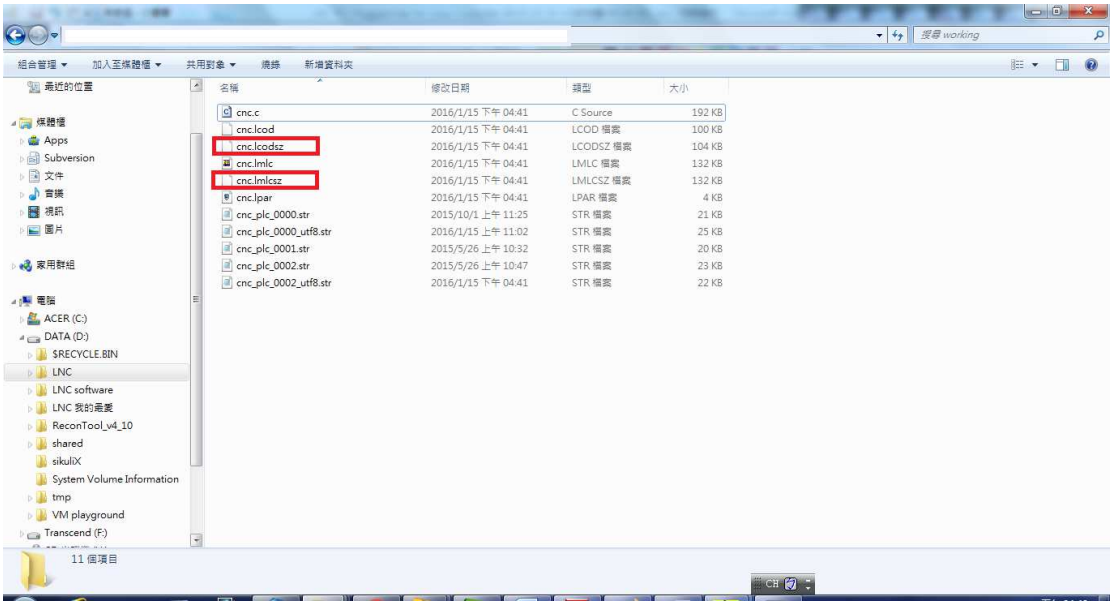


Fig. 36 The Encrypted Files

：Address symbol/comment string files would not be encrypted.

：This functionality needs to work with the controller. Check if the controller side supports the functionality before using.

3.6.17.  ：Copy Element or the Marked Area

Hot Key：Ctrl+C

Copy the element with focus or the marked area to system clipper board.

3.6.18.  ：Paste

Hot Key：Ctrl+V

Paste the copied elements in system clipper board to the current focus position.

：The action would not be taken if the current focus position does not allow. For example, try to paste across a comment element.

：If the copied area is in complete rows, pasting is to insert those rows above the focus.

3.6.19.  : Cut Element or the Marked Area

Hot Key：Ctrl+X

Cut the element with focus or the marked area to system clipper board.

3.6.20.  : Delete Element or the Marked Area

Hot Key：Del

Delete the element with focus or the marked area to system clipper board.

 ：The action would not be taken if the current focus position does not allow.

For example, try to delete the line elements on the row which contains END element.

3.6.21.  : Delete the Vertical Connection on the Left Side

Hot Key：Shift+Del

Delete the vertical connection on the left side of the current focus position as Fig. 37.

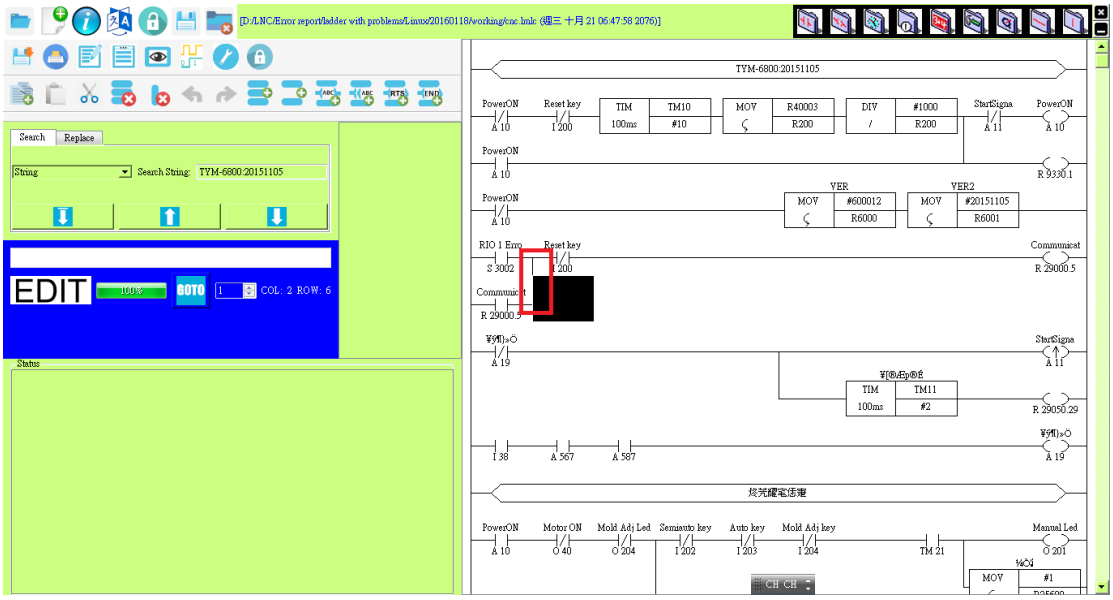


Fig. 37 Delete the Vertical Connection

3.6.22. ：Undo

Hot Key：Ctrl+Z

Cancel the last edit action.

：There is undo number limit.

3.6.23. ：Redo

Hot Key：Ctrl+Y

Redo the last canceled edit action.

：There is redo number limit.

3.6.24. ：Insert a Line Row above Focus

Insert a line row above focus as Fig. 38.

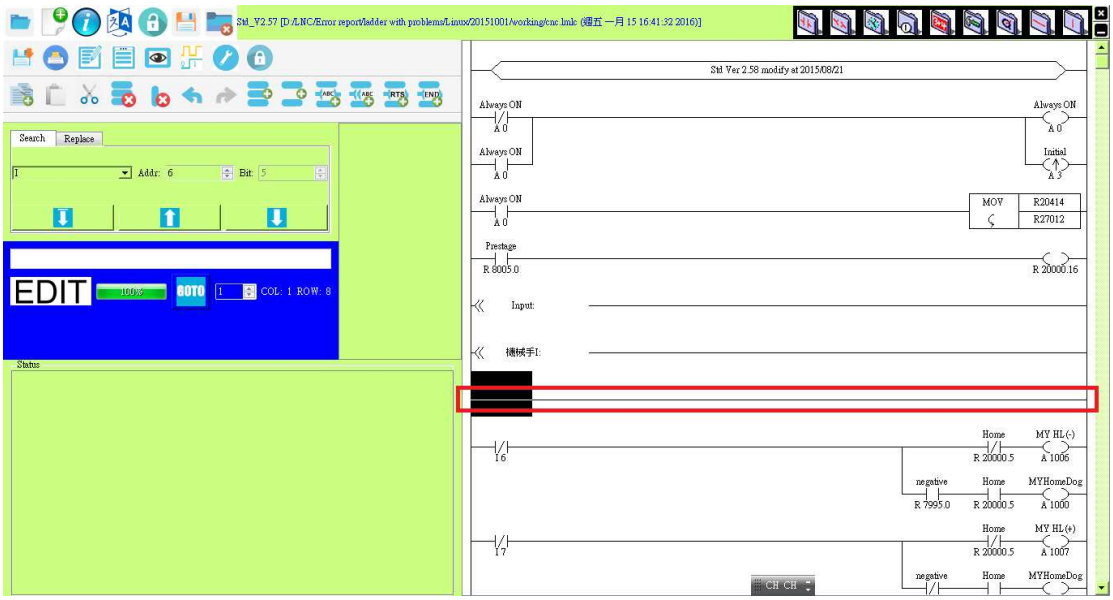


Fig. 38 Insert a Line Row

3.6.25. ：Insert an Empty Row above Focus

Insert an empty row above focus as Fig. 39.

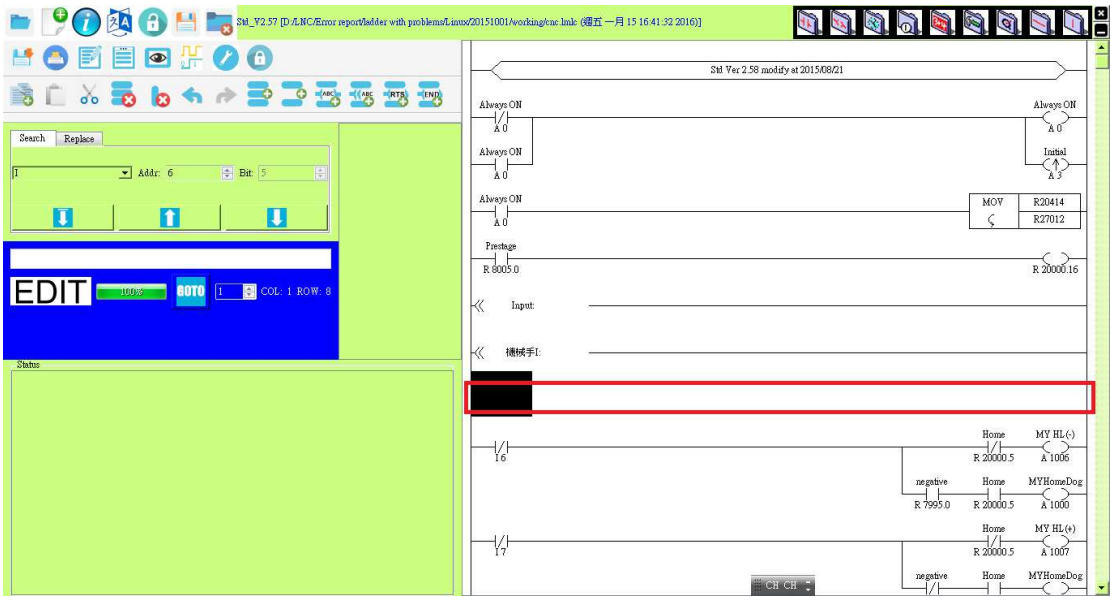


Fig. 39 Insert an Empty Row

3.6.26. ：Insert a Comment Element above Focus

Insert a comment element above focus as Fig. 40.

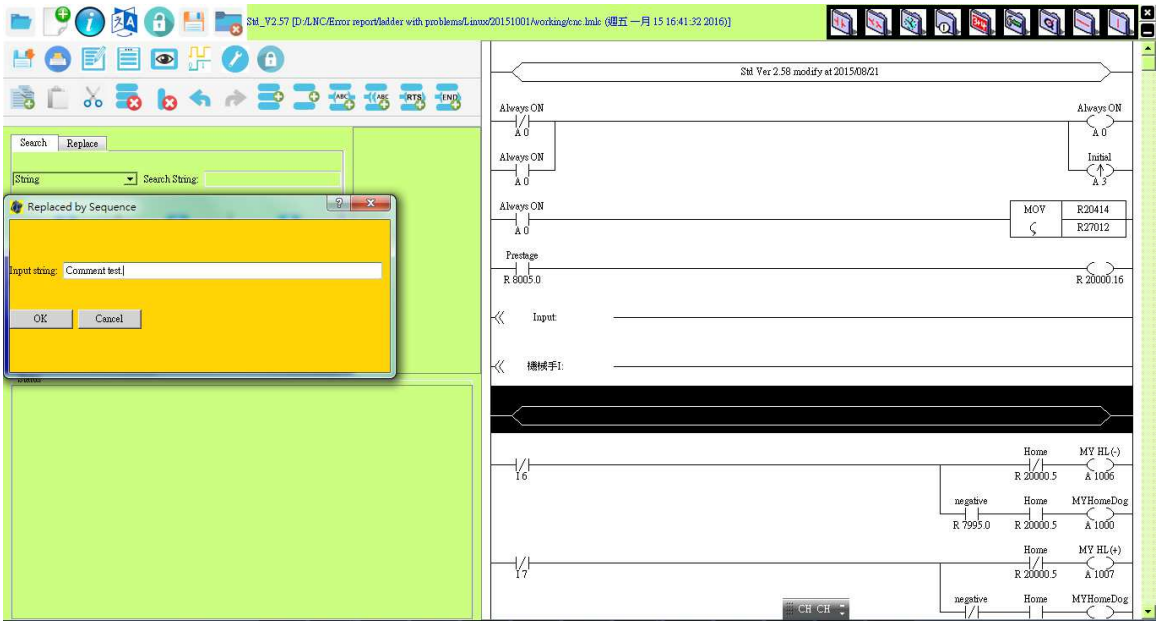


Fig. 40 Insert a comment element

3.6.27. ：Insert a Label Element Row above Focus

Insert a label element row above focus as Fig. 41.

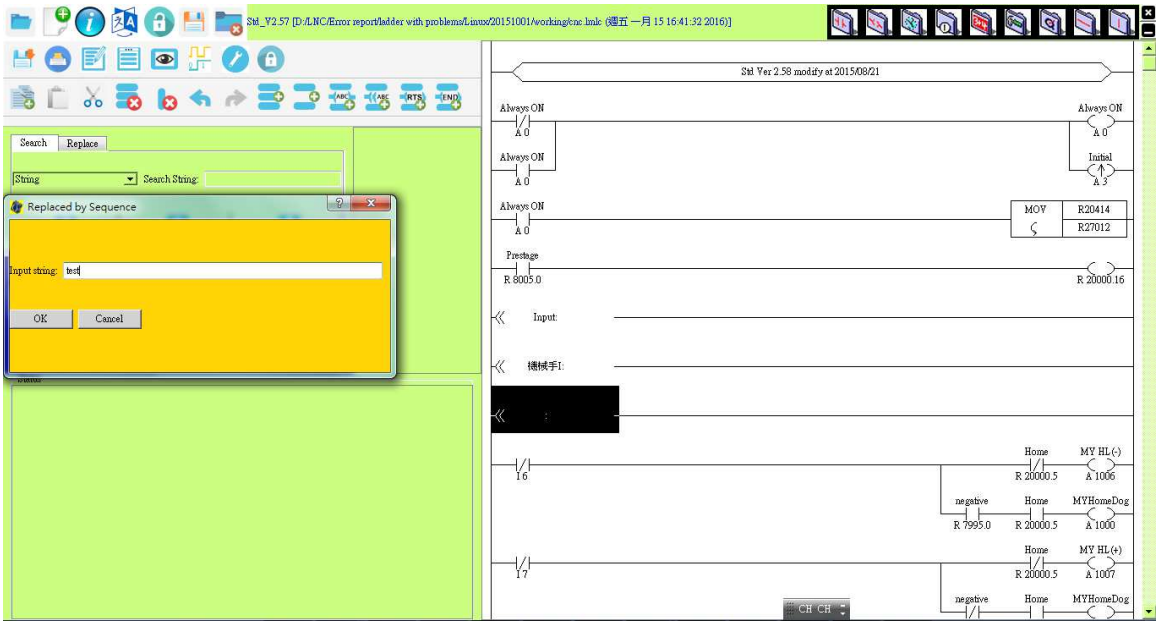


Fig. 41 Insert a Label Row

3.6.28.  : Insert an RTS Element Row above Focus

Insert an RTS element row above focus as Fig. 42.

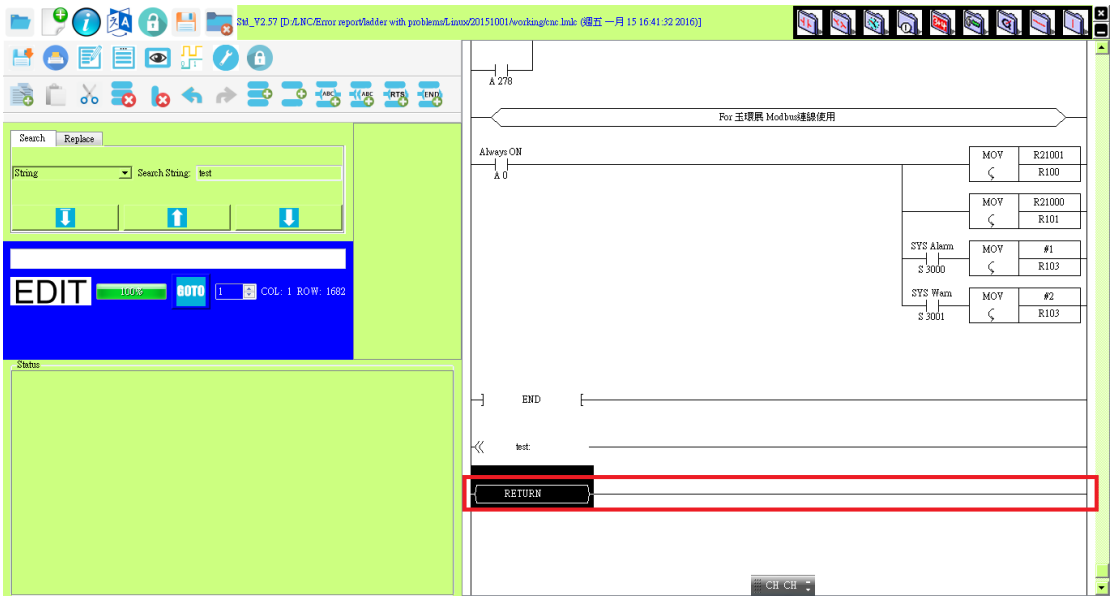


Fig. 42 Insert an RTS Element

: RTS element must pair with label element which is used as subroutine.

3.6.29.  ：Insert an END Element Row above Focus

Insert an END element row above focus as Fig.　43.

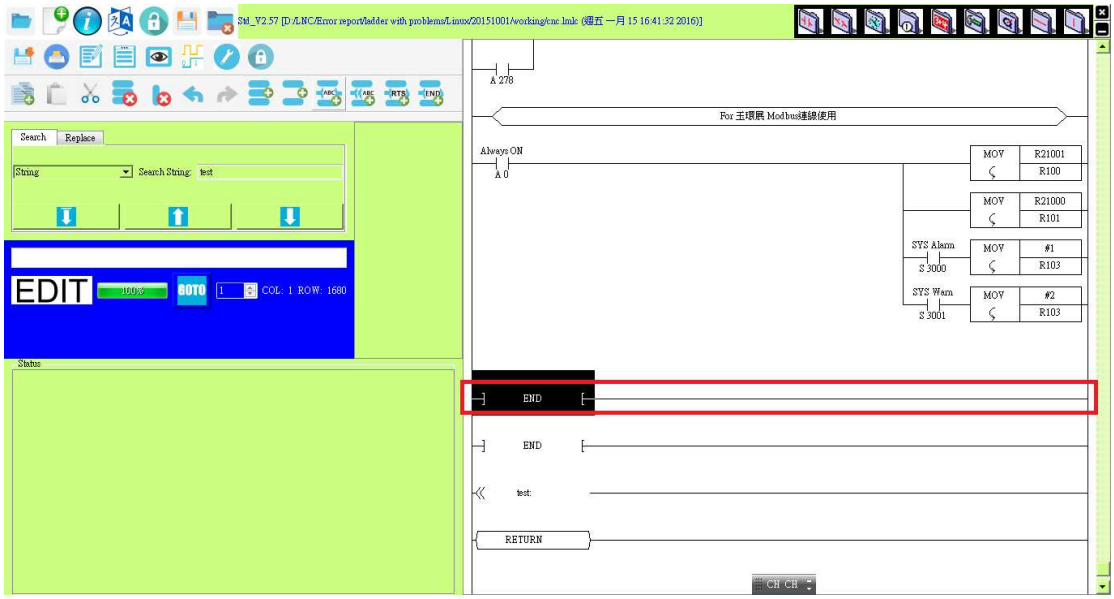


Fig.　43 Insert END Element Row

：Only one END element is needed within one ladder.

3.7. Element Toolbar

3.7.1.  Contact Class：

Hot Key：Ctrl+F2

Click function icon and then click mouse left button on the desired position then the contact dialog would pop up as Fig.　44.

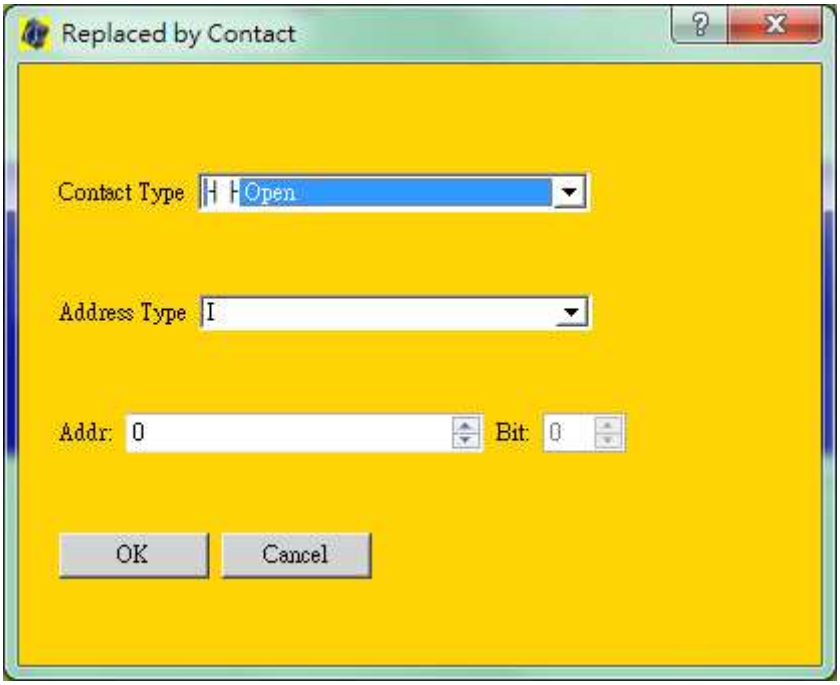


Fig. 44 Contact Dialog

Contact type selection is as Fig. 45：

- Open：Normal Open Contact
- Close：Normal Close Contact

Address selection is as Fig. 46：

- I：Outer environment input signal to PLC module, usually mapped to hardware digital input.
- O：PLC module output to outer environment, usually mapped to hardware digital output.
- C、S：Used by NC kernel and PLC module to interact.
- A：PLC auxiliary address.
- Timer：Active status is corresponding to the assigned timer timeout status.
- Counter：Active status is corresponding to the assigned counter reaching status.
- Register Bit：Active status is corresponding to the assigned register bit.

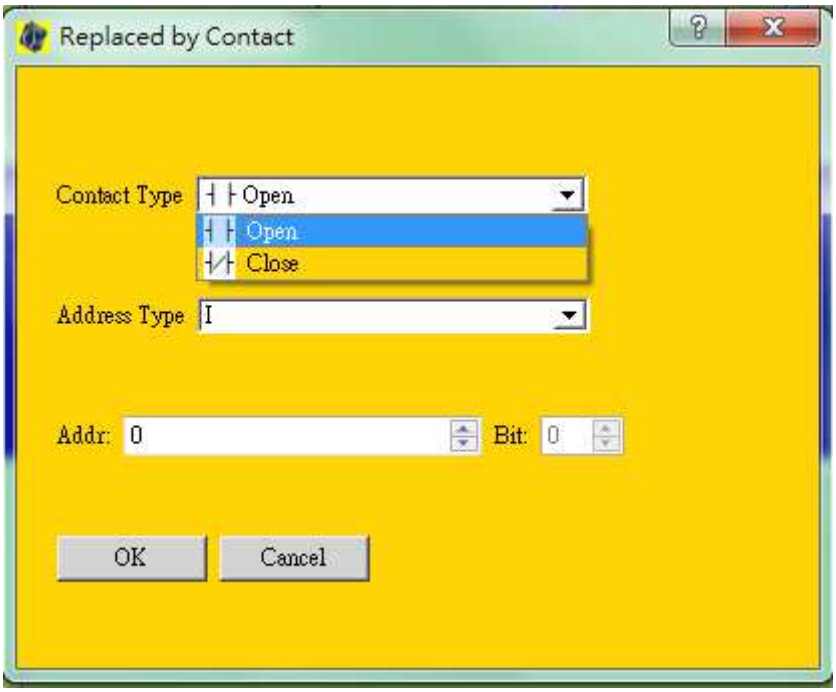


Fig. 45 Contact Type Selection

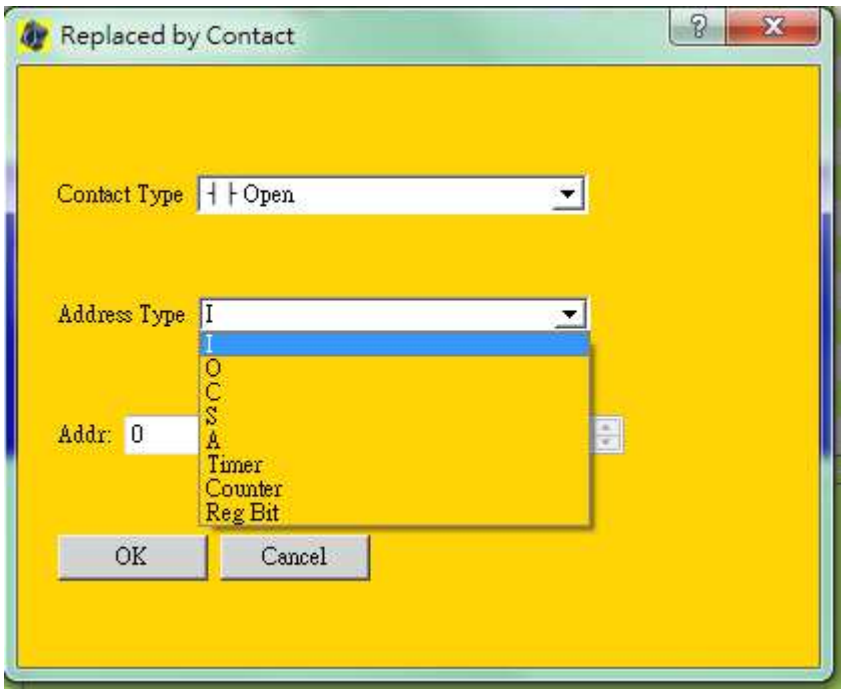


Fig. 46 Address Selection

Name	Description
Normal Open	Active when assigned address is True
Normal Close	Active when assigned address is False

Timer Normal Open	Active when assigned timer timeout
Timer Normal Close	Active when assigned timer has not timeout
Counter Normal Open	Active when assigned counter reaches the count.
Counter Normal Close	Active when assigned counter has not reached the count

Table 3 PLC Contact Functionality

3.7.2.  Coil Class：

Hot Key：Ctrl+F3

Click function icon and then click mouse left button on the desired position then the coil dialog would pop up as Fig. 47。

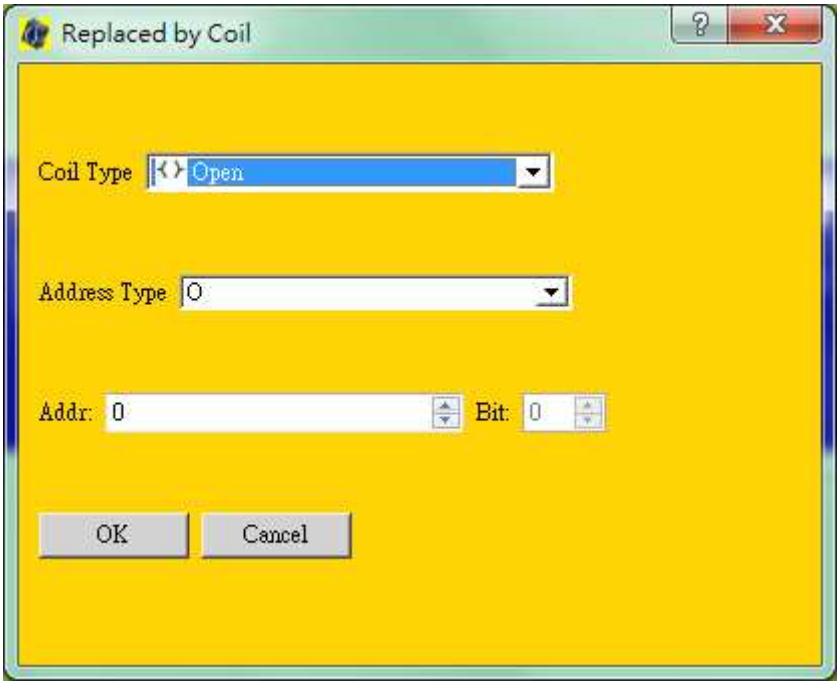


Fig. 47 Coil dialog

Coil type selection is as Fig. 48：

- Open：Normal Open Coil.
- Close：Normal Close Coil.
- Rising Edge：Rising Edge Triggered Coil.
- Falling Edge：Falling Edge Triggered Coil.
- Set ：Latch Coil.
- Reset：Reset Latch Coil.

Address selection is as Fig. 49：

- O：PLC module outputs signal to outer environment, usually mapped to

hardware digital output.

- C、S：Used by NC kernel and PLC module to interact.
- A：PLC auxiliary address.
- Register Bit：Active status is corresponding to the assigned register bit.

：Rising Edge/Falling Edge triggered coil doesn't support register bit address type.

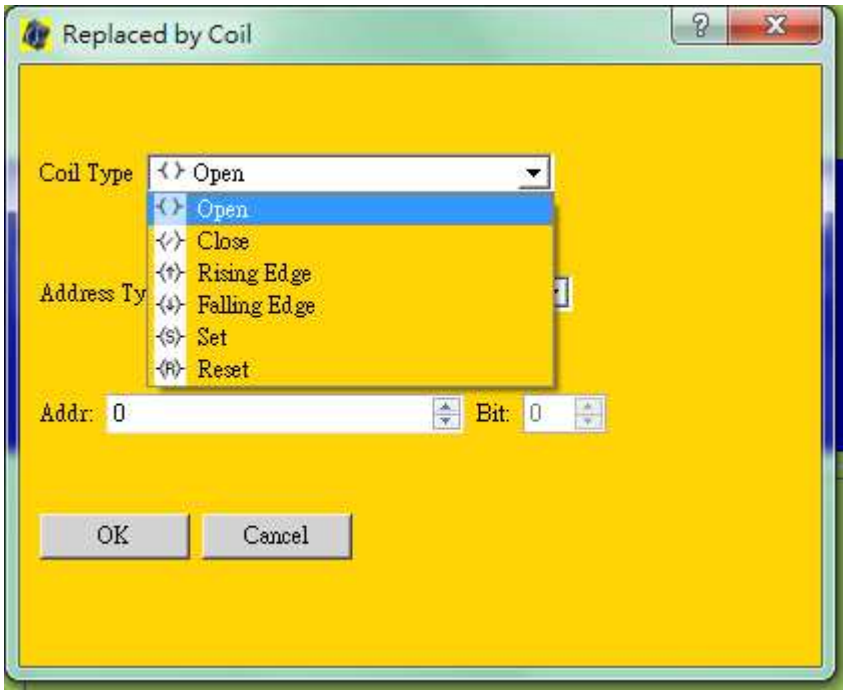


Fig. 48 Coil Type Selection

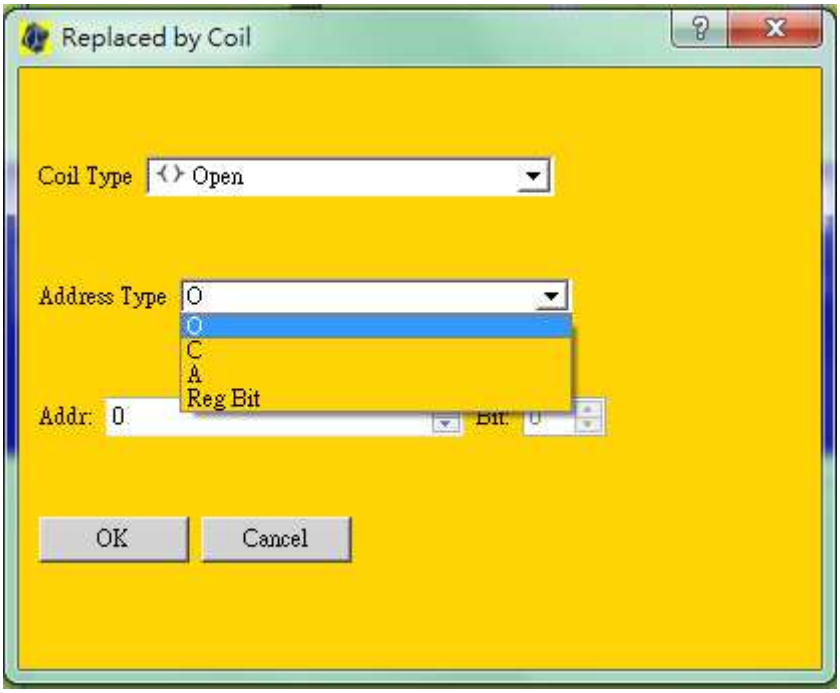


Fig.　49 Address Selection

Name	Description
Normal Open	Output when the assigned address is True
Normal Close	Output when the assigned address is False
Rising Edge Triggered	Output when the assigned address is changed from False to True
Falling Edge Triggered	Output when the assigned address is changed from True to False
Set	Output continuously when the assigned address even turns from False to True
Reset	Reset the latch coil when enabled

Table 4 PLC Coil Functionality

3.7.3.  Timer Class：

Hot Key：Ctrl+F4

Click function icon and then click mouse left button on the desired position then the timer dialog would pop up as Fig.　50。

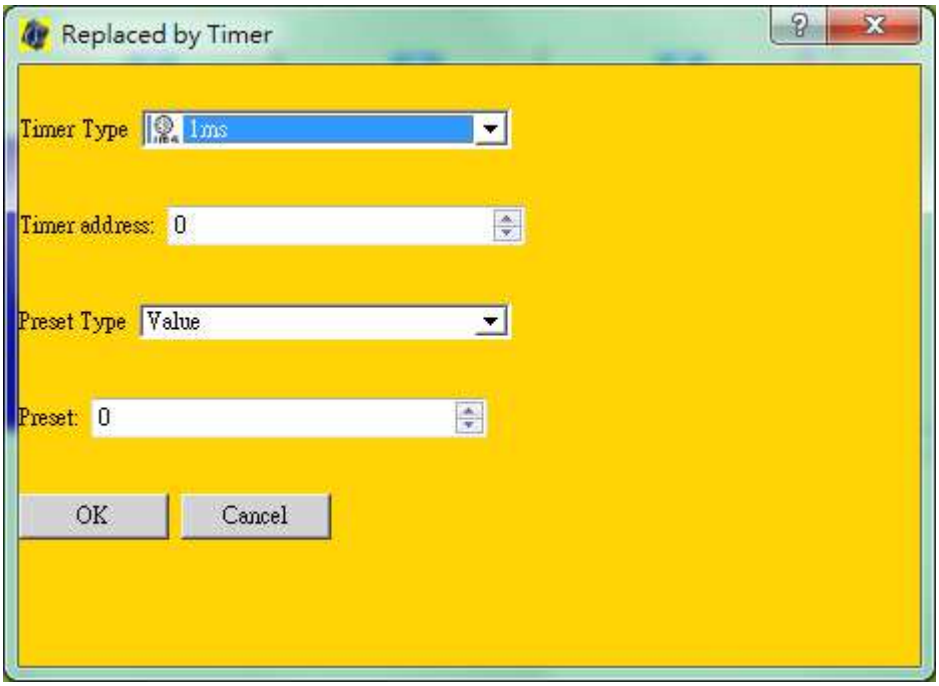


Fig.　50 Timer Dialog

Timer type selection is as Fig.　51：

- 1ms：Timer step is in 1ms.

：The interrupt time of the controller must be set to 1ms to make this type accurate.

- 10ms：Timer step is in 10ms.
- 100ms：Timer step is in 100ms.
- 1s：Timer step is in 1s.
- RTimer 1ms：Timer step is in 1ms.

：The interrupt time of the controller must be set to 1ms to make this type accurate.

- RTimer 10ms：Timer step is in 10ms.
- RTimer 100ms：Timer step is in 100ms.
- RTimer 1s：Timer step is in 1s.

One shot time period can be set by two ways as Fig.　52：

- Value：Directly input time period in 0 ~ 2147483647.
- Register：Use the value within the register as the time period.

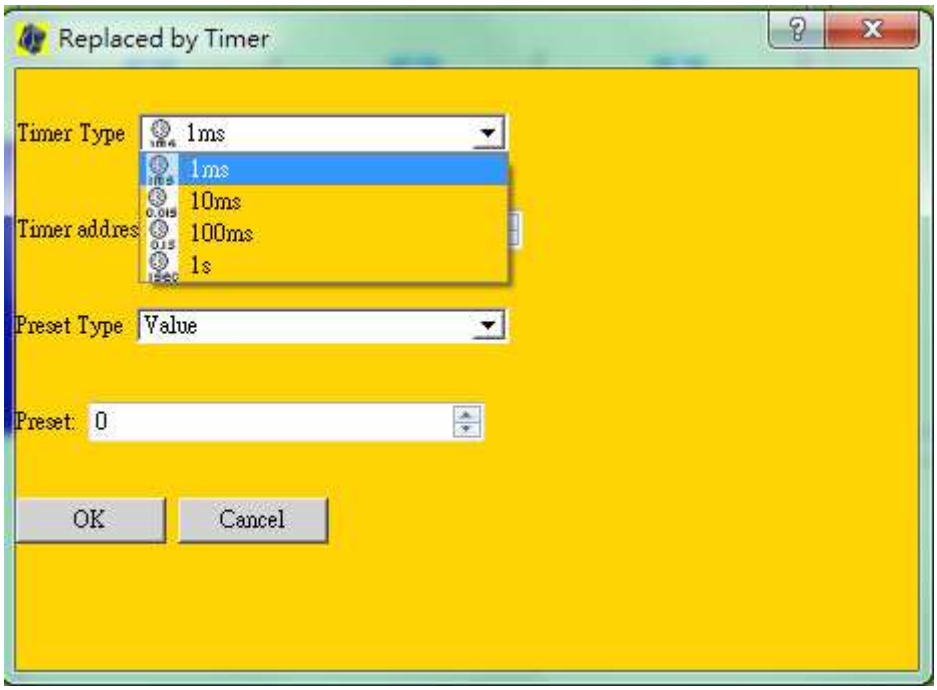


Fig.　51 Timer Type

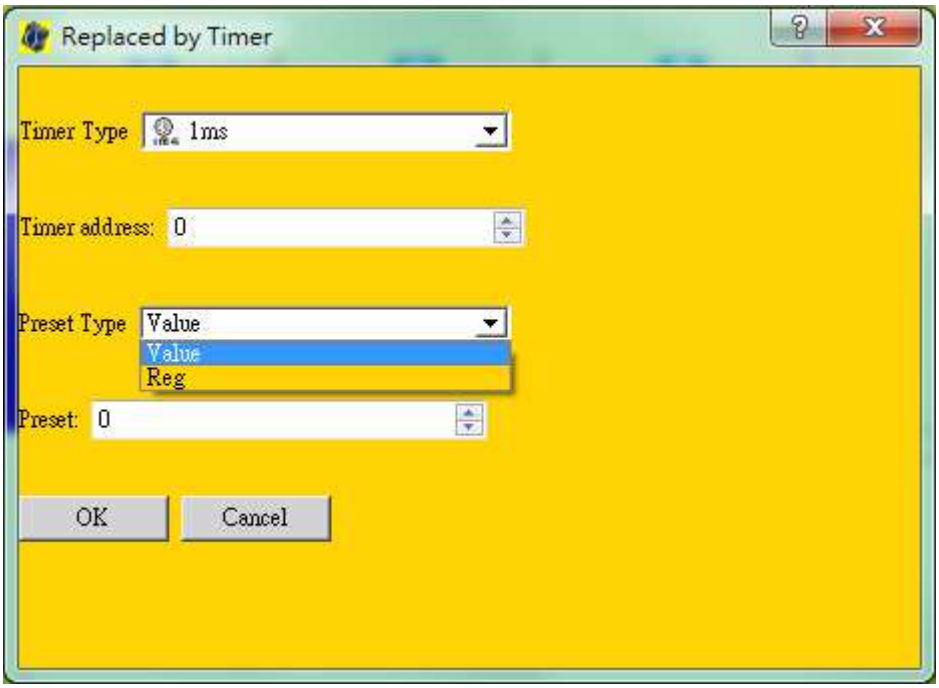


Fig.　52 Set Time Period

Name	Description
1ms Timer	Time step is 1ms
10ms Timer	Time step is 10ms
100ms Timer	Time step is 100ms
RTimer 1ms Timer	Time step is 1ms
RTimer 10ms Timer	Time step is 10ms
RTimer 100ms Timer	Time step is 100ms
RTimer 1s Timer	Time step is 1s

Table 5 PLC Timer Functionality

3.7.4. Counter Class：

Hot Key：Ctrl+F5

Click function icon and then click mouse left button on the desired position then the counter dialog would pop up as Fig.　53。

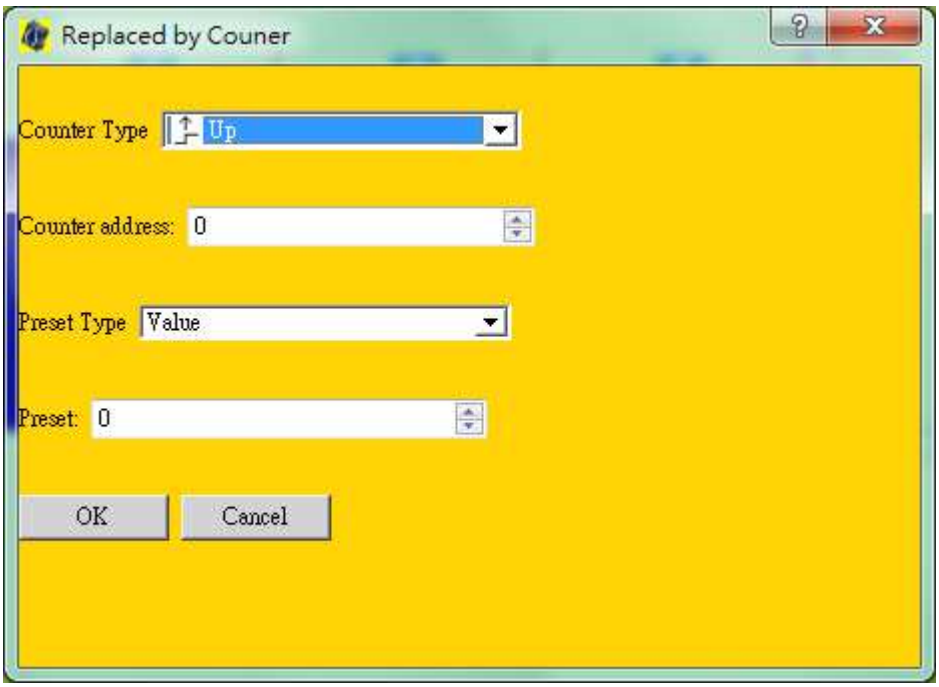


Fig. 53 Counter Dialog

Counter type selection is as Fig. 54：

- UP：Count from 0 to the set value increasingly.
- Down：Count from the set value to 0 decreasingly.
- Ring UP：Count from 0 to the set value and start over from 0 again till disabled.
- Ring Down：Count from the set value to 0 and start over from the set value again till disabled.
- Reset：Reset the counter to 0 for up counter and the set value for down counter.

The set value for the counter can be set as Fig. 55：

- Value：Directly input value in 0 ~ 2147483647.
- Register：Use the value in the register as the set value.

The user does not need to set value for reset type counter.

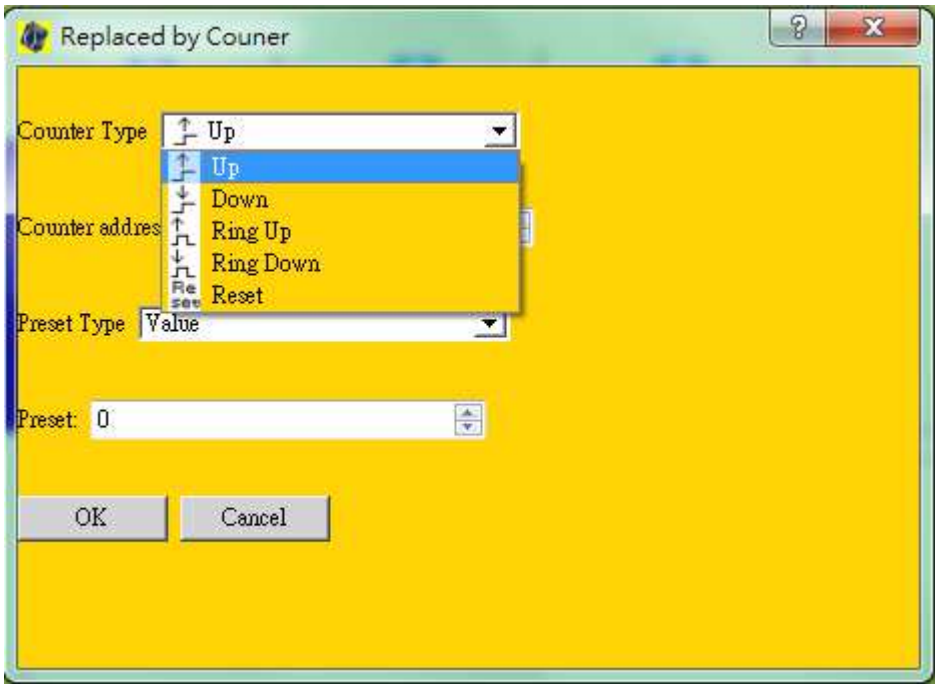


Fig. 54 Counter Type Selection

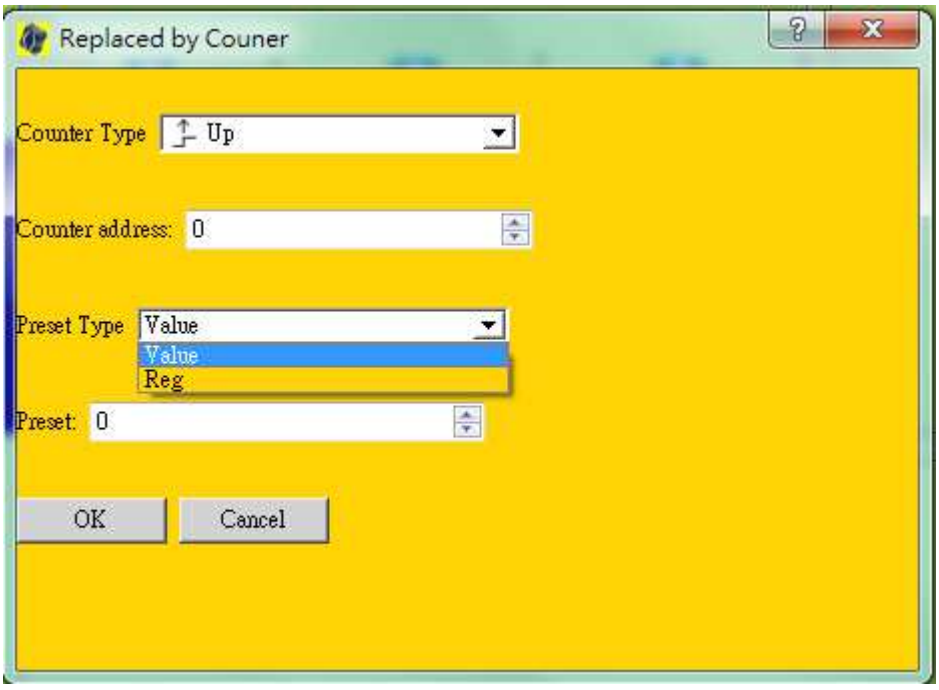


Fig. 55 Counter Setting

Name	Description
Up Counter	Count from 0 to the set value

Down Counter	Count from the set value to 0
Ring Up Counter	Count from the 0 to the set value and start over again
Ring Down Counter	Counter from the set value to 0 and start over again
Reset Counter	Reset counter

Table 6 PLC Counter Functionality

3.7.5. Arithmetic Class：

Hot Key：Ctrl+F6

Click function icon and then click mouse left button on the desired position then the arithmetic function element dialog would pop up as Fig. 56。

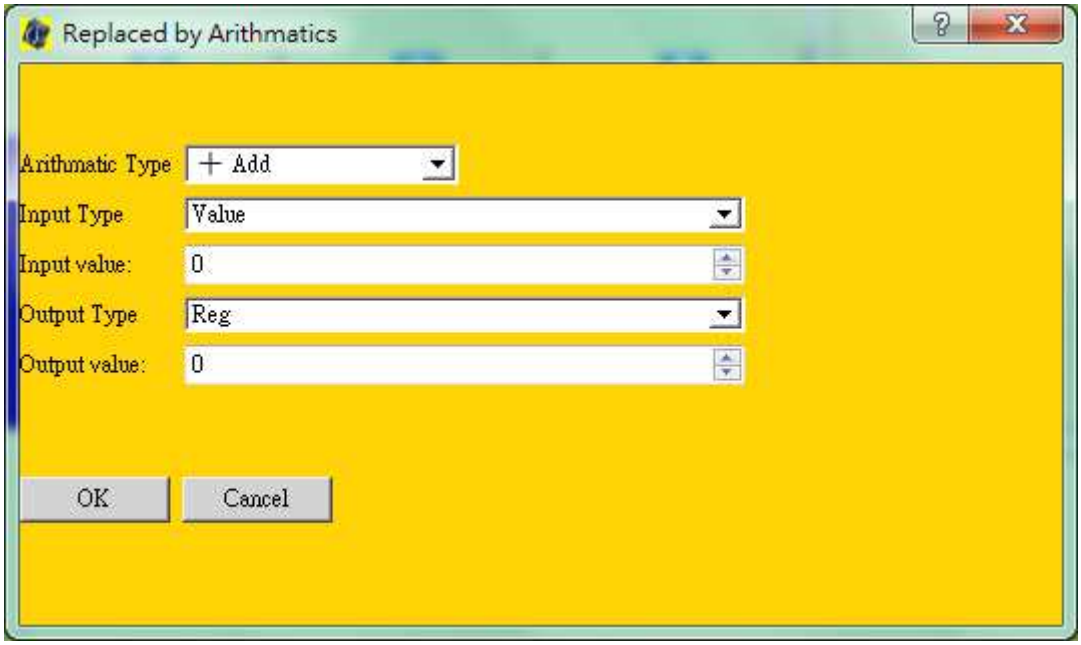


Fig. 56 Arithmetic Function Element Dialog

Type selection is as Fig. 57：

- +Add：Do addition for two operands.
- -Sub：Do subtraction for two operands.
- ×Mul：Do multiplication for two operands.
- ÷Div：Do division for two operands.
- %Mod：Do modulo for two operands.
- ↶Move：Move value between two operands.
- >More than：Is one operand greater than the other one..

- <Less than：Is one operand lesser than the other one.
- =Equal to：Is one operand equal to the other one.
- &And：Do bit and operation between two operands.
- |Or：Do bit or operation between two operands.
- ^Xor：Do bit xor operation between two operands.
- Mulrini：Set value to more than one registers.
- Mulrcpy：Copy a continuous range of registers to a non-overlapped area.
- I/Rmap：Map 32 continuous I addresses to one register.
- O/Rmap：Map 32 continuous O addresses to one register.
- Shll：Do logic left shift operation.
- Shrl：Do logic right shift operation.
- >=More than or Equal to：Is one operand greater than or equal to the other one
- <=Less than or Equal to：Is one operand lesser than or equal tothe other one.
- ≠Not equal to：Is one operand not equal to the other one.
- I/RmapN：Map N continuous I addresses to one register.
- O/RmapN：Map N continuous O addresses to one register.

Input operand type is as Fig. 58：

- Value：Directly input value in ±2147483647.
- Register：Use register's value.

Note：Move element has operand type of register pointer.

Note：I/Rmap and O/Rmap has I, O address as input.

Output operand type is as Fig. 59：

Register：Use register as operand.

Note：Move element has operand type of register pointer.

Mulrini, Mulrcpy , I/RmapN and O/RmapN have three operands as Fig. 60 and Fig. 61.

Mulrini：The first operand is the start address of the registers. The second is the number of registers to set. The third is the value to set into the registers.

Mulrcpy：The first operand is the start address of the source register area. The second is the start address of the target area. The third is the number to copy. The source and target area cannot overlap.

The third operand of I/RmapN and O/RmapN is the number of I or O to be mapped.

The element would continuously do the operation till it's disable by the the

ladder logic.

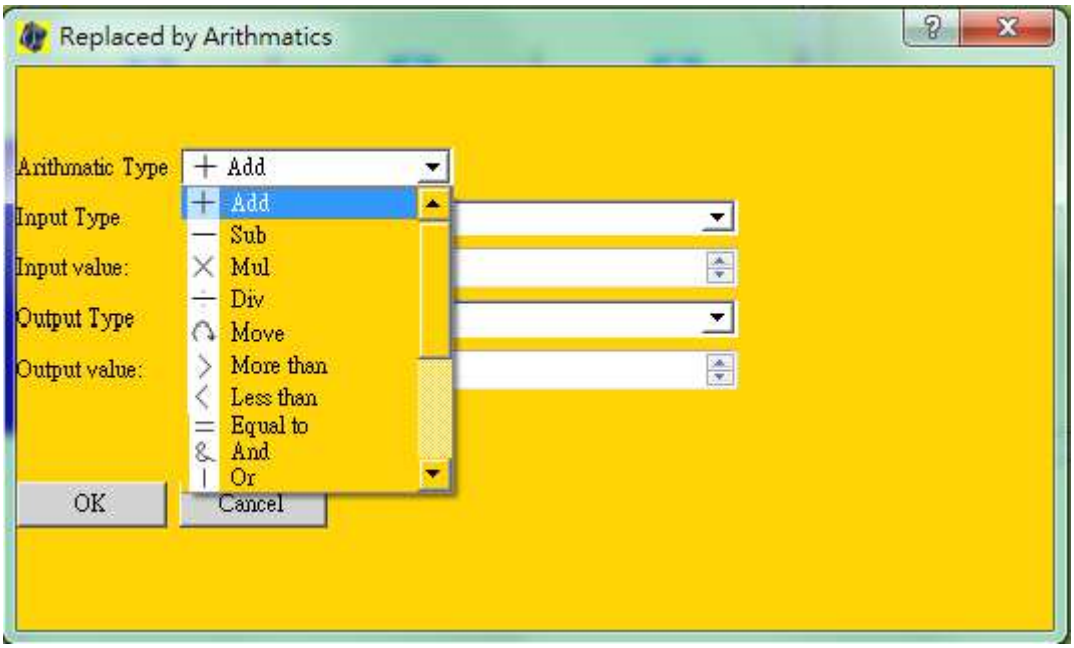


Fig. 57 Arithmetic Function Element Type

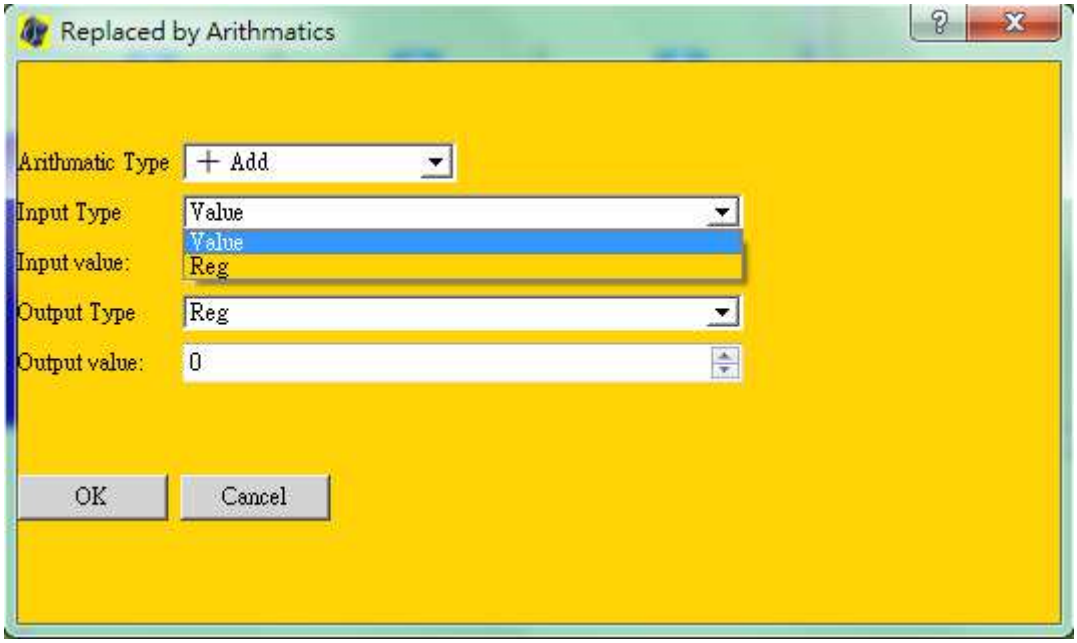


Fig. 58 Input Operand Type

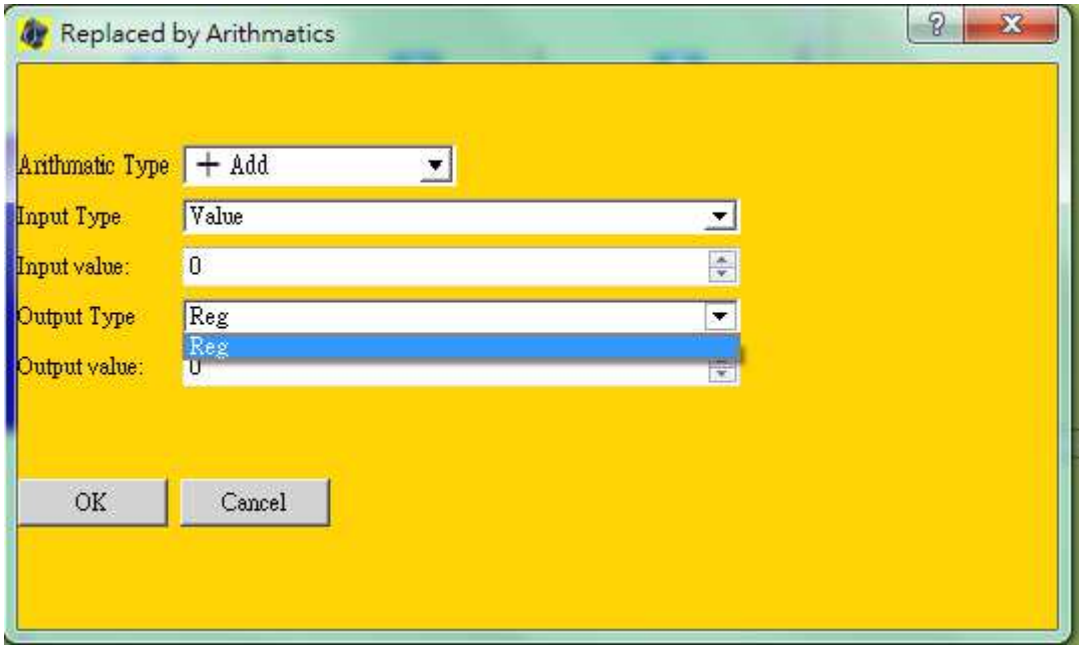


Fig. 59 Output Operand Type

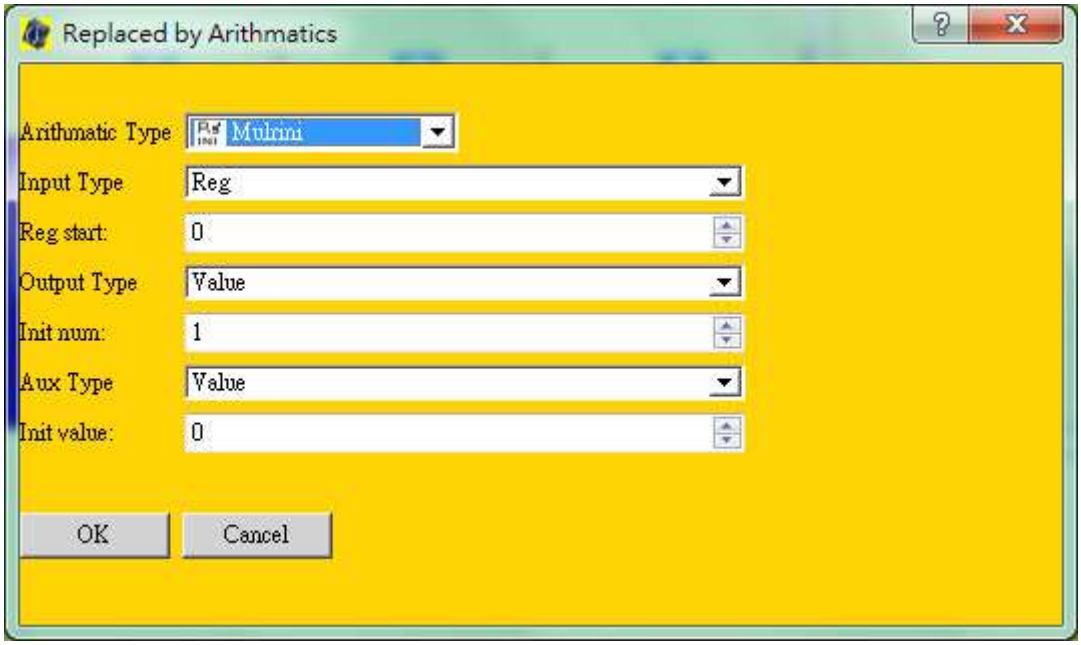


Fig. 60 Mulrini Setting

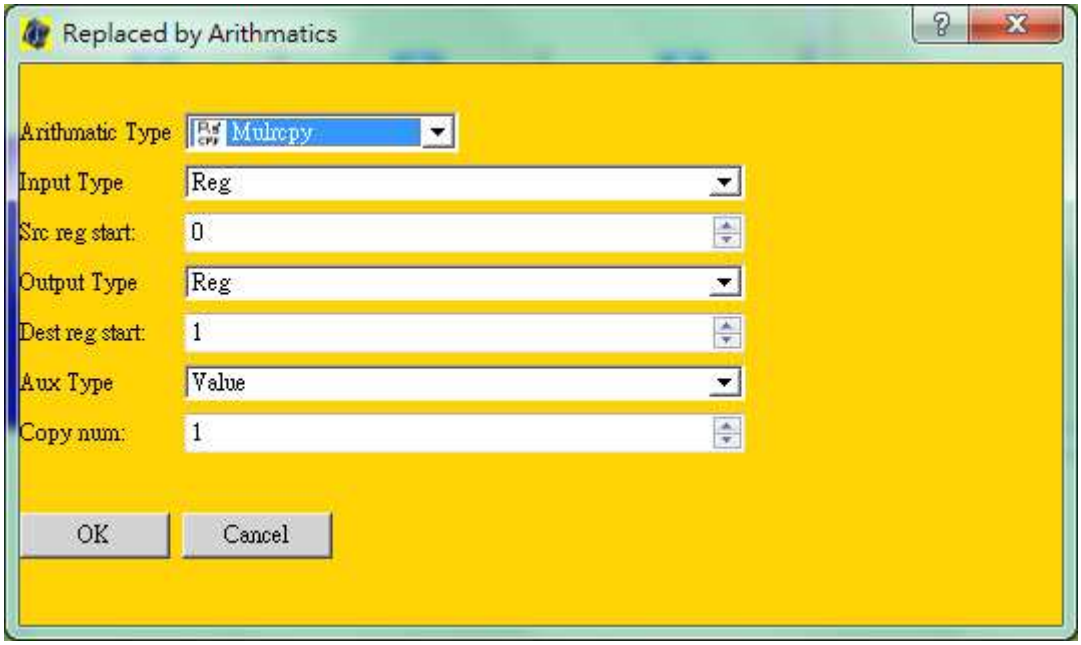


Fig. 61 Mulrcpy Setting

Name	Description
Add	Add value or value of register to one register
Subtract	Subtract value or value of register from one register
Multiply	Multiply one register by value or value of register
Divide	Divide one register by value or value of register
Modulo	Divide one register by value or value of register then put the remainder to output
Move	Move value between two operands
Compare Large	Is one operand larger than another
Compare Less	Is one operand lesser than another
Compare Equal	Is one operand equal to another
AND	Bit and operation
OR	Bit or operation
Exclusive OR	Bit xor operation
Mulrini	Set value to more than one register
Mulrcpy	Copy register area to another area

I/Rmap	Map continuous 32 I address to one register
O/Rmap	Map continuous 32 O address to one register
Shll	Logic left shift
Compare Large than or Equal to	Is one operand larger than or equal to another
Compare Less than or Equal to	Is one operand lesser than or equal to another
Compare Not Equal to	Is one operand not equal to another
I/RmapN	Map continuous N I address to one register
O/RmapN	Map continuous N O address to one register

Table 7 PLC Arithmetic Function Element Functionality

3.7.6.  Tool Class：

Hot Key：Ctrl+F7

Click function icon and then click mouse left button on the desired position then the tool function element dialog would pop up as Fig. 62。

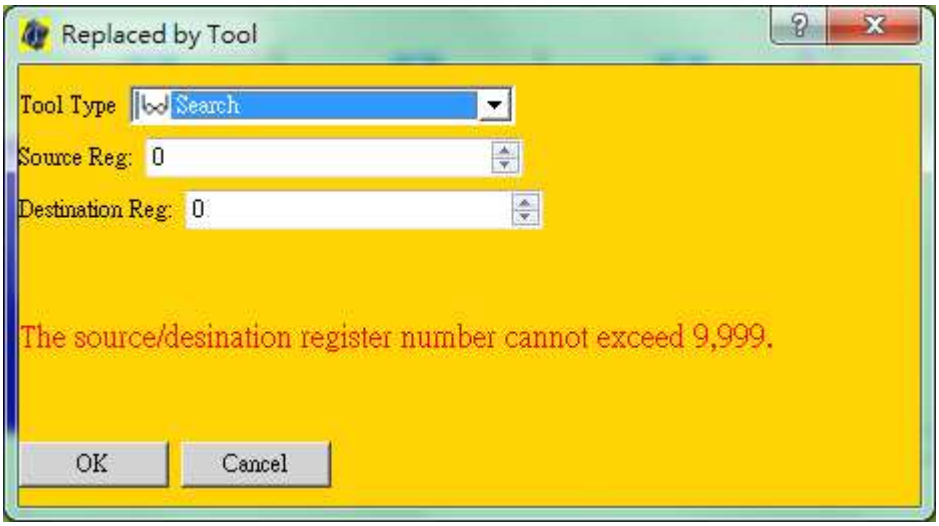


Fig. 62 Tool Dialog

Tool function element selection is as Fig. 63：

- Search：Search element.

- Rotate：Rotation step calculation.

Input operand：

Register：Use register from 0 to 9999.

Output operand：

Register：Use register from 0~9999.

：The register addresses of input operand+1 and output operand+1 would also be used.

The element would continuously do the operation till it's disable by the the ladder logic.

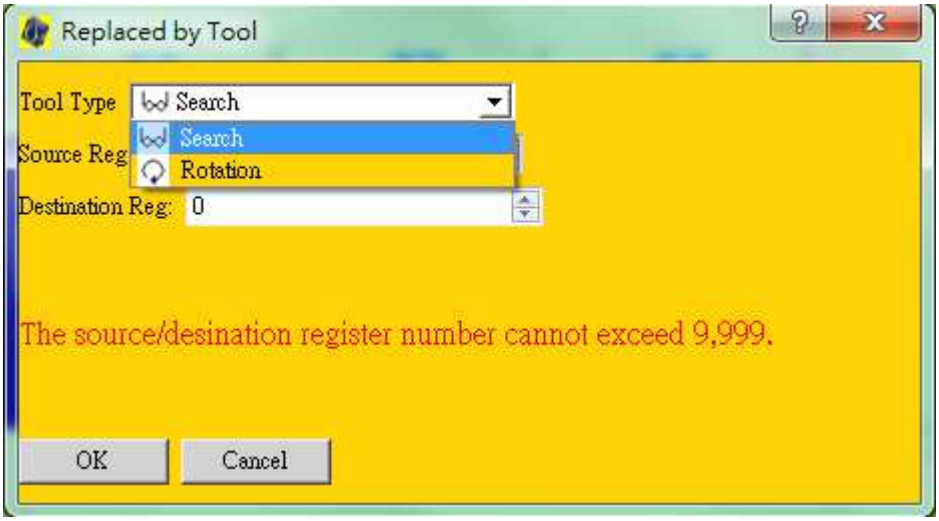


Fig. 63 Tool Element Type

Name	Description
Search	Search the tool number within the magazine
Rotate	Calculate the rotation step to a target position in a circle

Table 8 PLC Tool Functionality

3.7.7. Sequence Class：

Hot Key：Ctrl+F8

Click function icon and then click mouse left button on the desired position then the sequence function element dialog would pop up as Fig. 64。

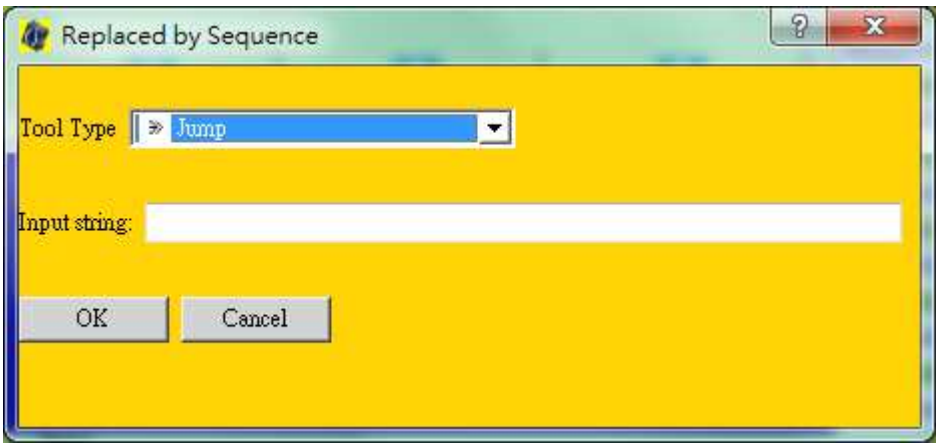


Fig. 64 Sequence Dialog

Sequence function element selection is as Fig. 65.

- Jump：Jump to the paired label element and continue the PLC logic from there.
- Jump Sub Routine：Jump to the paired label element and continue the PLC logic from there. After meeting the RTS element, the PLC logic would return to the next element of Jump Sub Routine Element to go on.
- RET：Return to the caller. Finish ladder logic for this run if RET is within top layer(the logics before END), or return to the top layer if it is in subroutine.

：Jump and Jump Sub Routine must pair with label elements in appropriate position.

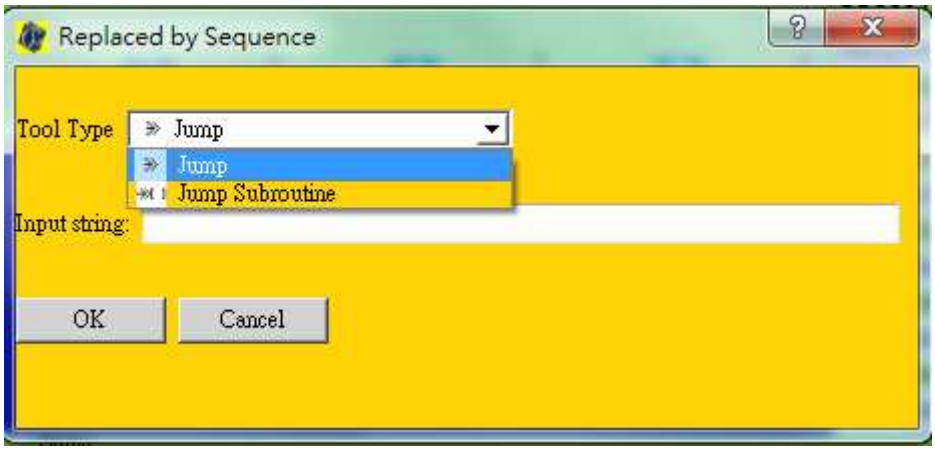


Fig. 65 Sequence Function Element Selection

Name	Description
JUMP	Jump to the paired label
Call Subroutine	Jump to the paired label and return after subroutine

	ending
RET	Return to the caller

Table 9 PLC Sequence Element Functionality

3.7.8. Horizontal Connect：

Hot Key：Ctrl+F9

Click mouse left button on the desired position then a horizontal connection is inserted.

3.7.9. Vertcal Connect：

Hot Key：Ctrl+F10

Click mouse left button on the desired position then a vertical connection is inserted.

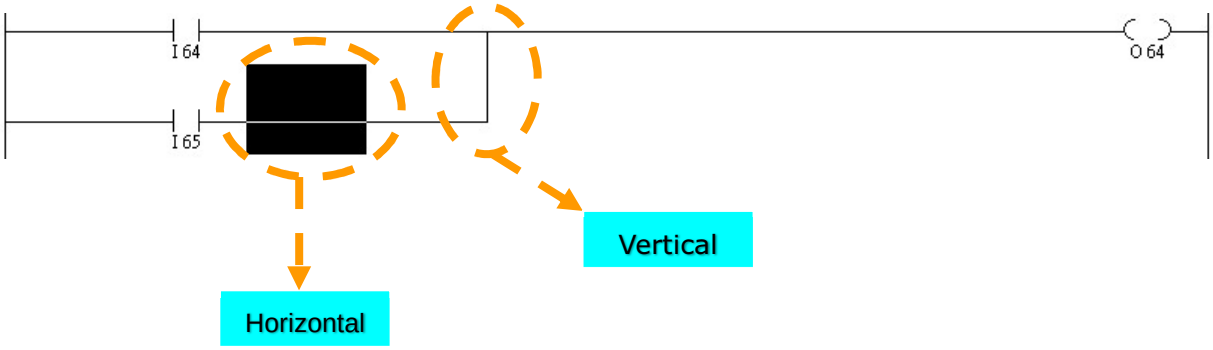


Fig. 66 Connection

Name	Description
Horizontal Line	Connect elements in horizontal direction
Vertical Line	Make a branch

Table 10 PLC Connection

3.8. Set the Element

Double click on the element, or hit Return/Enter to pop up right key menu and use Set as Fig. 67 . The element setting dialog would pop up as Fig. 68.

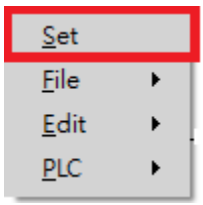


Fig. 67 Element Set

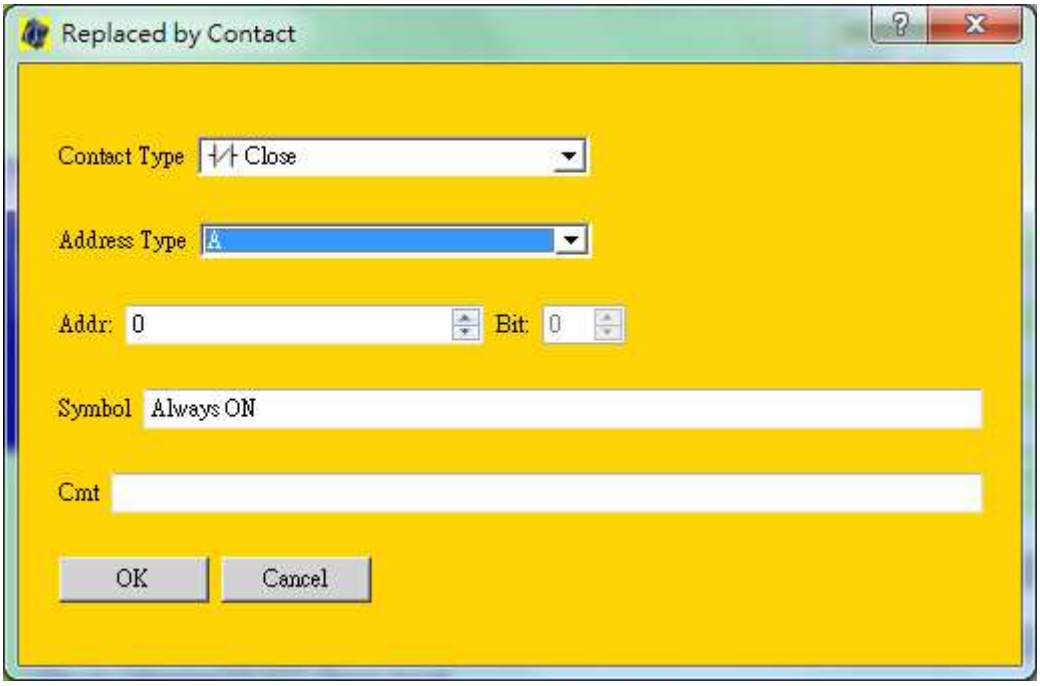


Fig. 68 Element Set Dialog

3.9. Search and Replace

The search and replace area is as Fig. 69, use page tab to select between search page and replace page.

Fig. 69 Search and Replace Area

3.9.1.Search Page

Search data type is as Fig. 70 and listed below:

- I：Search I address.
- O：Search O address.
- C：Search C address.
- S：Search S address.
- A：Search A address.
- Timer：Search Timer address.
- Counter：Search Counter address.
- Register：Search Register address.
- Register Bit：Search Register Bit address.
- String：Search the matching string in comment/label/Jump/Jump Sub Routine.
- Value：Search the value used in the elements.

：The elements with register bit of the same register address are also considered match at searching for register.

：String search would not search in address symbol/comment strings.

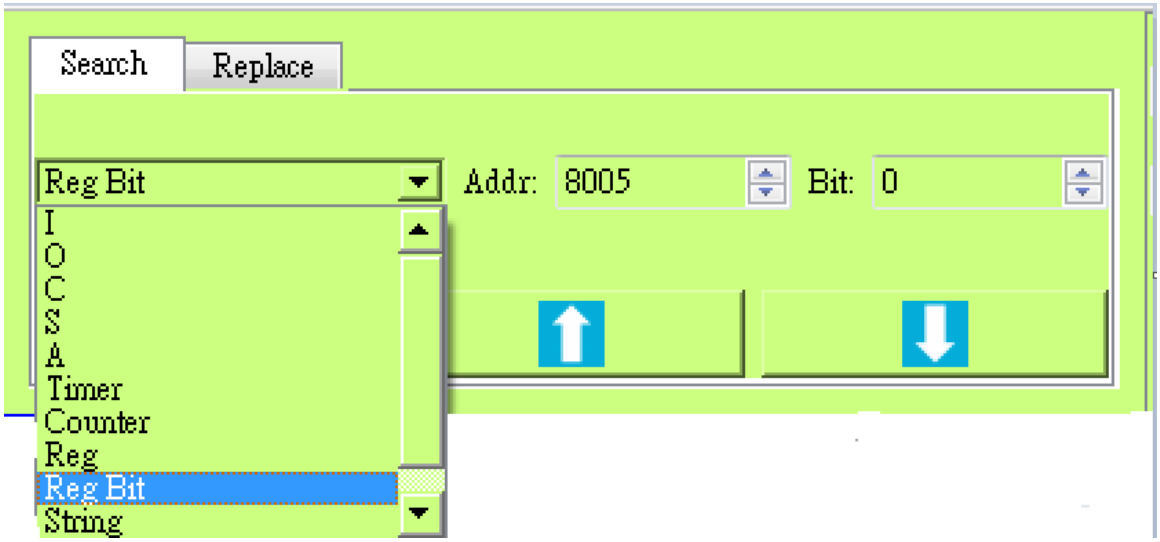


Fig. 70 Search Type

The searching data type and the address would be automatically updated to the set address of one element when the element gets the focus as Fig. 71。

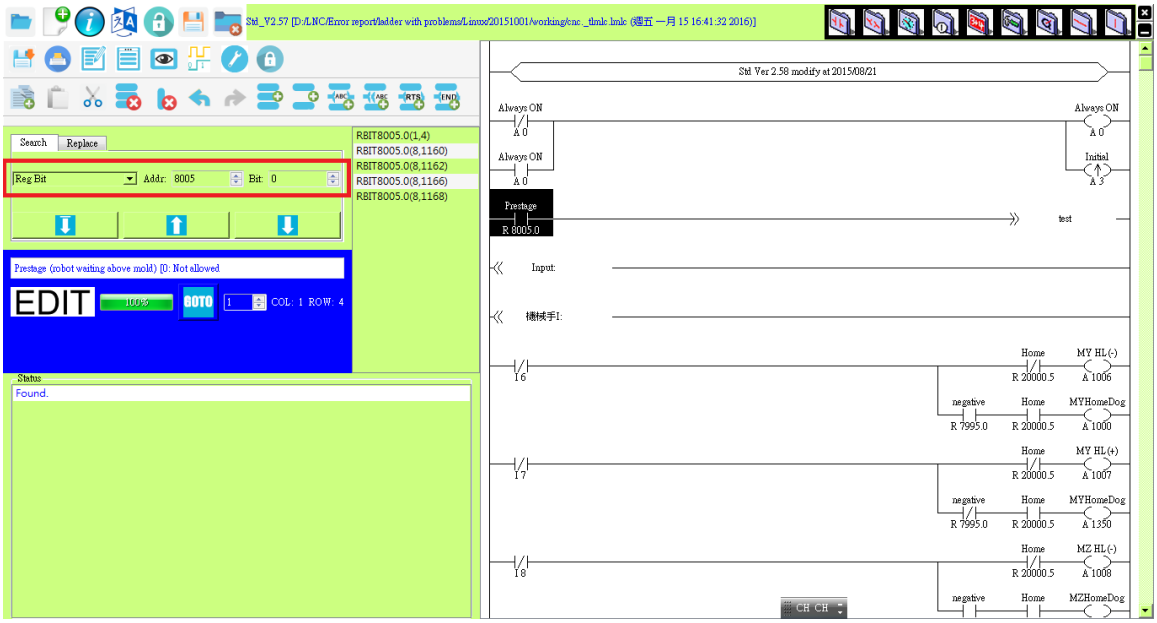


Fig. 71 Searching Type and Address Update for Element with Focus

Searching Direction is as below：

 ：Search from the beginning of the file.

 ：Search down from the next element of the current element.

Hot Key：F3

 ：Search up from the previous element of the current element.

Hot Key：Alt+F3

3.9.2.Replace Page

Replace page is as Fig. 72 and there is additional area to set the replace address.

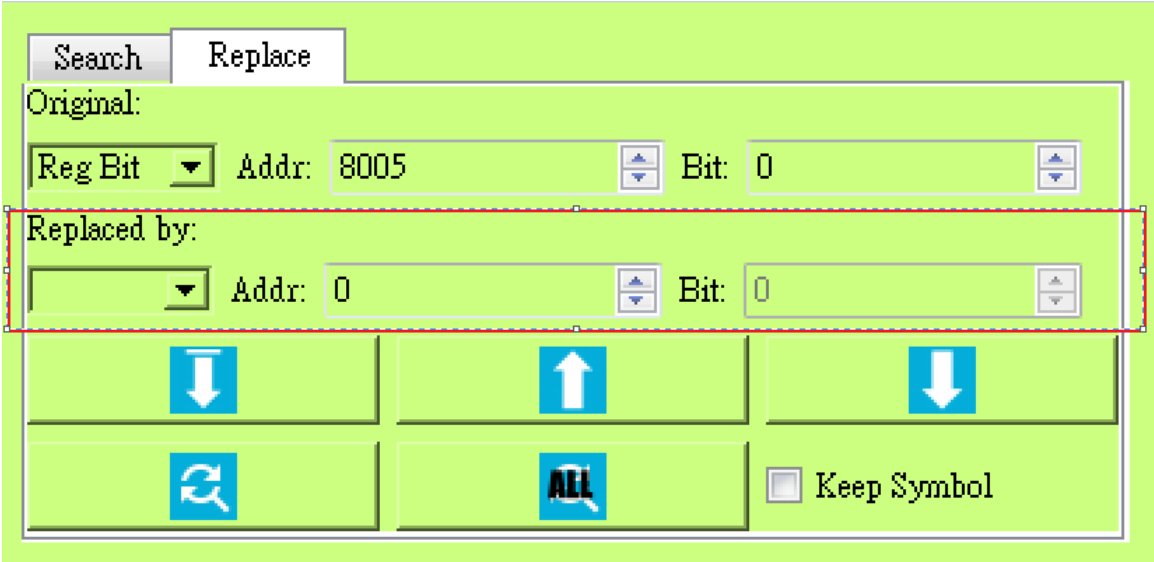


Fig. 72 Replace Page

：Replacing does not support String and Value data type.

Searching functionality refer to **錯誤! 找不到參照來源。**, the buttons for replacing is as following：

：Replace the current searched element only.

: Replace all the elements which match the searching condition in the file.

：The positions of the replaced elements would be listed in application output area for replacing all.

：Replacing may not be possible in some cases like using I address to replace an O address in coil element.

Keep Symbol：Keep symbol/comment of the replaced address to the replacing address.

Example: If the symbol of the replaced address is “Home”, and the replacing address does not have symbol set. If the keep symbol option is activated, then the symbol of the replacing address would become “Home” after replacing.

3.10. Edit Status Area

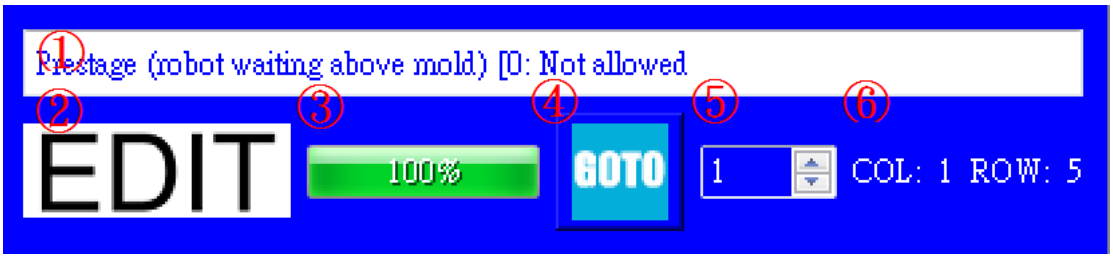


Fig. 73 Edit Status Area

The information displayed is as Fig. 73.

① Comment of the element with focus.

② Current mode.

③ Progress bar for time-consuming task.

④ Jump to row which is set by ⑤.

⑤ Set the row number to jump.

⑥ Show row and column of the current focus.

3.11. Mouse Right Key Menu

- File sub menu is as Fig. 74.

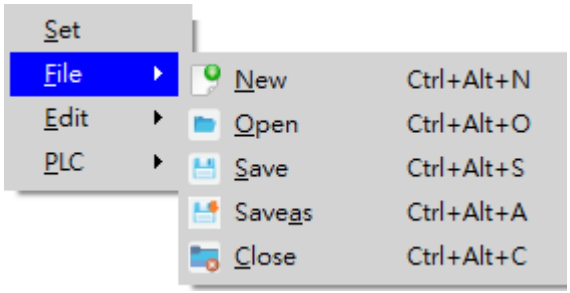


Fig. 74 File Sub Menu

- Edit sub menu is as Fig. 75.

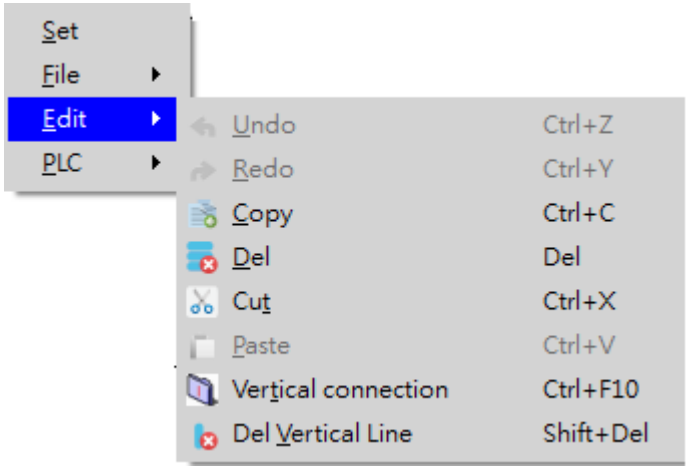


Fig. 75 Edit Sub Menu

- PLC sub menu is as Fig. 76.

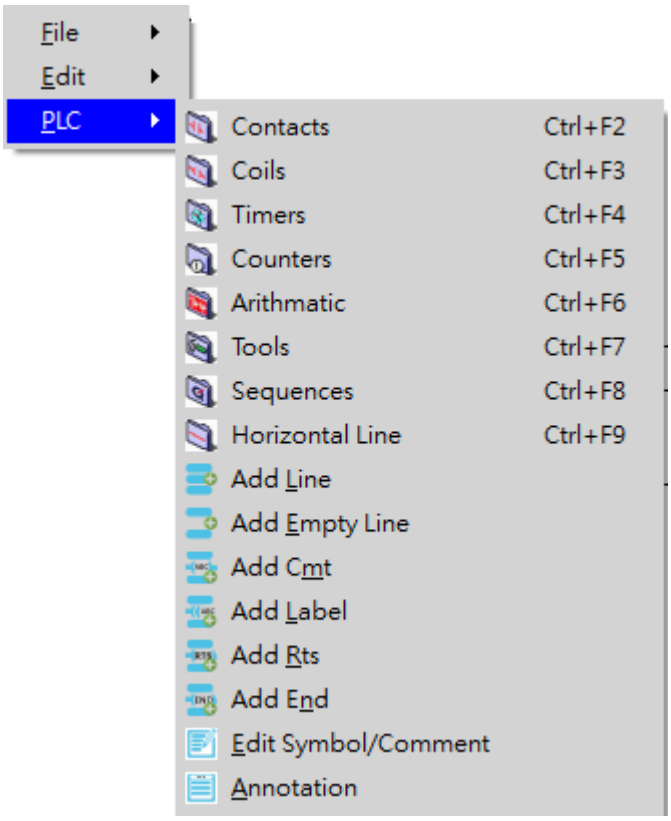


Fig. 76 PLC Sub Menu

4. PLC Ladder Elements

4.1. ⊣ ⊢ Normal Open Contact

The timing diagram for normal open contact is as Fig. 77. The output status is doing AND operation between input and the value of the assigned address.

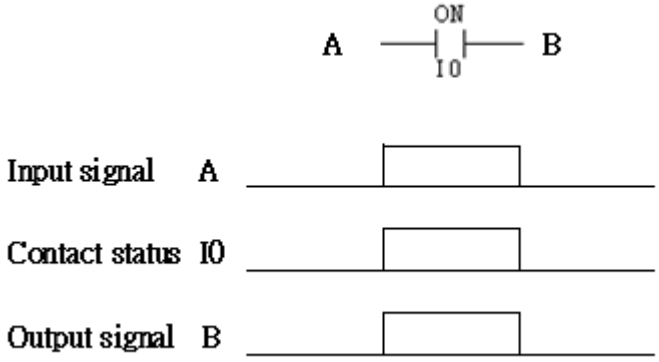


Fig. 77 Normal Open Contact

4.2. ⊣/⊢ Normal Close Contact

The timing diagram for normal close contact is as Fig. 78.

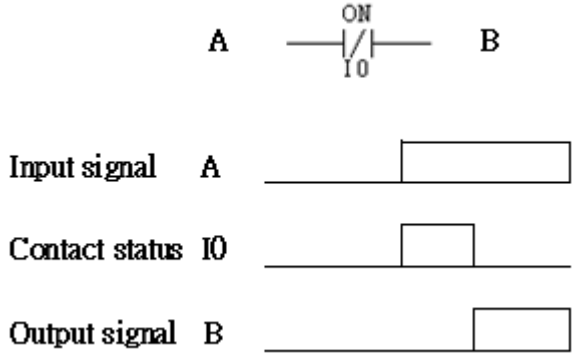


Fig. 78 Normal Close Contact

4.3. —( )— Normal Open Coil

When input is ON, the element writes ON to address O0, otherwise writes OFF as Fig. 79。

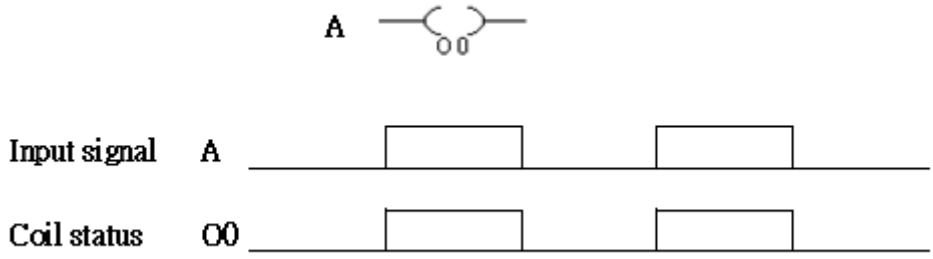


Fig. 79 Normal Open Coil

4.4. —(/)— Normal Closed Coil

When input is ON, the element writes OFF to address O0, otherwise writes ON as Fig. 80.

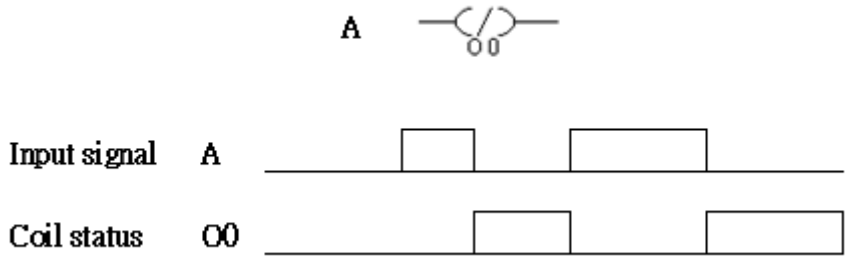


Fig. 80 Normal Close Coil

4.5. —(↑)— Positive Edge Triggered Coil

When input changes from OFF to ON, the element writes ON to address O0 for one PLC scanning period.

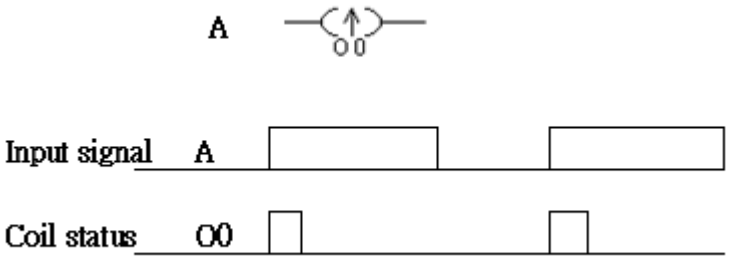


Fig. 81 Positive Edge Triggered Coil

4.6. —(↓)— Negative Edge Triggered Coil

When input changes from ON to OFF, the element writes ON to address O0 for one PLC scanning period.

Note：The total number of edge triggered coils in a ladder cannot exceed 4096.

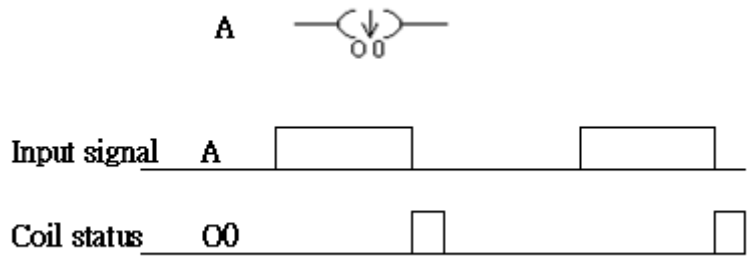


Fig. 82 Negative Edge Triggered Coil

4.7. —(S)—Latch Coil

When input is ON, the element writes ON to address O0 continuously even the input turns OFF.

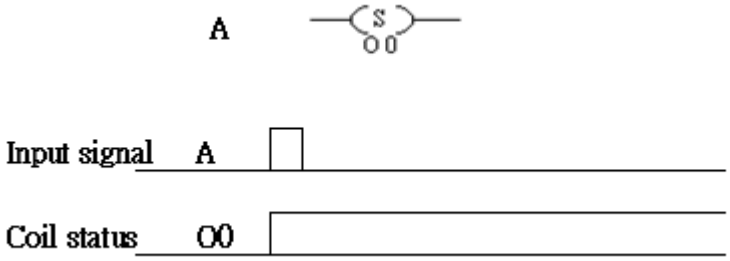


Fig. 83 Latch Coil

4.8. —(R)—Unlatch Coil

When the input is ON, the element writes OFF to address O0.

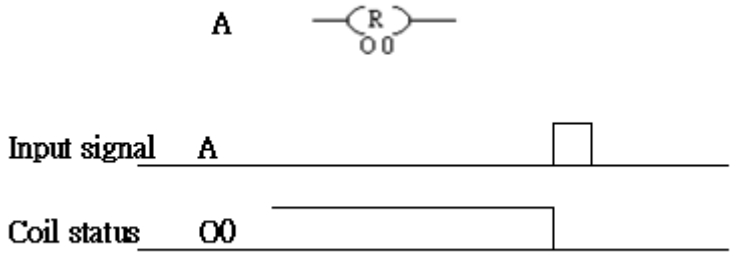


Fig. 84 Unlatch Coil

4.9. Timer Relay

- When I0 changes from OFF to ON and keeps ON, the timer #1 would start to time in time period of 100 ms.

- The timer#1 would enable O0 coil after reaching the set time (1s).
- There are total 256 timers and each timer has the corresponding contact signal. The corresponding contact signal would turn ON when the set time is reached and can be used as normal open or normal close contact.
- The timer would stop if the input becomes OFF during timing.
- The timing period can be set to 1ms, 10ms, 100ms and 1s.
- The timeout value can be set by register.

: The interrupt period must be set to 1ms to make the timer with 1ms period accurate.

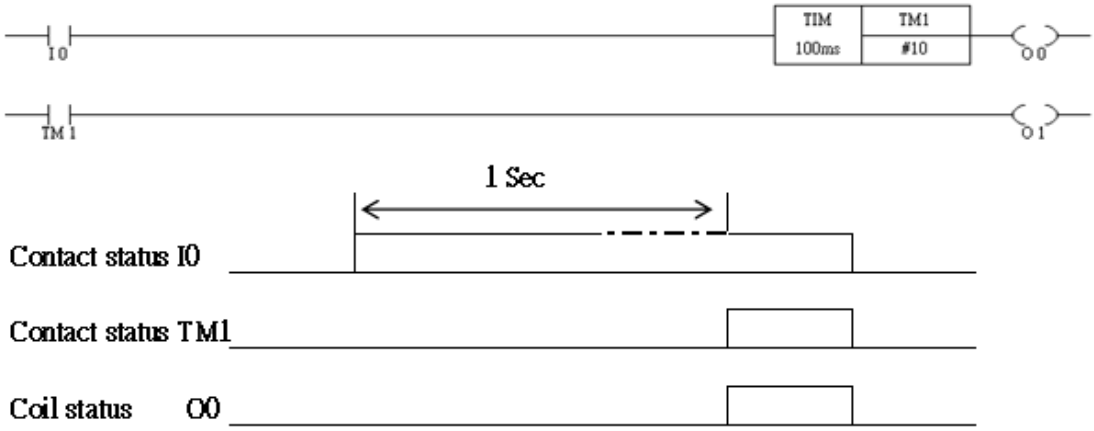


Fig. 85 Timer

4.10. RTimer Relay

- User can adopt RTimer PLC elements by assigning a range of register for the RTimer inner timing so that the number of timers within a ladder can be increased.
- The five register from the assigned address R would be used for inner timing so that the registers cannot be within system write protecting range.
- The 0 bit of the assigned register address R would be the corresponding timer contact for the RTimer. R.0 would be OFF when timing is started and turned ON after timeout.
- The address R+1 is used for recording set time and address R+2 records the current counting.
- After I0 is turned ON and keeps ON, RTimer R0 would start to do timing based on 100 ms resolution. R0.0 is turned OFF, R1 records 10 and R2 records current time counting.
- When the timer reach timeou (1 sec), O0 coil would be turned ON and R0.0 is turned ON.
- If I0 is turned OFF during timing, the timer stops.

- The timing period can be set to 1ms, 10ms, 100ms and 1s.
- The timeout value can be set by register.
- The assigned R address of the RTimer would response to search of register and timer address which are equal to R but can only be replaced by register address.
- Do not use the same register range for more than one RTimer within a ladder.
- The symbol and comment of RTimer are assigned to R.0.

: The interrupt period must be set to 1ms to make the timer with 1ms period accurate.

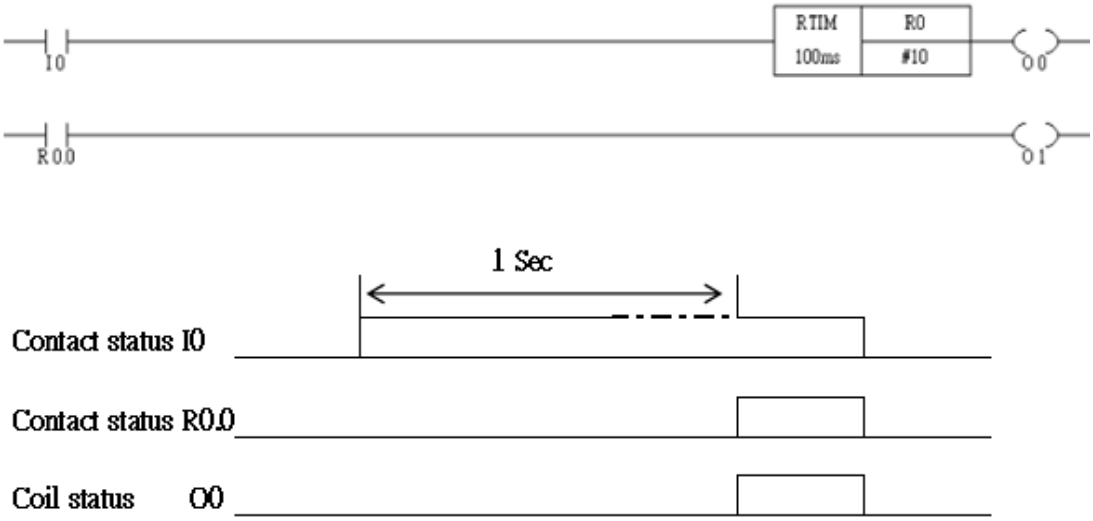


Fig. 86 RTimer

4.11. Up Counter

- When I0 changes from OFF to ON and keeps ON, the counter#1 which is set to up counter increases its count continuously till reaching 10.
- The counter#1enables coil O0 after reaching the set value.
- There are total 256 counters and each counter has the corresponding contact signal. The corresponding contact signal would turn ON when the set count is reached and can be used as normal open or normal close contact.
- The set value for counter can be from register.

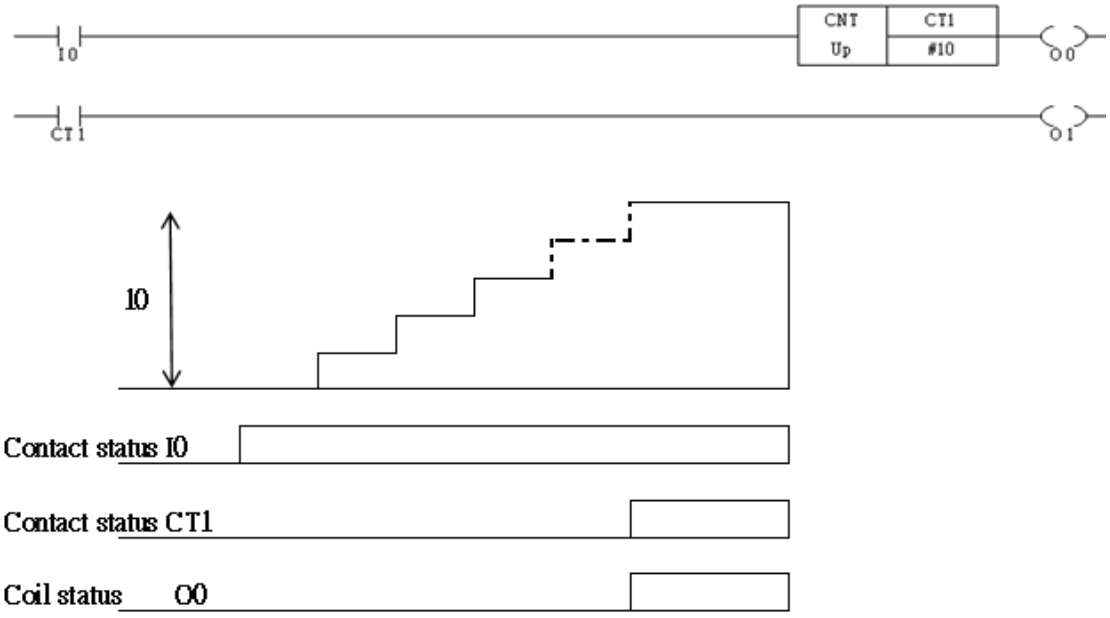


Fig. 87 Up Counter

4.12. Down Counter

- When I0 changes from OFF to ON and keeps ON, the counter#1 which is set to down counter decreases its count continuously till reaching 0.
- The counter#1enables coil O0 after reaching 0.
- There are total 256 counters and each counter has the corresponding contact signal. The corresponding contact signal would turn ON when the set count is reached and can be used as normal open or normal close contact.
- The set value for counter can be from register.

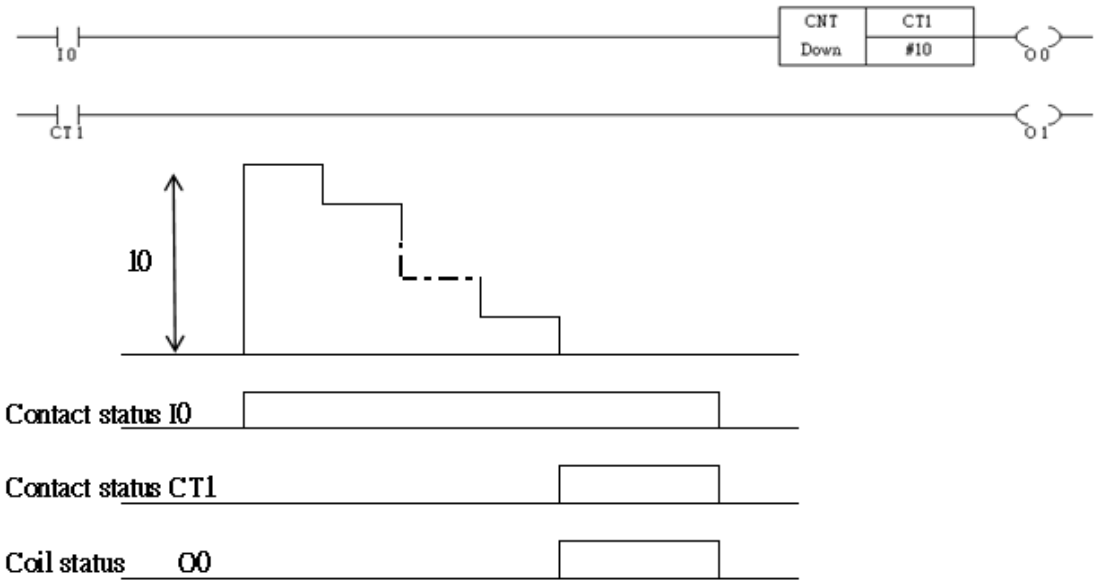


Fig. 88 Down Counter

4.13. Ring Up Counter

- When I0 turns ON from OFF and keeps ON, timer#1 which is set to ring up counter increases its count continuously. The counter would count from 0 again after it reaching the set value cyclically.
- The counter enables coil O0 on each time when it reaches the set value.

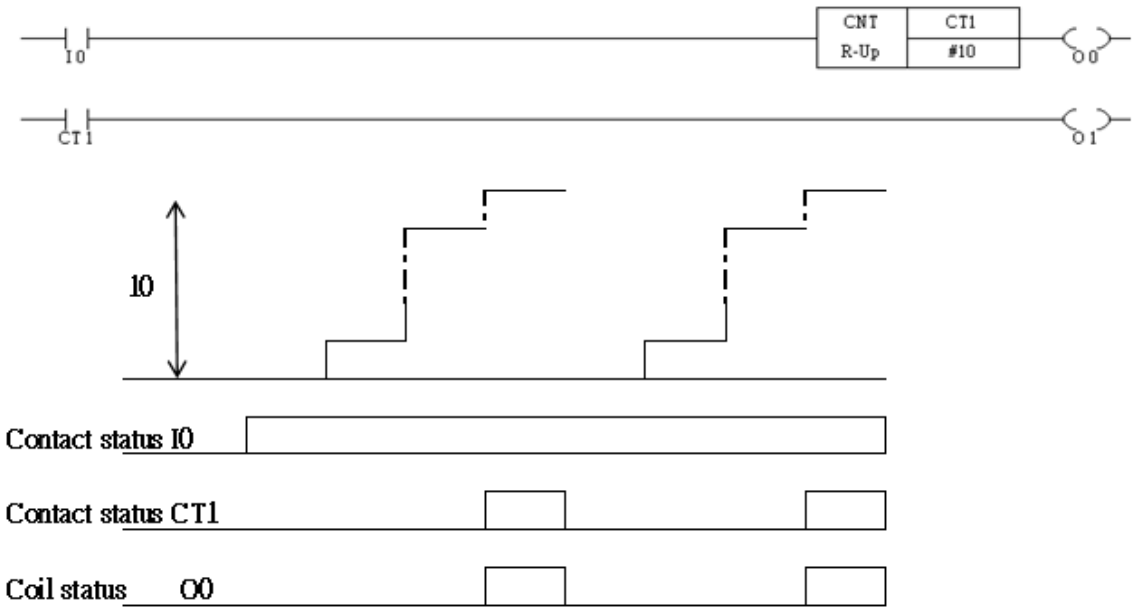


Fig. 89 Ring Up Counter

4.14. Ring Down Counter

- When I0 turns ON from OFF and keeps ON, timer#1 which is set to ring down counter decreases its count continuously from the set value. The counter would count from the set value again after it reaching 0 cyclically.
- The counter enables coil O0 on each time when it reaches 0.

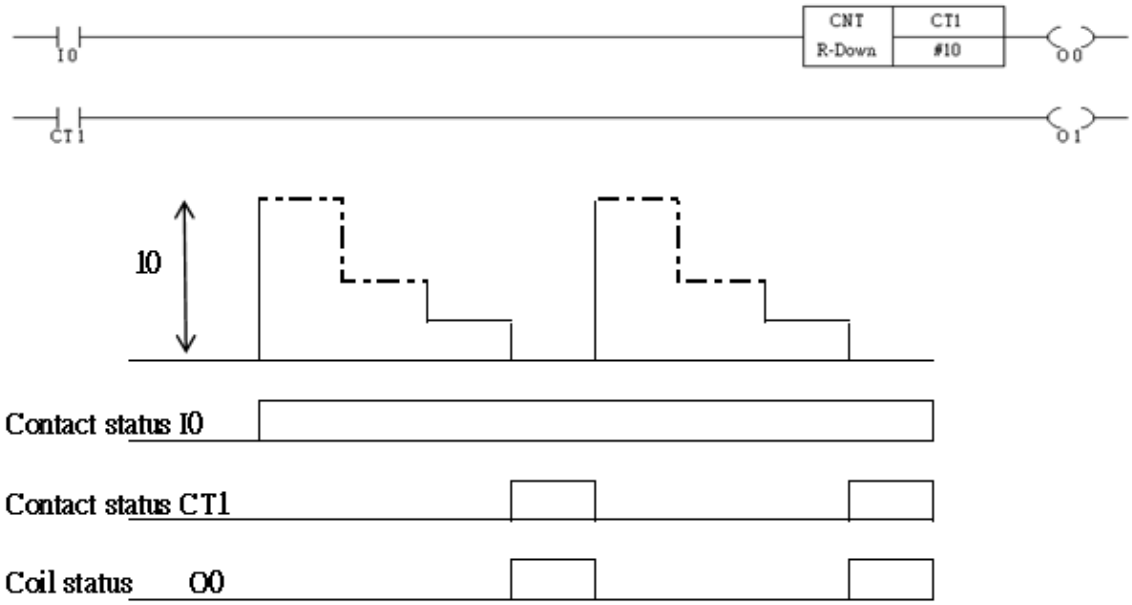


Fig. 90 Ring Down Counter

4.15. Reset Counter

- When I0 turns ON, reset the count of the counter#1.
- If the counter is up type counter, reset to 0.
- If the counter is down type counter, reset to the set value.
- The corresponding contact signal turns OFF at reset.

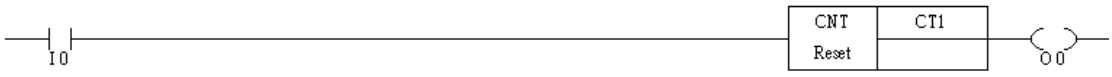


Fig. 91 Reset Counter

4.16. Add

- When I0 turns ON from OFF, the element keeps adding the value of R001 to R002. The register size is 4 byte long.
- Data range：-2147483648<=X<=+2147483647
- Operation：R002<-R002+R001。

- The addend can be set to value instead of register.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

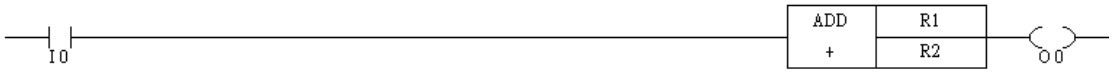


Fig. 92 Add

4.17. Subtract

- When I0 turns ON from OFF, the element keeps subtracting the value of R001 from R002.
- Operation：R002<-R002-R001
- Subtrahend can be set to value instead of register.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

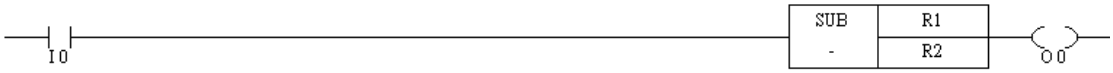


Fig. 93 Subtract

4.18. Multiply

- When I0 turns ON from OFF, the element keeps multiplying the value of R001 to R002.
- Operation：R002<-R002×R001
- Multiplier can be set to value instead of register.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

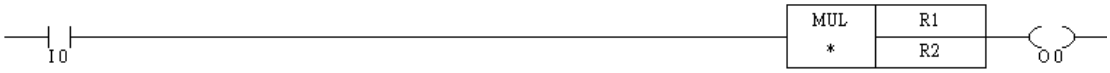


Fig. 94 Multiply

4.19. Division

- When I0 turns ON from OFF, the element keeps dividing the value of R002 by R001.
- Operation：R002<-R002/R001
- Divisor can be set to value instead of register.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

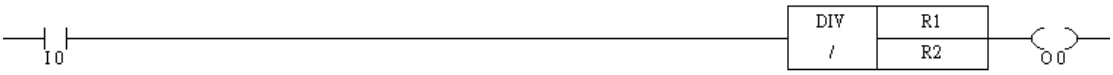


Fig. 95 Division

4.20. Modulo

- When I0 turns ON from OFF, execute R002 modulo R001. Remainder would be put into R002..
- Operation：R002<-R002 % R001
- Divisor can be set to value instead of register.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

Fig. 96 Modulo

4.21. Logic AND

- When I0 turns ON from OFF, the element keeps doing logic AND between R001 and R002. The result would be put into R002.
- Operation：R002<-R002&R001, & means AND operation.
- The input operand can be set to value instead of register.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the

input.

- AND truth table is as Fig. 97.

X	Y	Z
1	1	1
1	0	0
0	1	0
0	0	0

Fig. 97 AND Truth Table

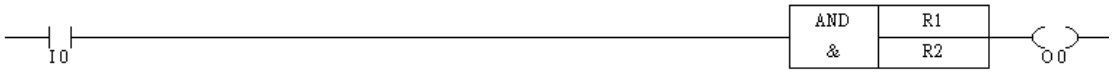


Fig. 98 Logic AND

4.22. Logic OR

- When I0 turns ON from OFF, the element keeps doing logic OR between R001 and R002. The result would be put into R002.
- Operation：R002<-R002|R001. | means OR operation.
- The input operand can be set to value instead of register.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.
- OR truth table is as Fig. 99.

X	Y	Z
1	1	1
1	0	1
0	1	1
0	0	0

Fig. 99 OR Truth Table

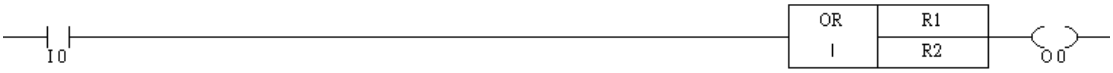


Fig. 100 Logic OR

4.23. Logic XOR

- When I0 turns ON from OFF, the element keeps doing logic XOR between R001 and R002. The result would be put into R002.

- Operation：R002<-R002^R001. ^ means logic XOR.
- The input operand can be set to value instead of register.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.
- XOR truth table is as Fig. 101.

X	Y	Z
1	1	0
1	0	1
0	1	1
0	0	0

Fig. 101 XOR Truth Table

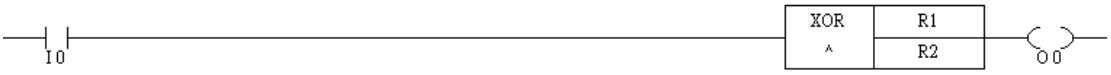
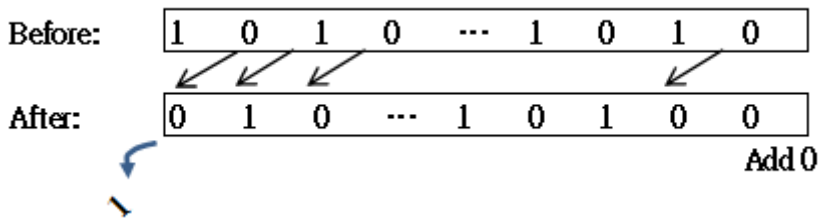


Fig. 102 Logic XOR

4.24. Logic Left Shift

- When I0 turns ON from OFF, the element keeps doing logic left shift on R002 by R001. The least significant bit would be supplemented by 0 for each left shift.
- Operation：R002<-R002<<R001
- The shift number can be set to value and 0<=X<=32.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.



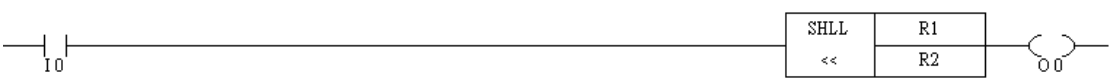
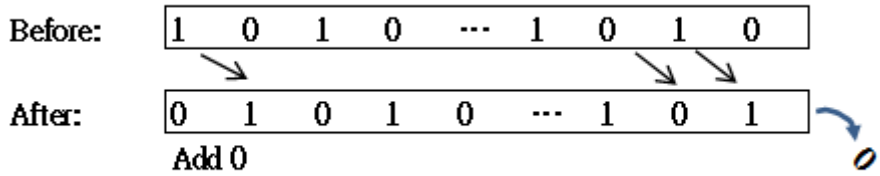


Fig. 103 Logic Left Shift

4.25. Logic Right Shift

- When I0 turns ON from OFF, the element keeps doing logic right shift on R002 by R001. The most significant bit would be supplemented by 0 for each right shift.
- Operation：R002<-R002>>R001
- The shift number can be set to value and 0<=X<=32.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.



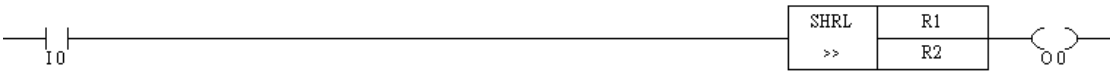


Fig. 104 Logic Right Shift

4.26. Compare：Larger Than

- When I0 turns ON from OFF, the element keeps doing comparison between R001 and R002. If R001 is larger than R002, then the element output TRUE, otherwise outputs FALSE.
- The value to be compared can be set as value instead of register.
- The element would enable coil O0 if the comparing result is TRUE and disable coil O0 if the comparing result is FALSE when it is enabled.
- The element would be executed for each PLC scan if it is enabled by the input.

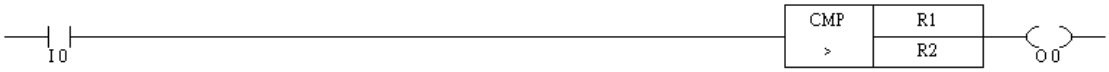


Fig. 105 CMP Larger Than

4.27. Compare：Smaller Than

- When I0 turns ON from OFF, the element keeps doing comparison between R001 and R002. If R001 is smaller than R002, then the element output TRUE, otherwise outputs FALSE.
- The value to be compared can be set as value instead of register.
- The element would enable coil O0 if the comparing result is TRUE and disable coil O0 if the comparing result is FALSE when it is enabled.
- The element would be executed for each PLC scan if it is enabled by the input.

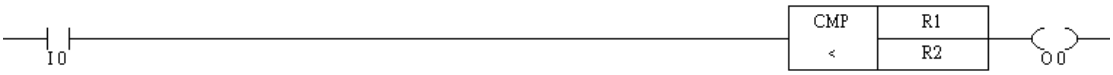


Fig. 106 CMP Smaller Than

4.28. Compare：Larger than or Equal to

- When I0 turns ON from OFF, the element keeps doing comparison between R001 and R002. If R001 is larger than or equal to R002, then the element output TRUE, otherwise outputs FALSE.
- The value to be compared can be set as value instead of register.
- The element would enable coil O0 if the comparing result is TRUE and disable coil O0 if the comparing result is FALSE when it is enabled.
- The element would be executed for each PLC scan if it is enabled by the input.

Fig. 107 CMP Larger than or Equal to

4.29. Compare：Smaller than or Equal to

- When I0 turns ON from OFF, the element keeps doing comparison between R001 and R002. If R001 is smaller than or equal to R002, then the element output TRUE, otherwise outputs FALSE.
- The value to be compared can be set as value instead of register.
- The element would enable coil O0 if the comparing result is TRUE and disable coil O0 if the comparing result is FALSE when it is enabled.

- The element would be executed for each PLC scan if it is enabled by the input.

Fig. 108 CMP Smaller than or Equal to

4.30. Compare：Equal to

- When I0 turns ON from OFF, the element keeps doing comparison between R001 and R002. If R002 is equal to R001, then the element output TRUE, otherwise outputs FALSE.
- The value to be compared can be set as value instead of register.
- The element would enable coil O0 if the comparing result is TRUE and disable coil O0 if the comparing result is FALSE when it is enabled.
- The element would be executed for each PLC scan if it is enabled by the input

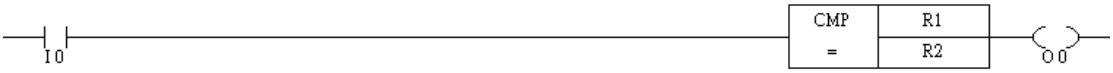


Fig. 109 CMP Equal To

4.31. Compare：Not Equal to

- When I0 turns ON from OFF, the element keeps doing comparison between R001 and R002. If R002 is not equal to R001, then the element output TRUE, otherwise outputs FALSE.
- The value to be compared can be set as value instead of register.
- The element would enable coil O0 if the comparing result is TRUE and disable coil O0 if the comparing result is FALSE when it is enabled.
- The element would be executed for each PLC scan if it is enabled by the input

Fig. 110 CMP Not Equal To

4.32. Move

- When I0 turns ON, the element moves the value within R001 to R002.
- Operation：R002=R001
- The moved value can be set as value instead of register.

- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

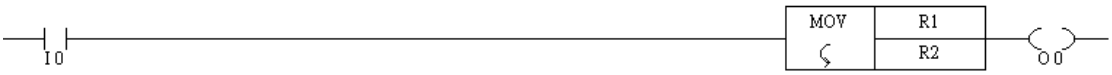


Fig. 111 MOVE

4.33. Register Move to Register pointer

- When I0 turns ON, the element move the value of R001 to the register whose address is the value of R002. If the value of R002 is equal to 3. The value of R001 would be moved to R003.
- Operation：*R002=R001
- The moved value can be set to value instead of register.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.
- If the register address stored in the target register exceeds the allowed range, the element would not do the moving and outputs OFF.

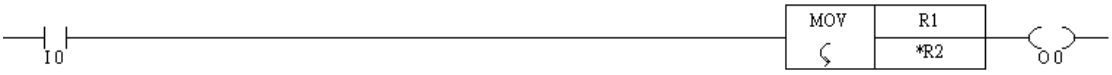


Fig. 112 MOVE from register to register pointer

4.34. Register pointer Move to Register

- When I0 turns ON, the element move the value of register whose address is the value of R001 to R002. If the value of R001 is equal to 4. The value of R004 would be moved to R002.
- Operation：R002=*R001
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.
- If the register address stored in the source register exceeds the allowed range, the element would not do the moving and outputs OFF.

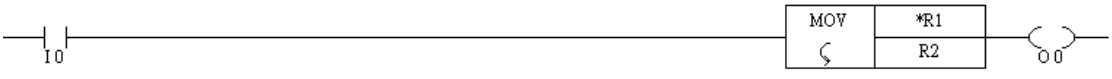


Fig. 113 MOVE Register Pointer to Register

4.35. SCH

- The element searches tool number for its position on the magazine.
- Operation：Tool number is in R0. The magazine table size is in R1. The starting address of the magazine table is in R3. The searching result is in R2.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

：Position index is from 0 to (magazine table size-1).

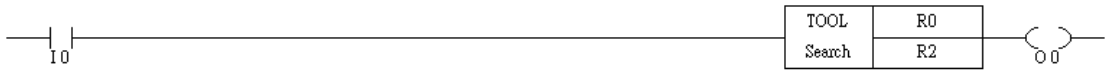


Fig. 114 SCH

4.36. ROT

- The element calculates the shorted path to take on a circular magazine.
- Operation：Current position on circular magazine is in R0，The target position is in R1. The magazine size is in R3. The calculated result is in R2.
- Turn in reverse direction if the result is negative.
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

：The position index is from 0 to(magazine table size-1).

：The element is for circular magazine.

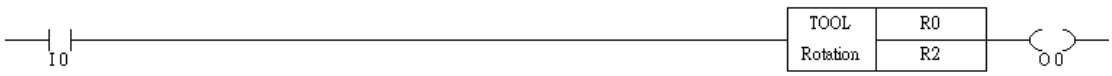


Fig. 115 ROT

4.37. MULRINI

- When I0 turns ON, the element sets value of the ten register from R000 to R009 to 1.
- Operation：[R000..R009]=1
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

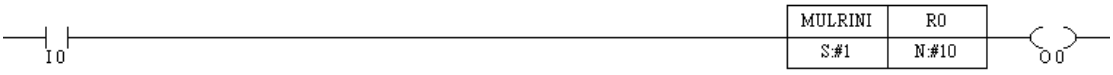


Fig. 116 MULRINI

4.38. MULRCPY

- When I0 turns ON, the element copies the register area from R000 to R009 to the area from R020 to R029 continuously.
- Operation：[R000..R009]=[R020..R029]
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

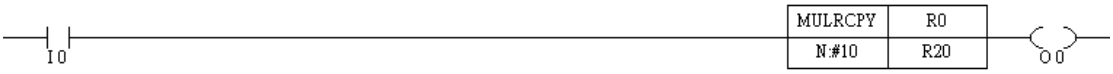


Fig. 117 MULRCPY

4.39. I/RMAP

- When I0 turns ON, the elements maps the value of I001 to I0032 to R001. The I001 is mapped to the least significant bit of R001 and I032 is mapped to the most significant bit.
- Operation：R001=I032<<(31) OR (I031)<<(30) OR …OR (I001)
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

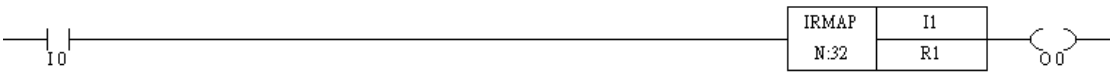


Fig. 118 I/RMAP

4.40. O/RMAP

- When I0 turns ON, the element maps the value of O000 to O0031 to R001. The O000 is mapped to the least significant bit of R001 and O031 is mapped to the most significant bit.
- Operation：R001=O031<<(31) OR (O030)<<(30) OR …OR (O000)
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.

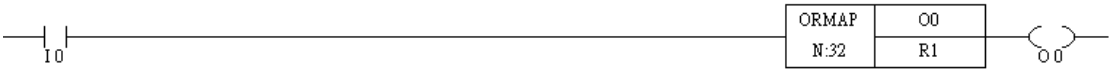


Fig. 119 O/RMAP

4.41. I/RMAPN

- When I0 turns ON, the element maps the value of 10 logic input addresses from I001 to I0010 to R001. The I001 is mapped to the least significant bit of R001 and I010 is mapped to the most significant bit.
- Operation ： R001=I010<<(9) OR (I009)<<(8) OR …OR (I001) OR 0x00000000
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.
- The element would be executed for each PLC scan if it is enabled by the input.
- : The value of the designated register is cleared to 0 before mapping.

Fig. 120 I/RMAPN

4.42. O/RMAPN

- When I0 turns ON, the element maps the value of 10 logic output addresses from O001 to O0010 to R001. The O001 is mapped to the least significant bit of R001 and O010 is mapped to the most significant bit.
- Operation ：R001=O010<<(9) OR (O009)<<(8) OR …OR (O001) OR 0x00000000
- The element would enable coil O0 if it is enabled and disable coil O0 if it is disabled.

- The element would be executed for each PLC scan if it is enabled by the input.
- : The value of the designated register is cleared to 0 before mapping.

Fig. 121 O/RMAPN

4.43. JMP (Jump)

- The element can jump to position before END element.
- After I0 turns ON, the PLC logic would jump to label with string of “stop”.

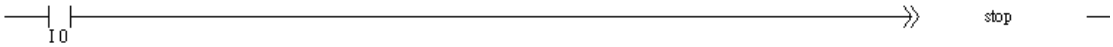


Fig. 122 JMP

4.44. JSR (Jump to Subroutine)

The subroutine area must be after END element. After I0 turns ON, the PLC logic would jump to subroutine ToSub1.

Fig. 123 JSR

4.45. RET

When this element is executed, the execution would return to the caller. If RET is in the top logics (before END), then the ladder logics of this run is finished. If RET is in the subroutine, the execution would return to the top logics, starting from next row where the subroutine is called.

Fig. 124 RET

4.46. RTS (Return Subroutine)

The element is paired with the subroutine label element. The PLC logic would return from subroutine if the element is met.

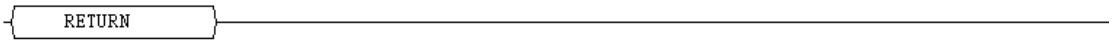


Fig. 125 RTS

4.47. Label

The element is paired with JMP or JSR as target position.
The label string is 6 byte long.

Fig. 126 Label

4.48. END

The element marks the end of a ladder. The area after this element is for subroutines.

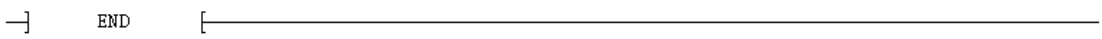


Fig. 127 END

4.49. Example for SCH

- The magazine table is from R100 to R103. The size is 4. The position index is 0 to 3.
- R2：Magazine table size.
- R4：The start address of magazine table.
- R1：Tool index for searching.
- After I0 turns ON，the value of R3 is equal to 3, which means tool #248 is at position index 3.

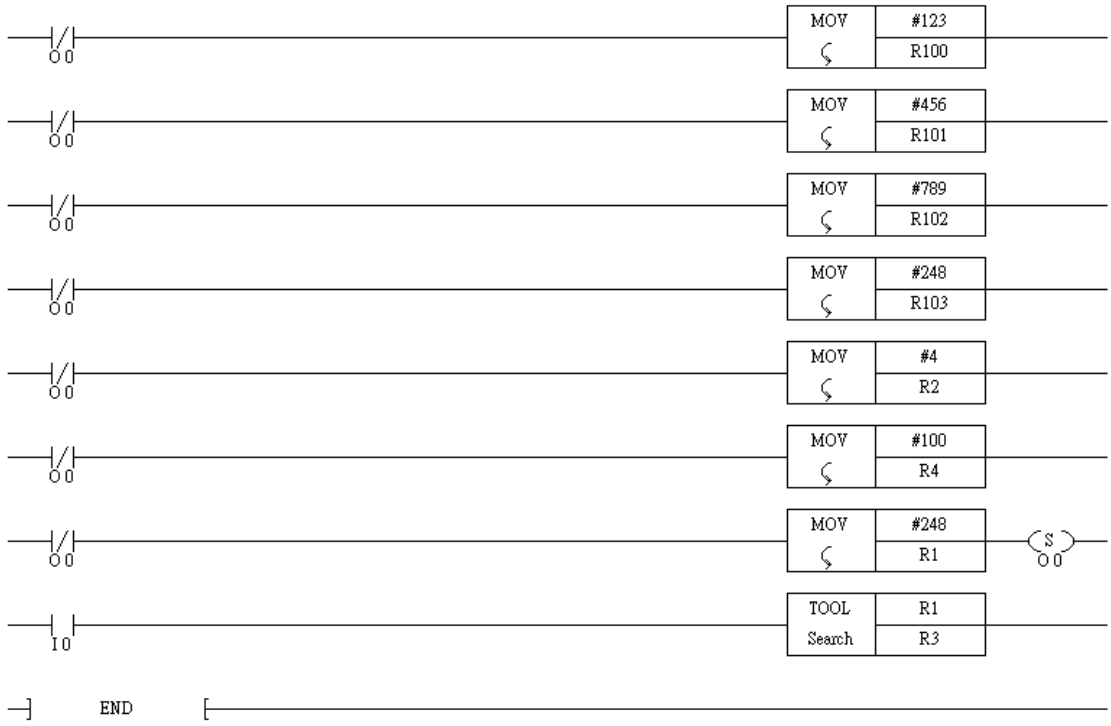


Fig. 128 SCH Example

4.50. Example of ROT

- R1：Current position on circular magazine.
- R2：Target position.
- R4：Magazine size.
- After I0 turns ON，R3 value is equal to -1 which means the shortest path to the target position is to reverse one step.

Fig. 129 ROTExample

4.51. Example of JMP

- After I0 turns ON, the accumulation on R0 would be skip.
- The paired label element must be before END element.

Fig. 130 JMPExample

4.52. Example of JSR

- After I0 turns ON, accumulation on R1 is started.
- The logic would continue from next element of JSR element after finishing Sub1 so that accumulation on R0 is always executed.
- The label used as subroutine must be after END element.

Fig. 131 JSR Example

4.53. RET Example

- When I0, I10 are both OFF, R0 and R1 keep being updated.
- When I10 is ON but I0 is OFF，R1 stops being updated but R0 keeps being updated.
- When I0 is ON, the ladder logics would return immediately and R0 and R1 stop being updated.

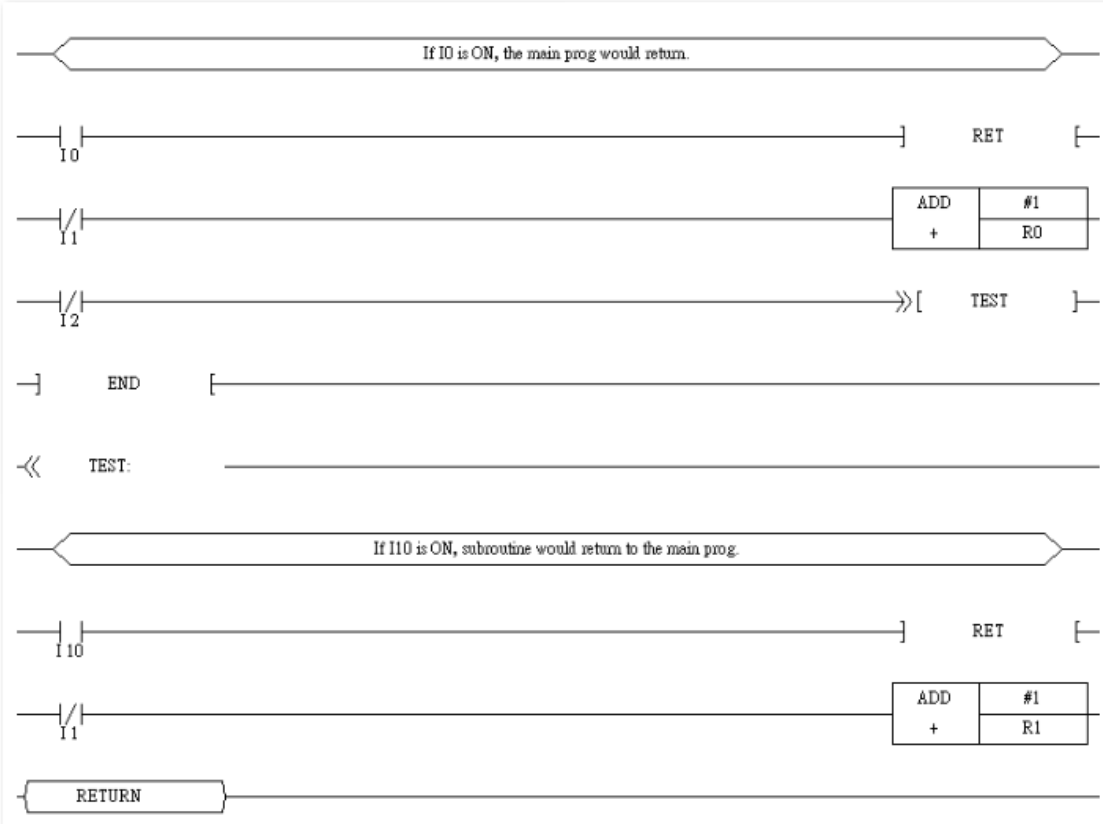


Fig. 132 RET Example

5. Simulation Mode

5.1. Start the Simulation Mode

The simulation mode can simulate the PLC logic on host computer without interacting with other modules in the controller. The user can verify the correctness of the pure PLC logic. How to start refer to 3.6.14.

5.2. The Introduction of the Simulation Mode Layout

The simulation layout is as Fig. 133, only the areas that differ from edit mode are explained as following.

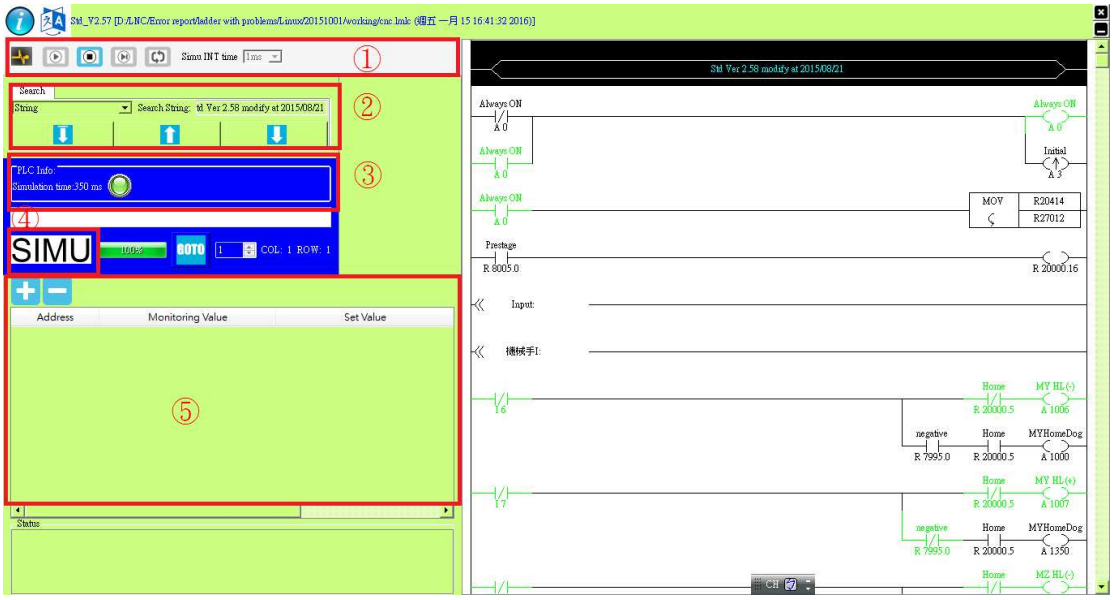


Fig. 133 Simulation Mode

① Simulation Mode toolbar.

② Search area. Only search is allowed in simulation mode.

③ Simulation time duration and operation light (Green for operating, Red for stop) are added to status area.

④ The working mode is shown as simulation mode.

⑤ Address monitor/setting area.

The ladder would show the element activation status by the color set by the user if the element has active/inactive state.

5.3. Introduction of Simulation Mode for Ladder in LMLC Format

5.3.1. Mouse and Keyboard

Mouse left button：

Double click on contact and coil elements to toggle the active status.

：The active status of the contact and coil element may not be changed because of the PLC logic under simulation operating condition.

Mouse right button：

None.

Wheel：Move focus up and down.

Move the cursor to the icon on the toolbar would display the tip for the icon.

Up, Down, Right, Left Key：Move focus up, down, left and right.

Page Up/Down Key：Move focus up or down by one page.

5.3.2.Toolbar

5.3.2.1.  End Simulation Mode

Switch back to the edit mode.

5.3.2.2.  Start the Simulation

Start the simulated interrupt to run the PLC logic. The operation light turns green.

5.3.2.3.  Stop the Simulation

Stop the simulated interrupt. The operation light turns red.

5.3.2.4.  Single Run

Only run simulated interrupt once and stop.

 : Not possible during simulation operating

5.3.2.5.  Reload

Reset simulation environment like to initialize PLC address area, etc.

 : Not possible during simulation operating.

5.3.2.6.  Set the Simulated Interrupt Period

The simulated interrupt period would effect the timer elements.

5.3.3. Address Monitor/Setting Area

The user can add or remove the addresses for monitor/setting.

5.3.3.1.  Add Address

Click to pop the dialog as Fig. 134. Choose the address as Fig. 135. The user can choose the way to display the register type as Fig. 136. The address would be added to the list as Fig. 137.

Fig. 134 Add Address

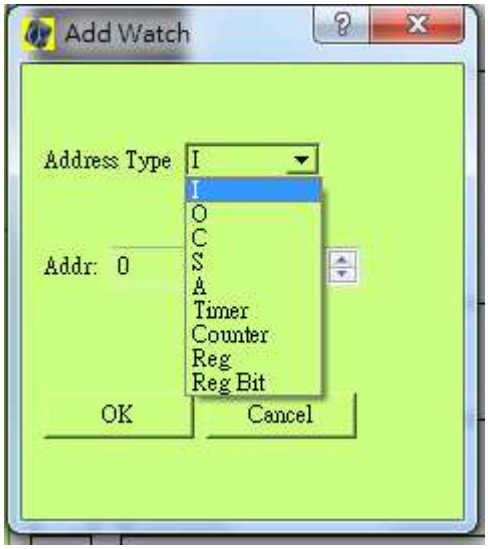


Fig. 135 Select Address Type

Fig. 136 The Way of Displaying Register

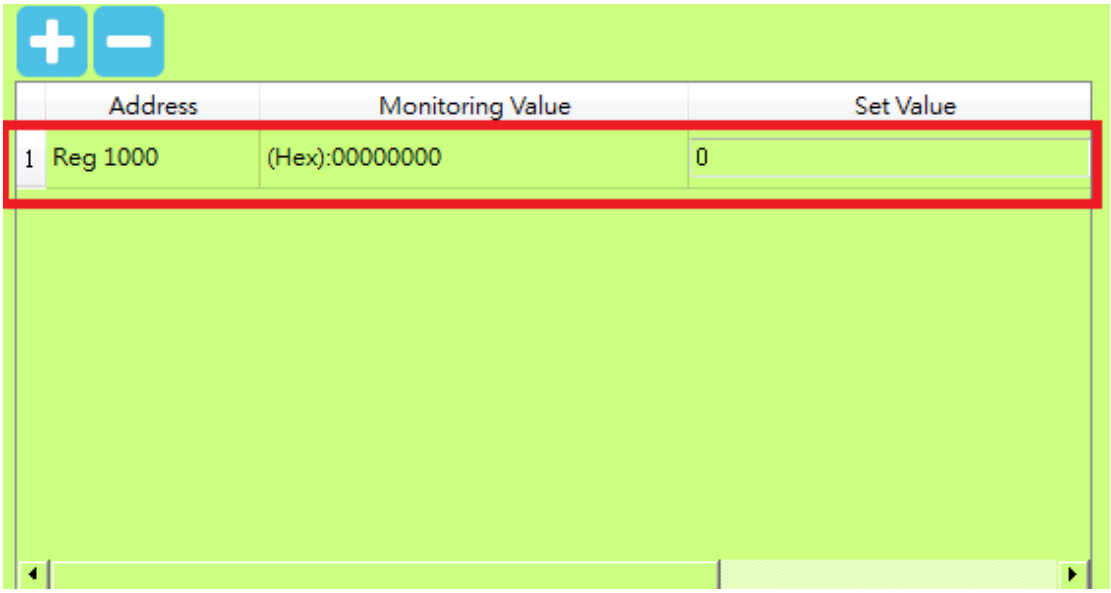


Fig. 137 Added Address

5.3.3.2. Remove Address

Click this icon to remove the address with focus on the list.

Operation of the Address

The Monitor Value column shows the current value of the address. The user can set the value on Set Value column.

: The monitored value may not reflect the set value while simulation is operating. The value may be overwritten by running PLC logic.

 ：Set register value in decimal only.

6. Monitor Mode

6.1. Start the Monitor Mode

The user can monitor the PLC logic on the remote controller with the monitor mode. How to start refer to 3.6.13.

：The monitor mode is not in real time. The monitored status may be asynchronous with the remote side.

：The ladder files must be synchronized to get correct monitor result.

：Automatically return back to the edit mode if the connection is broken.

6.2. The Introduction of the Simulation Mode Layout

The layout for monitor mode is as Fig. 138. The areas which differ with edit and simulation mode is explained as follows.

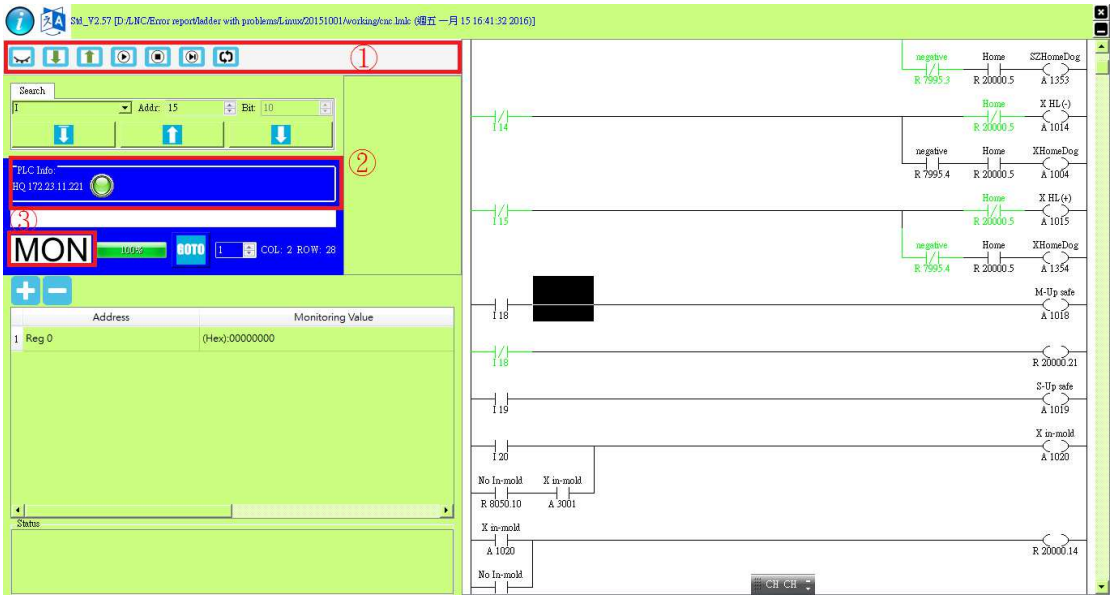


Fig. 138 監看模式

① Toolbar.

② Connected control IP and PLC operation status light on remote controller are added to the status area. The PLC operation light is green if the PLC logic is running, red if the PLC logic is stop and yellow during reloading.

③ The working mode is shown as monitor mode

The ladder would show the element activation status by the color set by the user if the element has active/inactive state.

6.3. Introduction of Monitor Mode for Ladder in LMLC Format

6.3.1.Mouse and Keyboard

Mouse left key：

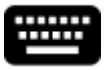
None.

Mouse right key：

None.

Wheel：Move focus up and down.

Move the cursor to the icon on the toolbar would display the tip for the icon.



Up, Down, Left, Right Key：Move focus up, down, left and right.

Page Up/Down Key：Move focus one page up and down.

6.3.2.Toolbar

6.3.2.1.  End Monitor Mode

Return to the edit mode.

6.3.2.2.  Download the Files on Remote Side and Open

Click and the user can select the directory to contain the downloaded files as Fig. 139. The application shows overwriting warning as Fig. 140. The user can select files for downloading as Fig. 141. The application automatically opens the downloaded LMLC file when the downloading successfully finishes.

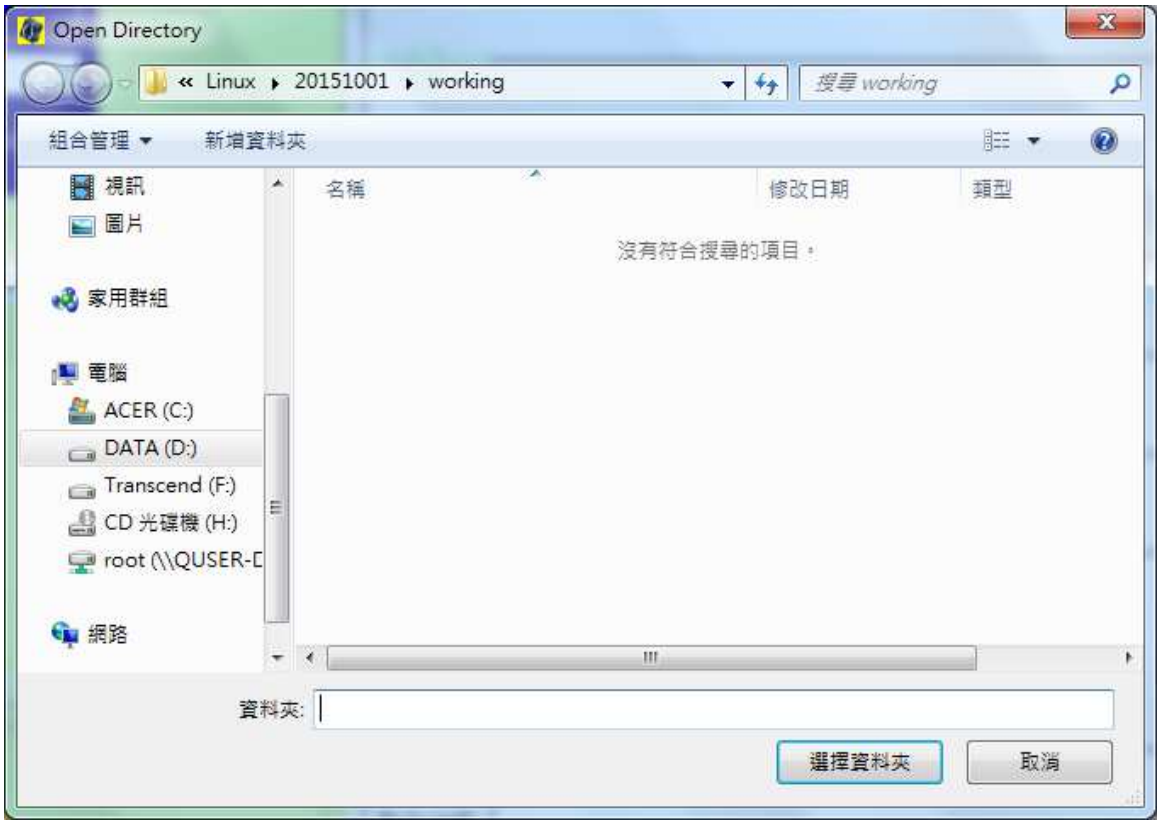


Fig. 139 Select Local Directory

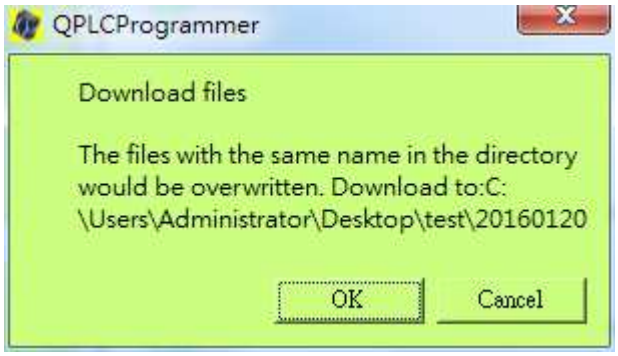


Fig. 140 Overwriting Warning

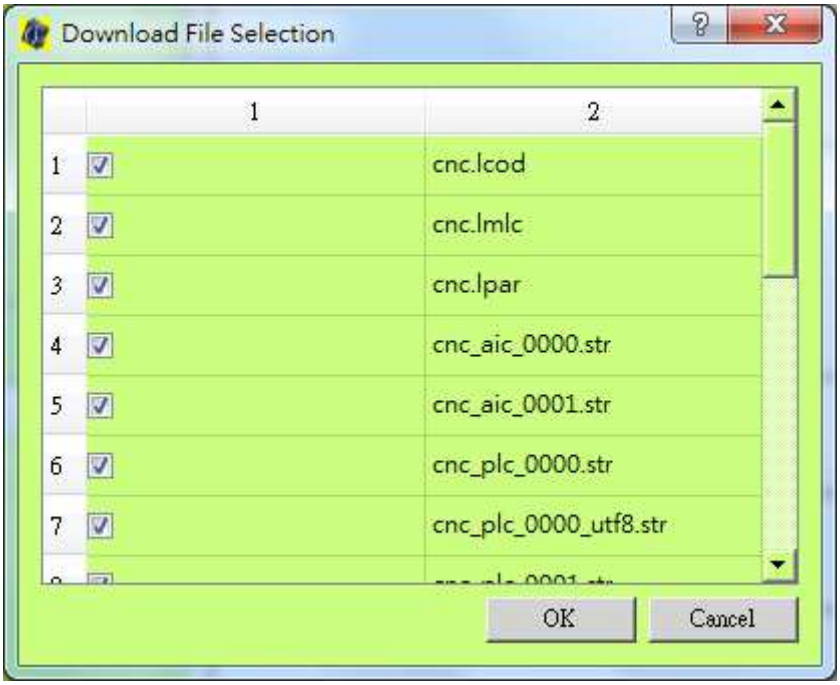


Fig. 141 Select Remote Files

6.3.2.3. Upload Local Files to Remote Controller

Click and the warning message is shown as Fig. 142. The user can select the files to upload as Fig. 143. The selected files will be uploaded after the user confirming.

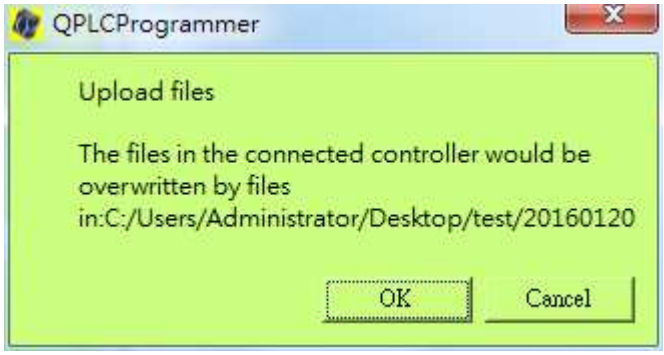


Fig. 142 Upload Warning

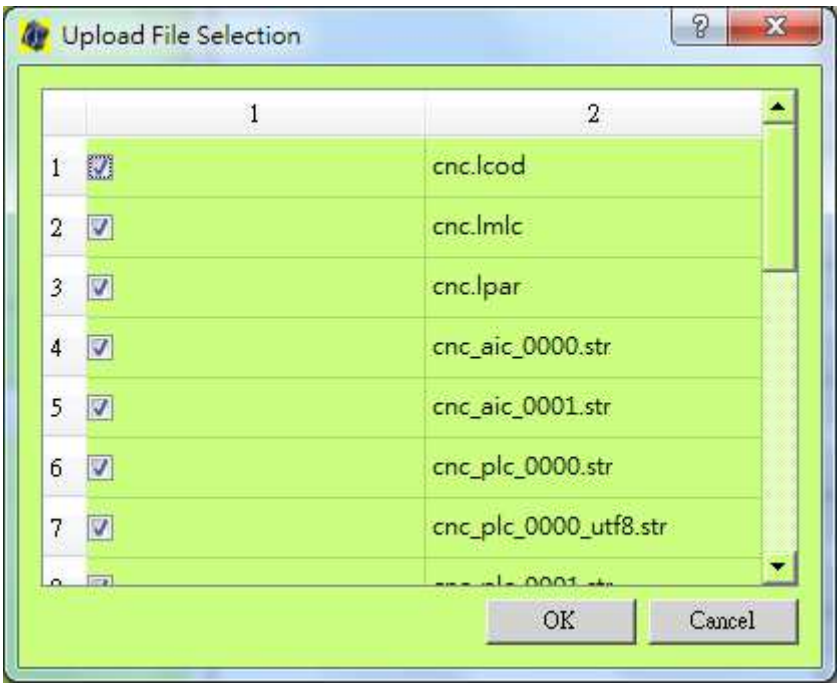


Fig. 143 Local File List

: The user must signal the remote controller to reload PLC for the uploaded binary executable file taking effect.

6.3.2.4. Start PLC on the Remote Controller

Inform the remote controller to start the PLC logic. The remote PLC operation light should turn to green. If the light is not green, check the remote controller.

6.3.2.5. Stop PLC on the Remote Controller

Inform the remote controller to stop the PLC logic. The remote PLC operation light should turn to red. If the light is not red, check the remote controller.

6.3.2.6. Single Run PLC on the Remote Controller

Inform the remote controller to run the PLC logic only for one interrupt.

6.3.2.7.  Reload PLC on the Remote Controller

Inform the remote controller to reload PLC. The remote PLC operation light should turn to green eventually after a short period time.

Monitored Address Area

Refer to 5.3.3 but setting value under monitor mdoe is not possible.

7. Set the URL Shown on Start Layout

It is possible to customize the URL to which the application would try to connect at starting. The steps are as follows：

- Get copy of progsettings.ini from the installation folder of a copy of PLC Programmer.

- Modify the value of the two keys manually.

main_url, this is the URL to which the application would try to connect at starting.
Example：main_url=http://www.google.com.tw/
backup_url, this is a local html file when connection to Internet is not possible.
Example：backup_url=test.html

- Put the following files within the same folder as Fig. 144.

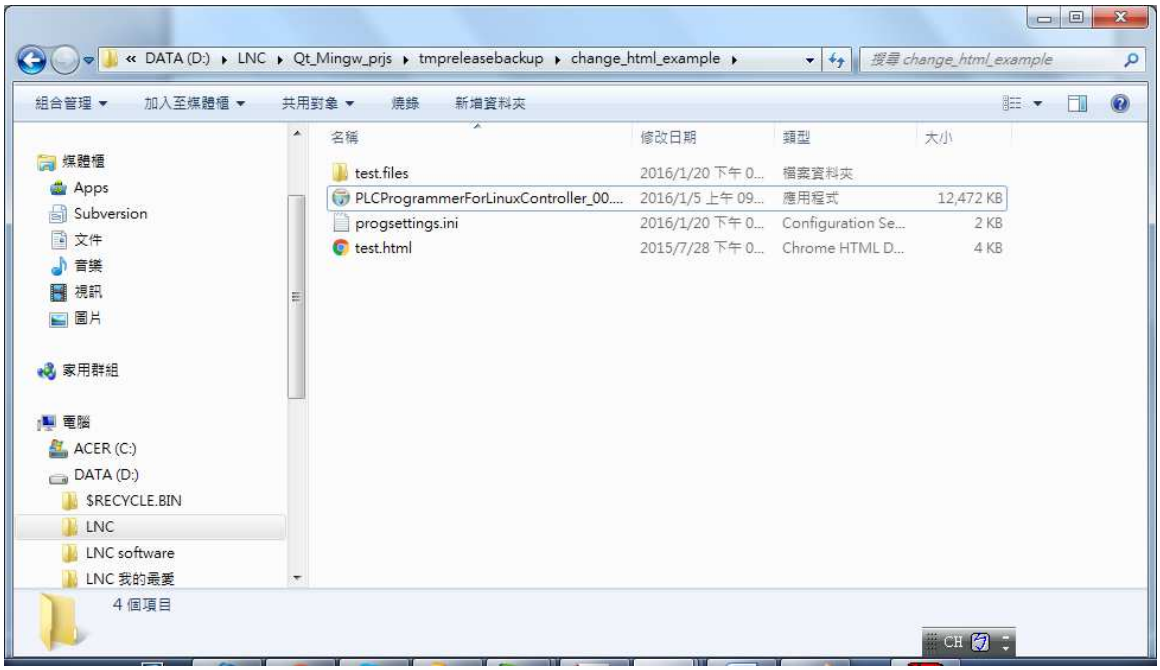


Fig. 144 Customize the Start URL

：If the local html file needs additional files, those files can be put within a folder named as <html file name>.files. This folder is named as test.files in the example since the local html file is test.html.

- The starting layout would be as Fig. 145 after installation.

Fig. 145 Customized Starting URL

8. Trouble Shooting

8.1. Compiling Errors and Warnings

Compiling errors would stop generating the executable binary file. On the contrary, compiling warnings would not stop the generation of the executable binary file since it may be the user special usage or not severe issue.

8.1.1. Errors

8.1.1.1. Branch before Any Contact Address is Loaded

Pattern is as Fig. 145. There is no contact element before branch.

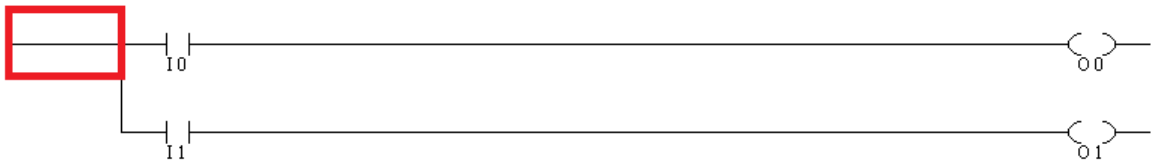


Fig. 146 Pattern

8.1.1.2. Duplicated Timer

Using the same timer address more than once within a ladder inclines to unexpected result.

8.1.1.3. Duplicated Counter

Using the same counter address more than once within a ladder inclines to unexpected result.

8.1.1.4. Connection Configuration between the Elements is inconsistent

The record for a vertical connection between elements is not consistent. Delete or redo the vertical connection on the reported position.

8.1.1.5. There is function element before any contact address is loaded

Pattern is as

Fig. 147. There is no contact element before function elements.

Fig. 147 Pattern

8.1.1.6. Illegal Label Position

The label paired with JMP element is after END element.

8.1.1.7. Illegal Position for Return from Subroutine

RTS element is before END element.

8.1.1.8. Jump is not paired with a Label

There is no label before END element to pair with a JMP element.

8.1.1.9. Jump to Illegal Position

The label element which has the same string with JMP element is after END element.

8.1.1.10. Subroutine Label is at Illegal Position

The label element which has the same string with JSR element is before END element.

8.1.1.11. Jump to Subroutine is not paired with a Label

There is no label element after END element to pair with JSR element.

8.1.1.12. Subroutine is not paired with Return

The label element used as subroutine does not pair with RTS element.

8.1.1.13. Duplicated Label

There is more than one label with the same string.

8.1.1.14. Open Circuit

There is broken connection in ladder as Fig. 148.

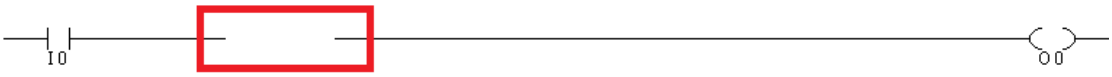


Fig. 148 Pattern

8.1.1.15. A Row is consist of only Horizontal Connect Elements

There is row only consist of horizontal connection element as Fig. 149.

Fig. 149 Pattern

8.1.2.Warnings

8.1.2.1. Duplicated Coil

There is more than one coil with the same address. The user should use with caution if the pattern is used.

8.1.2.2. Short Circuit

There are only horizontal connection elements between branches. It may be redundant connection or invalidate other OR branches as Fig. 150.

: Some short circuit pattern may cause compiling failure.

Fig. 150 Pattern

8.1.2.3. More Than One End for the Ladder

There is more than one END element in a ladder.

8.1.2.4. Not Recommended Upward Branch

There is upward branch as Fig. 151. This is stronly not recommended.

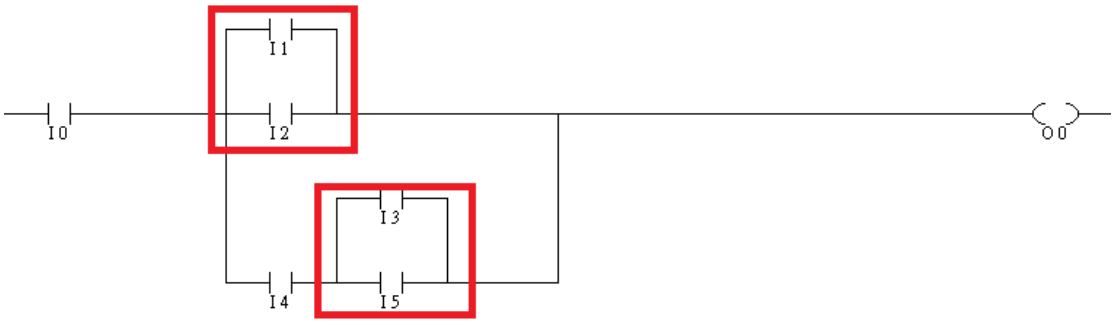


Fig. 151 Pattern

8.2. User Register Area

A ladder with element which contains user register area (address above 1,000,000) is not supported by the PLC programmer 02.XX.XX.XX version. A warning would be shown if such ladder is opened by PLC programmer 02.XX.XX.XX version as Fig. 152. Compiling and editing are also not possible to prevent generation the executable binary which can cause issue.

: PLC Programmer 02 version does not support user register area.

: The functionality must work with the controller which also supports it. The PLC logic would be skipped on the controllers which does not support the functionality.

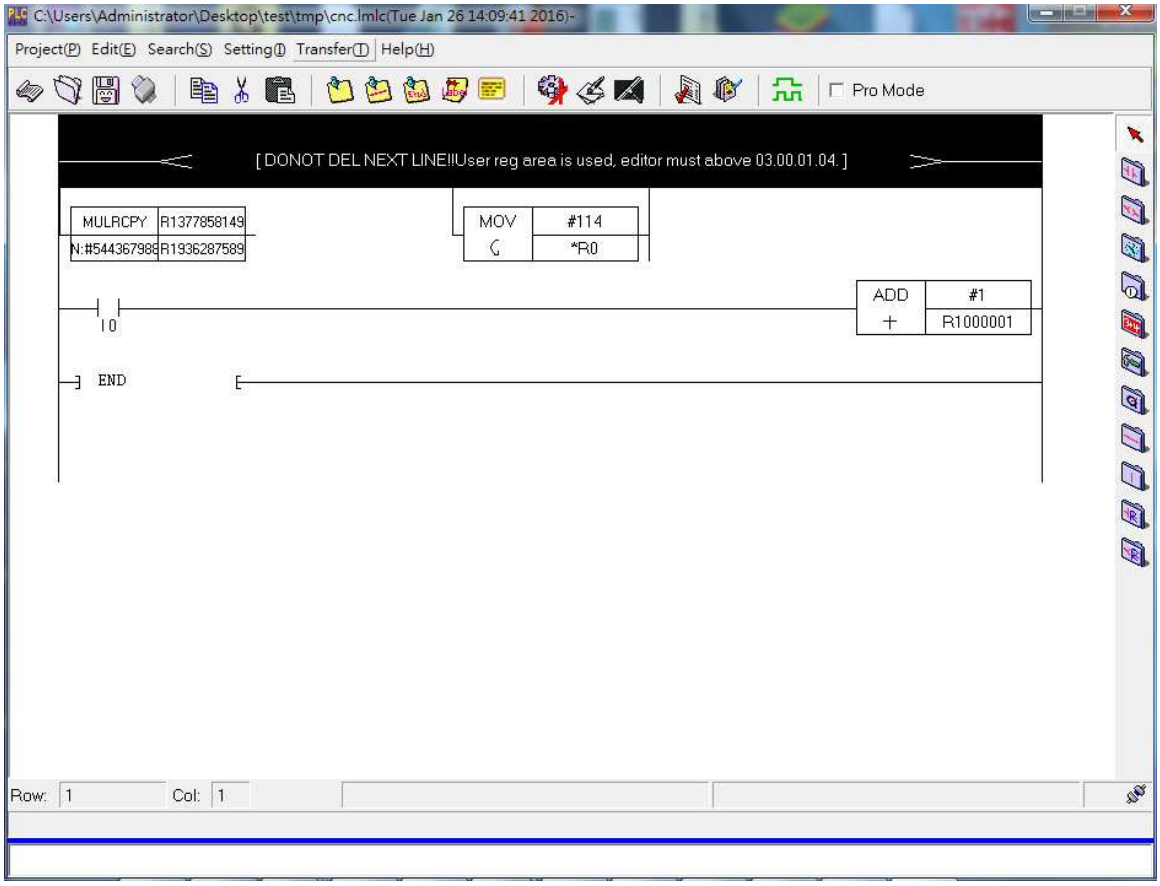


Fig. 152 User Register Area Warning